Journal of Cybersecurity and Information Management (JCIM) Vol. 16, No. 01, PP. 86-98, 2025

Innovative Resilient Systems Scheduling Methods for Explicit Critical Applications in Cloud Environments

Adel A. Alyoubi[1,*]

[1]College of Business, Department of Management Information Systems, University of Jeddah, Saudi Arabia

Email: aaaalyoubi1@uj.edu.sa

Abstract

The model mentioned in the study introduces a new Puzzle Optimization Algorithm-Based Fault Tolerant Scheduling (POAB-FTS) model specifically designed for the cloud computing setting. This pinpoints the significant challenge of achieving reliability, availability, and performance in resource scheduling in the context of failure cases, which is addressed by this novel technique. The POAB-FTS methodology integrates optimization using a game theory approach to perform actions that reduce execution time and failure probability while using a fitness function to provide better decision-making. This work entails an assessment of the main reasons behind task and hardware failures such as lack of resources, hardware defects, and suboptimal implementation. The model covers both active and passive fault tolerance approaches to workload balancing, migration before failure, and migration after failure points. Cooking schedules derived from the POAB-FTS technique are compared against the MAXMIN, ACO, and GTO-FTASS algorithms to present the makespan, failure ratios, and failure slowdowns—giving a comprehensive comparison of the method. As shown in this paper, the POAB-FTS framework can improve the system's fault-tolerance and adapt resource allocation based on the actual demand thereby stressing its capacity to act as a scalable and cost-efficient solution for the improvement of cloud computing infrastructures. On this contribution, a sound and optimal cloud resource management is made possible.

Keywords: Cloud Computing Environment; Puzzle Optimization Algorithm (POA); Pre-emptive Migration; Optimization Algorithms; Cost Efficiency; System Robustness; Resource Allocation and Execution Time Optimization

1. Introduction

Cloud computing has taken the mantle as a strategic shift in the handling of business activities through efficiencies strategies. This distributed computing infrastructure allows enterprises to organize a virtual resource pool of storage [1] devices, servers, and applications on a subscription model. With the help of flexible capacity procurement and low operational costs, cloud technology is now an essential part of business strategies that help the companies to be financially flexible and operationally effective. However, the reliability of cloud services is a significant consideration, which is why sudden system breakdowns with the [2] potential to cause mission-critical operational pauses or losses, revenue loss or reputational damage.

This introduces the problem of efficient scheduling in cloud computing as more clients turn to this service delivery model [3] to meet their needs for fault-tolerant services. Higher service-level agreements (SLAs) [4] are expected from businesses today to be reliable, always available and with the least possible disruption. Other attributes include; fault tolerance that is crucial now as it helps overcome effects of server, network or hardware failure [5] by continuing with tasks execution. On balance, the financial liabilities of rolling out such mechanisms are conversely offset by the capacity to minimize the economic loss during such periods and protect the customer base, as well as increase the ROI by maximizing resource utilization and preventing interruptions to businesses [6].

Cloud computing fault-tolerance techniques include proactive and reactive measures, which lie at the basis of optimisation of resources' usage. Workload [7] distribution and migration before the system exposes its weak points reduce the chance of failure due to planned preventative actions. On the other hand, reactive approaches, like restart systems and checkpoints,

DOI: https://doi.org/10.54216/JCIM.160107
Received: October 10, 2024 Revised: January 03, 2025 Accepted: February 01, 2025

complement initiative-taking systems by maintaining execution integrity in case of failure, or interrupting, by providing ways to manage such failures. [8] Each of the strategies has the critical business success factors like risk management, business continuity, and cost containment.

Again, the cost to the economy is not considered limited to how it relates to avoiding downtime. For instance, improving the schedule of tasks results to better resource utilization of the computation equipment resulting to minimization of infrastructure [9] expenses. Furthermore, scheduling is a good business technique in cutting down the time to execute projects, and operation time delay, which amounts to increasing production rates. These advancements give a competitive edge by offering better rates of product delivery and increasing customer's satisfaction. The other financial resources are; [10] It also improves the company's match with capital mobilization plan, goods and services budgeting is also enhanced as wastage and failure expenses are significantly reduced.

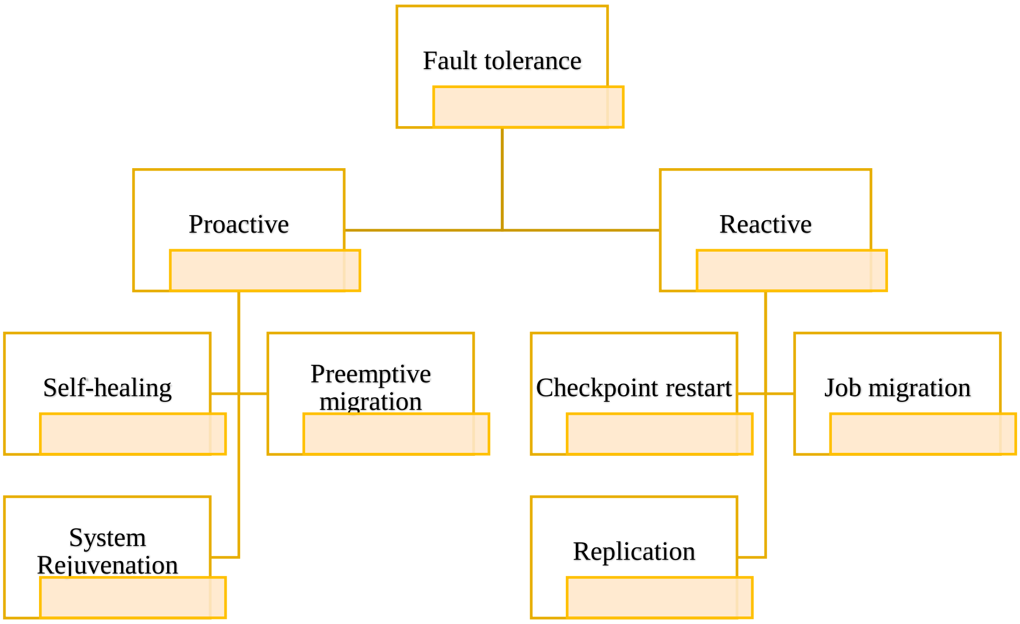

Figure 1. Fault tolerance approaches in cloud.

In order to resolve these issues, this research introduces a Puzzle Optimization Algorithm-Based Fault Tolerant Scheduling (POAB-FTS) model for efficient execution of scheduled tasks in cloud computing platforms [11]. Using the gaming approach towards optimization of the model, solutions can be enhanced for scheduling of resources in a fault-tolerance, and cost-optimized manner. The planned system also includes the concept of a fitness function designed to minimize the execution time and failure probability; this meets the goals of enterprises providing increased profitability and performance.

This research also establishes the loss implications of systems breakdowns in extensive data center structures bearing in mind the high susceptibility [12] to cloud service. The study shows that the hard drive, the network, and memory lead to operational disruptions most often. Working through these issues, the POAB-FTS model improves not only the general reliability of separate components, but the availability of the necessary services as well [13]. This, however, has the positive ramification of minimizing costs entailed by repetitive maintenance, hardware [14] exchanges, and service level non-adherence.

