**International Journal of AI, Big Data, Computational and Management Studies**
Noble Scholar Research Group | Volume 2, Issue 3, PP.53-63, 2021
ISSN: 3050-9416 | https://doi.org/10.63282/3050-9416.IJAIBDCMS-V2I3P107

# A Review of AI and Machine Learning Solutions for Fault Detection and Self-Healing in Cloud Services

Avinash Attipalli[1], Sunil Jacob Enokkaren[2], Varun Bitkuri[3], Raghuvaran Kendyala[4], Jagan Kurma[5], Jaya Vardhani Mamidala[6]
[1]University of Bridgeport, Department of Computer Science, USA.
[2]ADP, Solution Architect, USA.
[3]Stratford University, Software Engineer, USA.
[4]University of Illinois at Springfield, Department of Computer Science, USA.
[5]Christian Brothers University, Computer Information Systems, USA.
[6]University of Central Missouri, Department of Computer Science, USA.

**Abstract:** The reliability and availability of cloud services are thus critical issues as the usage of cloud computing as a fundamental aspect of present-day digital infrastructure continues to grow. As cloud environments become more complex and larger, the utility of the traditional fault detection schemes and approaches is inadequate, thus providing possibilities of system crashes, poor performance, and interruption of services. In this paper, researching the use of Artificial Intelligence (AI) and Machine Learning (ML) in detecting faults and self-healing in cloud services is performed. It highlights how the traditional rule-based monitoring is moving towards AI-based solutions that utilize large operational datasets in order to find anomalies and perform predictive maintenance. It classifies many types of AI / ML models, such as supervised models, unsupervised models and deep learning models, and explains their success or effectiveness in detecting faults and automating recovery operations. Moreover, it also considers the issues and prospects of including AI/ML in fault management of the cloud environment with the final goal of achieving system resilience and operational effectiveness.

**Keywords:** Artificial Intelligence, Machine Learning, Fault Detection, Self-Healing, Cloud Services.

## 1. Introduction

The concept of cloud computing has also emerged as the foundation of the current digital infrastructure where computing resources can be accessed on demand and provide scalability to both business enterprises and individuals alike [1]. With the increased use of cloud services by organizations as a way of running mission-critical applications, ensuring the availability of data and supporting remote operations, availability and reliability of services have remained crucial [2]. Nonetheless, the increasing size and complexity of cloud systems also increase their vulnerability to different kinds of faults such as faults on hardware, software bugs, and network disturbances. Fault detection in the cloud deals with detecting non-normal or abnormal behaviour that could result to service degradation or system failure in the cloud. The most common problems are crashing servers, memory leaks, network latency and software failures [3][4]. The typical rule-based monitoring systems tend to be unsuitable in such dynamic and heterogeneous environments because they are based on static thresholds and predetermined rules which are unable to predict novel or previously unknown failure patterns. Such shortcomings have led to the adoption of a paradigm shift in the adoption of Artificial Intelligence (AI) and Machine Learning (ML) techniques, capable of undertaking analysis on large volumes of log files, system events, and performance measurements independently and in real-time to identify anomalies in the system.

The techniques used in AI and ML have been shown to provide revolutionary methodologies to deal with the shortcomings of the traditional fault detection and recovery methods [5]. Supervised, unsupervised, and reinforcement learning algorithms form the learning background of these models to analyse operational data, reveal latent failure patterns, and use predictive diagnostics [6]. Consequently, they increase the relevance, rate and flexibility of fault detection and allow proactive and automated healing. In addition to fault identification, inclusion of AI/ML in self-healing processes empowers cloud systems to self-manage faults and their recovery with minimal input of human intervention [7]. These are smart orchestration systems, automatic resource provisioning and autonomic service reconfiguration [8]. Specifically, methods like reinforcement learning and knowledge-based systems are especially interesting because they can continually optimize policies of recovery on a broad variety of fault scenarios and learn through time. The use of AI and ML allows creating intelligent, resilient and autonomous cloud-based infrastructures. The purpose of this review is to bring out the state-of-the-art methods, categorize these methods according to their methodology and area of application as well as comment on how AI/ML has an opportunity to be used in the future regarding fault management and self-healing of cloud services.

### *1.1. Structure of the Paper*

The structure of this paper is as follows: Section II discusses the fundamentals of fault detection and self-healing in cloud environments. Section III reviews AI-based fault detection techniques. Section IV examines the role of AI and Machine Learning in cloud-based fault management. Section V presents self-healing architectures and mechanisms. Section VI reviews

relevant literature and case studies and finally, Section VII concludes with a discussion of challenges and future research directions.

