# Multi Objective Ameliorated Repetitive Resource Allocation Algorithm for Cloud Resource Scheduling and Allocation

Dipa D. Dharmadhikari[1](✉) and Sharvari Chandrashekhar Tamane[2]

[1] Department of CS and IT, Dr. Babasaheb Ambedkar, Marathwada University Aurangabad, Aurangabad, India
dharmadhikari2021@gmail.com

[2] MGM's Jawaharlal Nehru Engineering College MGM University, Aurangabad, India

**Abstract.** Mapping huge jobs onto cloud resources is a part of workflow scheduling, which increases scheduling effectiveness. Numerous researchers have been working hard to enhance the efficiency of scheduling in cloud computing as a result of this piqued interest. Scientific workflows, on the other hand, are huge data applications, therefore the executions are costly and time-consuming. Thus, a novel Multi Objective Ameliorated Repetitive Resource Allocation Algorithm that can quickly respond to unforeseen needs has been proposed in order to enhance the system's efficiency in allocating work tasks. Resource performance and resource proportion matching distances are also established in order to achieve resource optimization and the balanced use of all available resources. The results of the simulation show that the suggested method can efficiently complete Virtual Machine (VM) allocation and deployment and well manage incoming streaming workloads with a random arriving rate. Compared to small and medium workflow jobs, the suggested algorithm performs much better in big and extra-large workflow tasks. The experimental findings demonstrate that our algorithm is capable of balancing the consumption of all types of resources while allocating resources swiftly and optimally for unexpected demands.

**Keywords:** Resource Allocation · Task Scheduling · Cloud Computing

## 1 Introduction

The IT industry is changing as a result of cloud computing. Rather than unpacking computers and stacking them in a machine room, the cloud can download virtually equipment and related infrastructure. This makes it possible to build a datacentre in a matter of minutes with little technical expertise and for a small portion of the price of buying a single server. Sharing resources, software, and data is a developing trend in the world of technology [1]. In cloud computing, tasks from different systems are relocated there so that various systems can communicate with one another simultaneously [2]. Pay for only the services you utilise via cloud computing, or on a per-use basis [3]. The cost of the customer's hardware, software, and other services would go down as a result.

S. Tamane et al. (Eds.): ICAMIDA 2022, ACSR 105, pp. 403–414, 2023.
https://doi.org/10.2991/978-94-6463-136-4_34

The on-demand supply of infrastructure and other services is a promise made by cloud computing to help with the creation, deployment, and adaptive provisioning of applications. Additionally, it guarantees cost savings, resource management flexibility, and the capacity to deliver an almost endless amount of resources [4]. As a result, it has received widespread adoption, resulting in the migration of numerous applications to the Cloud. Even though cloud computing provides many advantages, like cost savings, scalability, stability, and ease of maintenance, there are also some disadvantages [5]. Although the performance of these services was not always as expected or promised, the earliest Cloud computing platforms and providers offered capabilities that fostered vendor lock-in [6].

The integrity and privacy of the information kept in the cloud are at risk as a result of various users and parties accessing it [7]. Data distribution over computers is offered by the cloud. When a user sends data to the cloud for processing, control over the information is given to a third party who might not be able to satisfy the user's security concerns. A user cannot physically access his or her data, therefore he is uninformed of its location and unsure of whether the integrity of his or her data is protected in the cloud or compromised [8]. It is crucial to make sure that the data being handled on the cloud is secure and that there is no data tampering when potentially unfamiliar parties may be present. The main barrier preventing a wider use of cloud computing is still concerns about the security of the data kept there [9]. Because cloud computing uses a variety of technologies, which include databases, networks, OSs, virtualization, resource scheduling, transaction management, load balancing, concurrency control, and memory management, it has several security encounters. Due to the widespread use of these technologies, even a minor security flaw in one of them has the potential to bring the entire system to a halt [10].

