

Load Balancing Based on Multi-Agent Framework to Enhance Cloud Environment

Shrouk H. Hessen^{1,2,*}, Hatem M. Abdul-kader¹, Ayman E. Khedr³ and Rashed K. Salem¹

¹Information Systems Department, Faculty of Computers and Information, Menoufia University, Shebin Elkom, Egypt

²Information Systems Department, Faculty of Computers and Artificial Intelligence, South Valley University, Hurgada, Egypt

³Information Systems Department, Faculty of Computers and Information Technology, Future University in Egypt (FUE), Cairo, Egypt

*Corresponding Author: Shrouk H. Hessen. Email: shrouk.hessen@fcih.svu.edu.eg

Received: 11 June 2022; Accepted: 01 August 2022

Abstract: According to the advances in users' service requirements, physical hardware accessibility, and speed of resource delivery, Cloud Computing (CC) is an essential technology to be used in many fields. Moreover, the Internet of Things (IoT) is employed for more communication flexibility and richness that are required to obtain fruitful services. A multi-agent system might be a proper solution to control the load balancing of interaction and communication among agents. This paper proposes a multi-agent load balancing framework that consists of two phases to optimize the workload among different servers with large-scale CC power with various utilities and a significant number of IoT devices with low resources. Different agents are integrated based on relevant features of behavioral interaction using classification techniques to balance the workload. A load balancing algorithm is developed to serve users' requests to improve the solution of workload problems with an efficient distribution. The activity task from IoT devices has been classified by feature selection methods in the preparatory phase to optimize the scalability of CC. Then, the server's availability is checked and the classified task is assigned to its suitable server in the main phase to enhance the cloud environment performance. Multi-agent load balancing framework is succeeded to cope with the importance of using large-scale requirements of CC and (low resources and large number) of IoT.

Keywords: Cloud computing; IoT; multi-agent system; load balancing algorithm; server utilities

1 Introduction

Cloud Computing (CC) is a technology that can allow the migration of many virtual machines with quick access anywhere and anytime efficiently [1,2]. This technology needs a load balancing algorithm to balance the load on the servers to achieve the high performance of using CC [1]. There



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

are many load balancing algorithms to be used for minimizing response time depending on the utilities of resources. These algorithms can help CC to improve its performance and scalability, which is considered a big issue for flexibility and usability [1,3,4].

Cloud computing is used in demand so users should pay for services to get access easily and flexibly [5,6]. Users face many challenges and issues to deal with cloud computing technology because of workload so CC technology requires many experiments to improve and develop its usability of this technology [7,8]. Load balancing algorithms become the key factor to achieve maximum performance for the cloud environment. These algorithms have some metrics and parameters that will be discussed next in the load balancing section to cover the challenges of cloud computing [1,5,9].

Storage and power computation are considered the main two lines to track the services provided to serve the requirements of users [10]. Cloud computing is not only a service provider but also is considered the highest required technology to serve various types of heterogenous work in our life [11]. When researchers and users define the concept of cloud computing, load balancing techniques should be used, which are related to others because of the high relation between CC challenges and load balancing benefits [12]. There is growth traffic over a network because of various users and their requirements may be happened in a cloud environment [4,13]. It is considered a pool of services trailed to be used for solving many problems of end-users and programmers who use different devices [14–16]. Every day, load balancing algorithms and methods are proposed to face the growth and challenges of the cloud environment because of the fast changes in data types and their load. The next paragraph will present the Internet of Things (IoT) used in the cloud environment and how an effective multi-agent system that contains different agents helps information technology achieve its target.

The Internet of Things (IoT) is employed as a flexible tool for global circulated services over the network used in many fields (medical, financial, education, engineering, etc.) [16–18]. IoT is a smart communication and complex device that supports low resources. It is considered the opposite face of cloud computing that needs a large-scale of computer storage and power. The high usage of IoT devices for accessing the resources of data stored and transmitted over a network in a cloud environment increases the opportunity of penetration for data resources and private information so the security aspect is needed during their storage [19,20].

Multi-agent system consists of a group of agents that co-operated with others to get the target of the decision. It can deal with autonomous data and applications used in many different fields. Multi-agent is considered an intelligent system to analyze the data and process it effectively to be ready for achieving the target of preparing the data collection [21].

Based on the great changes in the technological environment and the increasing human requirements, an intelligent system is needed to face the influence of tasks. The different tasks and their diversity belonging to types, sizes, and complexity should be analyzed and processed to be used for the scalability of the cloud environment [22]. There is the problem of dealing with dynamic information technology because of the large amount of data besides the differences in nature. A multi-agent system is used to deal with a large amount of data with different resources and types to prepare the data effectively [21].

This paper is organized as follows: Section 2 illustrates cloud environment performance and load balancing types and classification. The related work of load balancing algorithms and applied domains are investigated in Section 3. The proposed framework details are provided in Section 4. Section 5 presents the steps of load balancing algorithm design. Dataset collection, feature selections, testing, validation, and the details of experiment evaluation are discussed in Section 6. Finally, the conclusion and future work are in Section 7.

2 Literature Review

The cloud environment has related or connected requests that may be accessed by users [23]. These requests have their specific nodes which may be overloaded. Load balancing algorithms could solve the issues of the workload but may suffer from some challenges in the cloud [24,25]. These challenges can make a negative impact on the cloud environment [7]. Scalability is one of the most critical issues that should be solved to improve the response time belongs to storage and power. Load balancing algorithms, analyzing the performance, and modeling are required to improve the services of CC [26,27].