In the larger business perspective, the application of enhanced scheduling methodologies such as POAB-FTS assists organizations to handle scalability [15] and failure. The focus of the model on minimizing makespan and failure ratios improves resource utilization, decreasing overhead expenses but increasing system reliability. Besides, the integration of trust levels and [16] security enables compliance and governance features of the solution to meet compliance and governance compliance and emphasize the role of the solution in risk management and financial governance [17].

Lastly, by offering the POAB-FTS framework, the study helps in developing a fault-tolerant cloud-computing environment towards meeting the operational efficiency and financial viability of the cloud systems. This work fills a gap by providing steps for translating [18] technical advancement in cloud computing into tangible results that could lead to improvements in the competitive advantage and return on investment for firms.

DOI: https://doi.org/10.54216/JCIM.160107
Received: October 10, 2024 Revised: January 03, 2025 Accepted: February 01, 2025

2. Literature Review

The specifics of DERs incorporation and the growing importance of microgrids in contemporary power systems result in the most vital concerns of [19] stability, efficiency, and reliability. Such requirements are again hard to meet using conventional control techniques that may not be suited for addressing the emergent needs of these systems. Thus, ML and bio-inspired optimization solutions [20] are trending to improve the grid stability and predict the demand of power.

Therefore, this research work is titled “Stability and Control using Machine Learning (ML).’

Different approaches have been employed for stability prediction in power systems such as artificial neural network (ANN), random forests, and support vector [21] machine (SVM). These models claim better prediction results than the conventional techniques in conditions of an imbalanced dataset. This study has also demonstrated that a MD LSTMs model performed well in voltage stability forecasting since it provided an elevated level of accuracy. Furthermore, techniques such as the stacking have been applied to combine multiple classifiers in order to achieve better stability prediction results.

Microgrids Optimisation Citations Sun Microgrid, Bio Inspired Optimisation Architecture

We identified that bio-inspired algorithms, especially particle swarm optimization (PSO) and genetic algorithms (GA) [22] have shown potential in enhancing the stability of microgrids. These methods are applied for control parameter tuning in order to have improved grid voltage during disturbances. As such, they have shorter convergence time and better accuracy as compared to other conventional methods. Nevertheless, there are some problems: for example, long time to perform calculations, and slow convergence in some situations, especially when there are many variables. The use of integrated bio-inspired optimization, [23] and ANN models has been found to be more robust and robust with microgrids under different operating conditions.

Load demand forecasting models can be categorized as hybrid models.

It is particularly important to have the correctly predicted load demand of the grids. Finally, integration of evolutionary algorithms such as genetic algorithms or/and PSO with neural networks is identified as the best possible technique in predicting load demand [24]. Such models give perfectly accurate short-term load forecasting while taking into consideration features l.

Like seasonality, economic status and weather conditions. They are beneficial in many ways as they are efficient and fast; nevertheless, they are data and computationally hungry, making them unsuitable for applications in real time.

Table 1: An Overview of Essential Research Works

Ref. No	**Research Work**	**Findings**	**Methodologies Used**	**Advantages**	**Limitations**
[24]	Energy System Sustainability Forecasting	In terms of stability estimation, ML algorithms beat out approaches that are more conventional.	ANN, Random Forest, SVM, LSTM	Manage datasets with imbalances while maintaining high forecasting accuracy.	It is computationally expensive and requires enormous sets of information.
[25]	Improve the reliability of Microgrids	Algorithms that draw inspiration from biology enhance the stability of grid control settings.	PSO, GA, Hybrid ANN-PSO	Enhanced control of grid voltage and quicker settlement.	Collapse in complicated situations is slow.
[26]	Load Demand Prediction	Hybrid approaches are great for predicting loads soon.	GA-ANN, PSO-ANN	Extreme precision, responsiveness to variations in the weather and seasons.	Competingly costly and necessitating huge databases.
[27]	Voltage Stability Forecasting	When it comes to voltage stability, LSTM models are superior to methods that are more conventional.	LSTM, Decision Trees, Random Forest	Extreme precision in predicting the stability of voltage.	Demands a large amount of data for training and computing resources.

DOI: https://doi.org/10.54216/JCIM.160107
Received: October 10, 2024 Revised: January 03, 2025 Accepted: February 01, 2025

Lastly, it can be summarized similarly with the future work outlook that the employment of ML and bio-inspired optimization methodologies has a considerable potential to enhance the stability, efficiency, and accuracy of predictions in smart and microgrids. These methods can help manage these aspects of the contemporary power systems, which are otherwise difficult to achieve due to dynamism based on the above ideas and thoughts. But they have big data requirements and computational costs that are still a problem and thus need to be resolved before they become common.

3. Objectives of the Research

The proposed technique known as the Puzzle Optimization Algorithm-based Fault Tolerant Scheduling (POAB-FTS) is used to schedule resources within a Cloud Computing (CC) setting with consideration to system faults. Cloud computing systems are susceptible to failures including hardware, network, and software flaws, which disrupt a task. However, fault tolerance is taken into account in the POAB-FTS technique to provide better efficiency of the schedule results in terms of resource usage.

The Puzzle Optimization Algorithm (POA) basic idea is developed by consider certain tasks as “puzzles” and tries to solve them to find the best performant configurations into a potential population. The best solution in population, which is termed as the “best individual” in the process of providing solution information to the problem in hand. Due to consulting this best individual in the population, the algorithm increases the probability of resolving the puzzles (tasks) more efficiently. The strategy allows for the probability of faults to be incorporated into the scheduling mechanism to be taken into consideration, flexibility especially in the light of system failure.

The second aspect of POAB-FTS is called the Fault Tolerant Scheduling, which creates a scheduling strategy that can have low both execution time and failure probability. Execution time means the time that it takes to accomplish the tasks while failure probability takes into account the probability of a task failure through fault in the system. Thus, reducing the two elements has the goal of improving the global performance and the reliability of the cloud computing system by using the POAB-FTS technique.

POAB-FTS technique efficacy is tested through simulations and experiments. Such tests usually consist of executing a set of analogous data runs, where different fault conditions are initiated to understand how the number of resources is controlled, and the extent of the faults is minimized. From the outcome, much enhancement can be done on the POAB-FTS technique to improve on the fault tolerance and scheduling reliability in real world cloud computing scenarios.

4. Motivation of the Research

The rationale behind this research stems from essential concerns in cloud computing infrastructure, reliability, and fault tolerance. When deploying cloud-based solutions, which are becoming more popular about the scalability of businesses and cost effectiveness, unfamiliar problems such as hardware failure, unavailability of resources and delay in the execution appear. These failures can actually lead to shut down of business and immense loss of money and tarnished reputation.

This research aims at achieving new directions in fault-tolerant scheduling in which there are reduced faults and better resource utilization to ensure continuous provision of service. These problems are dealt with by the proposed Puzzle Optimization Algorithm-Based Fault Tolerant Scheduling (POAB-FTS) technique using proactive and reactive fault tolerance strategies, reduces the execution time and minimizes failure probabilities. In this research, I aimed at developing a solid and efficient cloud environment by establishing and analyzing efficient and reliable cloud settings, which will enhance the development and stability of organizations depending on cloud computing to execute essential operations.

