








Research Article

Artificial Intelligence Enabled Effective Fault Prediction Techniques in Cloud Computing Environment for Improving Resource Optimization

Junaid Hussain Abro ¹, Chunlin Li ¹, Muhammad Shafiq ¹, A. Vishnukumar ²,
Shivlal Mewada ³, Kamlesh Malpani ⁴ and Jonathan Osei-Owusu ⁵

¹School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan 430063, China

²School of Information Engineering, Qujing Normal University, Yunnan, China

³Dept. of Computer Science, Government College, Makdona (Vikram University), Ujjain, India

⁴Shri Vaishnav Institute of Management, Indore, India

⁵Department of Biological, Physical and Mathematical Sciences, University of Environment and Sustainable Development, Somanya, Ghana

Correspondence should be addressed to Jonathan Osei-Owusu; josei-owusu@uesd.edu.gh

Received 7 July 2022; Revised 1 August 2022; Accepted 9 August 2022; Published 22 September 2022

Academic Editor: Punit Gupta

Copyright © 2022 Junaid Hussain Abro et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The bulk of the fault tolerance techniques that are in use today laid their primary emphasis, in the event that a virtual machine fails, on the production of clones to replace it, rather than on the early prediction of the failure itself in advance. Several of the currently used techniques give migration priority over recovery in the event that a virtual machine (VM) fails. This is due to resource constraints and concerns with server availability. Examples of algorithms with a single objective include fault tolerance, migration prediction, and simply expecting failure. Another example is fault tolerance. In this research, we are aiming to determine the most effective strategy to transition from a system that is not operating well to one that does. It is essential to be able to predict the failure of a virtual machine in a timely manner due of issues such as squandered resources, energy, and cost. Since the beginning of cloud computing, there has been an issue with the dependability of virtual computers, often known as VMs. As an integral component of a fault tolerance system, preemptive measures are an absolute need in order to guarantee the continuation of service. As a consequence of this, it is vital to work toward enhancing and emphasizing the proactive failure prediction of virtual machines. The key motivations for this are decreased periods of downtime and enhanced scalability. A technique was utilized to transfer the resources that were predicted to fail from one virtual machine (VM) to another VM in a safe manner. Using the compression strategy reduced the amount of time needed to complete the migration, and resource utilization increased. This article provides artificial intelligence that enables effective fault prediction techniques in cloud computing to improve resource optimization.

1. Introduction

The method of using a network of networked and virtualized computers to provide on-demand access to a range of computer resources, including data storage and processing power, is called “cloud computing.” The word “cloud computing” refers to the activity. Other servers provide the same functionality as the master server, and each of these extra servers is connected to the master server. However, the

master server is the only server that may act as the master. The configuration of the cloud environment is subject to change and is decided by the needs of the organisation. Because of this, the deployment of services is based on the requirements of the company, which are then reflected in the configuration of the cloud environment. The cloud service may be set up in one of these three major configurations. Consumers have access to cloud-based resources when using software as a service, which can be accessed by paying a

monthly membership fee. This is the first thing that has to be addressed regarding software as a service. PaaS stands for “platform as a service,” and it provides customers with the ability to construct, run, and develop their own apps in the cloud using just the most fundamental configuration options. “Platform independence” is the term used to describe this capability. IaaS is an abbreviation that stands for “infrastructure as a service” and it refers to the accessibility of Internet-based computer resources (Infrastructure as a Service). In this method, users are provided access to servers, networking devices, and storage spaces [1]. Amazon Web Services and Microsoft Azure are examples of IaaS.

A technology that, in the very near future, has the potential to alter the accessibility of computers and storage in several ways. Cloud computing service providers benefit from an infrastructure or platform that is more efficient and less expensive thanks to the cloud. There are several reasons why adopting cloud services is beneficial. The quality and dependability of services have been significantly improved due to the widespread use of cloud technology [2]. The pay-as-you-go model of cloud computing operates on the principle that the customer is responsible for selecting the appropriate level of service for their needs. Users are able to leverage a cloud platform, software, or infrastructure to run their own apps by connecting their devices to the Internet [3]. By examining a service’s metrics and service level agreement (SLA), one is able to determine the amount of quality that the service provides [4].

