Providing an Improved Resource Management Approach for Healthcare Big Data Processing in Cloud Computing Environment

Fei Zhou[1*], Huaibao Ding[2], Xiaomei Ding[3]
School of Computer Engineering, Anhui Wenda University of Information Engineering, Hefei 231201, Anhui, China

***Abstract*—Due to the gathering of big data and the advancement of machine learning, the healthcare industry has recently experienced fast change. Acceleration of operations related to the analysis and retrieval of healthcare data is essential to facilitate surveillance. However, providing healthcare to the community is a complex task that is highly dependent on data processing. Also, processing health metadata can be very expensive for organizations. To meet the strict service quality requirements of the healthcare industry, large-scale healthcare data processing in the cloud confederation has emerged as a viable option. However, there are many challenges, including optimal resource management for metadata processing. Based on this, in the present study, a fuzzy solution for determining the optimal cloud using the resource forecasting technique is presented for health big data processing. During job processing, a fuzzy selection-based VM migration technique was used to move a virtual machine (VM) from a high-load server to a low-load server. The proposed architecture is divided into regional and global levels. After evaluating the local component, requests are sent to the global component. If the local component cannot meet the requirements, the request is sent to the global component. The hierarchical structure of the proposed method requires the generation of delivered requests before estimating the available resources. The proposed solution is compared with PSO and ACO algorithms according to different criteria. The simulation results show the effectiveness and efficiency of the model compared to alternative methods.**

Keywords*—Healthcare; big data; cloud confederation; service quality; Cloud Resource Management (CRM)

I. Introduction

The existing processes for collecting and processing the vital data of patients demand a large volume of workforce and calculations. Usually, these processes are error-prone and have large delays, which prevent the right data from being available in real-time [1]. As a novel technology, cloud computing with internet infrastructure and novel solutions provides considerable advantages in providing medical services electronically [2]. Indeed, the advent of cloud computing has brought a fundamental change in the path of the novel, developed, scaled, and up-to-date services of information technology [3]. With the fast development in processing and storage technologies and also the successfulness of the internet, the computing of resources has become cheaper, stronger, and more accessible than ever before, and governmental organizations have started to use architecture, platforms, and programs of cloud computing to provide the services and satisfying the needs of their subsidiaries [4]. However, there are still many challenges in optimal resource management and to what extent each data needs the computing resource are among these challenges [5].

The objective of the resource management system for the cloud is to satisfy the needs of the applications using it. Because it is possible, during the process, that some of the servers have high traffic load while the others have low or no load, the resource management mechanism should check the current situation of every resource in the cloud environment to provide the algorithms for better allocation of physical or virtual resources and thus reduce the operational costs in the cloud environment [6]. In this case, not only may the workload be spread among the available but underutilized servers, but the unused servers can be shut off altogether. Centralized work allocation to servers that do not take into account the individual solutions is obviously unfeasible in such large and complicated systems [7]. Also, according to the rapid growth in metadata in the advanced data centers and the urgent need to access good service quality, the necessity for providing solutions in order to increase the productivity of the available service providers in the metadata processing center is strongly felt. One way to achieve the desired productivity is to use resource management solutions [8]. For appropriate management of the resources of service providers, the load balance is required [9]. With an appropriate load balance, the requests will be distributed in a dynamic and balanced form among all nodes while ensuring the fairly and efficient allocation of every computing resource [10]. This process will result in increased user satisfaction and high productivity of the resources, which eventually increases productivity [11]. The load balancing aims to find an appropriate mapping of the tasks on the processors available in the system so that in each processor, approximately an equal number of tasks will be executed to minimize the overall runtime [12]. In recent years, due to population growth and advances in medical science, the number of discovered diseases has increased [13]. The information on each patient entered into hospital systems is so huge and complex that it has been classified as "big data," requiring plenty of resources to process [14]. It is important to pay attention to the fact that cloud computing has entered all fields, including the medical field, at a very high speed today [15]. Considering the high number of information related to a person that may be available in the hospital system, it is possible to use cloud processing instead of local storage of the person's information so that there is less need for local and physical facilities and the speed of obtaining and processing the person's information in time multiplied the need. One of the prominent points of this method is that there are no worries about losing information due to local system damage, and information can be stored, processed, or retrieved at any time, regardless of the hospital system load. One of the biggest challenges here is choosing the type of resources to store information, whether local resources are more optimal or resources that can be rented from other places, or more importantly, how much resources each data needs to be able to store it. It is possible to select the most optimal resource by predicting the number of resources each piece of information needs and allocating it to it. In this research, we intend to present a new approach to improve the management of healthcare big data resources, and by using this method, we will significantly improve resource management by facilitating and accelerating the processing of medical care data in cloud computing.