5. Proposed Method

The new method to be introduced here is the Puzzle Optimization Algorithm-Based Fault Tolerant Scheduling (POAB-FTS) technique, which is a proposed solution to the problems associated with fault tolerance in the cloud computing systems. Redundancy is important to maximize the availability of cloud-based systems in case of hardware or software crashes because such incidences are likely to result to system unavailability, resource wastage and inflated costs, respectively.

This paper proposes the POAB-FTS technique, which combines the proactive and reactive fault tolerance approach to schedule resources proactively. Preemptively it identifies areas that are likely to fail and rearranges how the work is going to be flowed to prevent disruption. As a reactive tool, it reduces the effects of failures through recovery procedures like replunging systems and restore points. It assesses an objective mechanical function that considers aspects of optimizing execution time, and low chances of failure to enhance system reliability and quality.

Taking cue from the puzzles, the algorithm approaches scheduling as an optimization dilemma; each of the resource-task analogs is a piece of the whole jigsaw. One of the main advantages of the approach is based on the effective application of collaborative problem solving to envision configurations for smooth execution of tasks.

DOI: https://doi.org/10.54216/JCIM.160107
Received: October 10, 2024 Revised: January 03, 2025 Accepted: February 01, 2025

While the POAB-FTS technique possesses all the beneficial features of the POAB-PROM model and computational optimization, experimental validation has indicated enhanced efficiency for minimizing failures, utilization of resources, and overall execution times compared to the current models. Perturbations in this innovation can indeed present the prospect of improving fault tolerance across various cloud computing related applications.

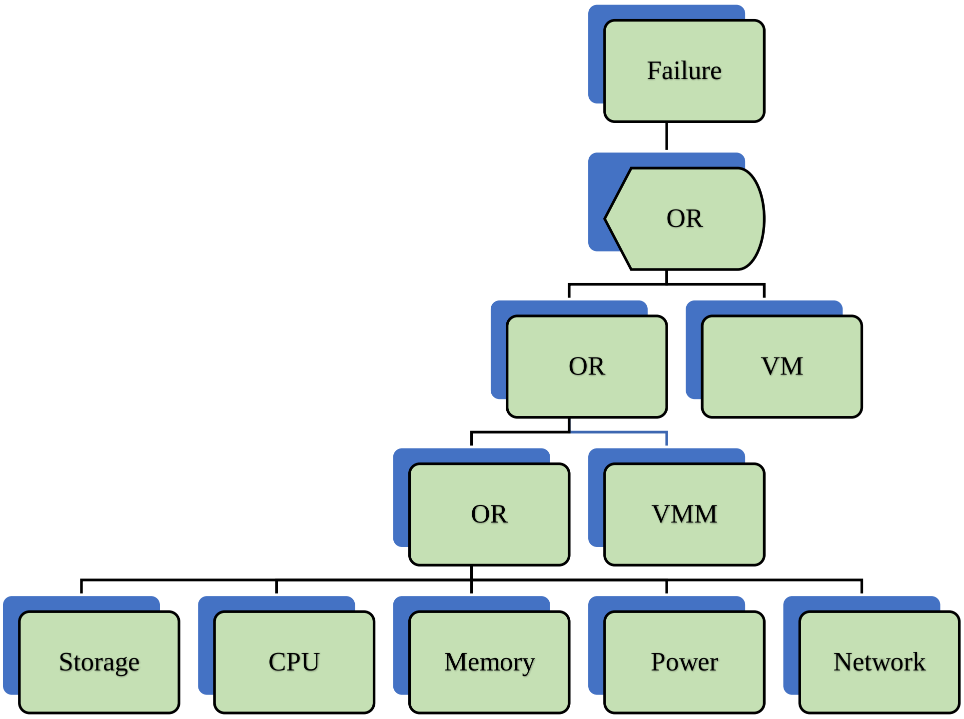

Figure 2. Fault tree characterizing server failures.

About hardware repair and server failure, the project aims to analyze the behavior with the help of a number of participants, approximately 100000 servers, and data on component replacement derived from a large store of servers. Among the data that is gathered, include the setup of the server, the time that a hard disk is issued for replacement and the time the disk is being replaced. The study reveals that around eight per cent of the machines break down annually, while 92 per cent of the equipment has never been serviced. They anticipate that it will cost approximately 2.5 million dollars to repair 100,000 servers, which implies a yearly failure rate of 8%. These are the areas that stand to suffer the most and it is the hard drive that is the centerpiece of the failure of servers. Disc failure is highest with young servers as about 5% of the servers have a disc failure in the first year of their commissioning date, and rates rise with machine age as 12% of the servers have hard disc failures when the machines are one year old, and 25% of servers have hard disc failures when the machines are two years old. When applying the Chi-squared automated interaction detector approach no major indications for faults in the system are discovered. A comparison was made between RPMs and the number of discs per server with 121 servers taken through the study. Based on the analysis performed in the current comparative study, it can be concluded that there is correlation between the failure characteristics of the servers that have a first failure and the realization of the number of RPM, which is equivalent to the total number of discs in that server.

According to this Figure 2, one might conclude that robust fault tolerance is required to improve the reliability of hard drives because in case individual parts fail the model predicts many failures in total number and to reduce this powerful fault tolerance technique is necessary. Furthermore, because applications should satisfy the high availability and reliability criteria, they should reduce their access to formerly fail hard drives classes since the chance to observe another failure on the same hard disc belongs to a bigger class.

During this research, a novel POAB-FTS approach was created for fault-tolerant programming in the Cisco Cloud environment. The objective of the POAB-FTS approach that has been provided is to plan assets while considering defects. During the process of solving all of the riddles, the best person in the population, in particular, is approached for guidance in POA. An overview of the POAB-FTS technique is shown in Figure 3, which can be found here.

DOI: https://doi.org/10.54216/JCIM.160107
Received: October 10, 2024 Revised: January 03, 2025 Accepted: February 01, 2025

