An Approach for Fault Tolerance in Cloud Computing Using Soft Computing Technique

[1]Neeraj Prakash Shrivastava, [2] Dr R. K. Srivastava

[1]Research Scholar, Mewar University, Chittorgarh , Rajasthan
[2] Professor & Head, Deptt. Of Computer Science, Dean (Engg & Technology), Dr. S.M. National Rehabilitation University, Lucknow, India

Abstract

Cloud computing provides IT services on a pay-as-you-go basis to customers globally. Cloud computing is a term that refers to network-based activities that seem to be delivered by physical server hardware but are really delivered by virtualization that is emulated by software operating on one or more physical computers. These networked-based applications and technologies are all prone to failure to varying degrees. To ensure robustness and reliability in cloud computing platform, failures must be appropriately identified and addressed. Proactive fault tolerance approaches have been employed in this research, and we suggest the usage of a Nave Bayes classifier to categorise nodes into those that are likely to fail and those that are not, and then use proactive fault tolerance strategies. Reliability is obtained by the employment of the Nave Bayes classifier in conjunction with fault tolerance strategies.

Keywords – Naïve Bayes, Fault Tolerance, Faulty node, Proactive technique

1. Introduction

1.1 Fault Tolerance in Cloud Computing

Fault tolerance in cloud computing is about creating a plan for resuming work when a few components fail or become unavailable This enables organisations to assess their infrastructure requirements and demands as well as deliver services when related equipment are inaccessible due to a variety of factors This does not imply that the alternative arrangement can offer 100% of the required service but it does mean that the system remains operational at a usable and most crucially a manageable level This is critical if organisations are to continue developing and increasing their production levels.

1.2 Cloud computing

The word "Cloud" originated from a system architecture in use by network engineers to indicate the location and interconnection of different network devices. This network architecture was shaped as a cloud.

While software fault tolerance attempts to apply physical fault tolerance knowledge to a different problem, it necessitates architectural variety in order to develop redundancy systems correctly.

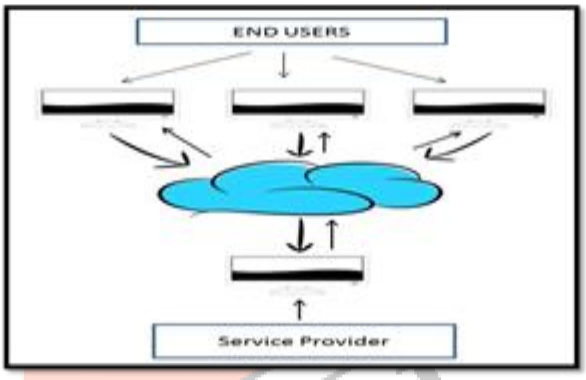


Figure 1 cloud computing

Cloud computing is described as the online storage and access of information and computer services. It does not save any data to your computer. It is the provision of computer services such as server, storage systems, connectivity, and databases on a demand basis. The primary aim of this strategy is to provide many people with access to computer servers. Additionally, users may access data stored on a remote server.

Software fault tolerance relates to a project's ability to detect and recover from hardware and software problems in the system on which it is running in order to keep delivering service in accordance with the requirements. For the next generation of technologically accessible and reliable computing systems, spanning from micro to data warehousing, software defect tolerance is necessary. However, being able to tolerate software defects is not really a solution, and it is vital to recognise that it is only one part of the next generation of systems.

To comprehend software fault tolerance effectively, it is necessary to grasp the source of the situation that software defect tolerance is intended to address. All software flaws are design flaws. The manufacture of software, or the replication of software, is believed to be flawless. The issue being mainly due to design flaws is very unlike practically any other systems in which high availability is a desirable quality. This intrinsic issue, that software defects are caused by human mistake in accurately understanding a requirement or executing an algorithm, generates complications that must be addressed in the basic approach to computer fault tolerance.