In response, multi-cloud computing emerged [11], promising to resolve the for mentioned problems. Applications can be launched one at a time, one cloud at a time, using this type of computing. In addition to preventing vendor lock-in, this enables consumer proximity by dispersing the application across numerous physical locations. Additionally, it makes it possible for app developers to choose cloud service providers with higher levels of service delivery. Additionally, it enables a better fit between requirements and preferences for applications, resulting in apps with increased performance. Cross-cloud computing, a particular type of multi-cloud computing, has also been presented [12, 13], promising to deploy apps each time in not one but several cloud providers. As a result, application developers can choose the best cloud services to realise the functionality of their apps, which has the unavoidable benefit of reaching actual optimality. Based on the for mentioned study, multi-cloud computing is currently gaining traction, leading to the development of numerous multi-cloud management platforms (MCMPs) [14].

The format of this article is as follows: The relevant works are discussed in Sect. 2. The proposed concept and its system architecture are presented in Sect. 3. Evaluation procedures and results are described in Sect. 4. Section 5 brings the paper to a conclusion.

## 2 Literature Survey

For increased security and privacy of cloud data, a hybrid system with a built-and-deployed multi-cloud hosting environment was designed by Pachala et al. [15]. There are three modules in the hybrid approach. (a) An autonomous byzantine protocol that can withstand server outages, security breaches, and cloud. (a) The DepSky architecture uses encoding and decoding methods for enhancing the dependability & secrecy of information stored. (c) Shamir secret sharing technique to enhance privacy & trustworthiness of data storage without affecting efficiency However, the method has to be verified for several sophisticated access control techniques in the future and then compared with current protocol settings.

Selvapandian et al. [16] presented a hybrid optimised resource allocation model that uses the particle swarm and bat optimization algorithms to assign resource while taking the resource status, bandwidth, and task needs into consideration. The effectiveness of the suggested model is assessed through simulation and contrasted with that of traditional optimization strategies. The suggested method allocates resources in 47 s for a set of 500 activities, using a minimum of 200 kWh of energy. Regarding missing deadlines, resource usage, energy consumption, and time allocation, the suggested approach performs noticeably better than conventional techniques. This research project can further be expanded by incorporating hybrid deep learning methodologies for enhanced performances.

A unique, trust-building brokering architecture for various cloud settings called Healthy Broker was proposed by Kurdi et al. This architecture was created especially for cloud-based, patient-focused eHealth services. Care team members can use it to perform safe eHealth transactions and acquire “need-to-know” access to pertinent patient data in accordance with data-protection laws. By evaluating the trust connection and monitoring it with a neutral, impermeable dispersed blockchain record, Healthy Broker also guards against potentially harmful behaviour. Two methods are used to evaluate trust. First, for transparency, a distributed ledger is used to log and audit all transactions and user input. Second, only comments made by reliable sources are considered. E-Health multi-cloud simulation was used to evaluate Healthy Broker. Future study on harmful behaviours should consider Sybil attacks and nefarious spies, among other forms of bad conduct.

Sahbudin et al. created a Web Client application that emphasises the data integrity problems that can occur while using the current multi-Cloud storage services. It was mentioned that a multi-Cloud combines various points of access to a cloud storage account or provider. The framework provides a cutting-edge platform that allows the user to simultaneously have flexibility and security. A variety of Web application experiments were conducted in a real multi-Cloud environment in order to evaluate the concept. Integrating the SSME environment at the operating system level, which will offer a storage abstraction in the form of a file directory structure, will further the research’s objective.

Bal et al. suggested combining resource allocation security with effective task scheduling in cloud computing using a hybrid machine learning (RATS-HM) technique. The following are the proposed RATS-HM techniques: First, a short scheduler for task scheduling (ICSTS) based on an enhanced cat swarm optimization method reduces makespan time and increases throughput. Second, bandwidth and resource load are included in a group optimization-based deep neural network (GO-DNN) for effective resource allocation under various design restrictions. Third, NSUPREME, a simple authentication technique, is suggested for data encryption to secure data storage. The usefulness of the suggested RATSHM technique is then demonstrated by simulating it using a different simulation setup and comparing the results to cutting-edge methods. The usefulness of the suggested model in a real-world scenario will be determined in the future using a significant amount of actual data in a real cloud environment.

From the survey it is clear that the multi-cloud network still faces certain issues in order to function effectively, and hence a novel approach has been adopted to address the aforementioned challenges, which is further covered in Sect. 3.