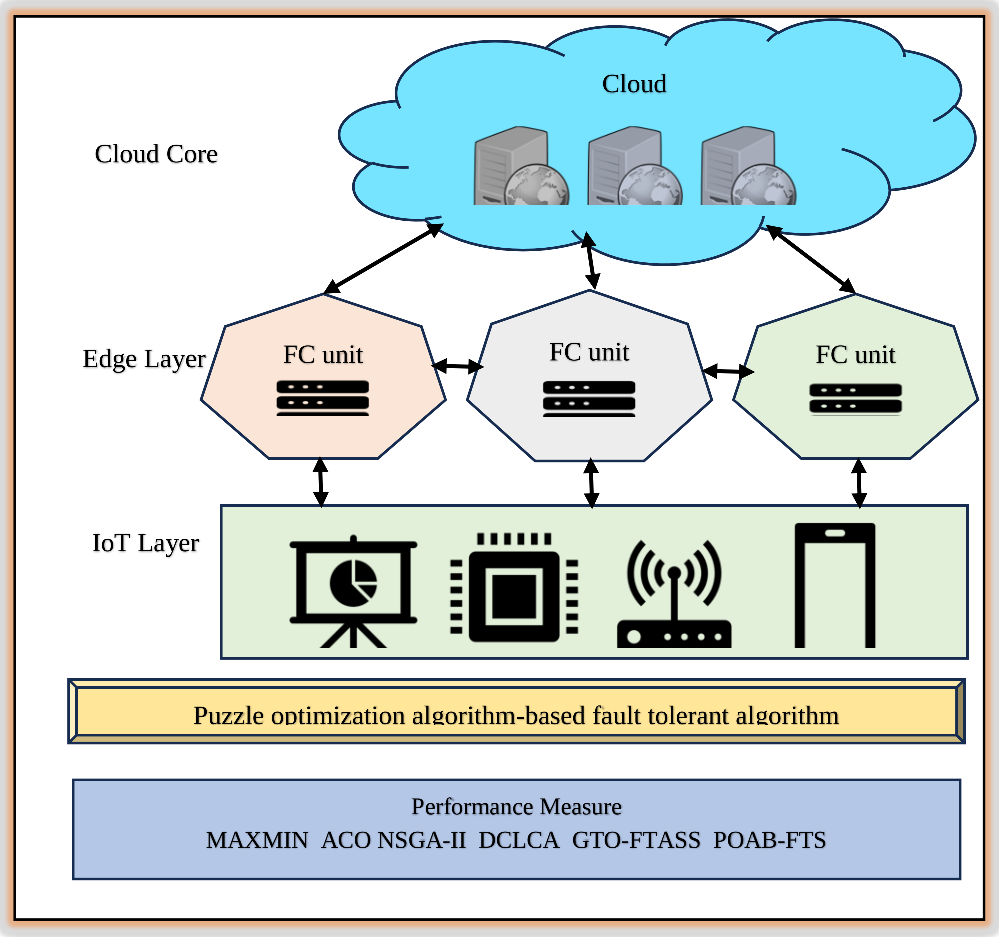

Figure 3. The POAB-FTS technique's overall process in its entirety

5.1. Fault Tolerant Model

Task failures may occur for a variety of reasons, including the inaccessibility of the substance, hardware problems, execution cost and duration surpassing threshold values, the framework operating out of memory or hard drive space, excessive utilization of resources, erroneous installation of key libraries, and other factors. It is assumed that these defects are autonomous and that they are either temporary or permanent. In order to develop a scheduling technique that is fault-tolerant, it is necessary to ensure that the purpose of each task that is generated by the method is accomplished before any fault occurs, even in the most negative of circumstances. We are aware that the replication method was widely applied for fault tolerance, which typically involves replicating work to a maximum of two copies and then distributing schedules to other hosts. Because of this, there is an increased likelihood of wasting resources and an increase in the usage of extraordinary amounts of energy. Consequently, in this scenario, the only jobs that can be reproduced are those that are prone to failure. It is possible to receive the three tasks that are successful in the queue of activities that are prone to failure, and it is possible to copy all of the jobs as three copies. In the next step, the vector reconstruction strategy is going to be used in order to recreate the super job in duplicate copies. It is possible to map the reconstructed supertasks to the hosts that are the best suitable for them, in addition to assigning them resources and scheduling them in separate hosts on an individual basis.

5.2. Strategy of POA

When it comes to solving optimization issues, the Population-Oriented Approach (POA) is a population-oriented strategy that involves mathematical modelling. The POA model belongs to the category of game-based models. Every single person in the population is considered a piece of a jigsaw to be solved in order to identify the variable that is influencing what is happening. A greater number of points are taken into consideration in order to finish the gathered puzzle superb parts. As an additional point of interest, it is exemplified by the primary function of "evaluation and value." The method that has been developed gives preference to the most qualified member of the population for providing guidance when completing all of the riddles.

DOI: https://doi.org/10.54216/JCIM.160107
Received: October 10, 2024 Revised: January 03, 2025 Accepted: February 01, 2025

Every single population that is included in the method that has been provided is a solution that is the greatest potential option for the optimization issue. Everyone in the population is responsible for determining the worth of the issue variable. Because of this, the matrix given in Equation (1) is applied for mathematical modelling.

$$A = \begin{bmatrix} A_1 \\ \cdots \\ A_2 \\ \cdots \\ A_n \end{bmatrix}_{n*m} = \begin{pmatrix} A_{1,1} & \cdots & A_{1,d} & \cdots & A_{1,m} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ A_{i,1} & \cdots & A_{i,d} & \cdots & A_{i,m} \\ A_{n,1} & \cdots & A_{n,d} & \cdots\cdots & A_{n,m} \end{pmatrix}_{n*m} \qquad (1)$$

Every individual in the population is responsible for determining the value of the primary function, which is the most effective answer to the issues that have been brought up. Therefore, the primary function may be assessed based on the function of those who are part of the population, and its significance can be estimated as follows.

$$A = \begin{bmatrix} E_1 \\ \cdots \\ E_2 \\ \cdots \\ E_n \end{bmatrix}_{n*m} = \begin{bmatrix} E(A)_1 \\ \cdots \\ E(A)_2 \\ \cdots \\ E(A)_n \end{bmatrix}_{n*1} \qquad (2)$$

On the basis of an assessment of the worth that was obtained for the primary purpose, the participant of the population who gives the superior value for the primary function was recognized as the best colleague of members. Equation (3) is utilized in order to determine which individual is the most qualified.

$$Y = A_k, E_k = min\,(E) \qquad (3)$$

Here A = puzzles population, A_i = ith puzzle, n = overall amount puzzle population, m = number of problem variable, $A_{i,d}$ = valus of dth parameter by ith puzzle, E = vector value for main function, E_i = value of puzzle in main function at ith location, Y = best member, A_k = kth puzzle with minimal detached equation.

According to the method that has been described, the population member is being improved twice. Through the use of the first phase, each and every person of the population is improved in accordance with the recommendations made by other people. During the subsequent stage, each participant of the populace tries to finish their problem by using puzzle pieces that have been recommended by other memberships. The following is a mathematical representation of the primary concept that underlies the first stage:

$$HG_i = A_h, h \in \{1,2,3,\ldots,n\} \qquad (4)$$

$$ex_{id} = \begin{cases} (HG_i - I * A_{id}), & E_h < E_i \\ (A_{id} - I * HG_{id}), & else \end{cases} \qquad (5)$$

$$I = round(1 + rand) \qquad (6)$$

$$A^{new} = A_i + r * eA_i \qquad (7)$$

$$A_i = \begin{cases} A_i^{new}, & E_i^{new} < E_i \\ (A_i), & else \end{cases} \qquad (8)$$

Each participant of the second phase participates in the process of upgrading their status by using the jigsaw pieces that are provided by other population organizations. The mathematical equation may be expressed as follows.

$$N_q = round\left(0.5 * \left(1 - \frac{t}{T}\right) * N\right) \qquad (9)$$

$$A_i^{new} = A_{h,e_j}, A_i = \begin{cases} h \in \{1,2,3,..,n\} \\ j \in \{1,2,3,.., n_p\} \\ e_j \in \{1,2,3,..,m\} \end{cases} \qquad (10)$$

In this case, N_q stands for the total number of proposed puzzle parts, r is the reiteration counter, T is the greatest number of repeats, $A_{i,e}^{new}$is the value representing the novel solutions for the dj'th length of the i'th puzzle, and $A_{h,e_j}$is the value representing the chosen solution from the h'th puzzle, where h is chosen at random. The technique is repeated, and the new position of the population is calculated once every member of the population has been upgraded using the first and second procedures. Let us pretend that the approach that has been suggested can be updated as it depends on repetition.