This study predicts and attempts to implement the necessary resources for processing RIMA model data (big data) on drug consumption in various American states between 2018 and 2021. The fuzzy DEMATEL method has the lowest prediction error as well as closely resembles the reality of a request's resources. Thus, we employ it to allocate local and remote assets for each request. This algorithm was developed with profits and profits as its primary motivation. In a nutshell, the following is the author's contribution to this article:

- Using the RIMA model to predict the required resources with a low error percentage.
- Using the fuzzy Dematel technique to choose a cloud data center to accelerate and facilitate big data processing in the cloud environment.
- Increasing productivity by maintaining user QoS and reducing response time.

This paper's remaining sections are structured as follows. The context is provided in Section II. Section III provides fuzzy DEMATEL methods for evaluating RIMA-based cloud and resource forecasting methods. Evaluation and simulation of the suggested model are presented in Section IV, followed by a discussion of the results. Conclusion and directions for further research is presented in Sections V.

II. LITERATURE REVIEW

About [16], a review of the common algorithms in the field of load balance in metadata is provided. One of the most popular algorithms is the Min-Min algorithm. At the outset, the algorithm is given a pool of jobs from which to choose. The first thing to do is determine the quickest possible way to do every activity. The quickest task on each available resource is then chosen, and its corresponding minimum finish time is chosen. The next step is to allocate the required amount of time for the task on the appropriate machine. Besides optimal scheduling, the approach has a main issue, which can lead to famine. In research [17], a hybrid task scheduling strategy has been provided for managing and processing medical metadata in cloud VMs. The tasks' migration from one server to another has been addressed. In this research, the genetic algorithm, along with the particle swarm optimization, was used to provide the load balance and allocate the available network resource to the tasks so that the distribution of tasks between the resources was performed fairly and the servers were able to execute the processing on different requests of users simultaneously. In this research, the proposed algorithm has been simulated by CloudSim software, and the results indicate the effectiveness of this approach in load balance. In [18], a load balance and scheduling model based on the cloud segmentation concept has been proposed. A hybrid algorithm was provided in which the RR2 scheduling algorithm and game theory were used. This algorithm selects a low-load server with a high execution speed. Time slicing is done better in RR because the better server is selected. The results show that the load balance has improved using the proposed algorithm. In study [19], a cloud confederation model which provides ideal selections for target cloud providers has been proposed to address the heterogeneous IoT metadata customers' demands. In addition, a multi-purpose optimization model has been provided. A general structure for the genetic algorithm has been developed to solve this model. Different evaluation tests have tested the proposed model. In study [20], a real model based on linear programming has been presented to identify the strategies and decisions in the cloud federation. The user requests are established on all cloud federation levels, and the users are easily directed. As a result, the obtained advantages motivate providers' participation by free insourcing to support the other federation members. The proposed model is highly scalable. In [18], multi-layer resource allocation has been provided. The main idea behind this was based on the allocation of resources in a layering approach, which configures the resource allocation for personal tasks on a cluster node based on the resource exploitation levels. The results show that MTRA has improved the performance and runtime by 18% and 10%, respectively. In [21], it is examined how end users choose MEC servers, how they offload their data optimally, and how MEC servers determine their prices optimally in a multi-MEC server and multi-end user scenario. An SDN controller first implements a reinforcement learning framework based on stochastic learning automata to enable end users to choose a MEC server to load their data. The whole MEC selection process considers the MEC server's method, its congestion and penetration in terms of completing end users' computing duties, and its declared price for its computing services. A non-cooperative game is developed between the end-users of each server to determine the end user's data loading portion to the chosen MEC server, and the existence and uniqueness of the corresponding Nash equilibrium are demonstrated. An iterative and simple method has been developed to implement the suggested framework. The performance of the suggested technique was assessed through modeling and simulation in various scenarios with both homogeneous and heterogeneous end users. In [22], author offers thorough analyses of prior cognitive computing research, problems, fixes, and prospects for further study. There is a focus on cognitive computing-based methods for resolving practical issues in the four extensively studied application fields of healthcare, cyber security, big data, and the Internet of Things. Author examined research from 2012 to 2020 in [23]. This study aims to offer a thorough analytical understanding of big data for health care. Three goals were met in this study's investigation: (a) identifying the most important themes in big data research on healthcare; (b) assessing the relationships between the themes in previous studies, and (c) creating time-series profiles for the topics. Healthcare research using big data has covered a variety of subjects. Based on the study's findings, a summary of researchers' interests in various techniques, technologies, and topic areas is given, along with a list of critical research gaps that need to be filled in the future. The advantage and disadvantages of different technologies are thoroughly described in research [24], along with the range of applications for each. Attacks, according to background information gleaned through data integration, can be avoided using various technologies, anonymity, and privacy during data collection. The majority of important medical data is kept in storage on a cloud computing platform. Encryption and auditing techniques are frequently employed to maintain the confidentiality and integrity of stored information. Access control techniques are also utilized during the data-sharing phase to control the objects with access to the data. Big medical and health data privacy protection is carried out under machine learning during the data analysis stage. Finally, acceptable management-level concepts have emerged due to broad privacy protection worries across the medical big data lifecycle across the sector. Table I discusses a summary of the comparison of existing techniques in resource management.