Large-scale and small-scale data centers may be linked to one another to create a network of virtual machines (VMs), which serve as the computing units in a cloud environment. This may be done in order to develop a cloud framework. Because of the current circumstances, these virtual machines (VMs) should be able to do their work as rapidly as humanly feasible. The principal use of this utility is to move or replace virtual machines without causing any disruption to the applications that are already running. For example, it may be used in a wide variety of applications to modify VM stacks, offer fault tolerance, and reduce power usage [2].

One of the benefits of virtualization is that it makes it easier to move data across different computer systems. Migration is the term used to describe the process by which a computer is transferred from one physical machine to another. Memory is being moved around at several points during this procedure. When migrating, both the destination and the date of the relocation are taken into consideration. In the lab, for instance, the piece of machinery with the biggest demand for resources is often moved to the apparatus used less frequently. The newly installed machinery served as the basis for making the ultimate target selection. Metrics such as the present utilization rate of the system and the efficiency of the system are helpful in determining the optimal location for a machine. Cloud computing is in high demand because virtualization makes it possible to dynamically allocate the required resources. This ability is what drives virtualization’s popularity [5].

The use of cloud computing offers a variety of advantages. One of them is the management of resources. This procedure’s objective is to effectively manage resources in a

manner that does not result in the freezing or crashing of any virtual machines. Throughput might be increased, the effort could be reduced, and flexibility could be increased if load balancing was used [5]. Primary and secondary materials are the two groups that make up the whole collection. The first sort of resources is known as logical resources, while the second kind is known as physical resources. The system can provide various services, including monitoring and managing volatile physical resources, efficient communication protocols, and support with application development. This service also provides logical resource management capabilities in addition to providing resource allocation and migration in the event of a failure. With the assistance of the distributed service, it will be able to increase the fault tolerance rate and, as a result, enhance the process of load balancing.

The scheduling of workloads in relation to the available resources is another difficulty associated with cloud computing. This has a considerable bearing on the situation. It is possible that various computer resources will be given to different customer workloads [6], and this will depend on the requirements of the customers. Distributing the job more evenly enhances the allocation of resources and boosts the level of pleasure experienced by customers. Failures may be reduced to a minimum if several copies of every resource are kept. The Primary-Backup approach is one that enables the creation of two copies of each activity. As the name suggests, the strategy is divided into two parts: the primary and the secondary. Grids and clusters have both been put to use in the past to manage the storage and distribution of a variety of resources. It is possible for the optimization of resources in a virtual environment to take place independently or as a component of a cloud service.

The great majority of scientists devote a significant portion of their research to the topic of fault tolerance. Failure tolerance is the degree to which a process may be handled effectively without having a detrimental effect on the results it produces. The system’s ability to tolerate complete destruction of either the computers or the resources will not be affected by this, and a trigger from the outside world has no influence on the operation that was planned. All of the problems in the world may be traced back to an excessive number of equipment and procedures. In addition, extra resources are needed to complete the operation using the now available software.

When a component of a system or the system as a whole does not perform as it should, the cause of this issue is referred to as a “fault” [7]. The existence of flaws in the system leads to the production of errors, which hinders the system’s ability to function as intended and provide desired results. When there are holes in a system, there will inevitably be places where it breaks down. In the case that at least one problem occurs, the system has to be able to continue working properly in spite of a decline in the execution quality. This article presents effective defect prediction approaches in a cloud computing environment that are enabled by artificial intelligence [6]. The goal of these strategies is to improve resource optimization.