Software fault tolerance techniques now in use are based on classic mechanical fault - tolerant. The shortcoming of this method is that conventional hardware failures tolerance was built mainly to overcome manufacturing flaws, with environment and other issues coming in second. Design variety was not a notion applied to hardware fault - tolerant solutions, and so N-Way redundancy systems fixed several single failures by reproducing the same equipment. While software fault tolerance attempts to harness physical fault tolerance knowledge to tackle a different issue, it generates a need for architectural variety in order to correctly develop redundancy systems.

Design diversity contributes to computer automatic failover only to the extent that it is feasible to generate varied and comparable specifications that enable programmers to construct software with enough distinct designs to avoid sharing identical failure modes. Diversification of design and self-contained failure modes has been demonstrated to be a pretty difficult topic. The difficulty is that when confronted with a complicated problem, the requirement for people to handle it correctly is insurmountable.

1.3. Main Concepts behind Fault Tolerance in Cloud Computing System

- **Replication**

The fault-tolerant system operates on the principle of maintaining multiple replicas of each services. Therefore, if a component of the system fails, it has other versions that may be used to keep it going. Consider a database clusters that consists of three servers that all have the same data. Each of them has a record of all the operations performed on it, such as data input, changes, and removal. The redundant servers will remain dormant until and until a fault tolerance system requires their availability.

- **Redundancy**

When a system component breaks or enters a degraded condition, it is critical to have alternative solutions in place. For instance, a website software that uses MS SQL as its database may have intermittent failures owing to a hardware failure. Then, when the primary database goes down, a new one must be made available under the redundancy idea. The server is connected to an emergency database that contains numerous redundant services.

1.4. Techniques for Fault Tolerance in Cloud Computing

When developing a fault tolerance system, all services must be prioritised. The database must be given extra consideration due to the fact that it powers multiple other components.

While software fault tolerance attempts to apply physical fault tolerance knowledge to a different problem, it necessitates architectural variety in order to develop redundancy systems correctly.

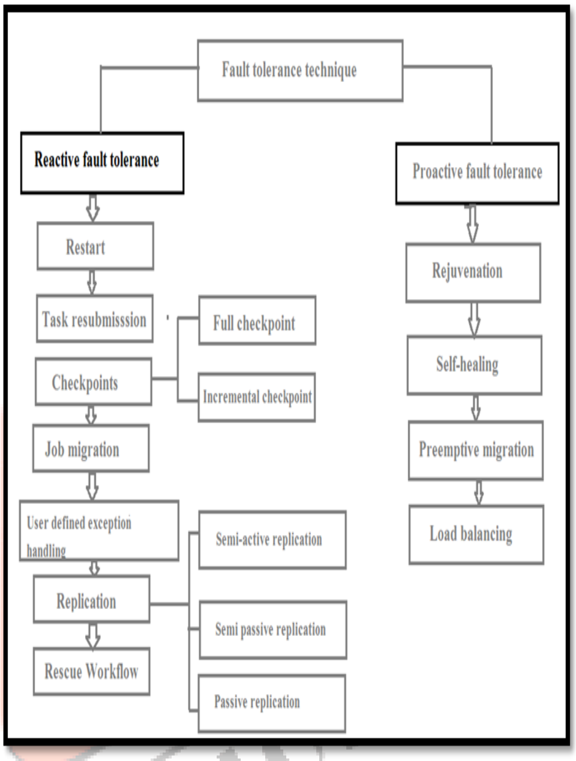


Figure 2 techniques of fault tolerance

Once the company has established its priorities, it must start work just on practice session. Consider the enterprise's forum website, which allows people to register and submit comments. When authentication services are unavailable due to an issue, users will be unable to sign in. The forum then becomes read-only and no longer serves its intended function. However, with fault - tolerance solutions, repair is assured, and the customer may do searches with minimum damage.

1.5. Major Attributes of Fault Tolerance in Cloud Computing

- **None Point Failure:** The principles of duplication and replicate establish that fault tolerance is possible, although at the expense of certain minor consequences. If a system does not have even a design flaw, it is not fault tolerant.

- **Accept the Fault Isolation Concept:** The occurrence of a fault must be handled independently of other processes. This assists the organisation in isolating itself from a previously failed system.