TABLE I. COMPARISON OF AVAILABLE TECHNIQUES IN RESOURCE MANAGEMENT

Work	Target parameter	Method	Type of technique	Advantages	Disadvantages
Miah et al. 2022 [25].	Achieving policies that help increase profits and productivity and overcome resource limitations.	Efficient resource management strategy, deployment of a cloud federation to coordinate resources	Model-based	1. It gives providers more flexibility in resource management and increased productivity. 2. Customers can maintain the machines for a long time if needed.	Outsourcing of spot requests is not considered in this method. Also, not having strategies that help to predict the availability of future resources for decision-making is one of the disadvantages of this method.
AbdulAhmeed et al. 2017 [26].	Explore a cooperative cloud approach to form a confederation to maximize CPs' profits and meet SLAs	Constructing a model of cloud federation to identify top-tier cloud service providers.	Multi-objective (MO) optimal model, genetic algorithm	Increase revenue for IOTDCs by sharing available virtual and physical resources	Heterogeneous request problems IOTDCs receive a large number of requests.
Mitsis et al. 2019 [27].	Data upload through MEC server.	SDN controller and reinforcement learning based on the concept of stochastic learning automata are used to load data on a MEC server.	SDN controller and reinforcement learning	Use of MEC technique, high security	Need to optimize the problem
Dhiman et al. 2022 [28].	Maximizing provider benefits in cloud federation	A linear numerical program for optimizing input workload distribution across federation members	Linear numerical programming algorithm, Branch & bound algorithm	Access to the highest incomes, minimizing energy consumption, reducing operating costs, scalability	Requires a lot of knowledge for modeling
Navroop Kaur et al. 2016 [29].	Appropriate cloud resource allocation has become a research problem due to the high growth of cloud big data.	Proposing an effective resource management system for big data streams	Dynamic Clustering Algorithm by SOM (Self-Organized Map)	Reduce waiting time	Need to optimize the problem
ThomasRyan et al. 2021 [30].	Create a method for managing resources that can function effectively in settings with varying levels of complexity.	Allocation of resources for individual tasks in a layered approach of dynamic resource allocation	MTRA technique (Multi Tier Resource Allocation (MTRA))	Performance improvement up to 18%, runtime improvement up to 10%, increased scalability	Increasing the complexity of the system