2. Literature Survey

In order to improve the accuracy of their predictions, a large number of researchers have focused their efforts on improving fault prediction. They have done this by employing a wide variety of methods. Cloud computing has not yet produced any methods for predicting activities that are unsuccessful when employing data from scientific processes. We conducted a comprehensive survey in this area to determine which methods are the most effective at predicting and identifying job failures. There are a large number of alternative approaches that may be taken in this regard.

In a scenario where faults are dispersed, several authors have proposed and implemented fault detection and prediction techniques. In their work, Fu and Xu [8] used these concepts to develop their model. In addition, they put in place a framework for failure proactive prediction to forecast the failure of components. Through the use of supervised learning algorithms, it has been possible to obtain a level of accuracy in predicting failures that range from 70.1 percent online to 74 percent offline.

Zhou et al. [9] have created a method for spotting faults in the actual workflow engine before they occur. Workflow management systems such as Pegasus need to be installed to employ the Failure Aware Workflow Scheduling technique recommended by Yu et al. to anticipate not only failures of online resources but also failures of tasks. Using decision trees or Bayesian networks to identify and forecast failures in healthcare data is an additional approach that was proposed by Guan and his colleagues [10]. After taking into consideration the numerous methods for defect prediction using healthcare data that have been offered by a number of authors, it is possible to improve the performance of workflow scheduling techniques in the context of evaluating data pertaining to scientific workflows. This is the case when carrying out an analysis of scientific workflows.

Using a heartbeat message protocol that Zhao et al. [11] established, replica failures may potentially be identified and remedied. Even though analogous work has been proposed by Jhawar et al. [12] using the heart-rate message protocol [12] in Cloud settings, these researchers have not taken into consideration the failure of scientific applications while attempting to identify VM crashes. Liu and Shabaz et al. [13] created a defect prediction approach for constructing dependable Cloud Systems using Bayesian classifiers and decision trees. This method was built by making use of the collection of health data. It has been suggested by Deepak et al. [14] that fault-tolerant workflow scheduling may reduce expenses by as much as seventy percent and that failure prediction algorithms may be used in the course of future studies in order to achieve this goal. It is necessary to create new machine learning and statistical approaches in order to enhance the performance of these methods moving ahead; comparatively, few academics have worked on cloud failure prediction.

According to Catal [15], when it comes to the ability to anticipate outcomes, machine learning models perform much better than statistical approaches. As a direct consequence of this, the use of machine learning strategies for proactive failure prediction is required. In their study [16],

researchers Malhotra and Jain discovered that Random Forest was more accurate than other methods in forecasting electrical failures. Zhang et al. [17] used multivariate analytic techniques to make their predictions on fault-prone classes using Ericsson Telecom AB's data. Utilizing an artificial neural network allows for more accurate forecasting of resource use in cloud-based systems (ANN).

It has been shown that incorporating Artificial Neural Networks (ANN) into cloud computing systems results in increased accuracy when it comes to forecasting the provisioning of resources. According to the findings of Catal's research, the Naive Bayes Model is the most effective technique for using machine learning to forecast software errors. In addition, the performance of Naive Bayes compared to that of other machine learning algorithms like ANN, LR, and Random Forest has not been compared. When evaluating this methodology, failures caused by resource overuse or failures caused by resource failures have also not been taken into account. None of the approaches described above have been successful in precisely predicting the outcome of a job despite making use of a number of different algorithms for machine learning.

When it comes to scientific applications, having precise failure forecasts might help lessen the impact of failures. It is feasible to arrange the many resources, applications, and services provided by the cloud in such a manner as to minimize the effect that any failures will have on the system. It is difficult to create accurate forecasts a sufficient amount of time in advance when dealing with complicated applications such as workflows; nonetheless, doing so is a precondition for applying an intelligent fault-tolerant strategy to scientific procedures.

It has been shown by Glass et al. [18] and others that prior models only supply one form of resource for each activity, and it should be taken into mind that there are connections between numerous resources and activities. In addition, the research presented in [19] describes the recovery strategy as a combination of self-reconfiguration and self-routing. This helps to clarify the reasoning behind the self-healing properties of point-to-point networks. Through the use of checkpointing, the permitted tasks alter the status of a shadow task that is not being utilized.