The pre-processing of data is needed for the cloud environment before starting to distribute the data over the network to balance the load [1]. This pre-processing is considered a problem-solving of a multi-agent system. The different agents can be interacted together for gaining integrated data, then the data could be extracted and split into training and test sets to gain relevant features of the multi-agent system to solve the users' problems in the cloud environment [21].

2.1 Cloud Performance

Load balancing is a critical step for achieving the highest performance in the cloud computing environment. The utilities of any hosted machine in cloud computing have many parameters to be evaluated to avoid workload [1,28]. These parameters play an important role in proposing load balancing algorithms in a cloud environment which could be discussed as follows [2,9,29]: response time indicates the time from the user starts to request the task and is completed on the server. Latency is the delay time over the network that occurred during starting the task from a computer and completed on the server. Bandwidth describes the maximum amount of data and traffic during the transfer between end-users and servers per unit of time. Hits are considered as load generators which is the most important parameter to measure scalability. Migration time is the time required to transfer the request from a virtual machine to another one based on workload. Fault tolerance is the capability to balance the load among nodes. Imbalance indicates the degree of differences between virtual machines. Span is the time required for a user-based resource allocator. After the metrics of load balancing are discussed, the performance of load balancing algorithms should validate the utilities of resources in the cloud to ensure the effectiveness of cloud load.

2.2 Load Balancing Algorithm Types and Classification

Load balancing algorithm could be divided into three main types as follow: Static Load Balancing (SLB), Dynamic Load Balancing (DLB), and Natural Inspired Load Balancing (NILB) [29]. SLB is suitable for a static environment that has information about the system and ignores the dynamic changes that could be happened [30]. DLB is developed to improve the challenges that could take place in the environment. These challenges forced researchers to use a dynamic algorithm to be adaptable to the changes for more flexibility and accessibility [30]. NILB is preferred to be used in the cloud computing environment because it could deal with the challenges of human nature activities. NILB algorithm is considered an artificial intelligence algorithm that could deal with the challenges which come from SLB and DLB [31].

Load balancing algorithms are classified according to (i) their system which is static, dynamic, and hybrid load balancing algorithms [7,32]. (ii) the features of data so the load balancing algorithms are divided into schedule or allocation tasks. (iii) the types of cloud load balancing algorithms, there are virtual machine, control unit processing, task, server, and network [13]. (iv) the methods of load

balancing (hardware or application) [29,33]. (v) the techniques that are used in the environment [7]. There are some proper algorithms classified as follows:

- Round Robin (RR) load-balancing algorithm: it is a static algorithm and is considered the easiest method that could deal with Virtual Machines (VM) [7]. In this method, all tasks assigned to VM are equal.
- Weighted Round Robin (WRR) is like RR which is used to enhance the previous one that could assign the weight for nodes of each VM. It could calculate the waiting time efficiently but is not sufficient for allocating the task [34].
- Min-Min Load Balancing (MMLB) Algorithm: it is a static algorithm that could calculate the time of the implemented task and assign it to the VM. This algorithm deals with many numbers of small tasks, so it is allowing to increase the performance faster in the case of small tasks [7].
- Max-Min Load Balancing Algorithm [33]: this algorithm is like MMLB which is based on the time of completing the task. It could deliver all tasks then calculate the time of each task and select the maximum one to be assigned to the specific machine. It differs from MMLB which could deal with many tasks and keep high performance [2].
- Honey-Bee Load Balancing (HBLB) algorithm [35]: it is proposed to solve the issues of cloud computing environment based on the concept belonging to food process bees. This algorithm is considered a solution to the problem of classification which could face the challenges of servers and services.
- Ant Colony Optimization Load Balancing (ACOLB) algorithm [35]: this algorithm can deal with the behavior of ants based on searching the food. It is successful that could achieve to schedule the task but could not decrease the response time which affected the performance.

3 Related Work

Genetic Algorithm (GA) could use an algorithm depending on the time of ending task and the priority of the arrival user to minimize response time [1]. They used a cloud analyst to show their results and implemented load balancing algorithms by simulation. In this algorithm, the requests of the users had been sent based on the memory of the central processing unit which belongs to each virtual machine considering the units of time. They could simulate the follow of the large requests that had come from users by dividing the users into small groups to allocate each group to its virtual machine. They could calculate the parameters that affect response time to be decreased and minimize the energy used.

An optimized load balancing method called Mutation based on Particle Swarm Optimization (MPSO) is used in [5]. Their work is proposed to increase the efficiency of the cloud during the process of search as they used allocating resources and de-allocating of cloud. Servers that have high power and storage need to be connected to the network. They could avoid transmission delay problems over the network because they used a load balancing allocator to deal with the cloud resources. This technique succeeded in decreasing the time and function efficiently as they deal with MakeSpan time, and the function called fitness to assign the task to the preferred data center. Their results proved that they proposed a significantly optimized technique to deal with the workload of a virtual machine by improving the MakeSpan time compared with particle swarm algorithms. However, their comparison should investigate many metrics to be more successful for the cloud environment.

Load balancing algorithm-based task classification and implementation on e-learning system architecture in the educational sector was compared with other load balancing algorithms [2]. The authors present three experiments using a task classification load balancing algorithm and evaluated the metrics that affect the performance of their algorithm. They could divide the process into two steps

to help them prepare the data before using the algorithm to measure its performance of the algorithm. The algorithm was implemented on servers that have low utilities based on allocation tasks. They used their algorithm and compared their results with many popular algorithms used in load balancing in the cloud environment. They conclude that the low utilities of servers suffer from the interrupted connection of the network. As a result, their technique is significant as it could scope on the use of low weighted server but not considering the differences of servers' utilities and the similarity between tasks.