III. PROPOSED METHOD

This section suggests a method for choosing the best cloud using the DEMATEL fuzzy technique as well as a method for resource prediction using the RIMA algorithm. The proposed architecture's structure is divided into two local and one global component. Before sending a request to the global part, the local function should be examined; if the local domain is unable to supply the necessary data, the request is forwarded to the global domain. Since the suggested procedure has a hierarchy architecture, the local data component receives the user request. The local analyzer examines and keeps track of it to forecast the resources. On this foundation, the RIMA algorithm first estimates the size of the available resources before creating the list of paid requests. The structure of the suggested architecture makes it clear that each IOTDC is presumed to have a Broker. The requests from users referring to each IOTDC are first received and examined by the Broker associated with that IOTDC. If there are sufficient resources to carry out the users' requests, do so and inform the IOTDC's users of the results. Even yet, imagine the Broker is unable to fulfill user requests with the IOTDC resources that are at his disposal. If so, it should utilize the Dematel fuzzy algorithm, which every Broker can access, to select the right IOTDC from the linked IOTDC. As a result, there is a Broker with the following components for each IOTDC:

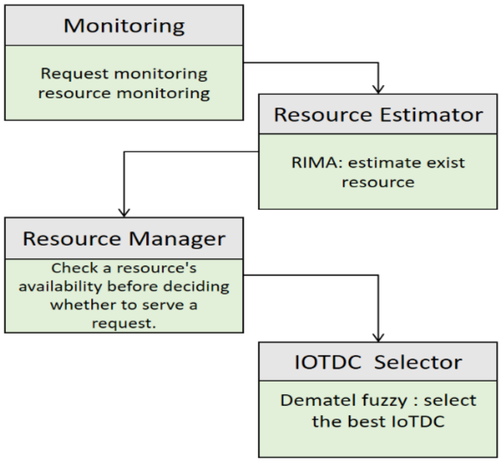


Fig. 1. Block diagram and broker component process.

The Broker's structure and its individual parts are shown in Fig. 1. A fuzzy selector-based VM migration technique is used after choosing the processing server to make sure that, should it become overloaded while processing or because of software or hardware violations, its VM can be moved to a low-load or available idle server, alleviating the overload as well as maximizing the server's useful life.

A. *Receiving the Requests*

In algorithm No. 1, requests are first added to a queue as well as then, in accordance with demand, are directed to the desired resource. The requests are positioned in the queue according to this algorithm. Each user's request is added to the queue (lines 2-4) because each user has a variety of requests. For all users, this step is repeated (line 1). The resources needed for each request are described in lines 6 through 8 that follow.

Algorithm 1: add Request to Queue

for i ← 1 to n do
for j ← 1 to Req_Count_j do
ADD $\mathbf{Req_j^i}$ **to Request_queue**
end for
While(**Request_queue is not empty**)
foreach req in Request_queue do
Send req to LCC
end for

This stage is in charge of compiling the requests that have been made as well as the available resources. Two request and resource stages are part of the monitor phase. The phase of resource monitoring is in charge of acquiring data regarding resource productivity, resource capacity usage, and maximum VM count. The request monitoring phase is in charge of compiling data on the volume of requests made by users. These two subcomponents combine the observed data, which is then saved for use by the knowledge base's other phases.

Algorithm No. 2 keeps tabs on the services provided by each LCC and GCC resource. The procedure is run up until free resources (line 1) are present. This method examined local and global resources, although local resource monitoring is given top attention. Lines 8-12 analyze the services provided by each of the GCC's available IOTDCs. First, the number of services provided by local resources is verified; if none are available, line 6 makes reference to the broader cloud federation's local resources.

Algorithm 2: Resource monitoring

1: while (reso. =0) {
2: for i ← 1 to n do
3: for j ← 1 to m do
4: if (IoTDC is available in our lcc)
5: $y^i = x_1 cdc_i + \cdots + x_j cdc_i$
6: else
7: if (IoTDC is available in gcc)
8: for i ← 1 to n do
9: for j ← 1 to m do
10: $G^i = x_1 cdc_i + \cdots + x_j cdc_i$
11: else
12: do resource = 0
}

B. *Prediction Phase*

The resource's forecast phase is in charge of determining how many resources will be needed to fulfil requests. Low supply happens when the available resources fall short of meeting demand, and high supply occurs when the available resources outpace actual needs. Therefore, the RIMA technique can be used to estimate the requirements for a given resource reliably. The third method makes use of a RIMA semi-code to estimate future service demands. Each request from the user has been viewed as an independent input, and features specific to that request have been defined using the x. Each request's amount of resource requirements is determined (line 4) based on the request's characteristics and the tracking factor of every modification. Z computes the upper bound of the genuine ask.