## 2. Fundamentals of Fault Detection and Self-Healing In Cloud Environments

Fault Detection and Self-Healing in Cloud Environments refer to the automated processes of identifying, diagnosing, and resolving system failures with minimal human intervention. Fault detection continuously monitors system metrics, logs, and behaviours to detect anomalies or service disruptions. Once a fault is detected, self-healing mechanisms are triggered to restore normal operations through automated actions, such as restarting services, reallocating resources, or initiating backups [9]. These smart, adaptable methods greatly decrease downtime and increase operational efficiency in ever-changing cloud settings by ensuring high availability, reliability, and resilience.

### *2.1. Fault Detection in Cloud Environment*

One definition of fault tolerance is a system's ability to respond quickly and effectively to unforeseen problems with hardware or software, while another is an approach to system design that allows a system to continue functioning even when a component fails [10]. A system may be able to keep running, although at a reduced capacity, after a breakdown thanks to fault tolerance technologies. Crash faults and Byzantine/Arbitrary faults are the two most common types of cloud-related errors. In the event of a crash, the system might crash completely; in the case of a Byzantine or arbitrary malfunction, it could divert from its usual functionality. When resources like storage, software, and hardware fail in a cloud environment, it impacts end users. There are three types of hardware faults: temporary, intermittent, and permanent.

### *2.2. Types of Faults*

The monitored system's fault occurrence can be ascertained by fault detection. This method relies on the interdependencies between several observable signals to identify process, actuator, and sensor failures [11]. Fault isolation and fault identification are closely related activities as well. Fault identification finds the size of a fault, while fault isolation finds its location and kind. One step in making a diagnosis is tracking down and removing the element that is causing the issue. During fault diagnosis, it is important to gather as much pertinent information as possible, such as the size, position, and time of identification of the defect, in order to determine the type of fault.

- **Physical Faults (hardware faults):** System errors that primarily affect hardware components, including the CPU, RAM, storage, and power outages.
- **Software Faults:** arise from bugs, incorrect configurations, or compatibility issues within operating systems, hypervisors, or cloud management software. Such faults may cause unexpected behaviour or service interruptions.
- **Network Faults (link faults):** The resources the user might have access to due to cloud computing being of the network nature are prone to errors such as packet loss, failure of the link among others.
- **Processor Faults (node faults):** These faults are due to software bugs, lack of resources and inappropriate consumption of the computational resources.
- **Service Expiry Faults:** Inconveniences caused by expiration of the service period of an application being used when it runs out of the service period.
- **Timing Faults:** Issues that will not allow an application to complete within the designated time frame.
- **Application Faults:** be caused by necessary runtime errors, memory leakage or management of microservices with cloud-native characteristics. This can deny the user experience or lead to partial failures of services.

### *2.3. Concepts of Self-Healing Systems in Distributed Architecture*

Cloud computing's self-healing systems are designed to find, fix, and recover from problems on their own, guaranteeing that services will be available and reliable at all times. These systems are defined by their autonomy, adaptability, resilience, and capacity to learn from past failures. The architecture typically comprises monitoring agents, anomaly detection modules, decision-making engines, and automated recovery mechanisms that collaboratively sustain system health. While traditional self-healing methods rely on rule-based scripts and predefined workflows, they often struggle with unforeseen or complex scenarios. Smart problem identification, root cause analysis, and proactive recovery are made possible by AI-based self-healing systems that use pattern recognition, ML, and predictive analytics. These systems work exceptionally well in large-scale and dynamic cloud environments. The healing agent in such systems can be an integrated component or an independent module, and the design often aligns with software agent architectures autonomous, interactive, and environment-aware particularly in multi-agent configurations [12]. Systems operate based on these strategies and policies as tools for observing and evaluating the state of the system where the system collects information, which consists of the type and quality of information which has a critical role in the decisions of adaptation. A closer breakdown of these self-healing methods together with a literature review is described below.

### *2.4. Traditional Methods vs. AI-Powered Approaches*

Progressive fault detection in cloud computing is simpler because it uses manual intervention and static rules, which are not effective in highly dynamic situations [13]. Conversely, AI-informed techniques employ ML to identify the complex patterns, adjust to varying circumstances and automate answers. It is more precise, scalable and has lower downtimes than the other

approaches, Studies comment DL models perform better than traditional methods that are used in cloud systems that have dynamic workloads [14]. Comparison of AI-based vs traditional fault detection is given below (Table I):