According to the importance of a new application that appeared recently which needs parallel programming implementation [14]. In their study, they succeeded to deal with the workload from the programmer's point of view. Using those applications could make a workload on CPU and GPU that comes from the different and complex usage of the newest application. The authors proposed a solution for heterogeneous workloads based on the utilities of CPU and GPU. They used a technique of machine learning-based feature selection methods to solve the problem of workload to optimize and assign the preferred algorithm to be used to balance the load. They could compare the results of their new method and others used before to improve the technique. They illustrated a developed machine learning method to choose the most suitable algorithm that should be used to balance the workload. The proposed work could distribute the load after computing and calculating all the requirements of the application and the utilities of the CPU used to run the application.

MAP-ACO [16] proposed Multi-Agent Pathfinding (MAP) based on the Ant Colony Optimization (ACO) algorithm to balance the consumption resources in cloud computing. The algorithm focused on the traffic of user requests over the network for available nodes queue, data reliability, and their routes. The routing model determined the required nodes and the best route to allocate multiple resources with different IoT structures. To find the optimal path, a lot of calculation is required to update the paths with a large number of parameters and buffering size which is hard to transfer large-scale data in real-time through wireless sensor networks and decentralized IoT.

A hybrid technique is conducted called Hybrid Hawks Optimization (HHO) load balancing algorithm and (Pigeon Inspired Optimization) PIO to produce the high cost of machines and consumption time [36]. They used their technique to avoid the challenges of the two load balancing algorithms that are depending on the level of the load on virtual machines. They overcome the issues which come from implementing the two algorithms separately. They optimized the load from overloaded VM to underloaded VM considering response time and utilities of resources. The workload of the memory or network needs a developed algorithm to cover the workload from different perspectives. The technique's main mission is to specify one VM acts as a load balancer then, other VMs constraints are validated and delivered to the specific one to balance the data between all VMs. Depending on the requests of users, the load balancing method of the virtual machine balancer could assign and allocate the task for its virtual machine. Otherwise, the VM balancer could not assign the task because of a non-available underloaded VM, could save the task and a message will be sent to a user to try again with another virtual machine [36].

Load balancing algorithm is used based on the agent which used the migration process for virtual machines [22]. This method is used to predict the time of the workload for the migration process. Their work was implemented on CloudSim by using Linux operating system. The author presented a framework of load balancing algorithm based on multi-agent to decrease the workload of the virtual machines by using the prediction of time series. They recommended the researchers for improving the response time and the errors that affect the load balancing of the cloud environment.

As a result of the related work, this paper is considered a preprocessing phase that could be investigated to enhance the performance of cloud computing by proposing a multi-agent framework

based on a load-balancing algorithm to enhance the performance of the cloud environment. The next section will illustrate the framework and the detailed steps to solve the scalability problem of the cloud environment.

4 Proposed Framework of Load Balancing for Cloud Environment

The load balancing framework as shown in Fig. 1 is developed for the cloud to cope with the different challenges faced cloud environment. In this section, we illustrate a load balancing framework based on a multiagent system that could be used to balance the workload among different servers to enhance performance for the cloud environment. There are two phases called (preparatory and main) to propose our framework as follows:

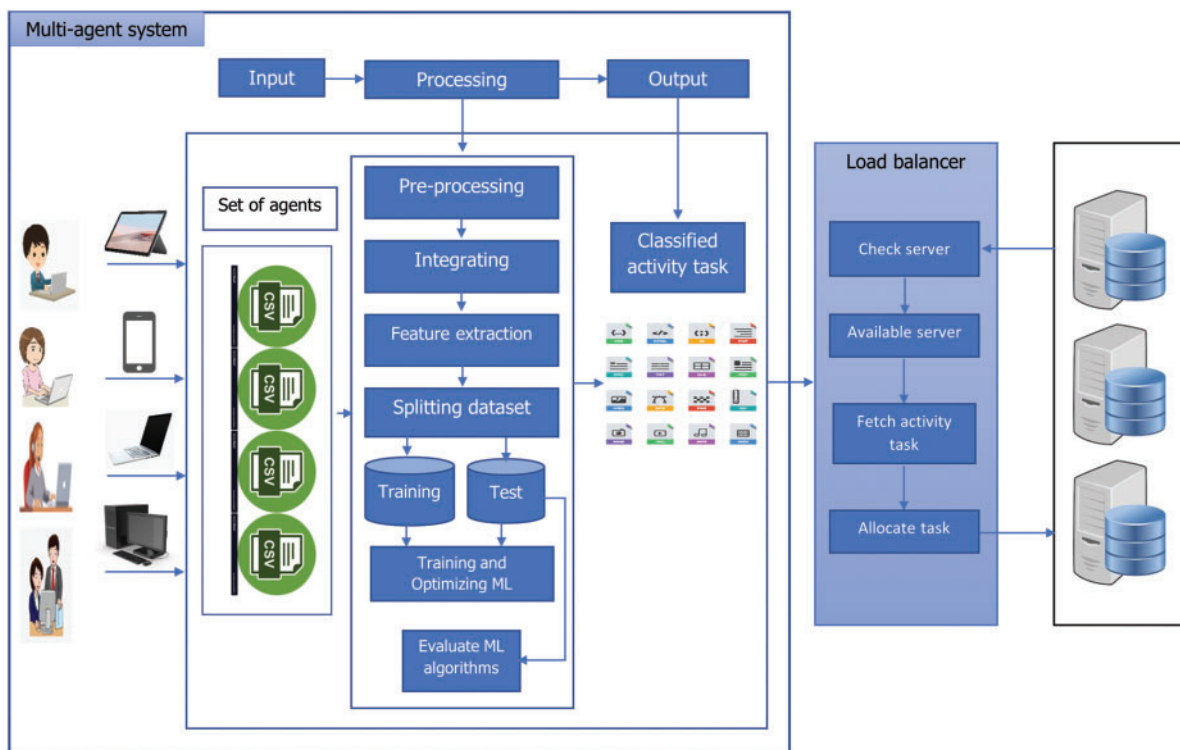


Figure 1: Proposed multi-agent framework of load balancing for cloud environment

4.1 The Preparatory Phase to Classify the Activity Task

The multi-agent system is used to integrate the actions between agents (e.g., students, employees, courses, staff, etc.). Using the feature selection method and machine learning algorithm to classify the tasks of users based on the workload of the resources is the preparatory step of the proposed framework. A multi-agent system is used to serve the transmitting tasks to its assigned server over the network to balance the load that could be happening in the cloud environment.

The preparatory phase consists of a set of agents that include student, course, activity, and virtual learning environment. The collection of data sets contains the student information and registration in addition to the courses that belong to students presented in a different style of learning. The assessments of students that should be taken to pass the exam and the semester that they could register

for their courses. This phase is considered the successful key factor for the next phase which classifies the workload that may be affected the performance of the cloud environment. The Random Forest algorithm-based feature selection method is used to predict the behavior of users in the activity task agent. The following steps will clarify the preparatory phase:

- Step 1: integrate the different types of data (student.csv, course.csv, activity task.csv, assessment.csv).
- Step 2: use an extraction method to split the data.
- Step 3: utilize machine learning algorithm and feature selection method.
- Step 4: use the classification method to obtain the relevant features depending on the workload.

4.2 The Main Phase for Proposing a Load Balancing Algorithm

The proposed algorithm of a multi-agent system is based on load balancing to ensure the performance of the servers and the scalability of the cloud environment. As the preparatory phase is successful to present the activity task using a multi-agent system, the main phase can be used to enhance the performance by checking the metrics affected by the cloud environment.

After the preparatory phase has achieved its mission, there is a classified activity task that will be ready to be distributed over servers by balancing the workload efficiently in the cloud environment. The main phase is responsible for distributing the Activity Tasks (AT) among several servers. The servers have different utilities to assign tasks to them by balancing the load considering the different utilities of the servers. The main step is distributing the workload without occurring overloading on servers and keeping the performance efficient. This phase deals with classified tasks based on the priority of workload that serve the load balancing algorithm to balance the load over the servers with high performance as follows:

- Step 1: check the utility of the servers according to the availability of the server's configuration (RAM, CPU cores, CPU power, disk space, and cache memory).
- Step 2: the task will be distributed to a Server Available (SA).
- Step 3: the available server will receive the activity tasks keeping the high performance.
- Step 4: when the performance of the server starts to decrease, or Migration Time (MT) starts, the status of a server becomes Server Busy (SB). At this moment, use the second available server.
- Step 5: assign the next activity task to the second available server.
- Step 6: the requests for activity tasks will be distributed again based on the availability of the server until completed.

The main contribution of the proposed framework is to efficiently assign tasks depending on their types and load to a specific server considering the priority of the weight. The classified activity tasks are assigned after the preparation of data to be suitable for the utilities of servers. Our work maintains the load among servers that have various utilities so the load balancing algorithm will be offered to distribute the load efficiently among the servers based on the metrics of the cloud performance. [Fig. 2](#) illustrates the flowchart of the load balancing agent. The feature selection method is used in the classification stage based on machine learning techniques [37] before the main phase of proposing a load-balancing algorithm for the cloud environment. More results will be illustrated in the result section for more clarification of the load-balancing algorithm. The algorithm is used to avoid the traffic flow, and decrease the response time, in addition, to sure the distribution of specific tasks for a specific server without overloading a machine.