## 3 Algorithm: Multi Objective Ameliorated Repetitive Resource Allocation Algorithm

A multi-objective evolutionary algorithm can solve a mathematical issue involving multi-objective optimization. The algorithm should also hasten the solution process and improve the quality of the solution set to ensure timeliness and optimise resource allocation for unforeseen requests. However, the resource allocation multi-objective optimization approach has the following drawbacks. The existing works take too long to compute the values of the goal functions; hence they cannot satisfy emergent resource needs. Individuals from the parent and offspring populations may repeat themselves during population evolution. When the parent and offspring individuals are combined, the repeating individuals are given the same hierarchical rank and have non-dominant connections. These individuals may have crowding distances that are bigger than those of the non-repeating individuals, which will cause many repetitive solutions to enter the ideal solution set. These individuals might be picked to make up the population of the following generation. We therefore suggest an Ameliorated Repetitive Resource Allocation Algorithm (ARRAA) that simultaneously computes the fitness values of individuals, eliminates repetitive individuals, and picks neighbouring great individuals in order to enhance the quality of the solution set. Additionally, this technique speeds up the solving process while enhancing the homogeneity of the solution set's distribution. As a result, the timely and efficient distribution of resources for urgent needs is further ensured.

### 3.1 Evaluation of the Fitness Function

Because there are too many individuals in the population, it takes a very long time to compute and evaluate the fitness values. The concurrent evaluation and calculation of individual fitness values by multi-core processors can hasten the convergence rate of the suggested approach. The following is how we determine and assess each individual's fitness values (i.e., objective functions).

$$fv_1 = \sum_{i=1}^{m} \sum_{j=1}^{n} M_{ij}$$

$$fv_2 = \sum_{i=1}^{m} \sum_{j=1}^{n} RE_{ij}$$

$$fv_3 = \sum_{i=1}^{m} \sum_{j=1}^{n} RN_{ij}$$

where,

RE is resource performance.

RN is resource proportion.

M is mapping element between PM and VM.

### 3.2 Individual Selection Based on Threshold

Some individuals remain even after all individuals are arranged in a non-dominant order and the repeated individuals who contribute to the solution set's non-uniformity distribution have been removed. As a result, we should continue to pick exceptional individuals to join the population of the following generation. Here, we measure the Euclidean distance between two nearby people to see if it is less than a predetermined distance. If so, the excellent individual is chosen using the following technique and a threshold by calculating the maximum Euclidean distance $Max_{ED(rank(i))}$ of two individuals of the non-dominant set and the threshold T which is calculated from

$$T = \frac{Max_{ED(rank(i))}}{2{*}S}$$

where S is the size of population.

Furthermore, it was shown that the majority of the available heuristic algorithms are not strong enough to simultaneously optimise competing objectives like execution cost and make span. A method that comprises of three sub-algorithms is proposed with the aim of decreasing the system execution cost, time, and fully utilising the resources, while ensuring that all tasks meet their deadlines namely: Task Excruciating Algorithm, Least VM (LVM) assortment, and Extreme VM (EVM) assortment.

The algorithm is started by identifying all the tasks in the Wline/Wqueue with a set of work-flow tasks W = (w1, w2, w3 wt4……….. wn) with their corresponding execution lengths (L1, L2………..Ln) in a Wline. These activities must be carried out using cloud resources (VMs). We presupposed that there are only two sizes of VMs in the suggested approach that is VMLeast and VMextreme. We considered the set of VMs with their corresponding sizes as (VMExtreme, VMLeast………. VMxx) based on instructions per second. The method determines the predicted finishing time (PFT) of each workflow task in the Wline when a task enters the queue. Then, based on their user-specified deadlines, all the jobs in the wline are re-queued in a new line called a deadline (Dline). The next phase involves comparing each process task's PFT to its user-specified deadline. If the PFT of Wi on VM is more than the deadline, the algorithm will split the work into smaller tasks and map the smaller tasks using the least VM selection technique (LVM). The LVM assortment method is used to assign tiny activities to less expensive virtual machines. In order to lower the execution cost, it is avoided to schedule minor jobs on expensive VMs. On the other side, the EVM selection method will be used to complete the work more quickly if the PFT of Wi is equal to the deadline. With the

EVM selection strategy, all process tasks with equal ECT are mapped to their deadlines on higher VMs in order to shorten the make span.