C. *Resource Management Phase*

The resource predictor reports back to the resource management of its origin after gauging the resource for the given request. The resource manager determines whether or not the new request can be fulfilled by the available resources after checking their availability. If no resource is available for the IOTDC host, DEMATEL is used to discover the best possible unified IOTDC. If the resource is accessible, it shouldn't be much larger than required for the request to avoid a high allocation.

Algorithm 3: RIMA Prediction

1: While (req ≥ 0) {
2: F or i ← 1 to 3 do
3: F or j ← 1 to n do
4: sum=sum + VL (j)
5: end for
6: μ = sum / n
7: for j ← 1 to n
8: $MA[i] = \mu + \Theta[j] * WL[n-j] + \mathcal{E}$
9: end for
10: for j ← 1 to n
11: $AR[i] = \phi[j] * WL[n-j] + \mathcal{E}$
12: end for
13: for j ← 1 to n
14: $L = Sqrt(\phi[j] * WL[n-j] + \mathcal{E}$
15: end for
16: $Predict[i] = L + \alpha + AR[i] + MA[i]$
17: end for
18: return [Predict] }

D. IOTDC Selector Phase

For each request, an appropriate resource allocation is made in the IOTDC selection step. Using the fuzzy DEMATEL method, this component chooses the best cloud data center [14]. The fuzzy triangle variables are needed to display the evaluation criterion. Fig. 2 displays the numerical range and each of these variables' linguistic representations.

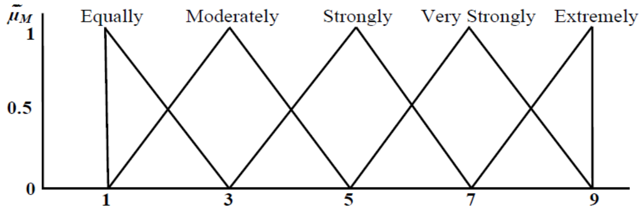


Fig. 2. Provides the linguistic representation of each of these variables and their numerical range.

To execute pairwise comparisons, the fuzzy numbers shown in Table II must first be defined in order for the responses to be appropriately delivered. Triangular fuzzy numbers have been employed in the analysis shown, as seen in the table. Since m is a triangular fuzzy number, it may be computed from equation (1) as follows:

$$S_i = \sum_{j=1}^{m} M_{g^i}^{j} \otimes \left[\sum_{i=1}^{n} \sum_{j=1}^{m} M_{g^i}^{j} \right]^{-1} \tag{1}$$

Within the pairwise comparisons matrix in the relationship mentioned above, M are triangular fuzzy numbers. In order to calculate the matrix S, we actually add each fuzzy number component individually and then multiply by the fuzzy inverse of the final sum. The normalized weights are computed in this phase similarly to the normal AHP approach, but with fuzzy numbers. The preliminary definitions should be made before introducing the suggested approach, followed by a presentation of the suggested algorithm. The investigation of fuzzy triangular numbers came first. Fuzzy triangular numbers are used in FAHP-based algorithms to verify criteria. The membership function for triangular numbers is stated as follows, and each triangle fuzzy number is displayed using three (n1, n2, n3).

All the operations that are defined for fuzzy numbers are performed in triangular fuzzy numbers on triples (n1, n2, n3). Equation (2) defines membership conditions for triangular fuzzy numbers.

$$\mu_N(x) = \begin{cases} \frac{x-n_1}{n_2-n} & x \in [n_1, n_2] \\ \frac{n_3-x}{n_3-n_2} & x \in [n_2, n_3] \\ 0 & otherwise \end{cases} \tag{2}$$

Table II also shows triangle representation and linguistic interpretation.