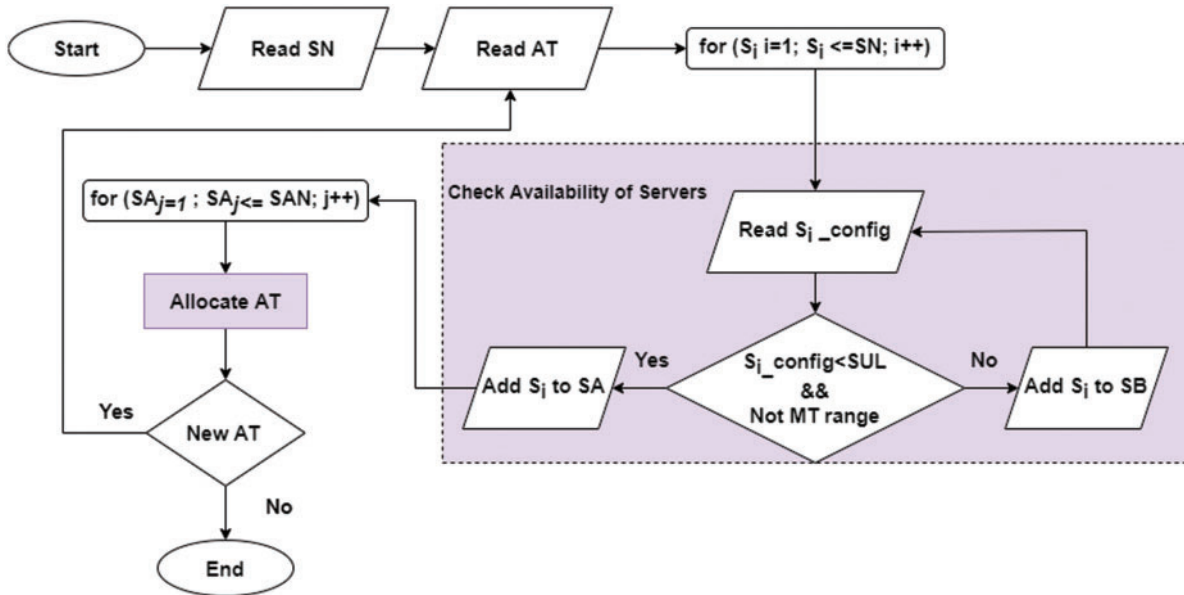


Figure 2: Flowchart of load balancing framework to enhance cloud performance

This algorithm is developed after implementing and testing the novel algorithm in [2] because it was implemented based on the history of (old and new) tasks and their weights for allocating and distributing them over low server utilities. The load balancing algorithm in [2] ignored the differences between servers in their implementation and applied the algorithm depending on the relation between old and new tasks. They presented their algorithm without any consideration of the performance of the cloud environment. Our proposed algorithm could distribute the workload among servers with different utilities and enhance the performance of the load-balancing algorithm in [2].

5 Proposed Load Balancing Algorithm

As shown in Algorithm 1, the proposed algorithm is dealing with several servers of $(SN) \in \{S_1, S_2, S_3, \dots, S_n\}$ where every server has its configuration (such as CPU power, RAM, CPU cores, desk space, and cache memory). First, check the status of the server to preset it as available (SA) or busy (SB). If it is available, assign the activity task $(AT) \in \{AT_1, AT_2, AT_3, \dots, AT_n\}$ to $\{SA_1, SA_2, SA_3, \dots, SA_j\}$. Then, if the server is busy or migration time occurs soon, assign the next AT to the second SA as illustrated in Fig. 3.

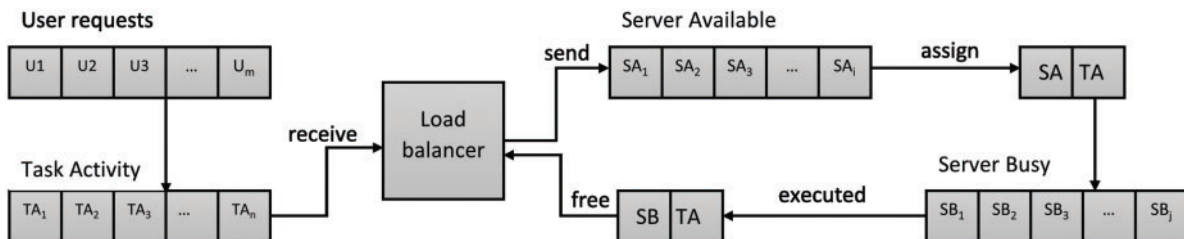


Figure 3: Proposed load balancer distribution

Algorithm 1: Proposed Load Balancing

```

Input: SN No. of servers  $\langle S_1, S_2, \dots, S_n \rangle$ 
Si_config = (CPU cores, RAM, CPU power, disk
space, and cache memory)
SUL Server Current Status
MTL Migration Time Limit
SB Busy Server Queue
SBN Busy Server Number
SA Available Server Queue
SAN Available Server Number
AT Activity Task
Begin
Read SN of server
Read AT
if AT is NOT Empty then
  for  $Si=1, Si \leq SN, i++$  do
    Read Si_config
    if  $Si\_config < SUL$  &  $\neq MTL$  then
      | ADD Si to SA []
    else
      | ADD Si to SB []
    end
  end
  end
  for  $SAj=1, SAj \leq SAN, j++$  do
    |  $SAj \leftarrow AT$ 
  end
  Go to Read Next AT
end

```

6 Results and Discussion

In the feature selection steps, Open University Learning Analytics Dataset (OULAD) [38] is used to investigate the proposed framework. It contains three main modules: activity, presentation, and demographics which include data about students and their activities in Virtual Learning Environment (VLE). Different machine learning algorithms are used: Decision Trees, Random Forest, Linear Regression, Naïve Bayes, and K-Nearest Neighbor, which are applied to select the high-ranked and relevant features [37]. Extra Trees Classifier feature selection methods are used to select more relevant features of student data. These features play an important action to propose a multi-agent framework based on load balancing to enhance the cloud environment. The Random Forest algorithm outperforms other machine learning algorithms [37]. Fig. 4 illustrates the confusion matrix of the classification using Random Forest. The classification model performance is evaluated based on the Extra Trees feature selection method to measure Accuracy (ACC), Precision (PRE), Recall (REC), and F1-Measurement (FM) as shown in Tab. 1.

The following figures showed the workload of each activity. Fig. 5 shows the results of the sum of clicks that student used to achieve their course and how they made a load in specific activity tasks. The oucontent has the highest load with 9,207,483 visits of this activity task, quiz activity tasks the second priority of rank as there are 5,610,534 visits of this type. The third one has 5,487,087 clicks and the fourth has 2,576,872 visits.