Fan et al. [20] used an adaptive learning technique that takes into account and remembers the decisions made by human administrators in their further development of the architecture-based method to self-adaptation. Although a Model Driven Technique was proposed, this method did not consider the practical uses of the system's functional and self-healing components.

Agent-based systems provide resilience through redundancy, enabling them to cope with unforeseen occurrences in their environments and accomplish the objectives they set for themselves. These systems consider a number of cooperative and persistent agents all working together.

It was suggested by Lohani et al. [21] to make use of a variety of techniques to get rid of discovered rules and improve the self-adaptive capabilities of the software. In order to eliminate any potential conflicts, the policy has to be described graphically, and the rule model should be produced

from the policy. A policy fulfills the purpose of a high-level policy by acting as a high-level proclamation of the rules. A policy-driven self-healing algorithm that automatically identifies, diagnoses, and recovers defects that develop throughout the course of context-aware systems adaptation processes has also been proposed by Kapoor et al. [22].

In order to successfully complete the action selection step, it is important to build a reinforcement learning system. On the basis of a rule-based methodology, Komi S. Abotosi and colleagues [23] have created a unique framework that they refer to as BP-FAMA. Within the context of this project, a set of business rules was reimagined as a graph depicting cause and effect relationships. However, there is a need to enhance the flavor. Włodzimierz Funika and his colleagues [24] devised a mechanism for self-healing systems to be monitored, and it is described here. The rules and actions might be preserved via the usage of ontology.

3. Methodology

According to the provided model, failure to complete a job may apparently be anticipated using scientific methods. For the sake of these scientific applications, a process or computation is shown in the form of a data flow and task dependencies. By proactively analyzing data from various scientific workflows with the help of cutting-edge failure prediction machine learning algorithms, it is possible to lessen the impact that failure has on these workflow activities when utilizing Cloud resources. This is accomplished by analyzing the data in real time. During the process of scheduling the scientific workflow, task failures could be caused by a variety of factors, including excessive use of resources, insufficient use of resources, execution time or cost exceeding the threshold value, improper installation of essential libraries, running out of memory or disc space, and other similar occurrences. The failure of tasks (of the central processing unit, random access memory, disc storage, and network bandwidth) as a result of resource overutilization is the primary emphasis of the recommended paradigm for this research. The objective of the methodology presented here is to build a model with the capability of tracking real-time data pertaining to scientific processes to spot problems on the job. The proposed model performs an analysis on many workflows that have been saved in cloud repositories to identify potential process issues before they arise. The proposed model is derived from the findings of an experiment, and it uses the machine learning approach that proved to be the most successful in forecasting failures. The flowchart for fault prediction techniques in a cloud computing environment for improving resource optimization is shown in Figure 1. The fault prediction technique follows three major steps: feature selection from input dataset using PCA algorithm, classification of the data using Naive Bayes, Random Forest and linear regression, and finally, failure prediction.

Logical regression [25] is one method among many others that may be used to predict a dependent variable based on a number of independent variables. In order to determine which classes had the greatest potential for failure, a multivariate model of resource consumption was used. This model took into account a large number of distinct characteristics.

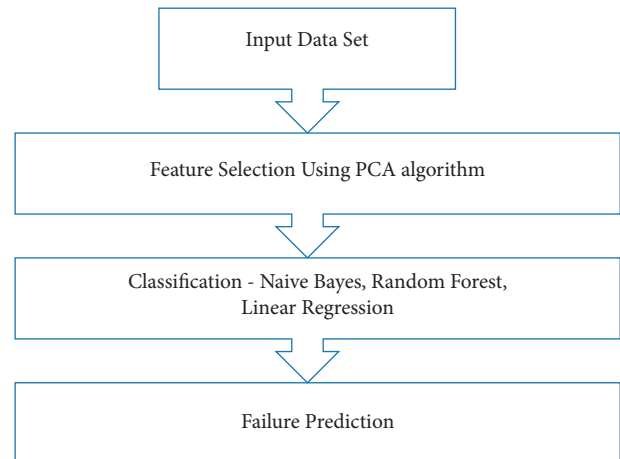


FIGURE 1: Fault prediction techniques in cloud computing environment for improving resource optimization.