Here, in order to shorten their scheduling times, we divide jobs into subtasks. The workflow tasks queue's newly arrived workflow tasks are first identified by the algorithm. The deadline and PFT for each task are then compared. The task is subsequently broken down into smaller tasks and added to the decision-ready queue. The algorithm is completed once all key workflow tasks have been successfully partitioned.

### 3.3 Algorithm: ARRA Algorithm

**Input:** workflow tasks (w1, w2, w3,……wn).

```
Output:
1 Start.
2 For wline = (w1, w2, w3,……wn).
3 VM queue = (vmj, vmK……… Vmn).
4 foreach VM in the VMqueue do.
5 get the IPS for every VM.
6 if IPS > ExtremeVM. Get IPS then.
7 ExremeVM = VM;
8 minVm = getVmSize.
9 foreach vm in vmAllocation.keyset do.
10 get the IPS for every VM.
11 if IPS of VMk in the VMAllocation.keyset < MaxVM then.
12 minVM = VM;
13 Compare PFT of wi to its deadline.
14 if the PFTwi > Dlwi then.
15 Split the wi into sub-tasks.
16 insert the partitioned task to the w line (wt1, wt2, wt1 + 1….wtn).
17 else.
18 if PFTwti = < Dlwi.
19 Add task in the wline for accomplishment.
20 Update the wline.
21 while wline is not empty do.
22 Repeat step 4 to 10.
23 end.
24 if $w_{line}$ is empty.
25 end.
```

The EVM selection method finds VMs in the VM queue that have a greater capacity for execution to complete jobs for which their PFTs are equal to their due dates. This approach aims to shorten the total execution time by reducing the waiting periods for processing processes with long execution times. The LVM selection method is used to map all tasks whose PFTs are below the user-specified completion dates. This is accomplished in order to avoid mapping smaller workflow jobs onto VMs with higher execution capacities, which would result in higher charges and perhaps increase execution cost. The algorithm first calculates the IPS of the VMs, which is then used to discover the

VMs with lower IPS (cheaper VMs) in the VM list in order to maximise profit. Then, all short-execution-time tasks are transferred to the less expensive VMs.

The quantity of workflow jobs and the size of the VMs determine the proposed algorithm's overall time complexity, which depends on execution time and cost complexity.

## 4 Results and Discussion

This research used CloudSim 3.0 module for simulation. CloudSim 3.0 provides power model and cloud simulation with predetermined performance (Fig. 1). The CloudSim power package is assessed for its ability to simulate VM-aware allocation method for power measurement and fault with developed algorithm programme. The power model is directly dependent on service consumption (Fig. 2). The created approach is evaluated in terms of programming with efficient power management strategy in servers utilising the fundamental DVFS (dynamic voltage and frequency scaling). Testing is carried out for request counts of 0, 200, 400, 600, 800, 1000, and 1200 for the suggested approach. In this experiment, three resource types—CPU, Memory, and disk—are taken into account (Fig. 3).

The ratio of the total number of CPUs that are accessible to the entire number of CPUs that are needed to execute the task is known as CPU usage.

$$C = \sum_{i=1}^{m} \frac{R_i}{A_i}$$

where Ai is the total number of CPUs that are available, and Ri is the CPU that has been requested to do the assignment. For VM requests ranging from 0 to 1200, it is anticipated from the figure that the CPU utilisation will be between 37 and 45%. For 1000 VM requests, the maximum value is reached at 45%, while for 400 VM requests, the minimum value is obtained at 37%.

Utilisation of memory is referred to as the percentage of the amount of memory requested to complete the work to the total amount of memory available in the cloud,