**Table 1: Traditional Fault vs. AI-Based Fault Detection with Aspects**

| Aspect | Traditional Fault Detection | AI-Based Fault Detection |
|---|---|---|
| Technique Used | Static thresholds, rule-based tracking, and human intervention | ML, data-driven anomaly detection, automated responses |
| Adaptability | Low adaptability to dynamic workloads and heterogeneous environments | High adaptability to dynamic workloads and changing operational contexts |
| Failure Pattern Recognition | Limited to known issues and explicit patterns | Can detect implicit, complex, and temporal failure patterns |
| Automation Level | Manual or semi-automated fault recovery | Fully automated fault detection and self-healing actions |
| Scalability in Large Ecosystems | Performing poorly in intricate, large-scale cloud infrastructures | Effective and scalable in massive, ever-changing cloud environments |
| Learning Capability | No learning from past incidents | Learns from historical data and adapts over time |
| Human Involvement | Required for monitoring, diagnosis, and response | Minimal to no human intervention required |
| Performance (from case studies) | Underperforms in environments with fluctuating workloads | Demonstrated higher accuracy and adaptability using DL models in dynamic scenarios |
| Resilience and Downtime | Lower resilience, higher downtime due to delayed or missed detection | Increased resilience, reduced downtime through proactive and predictive maintenance |
| Example Case Study Finding | Static thresholding underperformed | DL models outperformed static methods in cloud-hosted applications with dynamic workloads |

## 3. AI-Based Fault Detection Techniques For Fault Detection

Cloud systems that can repair themselves rely heavily on AI for tasks like issue detection and diagnostics. For the purpose of identifying and analysing network problems, they employ a wide range of supervised and unsupervised learning methods [15]. The kinds and characteristics that need to be detected, as well as the algorithms that can handle them, dictate the choice of algorithm. By training on labelled datasets that contain past fault data, supervised learning algorithms may accurately classify known faults. On the other hand, anomalies in unlabelled data can be more easily discovered using unsupervised learning models. In order to find the best method for fault detection, these models are usually tested using cross-validation. A basic framework for fault detection using ML is shown in Figure 1.

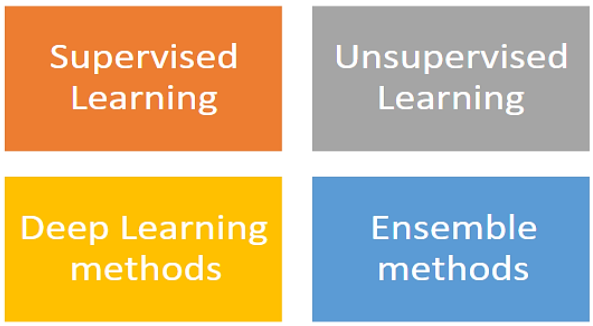


**Figure 1: AI-based Methods for Fault Detection for Self-Healing**

Here are the fault detection techniques based-AI are given below.

### *3.1. Supervised Learning*

One typical use case for supervised learning algorithms in self-healing networks is defect detection. These algorithms undergo training on labelled datasets that comprise instances of both typical and unusual network behaviour. Supervised learning models can use observable features to learn from past data and indicate if a network state is normal or abnormal. Common supervised learning algorithms employed by self-healing networks for fault detection include SVMs, DT, and RF. Example: SVMs divide typical from unusual network data points according to their distance from a hyperplane.

### *3.2. Unsupervised Learning*

The unsupervised learning methods are suitable for detecting the anomaly in the self-healing network since labelled data is not required. These methods process the fundamental structure of network telemetry data, to extract patterns that are suggestive

of a fault. Such technique can be helpful in case of identifying new errors or abnormalities in the behaviour of the system in complex cloud-based systems. Clustering algorithms, including K-means and DBSCAN, merge similar data points which form the data, and any outliers are potential abnormal data. Anomaly detection algorithms such as Isolation Forest and One-Class SVM allow finding data points that deviate greatly compared to the norm. Self-healing networks do not need any prior labels to find new or lurking issues due to the unsupervised learning strategies.

### *3.3. Deep Learning*

The capacity of DL methods, especially neural networks, to detect intricate patterns in data with many dimensions has brought them widespread attention. For self-healing network defect detection, CNNs and RNNs are often utilised. Problems like image-based fault identification in network topologies are well-suited to CNNs because of their superiority in analysing spatial correlations in network data. In contrast, RNNs excel at processing sequential data, which makes them great at spotting temporal outliers in network traffic patterns. By leveraging DL models, self-healing networks can detect subtle and complex faults that may evade traditional techniques.

### *3.4. Ensemble Methods*

To enhance the effectiveness of self-healing networks in fault detection, ensemble learning approaches include numerous base learners. Common ensemble strategies for combining model predictions include bagging, boosting, and stacking. The accuracy and resilience of fault detection in self-healing networks can be improved by the use of ensemble approaches, which combine the benefits of multiple algorithms. These techniques mitigate the risk of overfitting and improve generalization performance, enabling self-healing networks to adapt to diverse and dynamic network environments.