To identify which set of independent variables comprises the most useful set, it is possible to utilize either the forward selection or the backward elimination strategy.

When employing the Random Forest approach, it is possible to produce thousands of individual trees [26]. The advantages of the bagging method and the random selection methodology are combined in this method. Random feature selection examines each node in search of the finest split over a random subset of the features, while bagging algorithms have been used to constantly extract samples from various datasets with uniform probability distributions. When classifying a new item based on an input vector, the Random Forest takes into account the degree to which each tree has been penetrated by the vector. Each tree in the forest places a vote based on the classification of the input vector, and the forest ultimately selects the classification that was chosen because it earned the most votes from the other trees in the forest.

The Bayes theorem is a statistical formula that is used in the process of data classification when employing Bayesian approaches. A probabilistic model is assumed to apply to the numerical data and the categorical data. Compared to other methods of categorization, this one is very time efficient. It does this by integrating a number of different classification algorithms with a standard operating procedure and industry best practices in order to produce a classification model. It is often held that every person has all of the characteristics that are shared by the system. These commonly used classification methods are unfazed by the presence of challenging data characteristics such as large size and complexity, and they are a typical kind of tool. On the other hand, real-time data models focus more on the problem of missing values than traditional models do. The Nave Bayes classifier is used in sensor networks to make accurate predictions about the time of event outcomes [27].

4. Result Analysis

The sRNA Identification Protocol utilizing High-throughput Technology (SIPHT) and the Laser Interferometer Gravitational-Wave Observatory (LIGO) Inspiral

are just a couple of the many examples of scientific process applications that are analyzed and stored as part of the suggested approach. The features of resource use were evaluated by comparing the threshold values for CPU, bandwidth, random access memory, and disc usage. The threshold values have been set by looking at the past history of failed tasks that were caused by excessive use of virtual machines (VMs). In the event that the utilization parameter's value was more than the maximum threshold value, the operation would be deemed unsuccessful; otherwise, it would be successful. Following that, the model for predicting failure was constructed using a machine learning strategy with the smallest feasible RMSE, MAPE, and total accuracy errors. This course of action was decided upon.

Workflow Sim 1.0 has been put to use to investigate, optimize, and improve scientific workflow application failure patterns. This has been accomplished with the assistance of CloudSim. In this particular instance, both static and dynamic schedulers are linked together to form a chain. In order to develop apps for scientific workflows, XML schema was used, and workflow engine was employed to analyze the results of such workflows. In addition to it, clustering and fault tolerance are included. Workflow applications in the scientific community have been analyzed and documented with the help of CloudSim and Workflow Sim.

In the current investigation, both data extraction and prediction-based outcome-based algorithms were used, both of which are examples of the types of algorithms that may be carried out using Weka. The performance indicators and predicted failures of tasks for a number of different procedures have been documented.

In order to analyze resource consumption measures, the threshold values of CPU utilization, bandwidth utilization, random access memory utilization, and disc utilization were employed. The threshold value is determined by looking at the past failures of tasks that were attributed to an excessive amount of VM usage. If the value of a parameter is greater than a threshold that has been established in advance, the status of task failure will be issued; otherwise, the status of No Failure will be assigned.

Input data set has nine attributes: Task ID, VM ID, Data Centre ID, CPU Utilization, Bandwidth Utilization, RAM Utilization, Disk Utilization, Task Size, and Status.

In this study, the performance of a number of different algorithms is analyzed and compared based on three criteria: accuracy, sensitivity, RMSE, and specificity. Performance is shown in Figures 2–5.