1.6. Existence of Fault Tolerance in Cloud Computing

- **System Failure:** This might be a software or hardware problem. The software failure can result in a computer crash or a hanging state, which might be caused by a stack overflow or by another cause. Any neglect of the specific hardware devices will result in the breakdown of the hardware system.

- **Security Breach Occurrences:** Fault tolerance may arise for a variety of causes, including security breaches. The server being hacked has a detrimental effect on it and resulting in a data leak. Other causes for the requirement of fault tolerance include ransomware, phishing, and virus attacks.

1.7 Soft computing technique

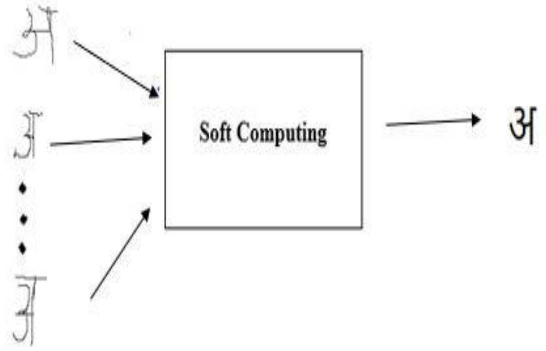
Soft computing is an approach where we compute solutions to the existing complex problems, where output results are imprecise or fuzzy in nature, one of the most important features of soft computing is it should be adaptive so that any change in environment does not affect the present process. The following are the characteristics of soft computing.

- ✓ It does not require any mathematical modeling for solving any given problem
- ✓ It gives different solutions when we solve a problem of one input from time to tim
- ✓ Uses some biologically inspired methodologies such as genetics, evolution, particles swarming, the human nervous system, etc.
- ✓ Adaptive in nature.

1.7.1 Three types of soft computing techniques

1.7.1.1 Artificial Neural Network

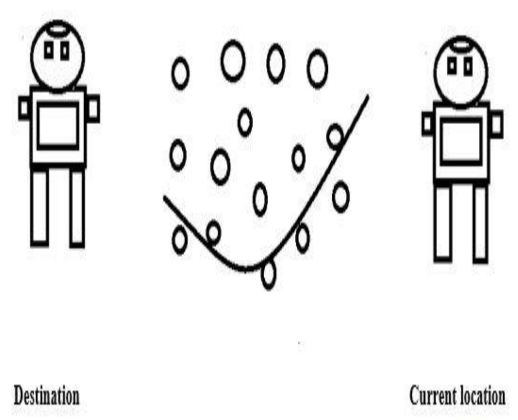
It is a parallel distributed network with connectionist modelling. There are two types: ANN (Artificial Neural Network) and BNN (Biological Neural Network) (Biological Neural Network). A unit is a neural network that processes a single element. The unit's components are input, weight, processing element, and output. It is analogous to our human neural system. The main advantage is that they solve problems in parallel; artificial neural networks communicate using electrical signals. The main disadvantage is that they are not fault-tolerant, which means that if one of the artificial neurons is damaged, it will no longer function.



Soft – computing

1.7.1.2 Fuzzy Logic

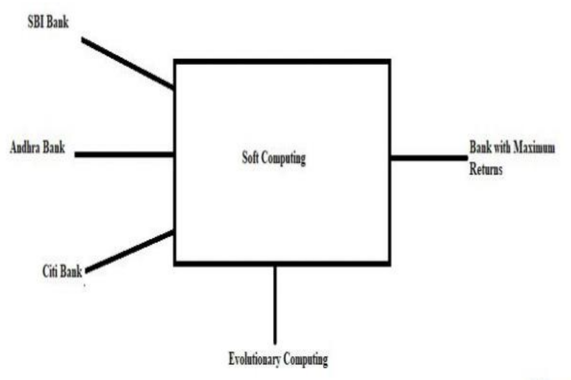
The fuzzy logic algorithm is used to solve models based on logical reasoning that are imprecise or vague. Latzi A. Zadeh first used it in 1965. With the closed interval [0,1], fuzzy logic provides a specified truth value. Where 0 denotes a false value and 1 denotes a true value.



fuzzy – logic

1.7.1.3 Genetic Algorithm in Soft Computing

The genetic algorithm was introduced by Prof. John Holland in 1965. It is used to solve problems based on principles of natural selection, that come under evolutionary algorithm. They are usually used for optimization problems like maximization and minimization of objective functions, which are of two types of an ant colony and swarm particle. It follows biological processes like genetics and evolution.