## 4. Role of AI and Machine Learning For Fault Management In Cloud

Cloud fault management relies heavily on AI and ML for automated, real-time failure identification and prediction. To detect outliers and prevent breakdowns before they happen, AI/ML models sift through mountains of data, in contrast to more conventional approaches [16]. This preventative method boosts operating efficiency, decreases downtime, and increases system reliability. AI and ML aid in the upkeep of stable cloud environments by continuously learning from both historical and real-time data, guaranteeing peak performance and an effortless user experience.

### *4.1. Artificial Intelligence and Machine Learning in Cloud Services*

The capacity of AI and ML to transform how the cloud is managed and exploited using smart automation, resource adaptation, and foresightful fault control is exciting and altering. Scalable and heterogeneous environments in cloud computing carry dynamic workloads which are not effectively monitored using conventional rule-based tools due to the possibility of missing more multifaceted types of failure [17]. AI and ML address this with the help of learning from the past logs, performance results, and machine behaviour. They then monitor anomalies, classify failures, and initiate self-healing actions based on the situation. Supervised learning models can predict hardware failure based on sensor data, while unsupervised models, such as clustering and autoencoders, can identify previously unseen anomalies in system logging [18]. Reinforcement learning can further enhance resilience in cloud services by optimizing recovery through feedback loops that modify approaches based on mitigation. These advantages put cloud providers in a position to improve service level agreements (SLAs) and incidents of downtime, and either stabilize operations of better respond to incidents of incident management., By having and AI/ML at different levels of the cloud computing stack from Virtual machine orchestration to application monitoring cloud service providers can maximize autonomy while operating efficiently, reliably, and ultimately having a cost-effective business model.

### *4.2. Challenges of AI and ML in Cloud Security*

Incorporating AI and ML into cloud security has the potential to greatly improve capabilities, but businesses must overcome certain obstacles:

- **Data Privacy and Security:** A lot of data is needed to train and run ML and AI systems, which makes people worried about the security and privacy of their data [19]. The International Association for Protection Professionals (IAPP) found that when it comes to AI solutions, 67% of organisations have trouble guaranteeing data protection.
- **Model Accuracy and Bias:** The quality of the training data determines how effective AI and ML models are. Using biased or inadequate training data causes 70% of AI models to exhibit biases, according to a study from the MIT Media Lab. Improper data quality can cause security judgments to be biased or threat detection to be erroneous.
- **Resource and Cost Constraints:** Large amounts of computing power and expert knowledge are needed to implement ML and AI systems. The expenses of creating and maintaining AI systems can be as much as 30% more than those of more conventional security solutions, according to research from McKinsey and Company.
- **Evolving Threat Landscape:** Cyber threats are ever-changing, which is a constant struggle for AI and ML systems. To keep up with emerging threats, AI models need regular updates. According to IBM research, 65 per cent of organisations need help updating their AI models to account for new threats.

## 5. AI and ML Approaches for Self-Healing In Cloud Services

The concepts of AI and ML in self-healing cloud services are complex algorithms that monitor and correct malfunctions in a system without human input. By means of the supervised, unsupervised, and DL techniques, the strategies comprise the use of operational data analysis to identify anomalies and manage corrective measures. Self-healing technologies may help cloud systems provide higher resilience, shorter downtime, and more balanced resource distribution, resulting in an overall increase in service availability and the formation of more efficient operational routines.

### *5.1. Self-Healing Architecture and Mechanisms*

A material is considered self-healing if it can fix its own defects and regain its original properties with the help of its own internal or external resources. Unlike regular polymers, those with self-healing capabilities can fix damage by transforming mechanical stress into chemical or physical processes [20]. These materials can greatly increase the durability of synthetics and have unique properties for each of their uses. The term "self-healing material" refers to man-made substances that can mend themselves, either once or repeatedly, protecting against material degradation while simultaneously increasing efficiency, strength, and dependability [21]. Inspired by the biological ability of living organisms to recover from injuries, such as skin regeneration or bone repair, these materials aim to mimic nature's resilience. While the concept of incorporating biological healing mechanisms into engineering has existed for nearly a century, research in this domain remains in its early stages.

### *5.2. Reinforcement Learning for Autonomous Recovery*

A two-stage system that can achieve self-adaptation by learning rules based on objectives offline and changing rules based on real-time environmental data and user goals online. A combination of reinforcement learning and using Rules has been developed and assessed using reasoning. The quantity of CPU allocation and the number of requests per unit are two dynamic features that fluctuate during runtime and are taken into account. Configuration properties, in accordance with dynamic properties, can take on varying values and transition between states. In this test, the RUBiS simulator is used. [22]. Applied reinforcement learning based on response time to an online bookstore. A new method for hybrid decision-making is introduced by integrating deep reinforcement learning into the ROS Hybrid Behaviour Planner (RHBP), which employs policy-based reinforcement learning. Using an artificial neural network to store the acquired knowledge, this method employs a specific kind of reinforcement learning. Previous papers used value functions in reinforcement learning, but this paper uses a neural network. It has been checked for workload evaluation in the RUBiS simulator.