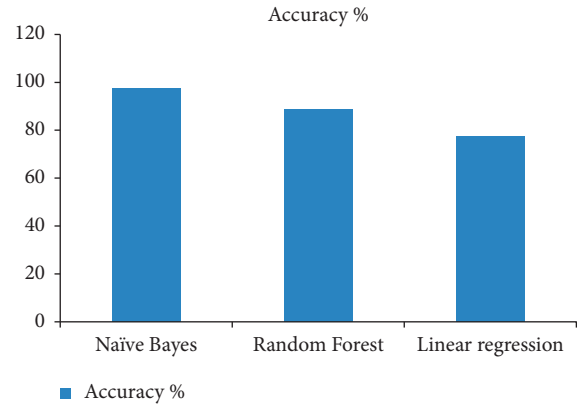


FIGURE 2: Accuracy comparison of classifiers for task failure prediction in a cloud environment.

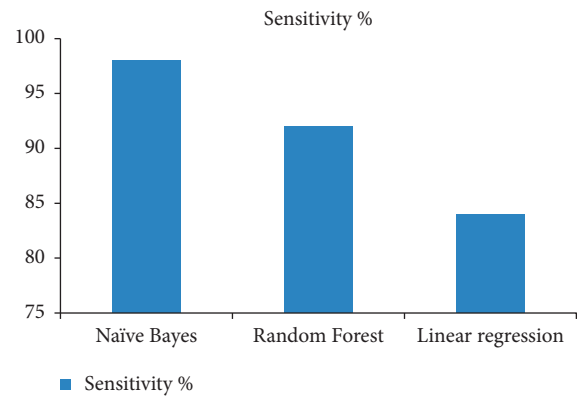


FIGURE 3: Sensitivity comparison of classifiers for task failure prediction in a cloud environment.

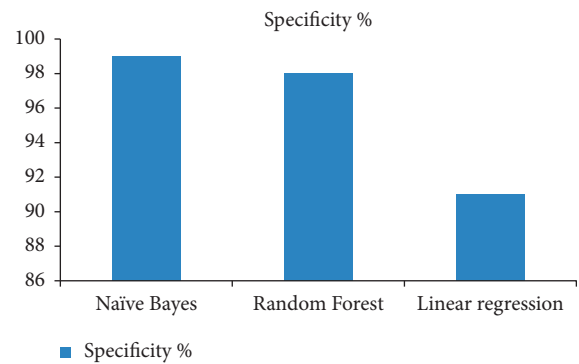


FIGURE 4: Specificity comparison of classifiers for task failure prediction in a cloud environment.

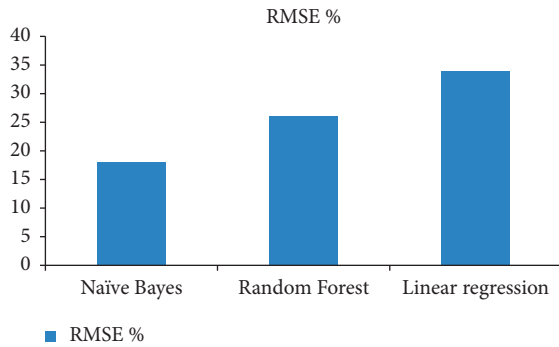


FIGURE 5: RMSE comparison of classifiers for task failure prediction in a cloud environment.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN},$$

$$\text{Sensitivity} = \frac{TP}{TP + FN}, \quad (1)$$

$$\text{Specificity} = \frac{TN}{TN + FP},$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