5.3. Procedure elaborates in POAB-FTS Procedure

The POAB-FTS method minimizes the processing time and likelihood of failure by computing a fitness function (FF). It is believed that the goal of cloud scheduling is to shorten the completion time, whereas the goal of the client is to lower the cost of accessing cloud resources by shortening the makespan hours; this is in order to construct the FF. Therefore, Equation (11) is used to determine the fitness levels of POAB-FTS using the FF:

DOI: https://doi.org/10.54216/JCIM.160107
Received: October 10, 2024 Revised: January 03, 2025 Accepted: February 01, 2025

$$fc = \max\left\{\bigcup_{i=1}^{m} c_i\right\} \qquad (11)$$

To increase the effectiveness of the model, the make-span should be reduced. This will outcome in a petite amount of time required to put the process into action. The execution time for all the tasks that need to be calculated on a virtual machine (VM) is what is meant to be characterized as the anticipated time of completion (ETC), which is obtained by the ETC matrix as follows. An ETC matrix that is connected with a number of tasks, denoted by Q = {Q_1, Q_2, . Q_n} and m VM, which is represented by $V = V_1, V_2, \ldots, V_m$} resource

$$ETC = Q.V = \begin{cases} Q_1V_1 & Q_1V_1 & \cdots & Q_1V_m \\ Q_2V_1 & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ Q_nV_1 & \cdots & \cdots & Q_nV_m \end{cases} \qquad (12)$$

$$p_f(Q_i, v_k) = \begin{cases} 0 & if\ SD_i <= QL_k \\ 1 - e^{-\alpha(SD_i - QL_k)} & if\ SD_i > QL_k \end{cases} \qquad (13)$$

The likelihood of failure of carrying out the job with a trust grade (TG) and a security requirement (SR) is often referred to as the combination of the two variables. The SR is a representation of the security requirements that must be met by the apps throughout the task submission process. The trust mechanism is responsible for determining the reliability of the virtual machine site, or more precisely, the TG. As a function of the difference amongst the machine's safety and the task requirement, a task failure method may be characterized as a term. The failure probability regarding the development of job X with a certain X value is described by Equation (16), which is compared to the X with a trust value of X. When the TG number is higher, the dependability of the resource virtual machine (VM) is more advanced. TG stands for the security guarantee that the resource VM provides.

6. Result and Analysis

The outcome of the case that uses the POAB-FTS technique reveals how effective this approach is when attempting to solve major difficulties in cloud computing settings. The method developed in this work proved to have better makespan, failure ratios, and failure slow down than previous methods for different volumes of tasks.

POAB-FTS's implementation of improvements in task scheduling, proactive, and reactive strategies for fault tolerance revealed that POAB-FTS was able to execute faster and more reliable. The fitness function being used to the model proved capable of reducing failure risks while at the same time optimizing resource utilization.

Comparatively, the present POAB-FTS technique was not only effective in enhancing the reliability and robustness attributes but in minimizing the utilization of resources and improving system output. These enhancements corroborate that POAB-FTS is an efficient strategy that supports the POAB model to propose efficient fault-tolerant cloud computing solutions to address the dependability issues of today's dynamic and complex cloud environments.

Accuracy: Over the measure of the total slips, it calculates the general accuracy of the model in the identification of a fault.

$$Acc = \frac{Correct\ positive + Correct\ negative}{Total\ cases} * 100 \qquad (14)$$

Precision: High value indicates the true positive ratio in relation to predicted positive results.

$$Pre = \frac{Correct\ positive}{correct\ positive + Incorrect\ positive} * 100 \qquad (15)$$

Recall: Shows how many of the actually positive cases the model was able to classify correctly.

$$Rec = \frac{Correct\ positive}{correct\ positive + Incorrect\ negative} * 100 \qquad (16)$$

Sensitivity: Sensitivity determines the percentage of true positive samples that have been classified accurately by the system. In the context of the GTO-FTASS approach, it represents the system's capability of identifying actual task failures and triggering the appropriate recovery processes.

$$Sen = \frac{Correct\ positive}{Correct\ positive + Incorrect\ negative} * 100 \qquad (17)$$

Specificity: Specificity acknowledges the ratio of true negative samples that have been correctly classified by the non-failing system. The last one measures the ability of the system in minimizing the chances of classifying healthy activities as failures.

$$Spe = \frac{Correct\ negative}{Correct\ negative + Incorrect\ positive} * 100 \qquad (18)$$

Failure Slowdown: Quantifies the time that was lost due to failures and compares the time when failures were present against the execution time that would have been achieved without them. GTO-FTASS, therefore, has a lower value where the system is more robust and recovers much faster when failures occur.

$$FSD = \frac{Time\ taken\ under\ Failures}{Time\ taken\ without\ Failure} * 100 \qquad (19)$$

Failure Ratio: Drives show the proportion of failed tasks. Lesser failure rates in GTO-FTASS mean better reliability as evidenced by high accomplishment rates of most tasks in contrast with non-GTO-FTASS algorithms.

$$FRR = \frac{Failed\ Tasks}{Total\ Tasks} * 100 \qquad (20)$$

Makespan: Stands for the total time taken to do all the activities on the project. Thus, a minimized makespan in GTO-FTASS indicates improved scheduling and resource utilization in tasks from the overall system standpoint.

$$MS = max\ (Task\ completion\ time) \qquad (21)$$

Table 1: Comparison of existing approach with the proposed approach

Method	**Accuracy (%)**	**Precision (%)**	**F1-Score (%)**	**Sensitivity (%)**	**Specificity (%)**
ACO	85.5	82.8	83.5	81.5	87.8
MAXMIN	87.6	84.5	85.3	83.8	88.2
DCLCA	89.2	86.7	87.5	85.8	90.2
GTO-FTASS	93.3	90.1	91.6	89.4	94.8
Proposed POAB-FTS	98.4	95.2	96.5	96.5	97.9