### *5.3. Cloud Computing and Infrastructure*

Access to a shared pool of elastic computing resources, such as servers, storage, and networking, can be quickly and efficiently offered and released with no administrative effort when customers use cloud computing. New cloud infrastructures are substantially complex, usually stacks of virtualized and containerized environments, distributed microservices and scaling aspects that could produce unforeseen reliability and performance issues, particularly in the event of failure or resource bottlenecks [23]. AI and ML were introduced to support these challenges with intelligent monitoring and action suggestions from a top-down perspective, or autonomous fault detection at the infrastructure level and remediation at the systems level or application level. For instance, ML models could analyze telemetry logs from virtual machines and containers to identify evidence of resource exhaustion or failure in real-time, enabling corrective scaling or healing actions or through a subordinate level of the remediation process. Cloud infrastructures can achieve self-healing operational efficiencies and service resilience through the use of virtualization and abstraction components, (As shown in Figure 2). Cloud Computing Model, the cloud architecture spans three major layers: Application, Platform, and Infrastructure, each of which presents distinct opportunities for embedding AI-based self-healing modules.

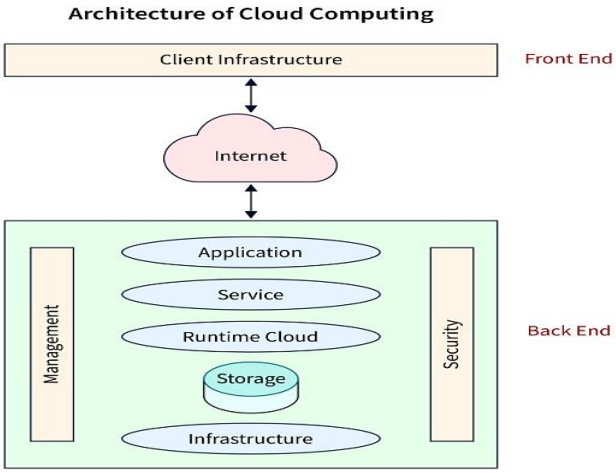


**Figure 2: Cloud Computing Architecture**

Modern data centres utilized for cloud computing are in fact intricate complexes, comprising tens of thousands of storage and memory devices. Many things can affect how stable cloud environments are. System reboots or failures are among these, and they can affect the memory or power provisioning infrastructure (such overhead power boards) that services rely on. People should still take precautions to make sure cloud systems can handle any future mistakes, even when fault detectors and service recovery mechanisms can be applied to each of the following probable failure sources.

## 6. Literature Review

The given section is a literature review of cloud computing defects and self-healing systems with a specific focus on the topic of utilizing ML and AI algorithms. The brief description of reviews of the studies has been presented in Table II. Porch et al. (2020) focus their efforts on utilizing supervised ML to detect symptoms of faults and determine their causes. By analysing the reference signal received power (RSRP) that users report during a given time frame, their technique can identify base station operational issues. Spotted certain problems with the base station lead to noticeable changes in RSRP readings and patterns of electromagnetic radiation in the area. In addition to allowing for the investigation of defects and the prevention of needless fault alerts, the framework's construction allowed for the differentiation of normal and non-normal operations in response to changing environmental conditions. The framework uses supervised ML to categorise the defect found once an abnormal operation is discovered [24].

Chen et al. (2020) offer a novel defect detection approach, which is based on active learning. This algorithm will provide excellent diagnosis performance with extremely few labelled training cases and this will save a lot of money. The major concept is selecting the most suitable unlabelled data to train and label. The final method of selecting them is uncertainty sampling. Experimental results showed that using the proposed methodology, many fewer labelled training cases would enable the same diagnosis accuracy compared to existing non-active methods, based on an LTE network fault model database. The proposed methodology is out to outperform the state-of-the-art in three other criteria as well on the same labelled instances figures [25]. Shetty and Sarojadevi (2020) This system employs an ML method with the goal of optimizing the utilization of cloud computing resources. Methods for scheduling tasks can model both static and dynamic scenarios. In order for this system to function, the conditions must be dynamic and ever-changing. They suggest using a machine learning method to schedule incoming tasks in a way that takes makespan, QoS, energy usage, execution time, and load balancing into account. This enables classification of the most suitable algorithm to use with every task request as opposed to random assignment of a scheduling algorithm [26].