Genetic – algorithm

1.7.2 Applications For soft Computing

The following are the applications of soft computing

- ✓ Controls motors like induction motor, DC servo motor automatically
- ✓ Power plants can be controlled using an intelligent control system
- ✓ In image processing, the given input can be of any form, either image or video which be manipulated using soft computing to get an exact duplicate of the original image or video.
- ✓ In biomedical applications where it is closely related to biology and medicine, soft computing techniques can be used to solve biomedical problems like diagnosis, monitoring, treatment, and therapy.
- ✓ Smart instrumentation is trendy nowadays, where intelligent devices automatically communicate with other devices using a certain set of communication protocols to perform certain tasks, but the problem here is there is no proper standard protocol to communicate. This can be overcome by using soft computing techniques, where the smart devices are communicated over multiple protocols, with high privacy and robustness.

- **1.8. Scope of the study**

By maintaining your technologies online & guaranteeing they are well-designed, fault tolerant design helps avoid security breaches. A carelessly built system may be quickly brought down by an assault, resulting in the loss of data, revenue, and confidence for your company. The primary disadvantage occurs when a component's fault tolerance limits the performance of another component that is reliant on it. Every such failure will result in the manufacturing of poorer items and a long-term rise in expenses. To overcome this limitation, we must investigate fault tolerance strategies employing soft computing technique.

2. Literature review

2.1. Geeta, & Prakash, S. (2018)

The dynamic scalability and connectivity of hardware and software provided by cloud computing make it an easily adoptable technology. Many unforeseen errors and failures might occur due to the dynamic nature of the cloud. Tolerance for unexpected device or software failures is one of the key features of a fault-tolerant system. This section focuses on cloud computing dependability, fault tolerance, and standard of care. Cloud computing has been more popular in recent years due to its ability to dynamically acquire and release computer resources at a low cost and with no administrative efforts or network operator contact. Cloud computing, despite several improvements, is still vulnerable to system failures. Concerns about the stability and accessibility of Cloud computing technology are growing among members of the community. Dynamically provisioning capabilities in a computing environment enables cloud client applications to meet the casually fluctuating capacity and services needs. There has been a great deal of research on the role of Quality of Service (QoS) in the emotive resource allocation in the Cloud computing paradigm.

2.2. Amin, Z., Singh, H., & Sethi, N. (2015).

Using Artificial Neural Networks, this research proposes a fault detection technique that fills in the gaps left by previously developed algorithms while also providing a fault resistant model.

2.3 Qasem, G. M., & Madhu, B. K. (2018).

Cloud data centres confront two major issues when it comes to service level agreements (SLAs). These are stability and accessibility. For overlapping problems, this study provides a Fuzzy minimum maximum neural net Classification technique, which can forecast failure and fix it before it happens, Training and testing data sets are fed into each other using a suggested Fuzzy min-max neural technique, which uses the cutoff value as a guide. When compared to earlier methodologies, the results reveal an improved performance.

2.4 Kumari, P., & Kaur, P. (2021).

Fault tolerance-related concerns in cloud computing are examined in this study, focusing on the important principles, architectural features and the most current approaches and strategies. This report's goal is to provide light on current methods to fault tolerance as well as any obstacles that remain. There are many interesting strategies that may be employed to come facilitate effective solution, and the study points out some of the most pressing research questions in this field.