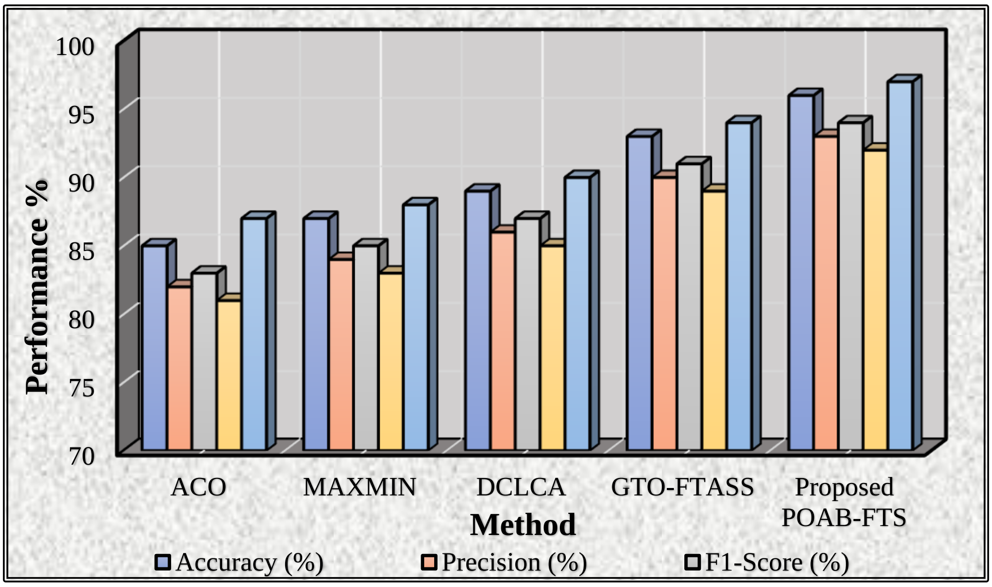

Figure 4. Illustration of comparing existing methods and proposed methods

The subsequent comparison of the proposed POAB-FTS technique with other approaches, such as ACO, MAXMIN, DCLCA, and GTO-FTASS, indicates its enhanced effectiveness and dependability. The results show that the proposed POAB-FTS model provides the highest accuracy of 98.4%, precision measure of 95.2%, F1-Score of 96.5, sensitivity of 96.5% and specificity of 97.9%. They show just how low false positive and negatives rates can be, and how high correct classifications can be for it. However, the result of GTO-FTASS is a better accuracy margin, which is 93.3% and F1-Score, 91.6% but still lower than the performance shown by POAB-FTS. Consequently, DCLCA, MAXMIN, and ACO produce lower values on all parameters, although the accuracy varies between 85.5 and 89.2 per cent. The results corroborate the

DOI: https://doi.org/10.54216/JCIM.160107
Received: October 10, 2024 Revised: January 03, 2025 Accepted: February 01, 2025

ability and reliability of POAB-FTS for fault-tolerant scheduling concerning resource allocation and execution in cloud computing scenario. Such enhancements against the previous methodologies underscore its ability to address systems that can be characterized as evolving or complicated and with high accuracy and reliability.

Table 2: Comparison of Makespan with the existing methods and proposed methods

No. of Tasks	ACO	MAXMIN	DCLCA	GTO-FTASS	POAB-FTS
20	3111	3137	2107	1806	1608
40	4940	4949	3110	2763	2653
60	5312	5250	3366	3434	3540
80	6199	5654	4213	4846	3608
100	6311	5897	4380	5100	3497

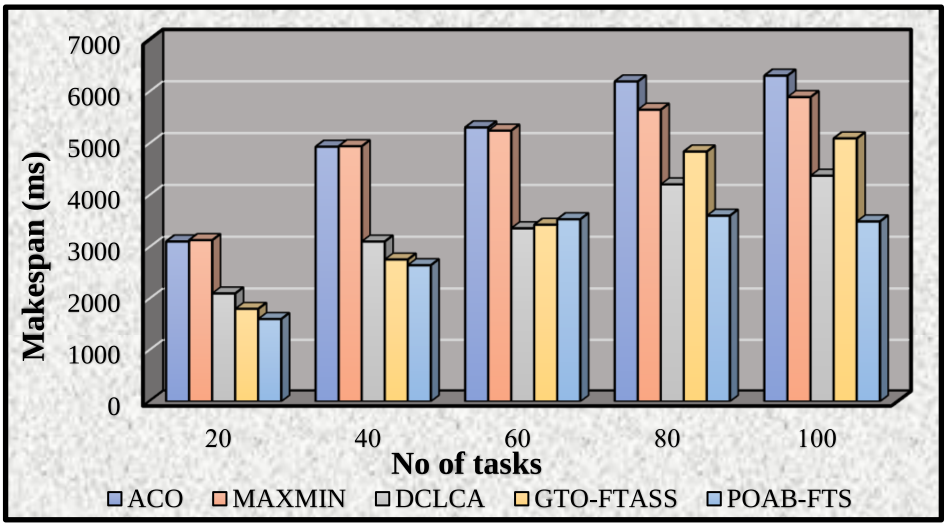

Figure 5. Illustration of Makespan comparing existing methods and proposed method

The purpose of Figure 5 is displaying makespan for all the other methods for unusual sizes of tasks is to prove the effectiveness of the proposed POAB-FTS technique. At the same time, POAB-FTS yields to GTO-FTASS in all investigated cases but has the minimum makespan of 1608 ms for 20 tasks that are far from GTO-FTASS, 1806 ms. At complexity increase, POAB-FTS still performs better than other schemes by having makespan values of 2653ms for 40 tasks, 3540ms for 60 tasks, and 3608ms for 80 tasks. The POAB-FTS takes 100 tasks and displays the smallest makespan: 3497 ms compared to GTO-FTASS: 5100 ms and DCLCA: 4380 ms. these results show that POAB-FTS is also effective for scalable application in reducing time taken during execution.

Table 3: Comparison of Failure ratio with the existing methods and proposed methods

No. of Tasks	ACO	MAXMIN	DCLCA	GTO-FTASS	POAB-FTS
20	65.84	63.1	55.1	52.5	37.5
40	47.8	45.2	40.1	22.1	20.1
60	38.1	31.5	25.1	20.1	17.5
80	35.5	30.1	24.5	19.1	10.2
100	31.5	28.1	15.2	15.01	3.1

DOI: https://doi.org/10.54216/JCIM.160107
Received: October 10, 2024 Revised: January 03, 2025 Accepted: February 01, 2025

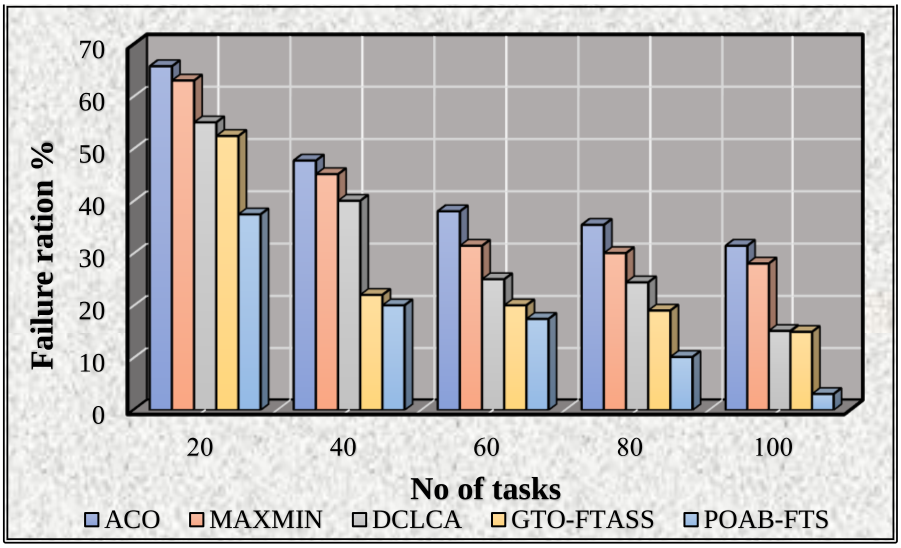

Figure 6. Illustration of Failure ratio comparing existing methods and proposed method