The main goal is to pick the best IOTDC from the set of confederation IOTDCs using Fuzzy DEMATEL. The organizational structure of the appropriate confederation has been informed by the following factors: calculation costs (C1), a distance of IOTDCi requested from IOTDCj confederation (C2), a departure from agreements (C3), as well as available resources (C4). The suggested approach for selecting the IOTDC is depicted in Algorithm (4). This method features a three-tiered, hierarchical structure; the ground level, the top level, and levels 1 and 2. The algorithms are bilevel, as denoted by the second line of code. The criteria count is assumed to be n at each tier. The first-level execution is given below; the second-level algorithm follows the same pattern. As such, the algorithm begins by reading the values from the fuzzy comparing matrix and then compares each Si pair in turn (lines 2-3). The order of importance of these criteria is then determined using the connection (3). An anomalous weight is calculated for each criterion. Then, using a relation, the weights are normalized (4). These steps are likewise carried out on the second floor, which is where you'll find IOTDCs. The resulting weight represents the importance of each IOTDC, and the one with the highest importance can be chosen.

Algorithm 4: Fuzzy DEMATEL

1: First, get tracking data.
2: for both 1st and 2nd graders
2-1: Input a fuzzy rating of the relevance of each criterion pair (Table II).
2-2: All criteria C1Cn (i1...n)
2-2-1: The formula is as follows: 2-2-1: Si (E.q 1) = Fuzzy Synthetic Extent
2-3: Two-Three: i, j, 1,..., n
2.4: Determine V if Si > Sj.
2-5: Use EQ to derive your actual weight for steps. 1-3
2-6: Use (EQ 2) to normalize the given Wight value.
3: Prioritize the return of the final Wight in the IOTDC.

TABLE II. TRIANGULAR FUZZY NUMBERS

Linguistic Interpretation	Fuzzy Numers	Membership Function	Interval	Linguistic Interpretation
Just Equal	$\tilde{1}$			(1,1,1)
Equally Important		µm(X)=(3-X)/(3-1)	1≤X≤3	(1,1,3)
Weakly Important	$\tilde{3}$	µm(X)=(X-1)/(3-1)	1≤X≤3	(1,3,5)
		µm(X)=(5-X)/(5-3)	3≤X≤5	
Essential Or Strongly Important	$\tilde{5}$	µm(X)=(X-3)/(5-3)	3≤X≤5	(3,5,7)
		µm(X)=(7-X)/(7-5)	5≤X≤7	
Very Strongly Important	$\tilde{7}$	µm(X)=(X-5)/(7-5)	5≤X≤7	(5,7,9)
		µm(X)=(9-X)/(9-7)	7≤X≤9	
Extremly Preferred	$\tilde{9}$	µm(X)=(X-7)/(9-7)	7≤X≤9	(7,9,9)

The weight vector is found using the following formula for i=k: k=1,2,...,n.

$$W' = (d'(A_1), d'(A_2), \ldots, d'(A_n))^T \quad (3)$$

Then, we normalize the vector W', and we get the following:

$$W = (d(A_1), d(A_2), \ldots, d(A_n))^T \quad (4)$$

E. Fuzzy Balancer

Despite efforts to choose the best available resource, certain servers may fail to execute commands in the processing of healthcare metadata for a number of reasons (like a delay in processing the preceding commands, software and hardware errors, congestion, or overload) [31, 39]. In these cases, it is necessary to migrate virtual machines (VMs) from a node with a heavy traffic load to another node and to identify as well as choose which VMs should be moved; a fuzzy selector is used. The linguistic variables and their corresponding membership functions are established on this foundation.

1) Membership functions: In this section, the membership functions are described based on the defined linguistic variables. Indeed, by using three defined input variables, a VM fuzzy selector has been provided. On this basis, the membership functions are needed to be defined. The range used for the membership functions has been defined based on the outputs of the CloudSim simulation of the PlanetLab project [32]. For every main parameter related to the resource, the membership functions related to the linguistic variables are defined as follows:

- The available storage (T_{RAM}): (low, medium, high)
- Execution time (T_{exe}) : (low, medium, high)
- Energy consumption (T_{energy}): (low, medium, high)

Now, based on the definition of the three parameters mentioned above, the VMs are categorized into classes (low, medium, high, and very high), and the machines which are located in the high and very high classes are the candidates to migrate to VMs with less and medium loads. In the following, tables and figures related to membership functions' parameters are shown. First, Table III and Fig. 3 show the membership functions related to the consumed RAM to define the low, medium, and high classes.

In Table IV and Fig. 4, the parameters of the membership functions related to energy consumption have been shown.

Finally, Table V and Fig. 5 show the parameters of the membership functions related to consumed energy.