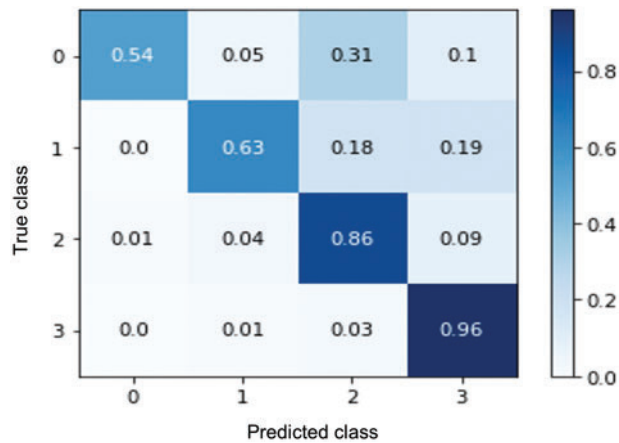


Figure 4: Confusion matrix for random forest classification model

Table 1: Random forest measurements

	Accuracy	Precision	Recall	F1-measurement
Cross-validation set	87.6%	88.05%	87.71%	87.7%
Testing set	86.72%	87.08%	86.72%	87.73%

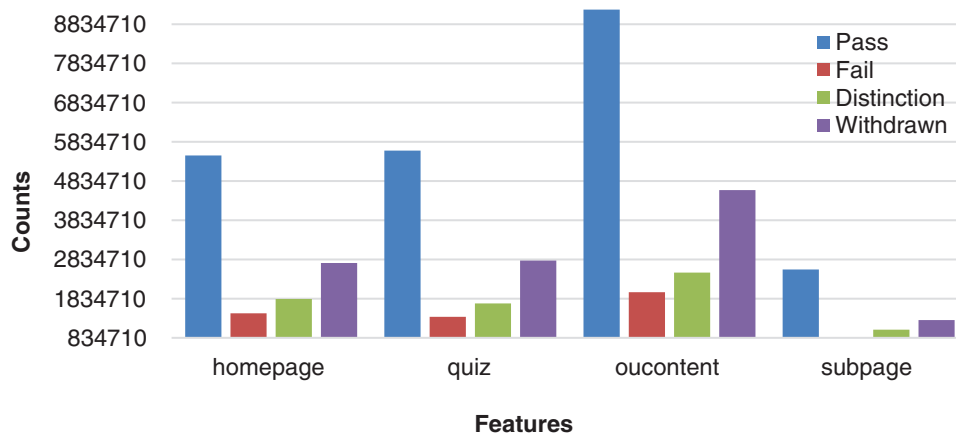


Figure 5: Load of first grouped features

In Fig. 6, forumng score is the first relevant feature of the total four features with 6,509,134 visits. The resource has recorded 815,562 visits to be the second rank of load. Ouwiki and Url recorded 7,62,601 and 438,101 clicks of load.

Fig. 7 shows the clicks that ranked with the priority of load as they record (88,582–’73,086’–’ 56,347 and 44,833) for oucollaborate, glossary, questionnaire, and page. From the previous results, we found that the most relevant features that affected the multi-agent framework based on load balancing are classified depending on their load related to the sum of clicks. This paper proposed a load balancing algorithm for the cloud environment and the multi-agent system that could also prove the relation

between the increasing sum of clicks of students that visit the activities and the increasing success percentage.