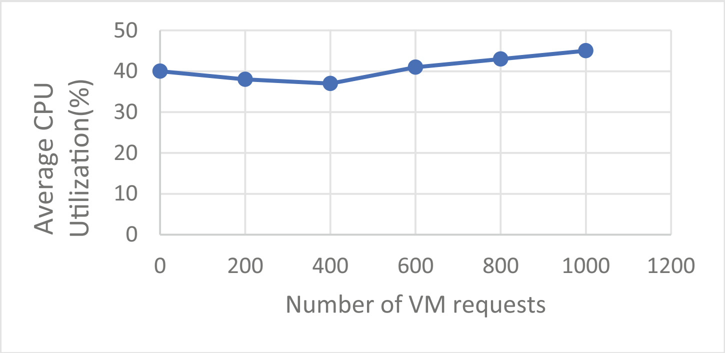

**Fig. 1.** CPU utilization

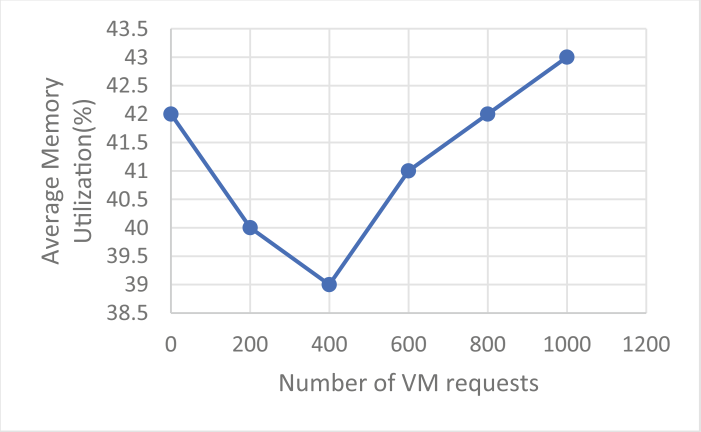

**Fig. 2.** Memory utilization

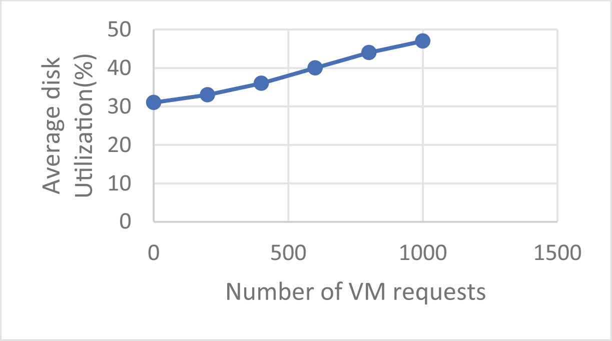

**Fig. 3.** Disk utilization

and it is computed as follows:

$$M = \sum_{i=1}^{m} \frac{x_i}{y_i}$$

where yi represents the total amount of memory that is accessible and xi represents the memory that is needed to complete the task. The plot shows that for 400 VM queries, the memory use is only 39%. The typical utilisation ranges from 39% to 43%, and memory usage tends to rise after 400 VM queries.

The disk utilization of the system is depicted from the above plot. It is identified that the average disk utilization tends to increase as the number of VM requests increase. The value ranges from 30% to 47% (Fig. 4).

In this experiment, we make comparisons of various categories having the same number of virtual machines (Fig. 5).