TABLE III. PARAMETERS OF RAM MEMBERSHIP FUNCTIONS

Parameters	RAM amount
a=0 , b=0 , c=651 , d=751	Low
a=701 , b=801 , c=901 , d=1001	Medium
a=900 , b=1101 , c=1801, d=1800	High

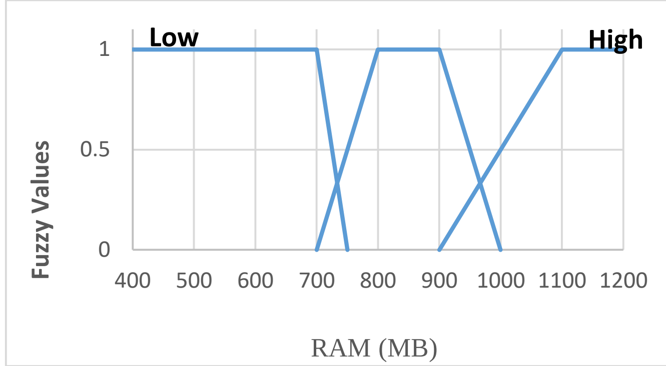


Fig. 3. RAM membership functions.

TABLE IV. PARAMETERS OF ENERGY CONSUMPTION MEMBERSHIP FUNCTIONS

Parameters	Energy amount
a=0 , b=0 , c=121 , d=136	Low
a=131 , b=141 , c=151 , d=156	Medium
a=151 , b=161 , c=171 , d=171	High

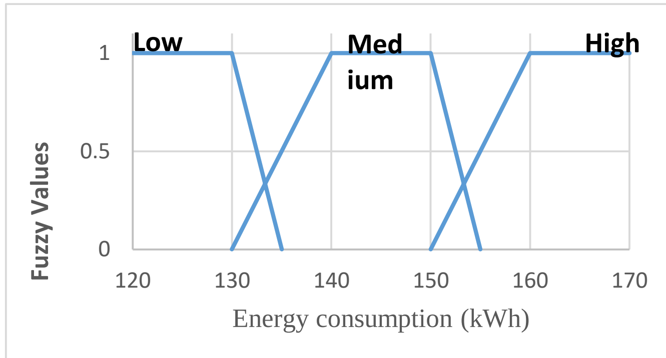


Fig. 4. Energy consumes membership functions.

TABLE V. PARAMETERS OF EXECUTION TIME MEMBERSHIP FUNCTIONS

Parameters	Amount
a=0 , b=0 , c=0.07 , d=0.08	Low
a=0.066 , b=0.07 , c=0.09 , d=0.2	Medium
a=0.08 , b=0.2 , c=0.12 , d=0.16	High

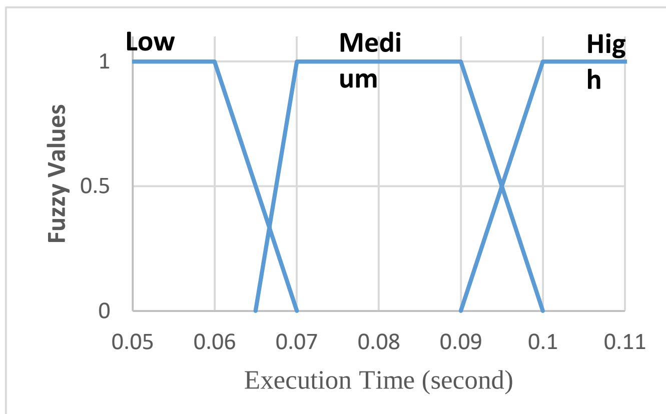


Fig. 5. Execution time membership functions.

1) Fuzzy rules: In this section, based on the membership functions described in the previous step, the fuzzy rules are defined in order to select the VMs for migration. These rules are used to select the VMs that faced overloading due to the additional load. The corresponding rules are shown in Table VI.