5. Conclusion

The bulk of the fault tolerance techniques that are in use today laid their primary emphasis, in the event that a virtual machine fails, on the production of clones to replace it rather than on the early prediction of the failure itself in advance. Several of the currently used techniques give migration priority over recovery in the event that a virtual machine (VM) fails. This is due to resource constraints and concerns with server availability. Examples of algorithms with a single objective include fault tolerance, migration prediction, and simply expecting failure. Another example is fault tolerance. In this research, we are aiming to determine the most effective strategy to transition from a system that is not operating well to one that does. It is essential to be able to predict the failure of a virtual machine in a timely manner due of issues such as squandered resources, energy, and cost. Since the beginning of cloud computing, there has been an issue with the dependability of virtual computers, often known as VMs. As an integral component of a fault tolerance system, preemptive measures are an absolute need in order to guarantee the continuation of service. As a consequence of this, it is vital to work toward enhancing and emphasizing the proactive failure prediction of virtual machines. The key motivations for this are decreased periods of downtime as well as enhanced scalability. A technique was utilized to transfer the resources that were predicted to fail from one virtual machine (VM) to another VM in a safe manner. By using the compression strategy, the amount of time needed to complete the migration was reduced, and resource

utilization increased. This article provides artificial intelligence that enabled effective fault prediction techniques in cloud computing to improve resource optimization.

Data Availability

The data can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] J. Domingos, "Failure Prediction for Cloud Applications through Ensemble Learning," in *Proceedings of the 2021 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pp. 319–322, Wuhan, China, October 2021.
- [2] X. Chen, C. D. Lu, and K. Pattabiraman, "Failure analysis of jobs in compute clouds: a google cluster case study," in *Proceedings of the 2014 IEEE 25th International Symposium on Software Reliability Engineering*, pp. 167–177, Naples, Italy, November 2014.
- [3] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: google trace analysis," in *Proceedings of the 3rd ACM Symposium on Cloud Computing*, p. 7, San Jose, CA, USA, October 2012.
- [4] M. Snir, R. W. Wisniewski, J. A. Abraham et al., "Addressing failures in exascale computing," *International Journal of High Performance Computing Applications*, vol. 28, no. 2, pp. 129–173, 2014.
- [5] R. Jhawar, V. Piuri, and M. Santambrogio, "A comprehensive conceptual system-level approach to fault tolerance in cloud computing," in *Proceedings of the 2012 IEEE International Systems Conference SysCon 2012*, pp. 1–5, Vancouver, BC, Canada, March 2012.
- [6] Microsoft, "Azure status history," 2022, <https://status.azure.com/en-gb/status/history/>.
- [7] A. Raghuvanshi, U. K. Singh, and C. Joshi, "A review of various security and privacy innovations for IoT applications in healthcare," *Advanced Healthcare Systems*, vol. 4, pp. 43–58, 2022.
- [8] S. Fu and C. Z. Xu, "Exploring event correlation for failure prediction in coalitions of clusters," in *Proceedings of the 2007 ACM/IEEE conference on Supercomputing*, vol. 41, p. 112, Reno, NV, USA, November 2007.
- [9] Y. Zhou, X. Hu, and M. Shabaz, "Application and innovation of digital media technology in visual design," in *International Journal of System Assurance Engineering and Management*, vol. 13, no. 1, Springer Science and Business Media LLC, New York, NY, USA, 2021.
- [10] Q. Guan, Z. Zhang, and S. Fu, "A failure detection and prediction mechanism for enhancing dependability of data centers," *International Journal of Computer Theory and Engineering*, vol. 4, pp. 726–730, 2012.
- [11] W. Zhao, P. Melliar, and L. E. Moser, "Fault tolerance middleware for cloud computing," in *Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing*, pp. 67–74, Miami, FL, USA, July 2010.
- [12] R. Jhawar, V. Piuri, and M. D. Santambrogio, "A comprehensive conceptual system-level approach to fault tolerance in