2.5 Dhingra, M., & Gupta, N. (2019).

To improve system reliability, this research presents an efficient adaptable fault tolerance technique using effective analysis architecture. This plan includes a decision process as well, in an effort to boost efficiency a tad above the usual. Finally, a fine - grain security checkpoint technique is used to reduce latency. According to the data analysed, the suggested method's execution time increases when compared to standard systems.

2.6. Amin, Z., Singh, H., & Sethi, N. (2015).
Pay-as-you-go models underpin cloud computing, which makes IT services available to anybody, anywhere. A cloud computing service is a service that seems to be delivered by a real database server, but in reality is usually served by software operating on one or more actual computers. Services and systems that are reliant on the internet are more or less susceptible to breakdowns. The failure should be recognised and dealt with efficiently in order to ensure robustness and reliability in cloud computing architecture. Various forms of fault - tolerant and fault tolerance solutions in the cloud computing system are discussed in this study.

2.7. Madhu, B. K. (2016).

It is necessary to enhance proactive fault prediction for virtual machines (VMs) in order to decrease downtime and deal with the expanding clouds. Fault Tolerance does not have a standard measure or interface. Because each solution has its own criteria for assessing virtual machine productivity and fitness, comparing and integrating them is almost impossible. Because different service providers are participating in a particular service, it is necessary to have a consistent measure for forecasting fault tolerance.

➢ **Objectives of the study**

- To analyse and assess different fault fault tolerance strategies in a cloud context;
- To investigate fault tolerance in cloud computing via the use of soft computing techniques.
- To achieve dependability via the use of the Nave Bayes classifier in conjunction with fault tolerance strategies.

3. Research Methodology

To accomplish the purpose of "research and analysis of different fault tolerance and damage detection approaches in cloud technology," a complete review of the literature on cloud computing and the various damage detection and fault - tolerant techniques utilised in cloud computing was conducted. A thorough analysis of the literature was conducted to determine the different types of neural networks that may be employed for defect identification.

Our suggested failure detector is based on the Naive Bayes Classification Algorithm, which use soft computing to calculate the predicted arrival time of a virtual machine.

4. Proposed Model

The proposed model comes within the category of Proactive Fault Management approaches. We may subject a cloud computing to several cycles of test and monitor and record which nodes fail throughout each round. Following that, we can apply prediction models, or more precisely, classification algorithms, to the data set of node failures, and also various factors including such load and infrastructural facilities that may affect the failure characteristics of a node, in order to predict and recognise the nodes that are most likely to fail based on the data collected. Once we have identified the fault-prone nodes, we may replace them, delete them, or establish a standby node that will automatically replace our defective node in the event of a failure, ensuring that the public cloud continues to operate without interruption. This paradigm enhances the cloud computing system's dependability and hence performs fault management flawlessly.

4.1. Classification algorithm:

We categorise our nodes using the Naive Bayes Classification Algorithm. The Naive Bayes classifier is among the most practical soft computing algorithms available. The Nave Bayes classifier is based on Bayes' theorem, which presupposes a high degree of independence (nave) between the qualities or features (predictors). A Naive Bayesian categorization model is especially advantageous for extremely big datasets since it is very easy to construct and does not involve complicated repetitious parameter setting or computation. We may use this approach to identify the nodes that are most likely to have a failure during a procedure. Thus, if we determine which node is likely to fail, we may use a proposed fault strategic plan to correct the problem. This whole operation will boost the system's dependability.

4.2. Implementation:

To simulate a real-world situation, we created many web pages (serving as servers or nodes) and then identified defective nodes among them using MATLAB simulation. We collected 15 web pages (nodes) and used the following methods to detect and correct problematic nodes.

4.3. Implementation tools:

- Python 3.x with jupyter notebook
- Dataset: In this project we use dataset from kaggle or UCI soft computing repository

4.4. Algorithms used:

- **Algorithm 1:**

Each server is composed of two sections of code; if the first section fails to function, the second section's code is called. The second section is responsible for generating the prime number. As a result, we can discover servers that return prime numbers and categorise them as broken using our technique. We analyse all potential combinations of our 15 nodes and, if a software failure occurs, it is displayed using the MATLAB technique.