Comparing the failure ratio for the tasks of different sizes has been demonstrated in the Figure 6 that shows the substantially better relative tolerance of the proposed POAB-FTS. POAB-FTS has the lowest failure ratio, 37.5% for 20 tasks, better than GTO-FTASS with 52.5% and other methods. The special advantage of POAB-FTS grows with the amount of tasks: thus, for 40 tasks it has 20.1%, for 60 – 17.5%, for 80 – 10.2% ratios, which are much, lower than competitors. For 100 tasks, the failure ratio observed with POAB-FTS is 3.1 %, which is lower than GTO-FTASS with 15.01% as well as DCLCA with 15.2%. The following results confirm how POAB-FTS is very reliable and has an outstanding fault-tolerant scheduling.

Table 4: Comparison of Failure slowdown with the existing methods and proposed methods

No. of Tasks	**ACO**	**MAXMIN**	**DCLCA**	**GTO-FTASS**	**POAB-FTS**
20	3.1	3.5	2.1	1.1	0.6
40	6.5	5.5	4.4	4.4	2.21
60	9.4	9.5	8.9	8.8	4.88
80	13.2	13.1	12.4	12.8	5.22
100	15.2	15.2	13.2	13.5	5.57

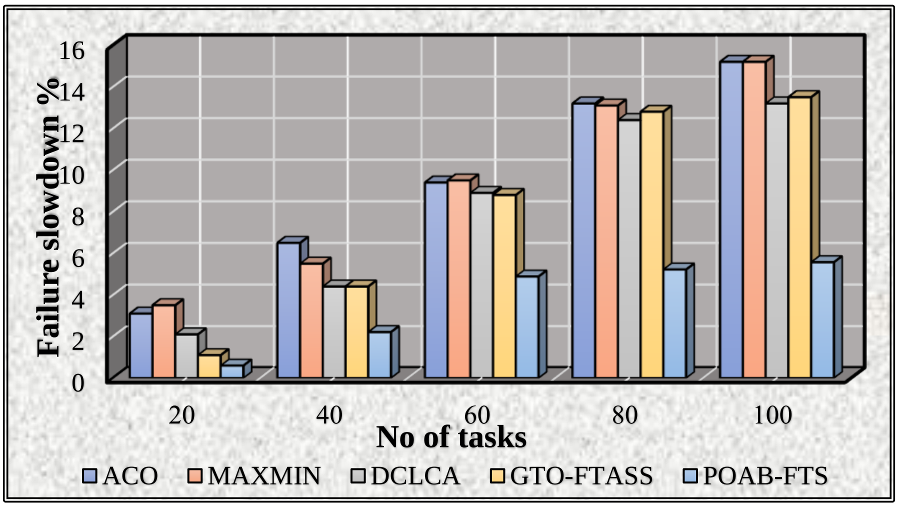

Figure 7. Illustration of Failure slowdown comparing existing methods and proposed method

DOI: https://doi.org/10.54216/JCIM.160107
Received: October 10, 2024 Revised: January 03, 2025 Accepted: February 01, 2025

The failure slowdown in Figure 7 shows how the POAB-FTS technique performs better than the other several methods in decreasing the number of delays cause due to failures. For 20 tasks, the lowest slowdown is figured for the POAB-FTS (equals 0.6), and it is significantly less than that of GTO-FTASS (equals 1.1) and DCLCA (equals 2.1). When the number of tasks rises to 40, 60, and 80, the pab scenario employing FTS yields, better results with the scores of 2.21, 4.88 and 5.22; hence better results than the competitive techniques, which vary between 4.4 and 13.2. Thus, POAB-FTS again emerges as the best algorithm at 100 tasks with failure slowdown of 5.57 as against 13.5 and 13.2 of GTO-FTASS and DCLCA respectively further legitimizing the use of fault tolerance.

7. Conclusion and Future Scope

The proposed Puzzle Optimization Algorithm-Based Fault Tolerant Scheduling (POAB-FTS) model presents an efficient solution for solving prominent issues in context of cloud computing including task failures and inefficient resource utilization. Through a game-based optimization methodology, the model maintains a flexible resource scheduling where the times of execution are also minimized while the reliability is increased which directly helps the businesses to keep the operations going and meeting the SLAs in the best possible manner. By decreasing overall downtime, efficiently utilizing resources, and decreasing the operational overhead, POAB-FTS model offers a better return on Investment (ROI) in contrast to other layout pattern strategies. In addition, the performance of fault-tolerance measures makes it possible to lower risks connected with interruptions in system activity and guarantee stable service provision that will help maintain buyer trust and market dominance. Benchmarking the proposed POAB-FTS framework with the best previous methods for failure ratio and execution time shows that the current model outperforms the previously proposed models and consolidates its position as a tactical key enabler of efficiency and cost-effectiveness. The study underscores the centrality of technology, financial and business objectives stressing appropriate resource deployment as well as cost control and increased reliability of services all of which shape value and competitiveness in the context of shifting dynamics of the cloud-computing environment.

Future work may incorporate the use of AI areal time predictive analysis of faults and remedial measures that facilitates self-healing clouds. Further, integrating the financial analytics tools into the conceptualization framework of schedule can help in achieving the balanced achieving the dual objectives of performance measurement and cost reduction in a dynamic way thus giving higher return on investment. Further, areas under investigation include opportunities with investigating cross-data center fault tolerance mechanisms and how blockchain may help provide additional data verification techniques and possibilities when dealing with inter-organizational collaborations, creating new business models, and revenue generating application possibilities.

References

[1] G. Jeeva Rathanam and A. Rajaram, "Trust Based Meta-Heuristics Workflow Scheduling in Cloud Service Environment," *Circuits and Systems*, vol. 7, pp. 520-531, 2016.

[2] N. Manikandan, N. Gobalakrishnan, and K. Pradeep, "Bee optimization based random double adaptive whale optimization model for task scheduling in cloud computing environment," *Computer Communication*, vol. 187, pp. 35–44, Apr. 2022.

[3] J. Liu, S. Wang, A. Zhou, S. Kumar, F. Yang, and R. Buyya, "Using proactive fault-tolerance approach to enhance cloud service reliability," *IEEE Trans. Cloud Computing*, vol. 6, no. 4, pp. 1191-1202, Oct.–Dec. 2018.

[4] C. E. Andrade, T. Silva, and L. S. Pessoa, "Minimizing flowtime in a flowshop scheduling problem with a biased random-key genetic algorithm," *Expert System Appl.*, vol. 128, pp. 67–80, Aug. 2019.