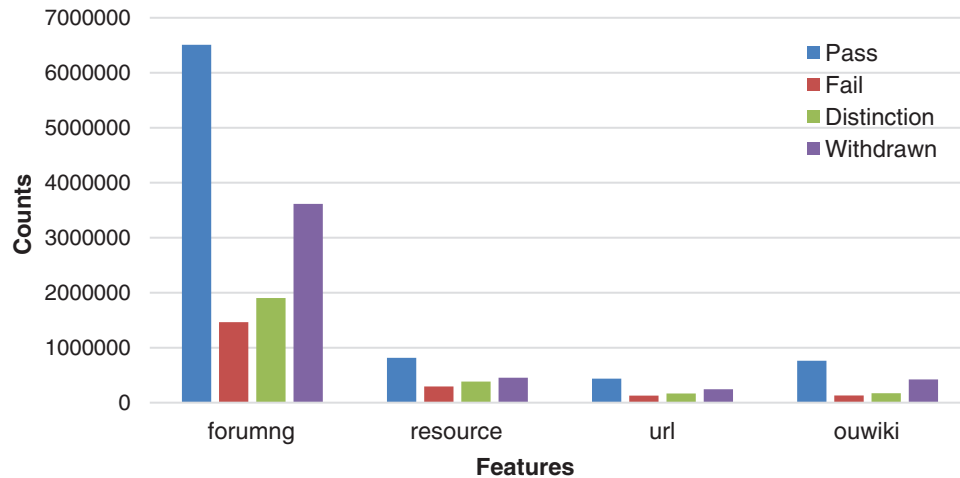


Figure 6: Load of second grouped features

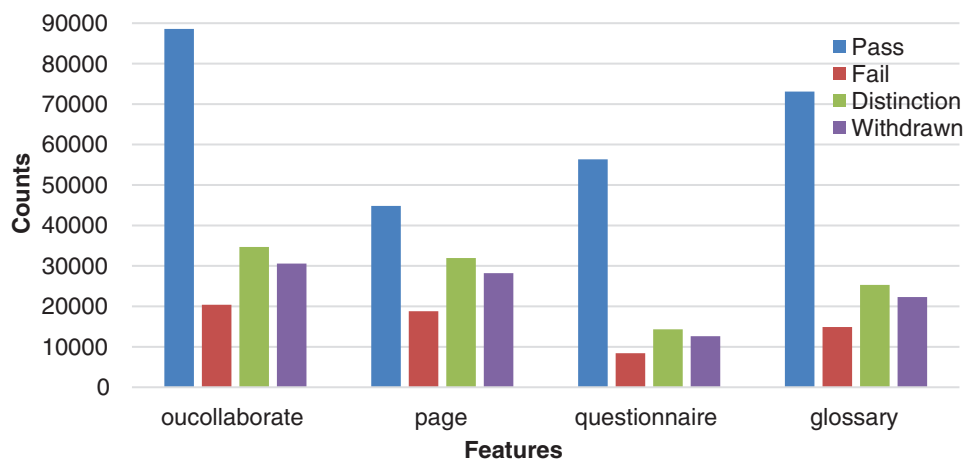


Figure 7: Load of third grouped features

7 Conclusion and Future Work

This paper could illustrate a framework considering a developed load balancing algorithm to serve the requirements of the cloud environment and remove the challenges of low resources for IoT devices. The target of our framework is maximizing the performance of the cloud computing environment under the effect of the workload and the high changes that daily occurred. The proposed algorithm is used to balance the workload of the multi-agent system between the various servers that have different utilities under the umbrella of IoT devices.

Our framework processed two main phases to gain high performance for a cloud environment, the first one is selecting the suitable mining technique based on machine learning algorithms to test the accuracy of the activity tasks and then selecting relevant features for multi-agent performance.

The second phase depended on the first one as the proposed algorithm was used after the preparatory process of data to optimize and distribute the workload over the network for a cloud environment.

Studying the relation between agents helps the proposed algorithm to achieve its mission. To clarify, classification mining technique-based machine learning selected more relevant features. Activity tasks based on relevant features of a multi-agent system are used to minimize the workload before running the proposed algorithm. Moreover, all tasks had been classified and weighted before assigning them for improving the results of the load balancing algorithm's performance. The preparatory step has been used to increase the performance which is necessary to achieve the main target to balance the workload around servers. Finally, this paper presented a load balancing algorithm to optimize the workload and check the utilities of the server then a task was assigned and located to a suitable server effectively.

The security of IoT devices is an important topic to be investigated considering the heterogeneous data and the enormous devices that are transmitted and received over the network in the future. The required distribution security by IoT applications will be investigated during the implementation of multi-agent systems with different load-balancing methods to improve the security sector.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] A. K. Jayswal and P. C. Saxena, "Multi-agent genetic algorithm for efficient load balancing in cloud computing," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 4, pp. 45–51, 2020.
- [2] A. E. Khedr, A. M. Idrees and R. Salem, "Enhancing the e-learning system based on a novel tasks' classification load-balancing algorithm," *PeerJ Computer Science*, vol. 7, pp. 1–28, 2021.
- [3] H. Bheda and H. Bhatt, "An overview of load balancing techniques in cloud computing environments," *International Journal of Engineering and Computer Science*, vol. 4, no. 1, pp. 9874–9881, 2015.
- [4] A. E. Khedr and A. M. Idrees, "Adapting load balancing techniques for improving the performance of e-learning educational process," *Journal of Computers*, vol. 12, no. 3, pp. 250–257, 2017.
- [5] R. Agarwal, N. Baghel and M. A. Khan, "Load balancing in cloud computing using mutation based particle swarm optimization," in *Proc. IC3A*, Lucknow, India, pp. 191–195, 2020.
- [6] A. E. Khedr, M. Nasr and H. Elmasry, "New balancing technique for green cloud computing and environmental sustainability," *International Journal of Advanced Research*, vol. 3, no. 9, pp. 201–215, 2015.
- [7] A. A. Alkhatib, A. Alsabbagh, R. Maraqa and S. Alzubi, "Load balancing techniques in cloud computing: Extensive review," *Advances in Science, Technology and Engineering Systems Journal*, vol. 6, no. 2, pp. 860–870, 2021.
- [8] A. E. Khedr, S. A. Kholeif and S. H. Hessen, "Enhanced cloud computing framework to improve the educational process in higher education: A case study of Helwan University in Egypt," *International Journal of Computers & Technology*, vol. 14, no. 6, pp. 5814–5823, 2015.
- [9] H. ELMasry, A. E. Khedr and H. M. Abdul-Kader, "Challenges and opportunities for intrusion detection system in cloud computing environment," *Journal of Theoretical and Applied Information Technology*, vol. 98, no. 20, pp. 3112–3129, 2020.
- [10] J. Nazir, M. W. Iqbal, T. Alyas, M. Hamid, M. Saleem *et al.*, "Load balancing framework for cross-region tasks in cloud computing," *Computers, Materials & Continua*, vol. 70, no. 1, pp. 1479–1490, 2021.