Algorithm 2:

This approach is based on the hamming code notion. It operates as follows:

5. Proposed Cloud Virtualized System Architecture And Implementation

5.1 Cloud Virtualized System Architecture

There are currently a few techniques for autonomic fault tolerance in the cloud environment. Shelp can withstand software failures for server applications running in the cloud.Virtual machine environment There is a requirement to autonomic fault tolerance in the cloud enviorment. If one of the servers fails, the rest of the system will fail as well. User requests should be automatically redirected to the system.

a backup server As a result, the cloud virtualized system architecture has been proposed and put into action HAProxy is being used. The app's availability and Using the proposed method, reliability can be maintained. The cloud virtualized system architecture depicted in VMs that run Ubuntu The 10.04 operating system and database application are both active. Server 2 serves as a backup server in the event of a primary server failure. HAProxy is set up on the.

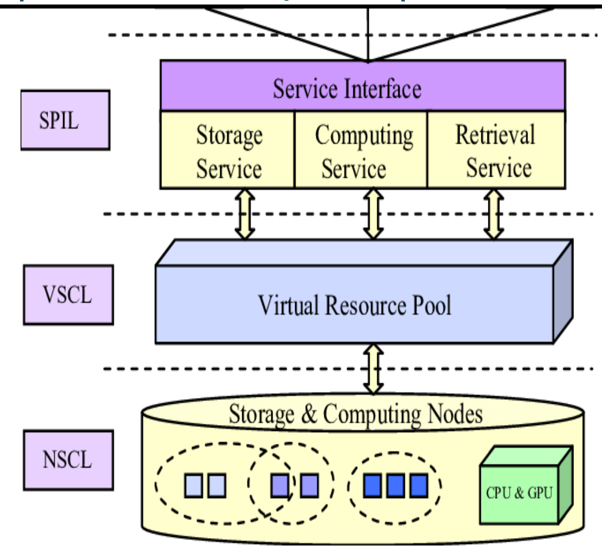


Fig: Cloud Virtualized System Architecture

5.2 Implementation and Experimental Results

HAProxy and MySQL were used to build a fault-tolerant system. In a fault-tolerant cloud environment, HAProxy is used to handle server failures. It provides a web-based statistics interface known as Statistics for HAProxy. Implementation consists of two steps. server 1 and server 2 are virtual machines that serve as web servers. hosting Apache Tomcat 6.0.32. HAProxy software version 1.3.15.2 is configured on third virtual machine. A simple database application written in Java is installed on the web servers. Xampp for Linux Windows XP (SP2) is used to install MySQL. Data in MySQL is replicated using replication technique for local backup. Replication enables data from one MySQL database server (the master) to be replicated to one or more MySQL database servers. Application can be accessed on any of the web server. Data consistency is also maintained through MySQL replication. The experimental results show that HAProxy can assist server applications to recover from server failures in just a few milliseconds with minimum performance overhead.