[5] G. Yao, Y. Ding, and K. Hao, "Using imbalance characteristic for fault-tolerant workflow scheduling in cloud systems," *IEEE Trans. Parallel Distrib. System*, vol. 28, no. 12, pp. 3671-3683, Dec. 2017.

[6] R. Zhang, F. Tian, X. Ren, Y. Chen, K. Chao, R. Zhao, B. Dong, and W. Wang, "Associate multi-task scheduling algorithm based on self-adaptive inertia weight particle swarm optimization with disruption operator and chaos operator in cloud environment," *Service Oriented Comput. Appl.*, vol. 12, no. 2, pp. 87–94, Jun. 2018.

[7] R. Vandana et al., "Detection of sleep apnea through heart rate signal using Convolutional Neural Network," *Int. J. Pharm. Res.*, vol. 12, no. 4, pp. 4829-4836, Oct.–Dec. 2020.

[8] G. Yao, Y. Ding, L. Ren, K. Hao, and L. Chen, "An immune system-inspired rescheduling algorithm for workflow in cloud systems," *Knowl.-Based Syst.*, vol. 99, pp. 39-50, 2016.

[9] A. Mubeen, M. Ibrahim, N. Bibi, M. Baz, H. Hamam, and O. Cheikhrouhou, "ALTS: An adaptive load balanced task scheduling approach for cloud computing," *Processes*, vol. 9, no. 9, p. 1514, Aug. 2021.

[10] R. Aron and A. Abraham, "Resource scheduling methods for cloud computing environment: The role of meta-heuristics and artificial intelligence," *Eng. Appl. Artif. Intell*, vol. 116, p. 105345, 2022.

DOI: https://doi.org/10.54216/JCIM.160107
Received: October 10, 2024 Revised: January 03, 2025 Accepted: February 01, 2025

[11] M. U. Sana and Z. Li, "Efficiency aware scheduling techniques in cloud computing: A descriptive literature review," *PeerJ Comput. Sci.*, vol. 7, p. e509, 2021.

[12] Z. Li, J. Ge, H. Hu, W. Song, H. Hu, and B. Luo, "Cost and energy aware scheduling algorithm for scientific workflows with deadline constraint in clouds," *IEEE Trans. Serv. Comput.*, vol. 11, no. 4, pp. 713-726, Jul./Aug. 2018.

[13] V. Roy and S. Shukla, "Effective EEG Motion artifacts Removal with KS test Blind Source Separation and Wavelet Transform," *Int. J. Biosci. Biotechnol.*, vol. 8, no. 5, pp. 139-154, 2016, DOI: 10.14257/ijbsbt.2016.8.5.13.

[14] H. Liu, P. Chen, X. Ouyang, H. Gao, B. Yan, P. Grosso, and Z. Zhao, "Robustness challenges in Reinforcement Learning based time-critical cloud resource scheduling: A Meta-Learning based solution," *Future Gener. Comput. Syst.*, vol. 146, pp. 18-33, 2023.

[15] S. Zhou, B. Yuan, K. Xu, M. Zhang, and W. Zheng, "The impact of pricing schemes on cloud computing and distributed systems," *J. Knowl. Learn. Sci. Technol.*, vol. 3, no. 3, pp. 193-205, 2024.

[16] M. Cinque, D. Cotroneo, L. De Simone, and S. Rosiello, "Virtualizing mixed-criticality systems: A survey on industrial trends and issues," *Future Gener. Comput. Syst.*, vol. 129, pp. 315-330, 2022.

[17] M. R. Shirani and F. Safi-Esfahani, "Dynamic scheduling of tasks in cloud computing applying dragonfly algorithm, biogeography-based optimization algorithm and Mexican hat wavelet," *J. Supercomput.*, vol. 77, no. 2, pp. 1214–1272, Feb. 2021.

[18] E. Khezri, R. O. Yahya, H. Hassanzadeh, M. Mohaidat, S. Ahmadi, and M. Trik, "DLJSF: Data-Locality Aware Job Scheduling IoT tasks in fog-cloud computing environments," *Results in Eng.*, vol. 21, p. 101780, 2024.

[19] Y. Kumar, S. Kaul, and Y. C. Hu, "Machine learning for energy-resource allocation, workflow scheduling and live migration in cloud computing: State-of-the-art survey," *Sustainable Comput. Informatics Syst.*, vol. 36, p. 100780, 2022.

[20] A. Tarafdar, M. Debnath, S. Khatua, and R. K. Das, "Energy and makespan aware scheduling of deadline sensitive tasks in the cloud environment," *J. Grid Comput.*, vol. 19, no. 2, pp. 1–25, Jun. 2021.

[21] X. Zhou, W. Liang, K. Yan, W. Li, I. Kevin, K. Wang, J. Ma, and Q. Jin, "Edge-enabled two-stage scheduling based on deep reinforcement learning for internet of everything," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3295-3304, 2022.

[22] M. R. Hossain, M. Whaiduzzaman, A. Barros, S. R. Tuly, M. J. N. Mahi, S. Roy, C. Fidge, and R. Buyya, "A scheduling-based dynamic fog computing framework for augmenting resource utilization," *Simul. Model. Pract. Theory*, vol. 111, Art. no. 102336, Sep. 2021.

[23] P. Kumar, A. Baliyan, K. R. Prasad, N. Sreekanth, P. Jawarkar, V. Roy, and E. T. Amoatey, "Machine Learning Enabled Techniques for Protecting Wireless Sensor Networks by Estimating Attack Prevalence and Device Deployment Strategy for 5G Networks," *Wireless Commun. Mobile Comput.*, vol. 2022, Article ID 5713092, 15 pages, 2022, doi: 10.1155/2022/5713092.

[24] M. Mukherjee, M. Guo, J. Lloret, R. Iqbal, and Q. Zhang, "Deadline-aware fair scheduling for offloaded tasks in fog computing with inter-fog dependency," *IEEE Commun. Lett*, vol. 24, no. 2, pp. 307–311, Feb. 2020.

[25] G. Gala, G. Fohler, P. Tummeltshammer, S. Resch, and R. Hametner, "RT-cloud: Virtualization technologies and cloud computing for railway use-case," in *Proc. IEEE 24th Int. Symp. Real-Time Distrib. Comput. (ISORC)*, Jun. 2021, pp. 105-113.

[26] M. Afrin, J. Jin, A. Rahman, A. Rahman, J. Wan, and E. Hossain, "Resource allocation and service provisioning in multi-agent cloud robotics: A comprehensive survey," *IEEE Commun*. Surveys Tuts, vol. 23, no. 2, pp. 842-870, 2021.

[27] A. Garbugli, L. Rosa, A. Bujari, and L. Foschini, "KuberneTSN: A deterministic overlay network for time-sensitive containerized environments," in *Proc. ICC 2023-IEEE Int. Conf. Commun*., May 2023, pp. 1494-1499.

DOI: https://doi.org/10.54216/JCIM.160107
Received: October 10, 2024 Revised: January 03, 2025 Accepted: February 01, 2025