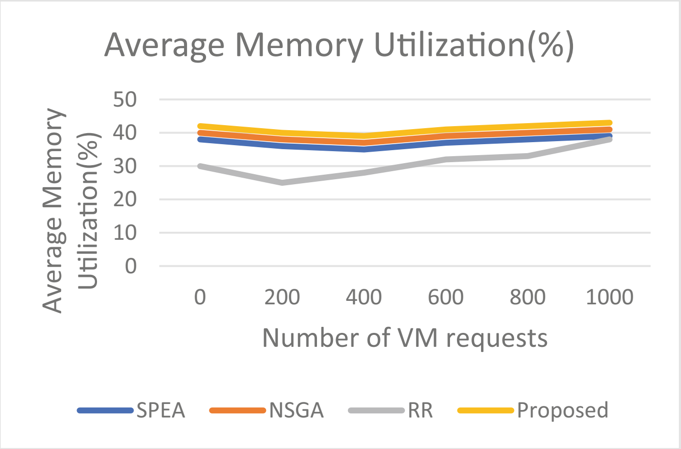

**Fig. 4.** Comparison of Memory utilization

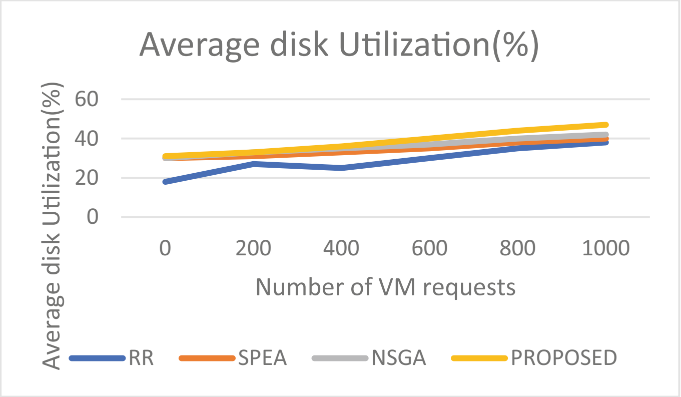

**Fig. 5.** Comparison of disk utilization

The system's efficiency in using the CPU, memory, and disc is compared to previous works like Round Robin, the Strength Pareto Evolutionary Algorithm, and the Non-dominated Sorting Genetic Algorithm (NSGA). The CPU, memory, and disc utilisation of the proposed algorithm slightly increases when the number of VM requests rises, which is favourable for a cloud platform's resource efficiency and stability. The suggested algorithm also takes into account how different resource categories' resource proportions match (Fig. 6)

Harmony Inspired Genetic Algorithm (HIGA), Harris Hawks Optimization and Simulated Annealing Algorithm (HHOSA), Cost-Effective Firefly based Algorithm (CEFA), Cuckoo Scheduling Algorithm are compared with the accuracy of proposed algorithm as shown in Fig. 7. The results depict that the proposed method performs better than the conventional method.

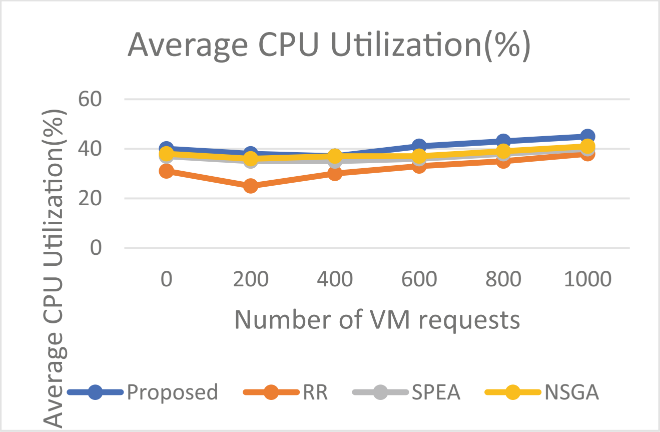

**Fig. 6.** Comparison of CPU utilization

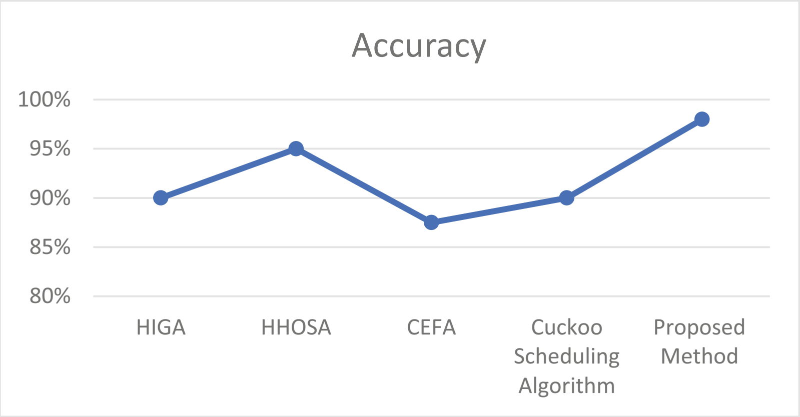

**Fig. 7.** Comparison of accuracy

## 5 Conclusion

The growth of cloud computing, big data, and artificial intelligence has led to a demand for cloud resources that exhibits diversity, and uncertainty. Undoubtedly, such sudden resource demands frequently occur on cloud platforms, requiring the swift and effective allocation of resources. This paper offers a multi-objective optimization cloud resource allocation technique to achieve resource optimization and balanced consumption of all resource kinds. A multi-objective mathematical model that minimises three objectives in order to maximise resource utilisation is offered to ensure the precision and efficacy of resource allocation. The solutions are found more quickly, and the quality and uniformity of the distribution of the solution set are improved by the algorithms that are provided. The CPU utilization for 1000 VM requests, the maximum value is reached at 45%, while for 400 VM requests, the minimum value is obtained at 37%. The typical utilisation ranges from 39% to 43%, and memory usage tends to rise after 400 VM queries. The disk utilisation value ranges from 30% to 47%.