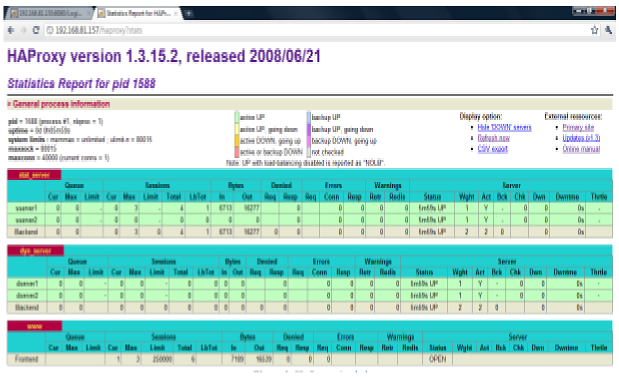


Fig HAProxy Statistics

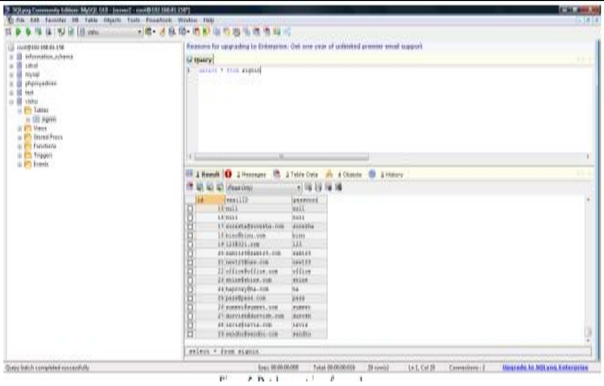


Fig Database entries of sever 1

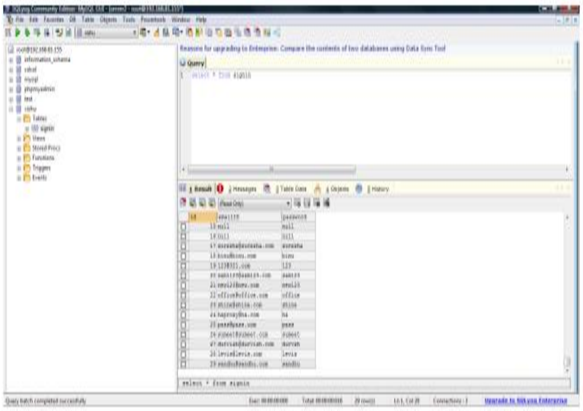


Fig Replicated Database entries of sever 2

Conclusion and future scope

There are several fault tolerance models available, each of which provides a unique method for enhancing the system's performance and dependability. Conventional fault-tolerance models have a number of disadvantages. The prospect of overcoming the shortcomings of all prior models and developing a more reliable model capable of reducing downtime and covering all areas of fault tolerance in cloud data centres is at the heart of this proposed study.

We used our suggested model in the proactive fault management area, predicting which nodes are most likely to have a problem using a Naive Bayes classification and then using fault tolerance strategies to assure the system's better dependability. We may improve the system's reliability by using our suggested model; correlation coefficients are used to

There evaluate the probability of failure of any component, in this instance a node. The MTBF factor is used to determine the system's reliability. We can minimise the number of node failures by roughly 87 percent using Nave Bayes. The proposed model shows superiority over conventional model.

is a possibility of utilising alternative categorization methods for the similar purpose, which can then be compared. Additionally, scalability, a QOS property, may be assessed in order to improve the services and minimise overhead.

References

1. Geeta, & Prakash, S. (2018). A review on quality of service in cloud computing. *Advances in Intelligent Systems and Computing*, *654*, 739–748. https://doi.org/10.1007/978-981-10-6620-7_72

2. Amin, Z., Singh, H., & Sethi, N. (2015). Review on Fault Tolerance Techniques in Cloud Computing. *International Journal of Computer Applications*, *116*(18), 11–17. https://doi.org/10.5120/20435-2768

3. Qasem, G. M., & Madhu, B. K. (2018). A Classification Approach for Proactive Fault Tolerance in Cloud Data Centers. *International Journal of Applied Engineering Research*, *13*(22), 15762–15765. http://www.ripublication.com

4. Kumari, P., & Kaur, P. (2021). A survey of fault tolerance in cloud computing. *Journal of King Saud University - Computer and Information Sciences*, *33*(10), 1159–1176. https://doi.org/10.1016/j.jksuci.2018.09.021

5. Dhingra, M., & Gupta, N. (2019). Reliability framework for cloud virtual nodes using fault tolerance techniques. *International Journal of Engineering and Advanced Technology*, *9*(1), 7586–7593. https://doi.org/10.35940/ijeat.B2502.109119

6. Amin, Z., Singh, H., & Sethi, N. (2015). Review on Fault Tolerance Techniques in Cloud Computing. *International Journal of Computer Applications*, *116*(18), 11–17. https://doi.org/10.5120/20435-2768

7. Madhu, B. K. (2016). *Proactive Fault Tolerance for Resilience Cloud Data Centers to Improve Performance Efficiency*. *4*(27), 1–5.