- cloud computing,” in *Proceedings of the EEE International System Conference Syscon 2012*, pp. 1–5, Vancouver, BC, Canada, March 2012.
- [13] Y. Liu and M. Shabaz, “Design and research of computer network micro-course management system based on JSP technology,” in *International Journal of System Assurance Engineering and Management*, Springer Science and Business Media LLC, New York, NY, USA, 2021.
- [14] P. Deepak, S. K. Garg, R. Buyya, Y. Yang, and K. Ramamohanara, “Robust Scheduling of Scientific Workflows with Deadline and Budget Constraints in Clouds,” in *Proceedings of the 2014 IEEE 28th International Conference on Advanced Information Networking and Applications*, pp. 858–865, Victoria, BC, Canada, May 2014.
- [15] C. Catal, “Software fault prediction: a literature review and current trends,” *Expert Systems with Applications*, vol. 38, no. 4, pp. 4626–4636, 2011.
- [16] R. Malhotra and A. Jain, “Fault prediction using statistical and machine learning methods for improving software quality,” *Journal of Information Processing Systems*, vol. 8, no. 2, pp. 241–262, 2012.
- [17] X. Zhang, K. P. Rane, I. Kakaravada, and M. Shabaz, “Research on vibration monitoring and fault diagnosis of rotating machinery based on internet of things technology,” *Nonlinear Engineering*, vol. 10, no. 1, pp. 245–254, 2021.
- [18] M. Glass, M. Lukasiewicz, T. Streichert, C. Haubelt, and J. Teich, “Reliability-Aware System Synthesis,” in *Proceedings of the 2007 Design, Automation and Test Europe Conference and Exhibition*, pp. 1–6, Nice, France, April 2007.
- [19] D. Koch, T. Streichert, S. Dittrich, C. Strengert, C. Haubelt, and Teich, “An operating system infrastructure for fault-tolerant reconfigurable networks,” in *Proceedings of the 19th International Conference on Architecture of Computing Systems*, pp. 202–212, Springer, Berlin, Germany, March 2006.
- [20] M. Fan, D. Su, M. W. Bhatt, and A. Mangal, “Study on non-linear planning model of green building energy consumption under multi-objective optimization,” *International Journal of System Assurance Engineering and Management*, vol. 13, no. 1, pp. 437–443, 2021.
- [21] T. K. Lohani, M. T. Ayana, A. K. Mohammed, M. Shabaz, G. Dhiman, and V. Jagota, “A comprehensive approach of hydrological issues related to ground water using GIS in the Hindu holy city of Gaya, India,” *World Journal of Engineering ahead-of-print (Issue ahead-of-print)*. Emerald, vol. 4, 2021.
- [22] A. Kapoor, A. Gupta, R. Gupta, S. Tanwar, G. Sharma, and I. E. Davidson, “Ransomware detection, avoidance, and mitigation scheme: a review and future directions,” *Sustainability*, vol. 14, no. 1, p. 8, 2021.
- [23] B. Mohamed and Y. Amghar, “Towards self-healing execution of business processes based on rules,” in *Proceedings of the Enterprise Information Systems, 11th International Conference, ICEIS 2009*, pp. 510–512, Milan, Italy, May 2009.
- [24] N. K. Singh, R. K. Singh, D. K. Khare, H. yadav, P. Jain, and M. W. Bhatt, “Optimized resource allocation and trust management schemes for non-orthogonal multiple access on the internet of vehicles,” in *Computers & Electrical Engineering*, vol. 102, Elsevier BV, Article ID 108184, 2022.
- [25] R. Khan, M. Shabaz, S. Hussain, F. Ahmad, and P. Mishra, “Early flood detection and rescue using bioinformatic devices, internet of things (IOT) and Android application,” *World Journal of Engineering*, vol. 19, no. 2, pp. 204–215, 2021.
- [26] A. Gupta and P. Prabhat, “Novel approaches in network fault management,” *International Journal Of Next-Generation Computing*, vol. 8, no. 2, 2017.
- [27] D. Mondal, D. K. Kole, and K. Roy, “Gradation of yellow mosaic virus disease of okra and bitter melon based on entropy based binning and Naive Bayes classifier after identification of leaves,” *Computers and Electronics in Agriculture*, vol. 142, pp. 485–493, 2017.