Tamashiro et al. (2020) IaaS architecture of cloud computing and fault modelling of simulators based on it are outlined. As the number of individuals seeking means to conduct business on the Internet continues to grow, cloud computing, online data storage, and the dispatch of online services have shown stratospheric increases in their popularity over the past couple of years. Due to these evolutions an increased importance is being placed on service fault tolerance, hardware-based or software-based. This is why it becomes so important that cloud environments place recognized and mitigating plans in such a manner that they create detection and response plans. Therefore, hypothetical vulnerability in cloud computing architecture was examined through cloud environment simulators in this study and the means of identifying and remedying the vulnerabilities were tested. Critical cloud administrators' most common problems and fault tolerance strategies formed the basis of this research [27].

Liang et al. (2019) a methodology for recognizing grey areas which is based on modelling the application scenarios. The technique is capable of automatically evaluating application-to-application performance interference and developing a model of the relation between application-to-application performance interference and grey faults in different application contexts. Lastly, it determines the faulty node by itself through relational model that can notice the changes of the environment that then disrupt the performance. The effectiveness and accuracy of the proposed approach are confirmed by data received through the Google cluster and the virtual storage cluster environment based on Docker. Moreover, the method is very accurate and can recognize a grey issue in only 6.4 seconds [28].

Joseph and Mukesh (2019) identified security aspects that can be adopted on the cloud-based infrastructure security as service (IaaS), tested an attack model that captured a virtual machine snapshots, and performed analysis by using supervised ML techniques. In order to distinguish between virtual machines that have been attacked and those that have not, supervised and unsupervised ML algorithms are fed the sequences of API calls from the memory snapshots of the affected machines. By feeding the self-healing algorithm the collected collection of memory snapshots from the compromised virtual machines, it is possible to restore their functioning [29]. Ghahremani and Giese (2019) There is no doubt that using simulators is the gold standard for assessing SHS performance. In order to evaluate the present status of practice for simulating SHS performance, a thorough literature analysis was carried out to determine what realistic fault injection scenarios are required. In this paper, they lay out the current state of affairs and argue that SHS performance evaluations need to be more meticulous and comprehensive [30].

The Table II summarises findings of the literature review outlining focus of each study, its methodology, and main findings. Challenges and directions that the future holds.

**Table 2: Summary of Literature on AI and Machine Learning Fault Detection**

| Reference | Study On | Approach | Key Findings | Challenges | Future Directions |
|---|---|---|---|---|---|
| Porch et al., (2020) | Fault detection in base stations | Supervised ML using RSRP signal data | Faults alter RSRP readings and EM patterns; framework distinguishes normal/abnormal to reduce false alarms | Differentiating faults under changing environments | Extend framework to include more signal features; adaptive learning for dynamic environments |
| Chen et al. (2020) | Active learning for fault diagnosis | Uncertainty sampling with active learning | Achieved 99% accuracy with fewer labels (66 vs 557); better performance with limited data | Selection of valuable data remains non-trivial | Explore other sampling strategies; real-time adaptation in operational networks |
| Shetty et.al. (2020) | Cloud task scheduling optimization | ML-based algorithm selection for dynamic scheduling | Efficient task allocation improves QoS, energy use, and execution time | Real-time prediction accuracy and scalability | Integration with edge computing and IoT platforms |
| Tamashiro et al. (2020) | Fault modelling in IaaS-based cloud simulators | Analyzed common failures and fault-tolerance techniques used in cloud management platforms; evaluated them using simulation frameworks | Cloud services must be designed with integrated failure detection and mitigation to improve fault tolerance in IaaS environments | Limited real-time applicability; simulation accuracy and scope may not fully represent real-world complexities | Enhance simulator fidelity; integrate real-time monitoring tools; apply AI/ML-based predictive fault modelling |
| Liang et al. (2019) | Grey failure detection using performance interference modelling | Developed an automatic detection model based on performance interference analysis in application scenarios; validated using Docker and Google cluster datasets | Grey faults can be detected within 6.4 seconds with high accuracy using performance relationship models | Difficulty in modelling complex real-world application interferences; scalability to large systems | Extend model to diverse cloud workloads and environments; incorporate DL to improve detection adaptability |
| Joseph et.al. (2019) | Self-healing security mechanism in private IaaS clouds | Used memory snapshot API call sequences with supervised and unsupervised ML for attack detection; implemented a self-healing algorithm for recovery | Effective in classifying attacked vs. non-attacked VMs and initiating automated recovery | Dependence on the quality of memory snapshot data; potential latency in recovery | Enhance learning models with larger datasets; reduce healing time; extend to hybrid/multi-cloud environments |
| Ghahremani et.al., (2019) | Performance evaluation of self-healing systems | Systematic literature review + simulation | Simulators widely used; evaluation often lacks rigour | Lack of realistic fault injection scenarios and metrics | Develop standardized benchmarks and testbeds for SHS |