## References

1. Mohammad, Omer Jasim, K.: Recent trends of cloud computing applications and services in medical, educational, financial, library and agricultural disciplines. Proceedings of the 4th International Conference on Frontiers of Educational Technologies (2018).
2. Tabrizchi, Hamed, and Marjan Kuchaki Rafsanjani: A survey on security challenges in cloud computing: issues, threats, and solutions. The journal of supercomputing 76(12), 9493-9532 (2020).
3. Lowe, Devesh, and Bhavna Galhotra: An overview of pricing models for using cloud services with analysis on pay-per-use model. International Journal of Engineering & Technology 73(12), 248-254 (2018).
4. Kritikos, Kyriakos, and Pawel Skrzypek: Simulation-as-a-service with serverless computing. 2019 IEEE World Congress on Services (SERVICES) 2642, (2019).
5. Heidari, Arash, and Nima Jafari Navimipour: Service discovery mechanisms in cloud computing: a comprehensive and systematic literature review. Kybernetes (2021).
6. Investments, A.I.: Are Cloud Platforms Ready for Multi-cloud?. Service-Oriented and Cloud Computing 56.
7. Han, Si, Ke Han, and Shouyi Zhang: A data sharing protocol to minimize security and privacy risks of cloud storage in big data era. IEEE Access 7, 60290-60298 (2019).
8. Awaysheh, Feras, M., et al.: Security by design for big data frameworks over cloud computing. IEEE Transactions on Engineering Management (2021).
9. Bello, Sururah, A., et al.: Cloud computing in construction industry: Use cases, benefits and challenges. Automation in Construction 122, 103441 (2021).
10. Zibouh, Ouadia, Anouar Dalli, and Hilal Drissi: Cloud computing security through parallelizing fully homomorphic encryption applied to multi-cloud approach. Journal of Theoretical and Applied Information Technology 87(2), 300 (2016).
11. Hong, Jiangshui, et al.: An overview of multi-cloud computing. Workshops of the international conference on advanced information networking and applications. Springer, Cham (2019).
12. Ahmed, Usama, Imran Raza, and Syed Asad Hussain: Trust evaluation in cross-cloud federation: Survey and requirement analysis. ACM Computing Surveys (CSUR) 52(1), 1-37 (2019).
13. Huang, Qinlong, et al.: Adaptive Secure Cross-Cloud Data Collaboration with Identity-Based Cryptography and Conditional Proxy Re-Encryption. Security & Communication Networks (2018).
14. Kritikos, Kyriakos, Paweł Skrzypek, and Marta Różańska: Towards an integration methodology for multi-cloud application management platforms. Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion (2019).
15. Pachala, Sunitha, Ch Rupa, and Sumalatha, L.: An improved security and privacy management system for data in multi-cloud environments using a hybrid approach. Evolutionary Intelligence 14(2), 1117-1133 (2021).
16. Selvapandian, D., and Santosh, R.: A Hybrid Optimized Resource Allocation Model for Multi-Cloud Environment Using Bat and Particle Swarm Optimization Algorithms. Computer Assisted Methods in Engineering and Science 29(1–2), 87-103 (2022).
17. Kurdi, Heba, et al.: Healthy broker: A trustworthy block chain-based multi-cloud broker for patient-centered ehealth services. Electronics 8(6), 602 (2019).

18. Sahbudin, Murtadha Arif Bin, Riccardo Di Pietro, and Marco Scarpa. A web client secure storage approach in multi-cloud environment. 2019 4th International Conference on Computing, Communications and Security (ICCCS). IEEE (2019).
19. Bal, Prasanta Kumar, et al. A Joint Resource Allocation, Security with Efficient Task Scheduling in Cloud Computing Using Hybrid Machine Learning Techniques. Sensors 22(3), 1242 (2022).