- [11] A. E. Khedr, S. A. Kholeif and S. H. Hessen, "Adoption of cloud computing framework in higher education to enhance educational process," *International Journal of Innovative Research in Computer Science & Technology*, vol. 3, no. 2, pp. 150–156, 2015.
- [12] K. Balaji, P. S. Kiran and M. S. Kumar, "Load balancing in cloud computing: Issues and challenges," *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 2, pp. 3224–3231, 2021.
- [13] A. Imielowski, "Load balancing algorithms in cluster systems," in *Proc. CST*, Rzeszów-Muczne, Poland, vol. 21, pp. 1–8, 2018.
- [14] U. Ahmed, J. C. W. Lin, G. Srivastava and M. Aleem, "A load balance multi-scheduling model for open CL kernel tasks in an integrated cluster," *Soft Computing*, vol. 25, no. 1, pp. 407–420, 2021.
- [15] N. Sultan, A. Khedr, A. Idrees and S. Kholeif, "Data mining approach for detecting key performance indicators," *Journal of Artificial Intelligence*, vol. 10, no. 2, pp. 59–65, 2017.
- [16] S. Amir and F. Kiani, "MAP-ACO: An efficient protocol for multi-agent pathfinding in real-time WSN and decentralized IoT systems," *Microprocessors and Microsystems*, vol. 79, pp. 1–9, 2020.
- [17] N. Qingwei, D. Tang, H. Zhu and H. Sun, "A multi-agent and internet of things framework of digital twin for optimized manufacturing control," *International Journal of Computer Integrated Manufacturing*, pp. 1–22, 2021.
- [18] G. Pratik, A. Mukherjee, B. Sarkar and L. Yang, "Multi-agent-based smart power management for remote health monitoring," *Neural Computing and Applications*, vol. 5, pp. 1–10, 2021.
- [19] N. Narang and S. Kar, "A hybrid trust management framework for a multi-service social IoT network," *Computer Communications*, vol. 171, pp. 61–79, 2021.
- [20] P. Kendrick, A. Hussain, N. Criado and M. Randles, "Multi-agent systems for scalable internet of things security," in *Proc. ICC '17*, New York, USA, pp. 1–6, 2017.
- [21] A. P. Maranda and L. Gebel, "Retrieval and processing of information with the use of multi-agent system," *Journal of Applied Computer Science*, vol. 24, no. 2, pp. 17–37, 2016.
- [22] J. Zhang, Q. Liu and J. Chen, "A multi-agent based load balancing framework in cloud environment," in *Proc. ISCID*, Hangzhou, China, pp. 278–281, 2016.
- [23] R. Ramya, B. Seetha, S. P. Vigneshwari and N. Angayarkanni, "Analysis of load balancing mechanisms in public cloud service," *International Journal of Recent Technology and Engineering*, vol. 7, no. 6, pp. 517–521, 2019.
- [24] M. Mesbahi and A. M. Rahmani, "Load balancing in cloud computing: A state of the art survey," *I. J. Modern Education and Computer Science*, vol. 8, no. 3, pp. 64–78, 2016.
- [25] S. Keshvadi, "A multi-agent based load balancing system in IaaS cloud environment," *International Robotics & Automation Journal Research*, vol. 1, no. 1, pp. 3–8, 2016.
- [26] H. C. Sharma and H. Bahuguna, "A survey of load balancing algorithms in cloud computing," *International Journal of Computer Engineering and Applications*, vol. 11, no. 9, pp. 1–11, 2021.
- [27] C. Roberta, G. Ciatto, V. Mascardi and A. Omicini, "Logic-based technologies for multi-agent systems: A systematic literature review," *Autonomous Agents and Multi-Agent Systems*, vol. 35, no. 1, pp. 1–67, 2021.
- [28] A. Zaouch and F. Benabbou, "Load balancing for improved quality of service in the cloud," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 7, pp. 184–189, 2015.
- [29] E. J. Ghomi, A. M. Rahmani and N. N. Qader, "Load-balancing algorithms in cloud computing: A survey," *Journal of Network and Computer Applications*, vol. 88, no. 12, 2016, pp. 50–71, 2017.
- [30] M. Alam and Z. A. Khan, "Issues and challenges of load balancing algorithm in cloud computing environment," *Indian Journal of Science and Technology*, vol. 10, no. 25, pp. 1–12, 2017.
- [31] A. Thakur and M. S. Goraya, "A taxonomic survey on load balancing in cloud," *Journal of Network and Computer Applications*, vol. 98, no. 4, pp. 43–57, 2017.
- [32] S. Afzal and G. Kavitha, "Load balancing in cloud computing—A hierarchical taxonomical classification," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 8, no. 1, pp. 1–24, 2019.
- [33] S. K. Upadhyay, A. Bhattacharya, S. Arya and T. Singh, "Load optimization in cloud computing using clustering: A survey," *International Research Journal of Engineering and Technology*, vol. 8, pp. 2455–2459, 2018.

- [34] M. I. Moly, A. Hossain, S. Lecturer and O. Roy, "Load balancing approach and algorithm in cloud computing environment," *American Journal of Engineering Research*, vol. 8, no. 4, pp. 99–105, 2019.
- [35] S. Sonawane, P. Arnikar, A. Fale, S. Aghav and S. Pachouly, "A review on various load balancing algorithms in cloud computing," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 6, no. 4, pp. 599–604, 2016.
- [36] G. A. Poornima and A. S. Radhamani, "A hybrid meta-heuristic for optimal load balancing in cloud computing," *J. Grid Computing*, vol. 19, no. 2, pp. 1–22, 2021.
- [37] S. H. Hessen, H. M. Abdul-kader, A. E. Khedr and R. K. Salem, "Developing multiagent e-learning system-based machine learning and feature selection techniques," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–8, 2022.
- [38] J. Kuzilek, M. Hlosta, and Z. Zdrahal, "Open university learning analytics dataset," *Scientific Data*, vol. 4, pp. 1–8, 2017.