## 7. Conclusion and Future Scope

In conclusion, cloud computing's quick development has changed the face of IT infrastructure by making resources available on demand and in scalable ways. Nevertheless, self-healing systems and problem detection have been significantly hindered by this transition. A holistic strategy combining conventional fault management methods with cutting-edge AI and ML approaches is necessary to guarantee cloud environments' high availability and reliability. Traditional techniques of monitoring frequently fall short when faced with the intricacies of contemporary cloud systems, which is a growing concern for organisations that depend heavily on cloud services. Anomaly detection and recovery automation rely on techniques like supervised and unsupervised learning,

DL, and ensemble methods. Keeping operational efficiency and downtime to a minimum requires the adoption of self-healing systems. These systems respond autonomously to recognised problems. Problems like as obtaining accurate models, obtaining high-quality training data, and allocating sufficient computing resources for real-time analysis remain despite progress in AI/ML applications. Future research should be directed towards an increased flexibility of AI/ML models to be used in a dynamic cloud environment, the framework of implementing self-healing portability, across a wide variety of cloud environments, and discuss the promise of reinforcement learning to optimize recovery strategies on a case-by-case basis.

## References


1. S. Garg, "Predictive Analytics and Auto Remediation using Artificial Intelligence and Machine learning in Cloud Computing Operations," *Int. J. Innov. Res. Eng. Multidiscip. Phys. Sci.*, vol. 7, no. 2, 2019.
2. M. Kaur and H. Singh, "A review of cloud computing security issues," *Int. J. Grid Distrib. Comput.*, vol. 8, no. 5, pp. 215–222, 2015, doi: 10.14257/ijgdc.2015.8.5.21.
3. M. K. Gokhroo, M. C. Govil, and E. S. Pilli, "Detecting and mitigating faults in cloud computing environment," in *2017 3rd International Conference on Computational Intelligence and Communication Technology (CICT)*, IEEE, Feb. 2017, pp. 1–9. doi: 10.1109/CIACT.2017.7977362.
4. S. Gupta and S. Prakash, "QoS and load balancing in cloud computing-an access for performance enhancement using agent-based software," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 11, pp. 641–644, 2019.
5. F. R. Salmasi, "A Self-Healing Induction Motor Drive With Model Free Sensor Tampering and Sensor Fault Detection, Isolation, and Compensation," *IEEE Trans. Ind. Electron.*, vol. 64, no. 8, pp. 6105–6115, 2017, doi: 10.1109/TIE.2017.2682035.
6. J. Jiao, M. Zhao, J. Lin, and K. Liang, "A comprehensive review on convolutional neural network in machine fault diagnosis," *Neurocomputing*, vol. 417, pp. 36–63, 2020, doi: 10.1016/j.neucom.2020.07.088.
7. S. S. S. Neeli, "Real-Time Data Management with In-Memory Databases: A Performance-Centric Approach," *J. Adv. Dev. Res.*, vol. 11, no. 2, p. 8, 2020.
8. W. Wang, N. G. Moreau, Y. Yuan, P. R. Race, and W. Pang, "Towards machine learning approaches for predicting the self-healing efficiency of materials," *Comput. Mater. Sci.*, vol. 168, no. February, pp. 180–187, 2019, doi: 10.1016/j.commatsci.2019.05.050.
9. V. Kalra and R. K. Sahu, "A Review on Fault Detection in WSNs," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 7, no. 4, pp. 63–67, 2017, doi: 10.23956/ijarcsse/v7i3/0140.
10. Z. Amin, H. Singh, and N. Sethi, "Review on Fault Tolerance Techniques in Cloud Computing," *Int. J. Comput. Appl.*, vol. 116, no. 18, pp. 11–17, 2015, doi: 10.5120/20435-2768.
11. D. Miljković, "Fault detection methods: A literature survey.," 2016.
12. D. Ghosh, R. Sharman, H. Raghav Rao, and S. Upadhyaya, "Self-healing systems - survey and synthesis," *Decis. Support Syst.*, vol. 42, no. 4, pp. 2164–2185, 2007, doi: 10.1016/j.dss.2006.06.011.
13. M. R. Mesbahi, A. M. Rahmani, and M. Hosseinzadeh, "Reliability and high availability in cloud computing environments: a reference roadmap," *Human-centric Comput. Inf. Sci.*, vol. 8, no. 1, 2018, doi: 10.1186/s13673-018-0143-8.
14. R. Chalapathy and S. Chawla, "Deep Learning for Anomaly Detection: A Survey," 2019, doi: 10.48550/arXiv.1901.03407.
15. P. Ravikumar, "Self-Healing Networks : An AI Approach to Network Fault Management," vol. 1, no. 2, pp. 1–10, 2015.
16. Y. Zhao, T. Li, X. Zhang, and C. Zhang, "Artificial intelligence-based fault detection and diagnosis methods for building energy systems: Advantages, challenges and the future," *Renew. Sustain. Energy Rev.*, vol. 109, pp. 85–101, 2019, doi: https://doi.org/10.1016/j.rser.2019.04.021.
17. A. Xenakis, A. Karageorgos, E. Lallas, A. E. Chis, and H. González-Vélez, "Towards distributed IoT/Cloud-based fault detection and maintenance in industrial automation," *Procedia Comput. Sci.*, vol. 151, no. 2018, pp. 683–690, 2019, doi: 10.1016/j.procs.2019.04.091.
18. U. A. Butt *et al.*, "A Review of Machine Learning Algorithms for Cloud Computing Security," *Electronics*, vol. 9, no. 9, 2020, doi: 10.3390/electronics9091379.
19. H. A. S. Ahmed, M. H. Ali, L. M. Kadhum, M. F. Bin Zolkipli, and Y. A. Alsariera, "A review of challenges and security risks of cloud computing," *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 1–2, pp. 87–91, 2017.
20. M. D. Almutairi, A. I. Aria, V. K. Thakur, and M. A. Khan, "Self-healing mechanisms for 3D-printed polymeric structures: From lab to reality," *Polymers (Basel).*, vol. 12, no. 7, pp. 1–27, 2020, doi: 10.3390/polym12071534.
21. G. I. Kadhom and A. M. Jaafar, "Semi-alive architecture 'from healing to self-healing in architecture,'" *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 881, no. 1, 2020, doi: 10.1088/1757-899X/881/1/012015.
22. T. Zhao, W. Zhang, H. Zhao, and Z. Jin, "A Reinforcement Learning-Based Framework for the Generation and Evolution of Adaptation Rules," in *2017 IEEE International Conference on Autonomic Computing (ICAC)*, 2017, pp. 103–112. doi: 10.1109/ICAC.2017.47.
23. A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of Cloud computing and Internet of Things: A survey,"

*Futur. Gener. Comput. Syst.*, vol. 56, pp. 684–700, 2016, doi: https://doi.org/10.1016/j.future.2015.09.021.

24. J. B. Porch, C. H. Foh, H. Farooq, and A. Imran, "Machine Learning Approach for Automatic Fault Detection and Diagnosis in Cellular Networks," in *2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2020, pp. 1–5. doi: 10.1109/BlackSeaCom48709.2020.9234962.
25. M. Chen, K. Zhu, R. Wang, and D. Niyato, "Active Learning-Based Fault Diagnosis in Self-Organizing Cellular Networks," *IEEE Commun. Lett.*, vol. 24, no. 8, pp. 1734–1737, 2020, doi: 10.1109/LCOMM.2020.2991449.
26. C. Shetty and H. Sarojadevi, "Framework for Task Scheduling in Cloud using Machine Learning Techniques," in *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*, 2020, pp. 727–731. doi: 10.1109/ICISC47916.2020.9171141.
27. C. B. O. Tamashiro, R. Spolon, R. S. Lobato, A. M. Junior, and M. A. Cavenaghi, "Modelagem de falhas em nuvem Cloud Fault Modeling," in *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, IEEE, Jun. 2020, pp. 1–6. doi: 10.23919/CISTI49556.2020.9141061.
28. B. Liang, N. Chen, Y. Xie, and Y. Chen, "Grey Fault Detection Method Based on Application Interference Model in Cloud Storage," in *2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*, IEEE, Aug. 2019, pp. 36–43. doi: 10.1109/SmartIoT.2019.00015.
29. L. Joseph and R. Mukesh, "To Detect Malware attacks for an Autonomic Self-Heal Approach of Virtual Machines in Cloud Computing," in *2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, IEEE, Mar. 2019, pp. 220–231. doi: 10.1109/ICONSTEM.2019.8918909.
30. S. Ghahremani and H. Giese, "Performance Evaluation for Self-Healing Systems: Current Practice and Open Issues," in *2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, 2019, pp. 116–119. doi: 10.1109/FAS-W.2019.00039.
31. Kalla, D., and Samiuddin, V. (2020). Chatbot for medical treatment using NLTK Lib. *IOSR J. Comput. Eng*, *22*, 12.
32. Kuraku, S., and Kalla, D. (2020). Emotet malware a banking credentials stealer. *Iosr J. Comput. Eng*, *22*, 31-41
33. Sreejith Sreekandan Nair, Govindarajan Lakshmikanthan (2020). Beyond VPNs: Advanced Security Strategies for the Remote Work Revolution. International Journal of Multidisciplinary Research in Science, Engineering and Technology 3 (5):1283-1294.