TABLE VI. FUZZY RULES

Execution	RAM	Energy	VM Selection
H	H	H	VH
H	H	M	VH
H	H	L	VH
H	M	H	VH
H	M	M	H
H	M	L	H
H	L	H	H
H	L	M	M
H	L	L	M
M	H	H	VH
M	H	M	H
M	H	L	M
M	M	H	H
M	M	M	M
M	M	L	L
M	L	H	M
M	L	M	L
M	L	L	L
L	H	H	H
L	H	M	M
L	H	L	M
L	M	H	M
L	M	M	L
L	M	L	L
L	L	H	L
L	L	M	L
L	L	L	L

VH: Very High L: Low M: Medium H: High

IV. EVALUATION

ClouSim [32] has been utilized for the simulation and evaluation of the proposed solution. The scalability of this simulation is high and can simulate very big clouds. At the moment, despite the simulation, a cloud is able to simulate the cloud environments composed of some other clouds. Also, in this research, the surveyed data obtained from the CoMon project, which indeed is a monitoring infrastructure for PlanetLab servers, have been used to create the workload [33, 34]. In this scenario, data from checks performed on the physical equipment at intervals of 300 milliseconds have been taken into account. Three separate assessments were conducted using varying quantities of virtual assets on this premise. In order to accomplish this, initial guesses were made that the number of virtual computers in each of the three assessments would be 898, 1033, and 1358. Two optimization techniques, Particle Swarm Optimization [17, 35] and Ant Colony Optimization (ACO) [36, 37], were used to examine the effectiveness of the proposed solution across a range of metrics, including energy usage, runtime, SLA violations, and migrations. The evaluation's findings are depicted in Fig. 6 to 9. The energy we use is depicted in Fig. 6 below. As seen in the figure, the suggested approach significantly reduces energy use. Because one of its primary goals is efficient resource management and it doesn't generate congestion in the processing servers, it was anticipated that the solution would reduce energy consumption. With the aid of the proposed fuzzy selector, the suggested method successfully avoided congestion and, by extension, increased energy usage. Fig. 7 displays the runtime of the solutions over three distinct assessments. It is clear at this point that the processing duties have been completed more quickly, thanks to the offered method. As a result of the RIMA algorithm's careful monitoring, the accurate resources prediction needed, and the best migrations, the answers for all three tests now have shorter execution times. Increased productivity may be the result of the solution's newfound capacity for accurate resource prediction, as well as the suitable migrations made by the use of the fuzzy selector. Fig. 8 also displays the total number of VM migrations performed by each solution, allowing us to examine the migrations made by each solution to achieve load balancing and load reduction. In all three tests, the migration counts of the suggested solutions were much lower than those of the other two. Still, this is not the cause for raising and optimizing the other parameters when taking into account execution time and energy usage. It's important to remember that if VM migrations aren't performed optimally and correctly, it might cause a significant overload in the network and waste server resources. Fig 8 shows that while the proposed solution has fewer migrations, those migrations are optimized and done timely, causing minimal additional network overload and avoiding unnecessary engagement of server resources. It also suggests that a large number of migrations do not necessarily indicate the superiority of that solution. The cause of the migrations' optimality is directly related to the usage of fuzzy selectors, in which the server state has been studied as well as carried out in preparation for migration.

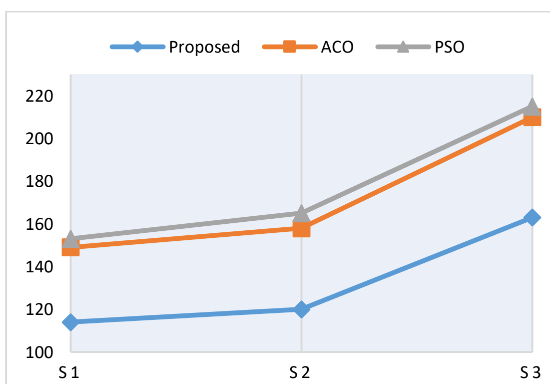


Fig. 6. Energy consumed.

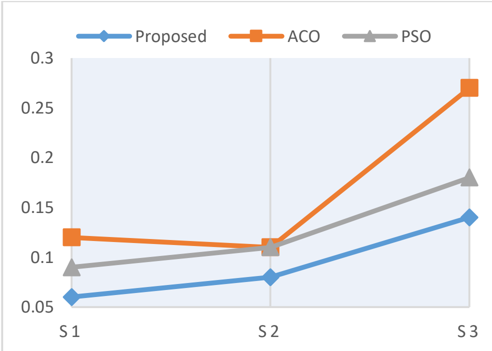


Fig. 7. Average execution time.

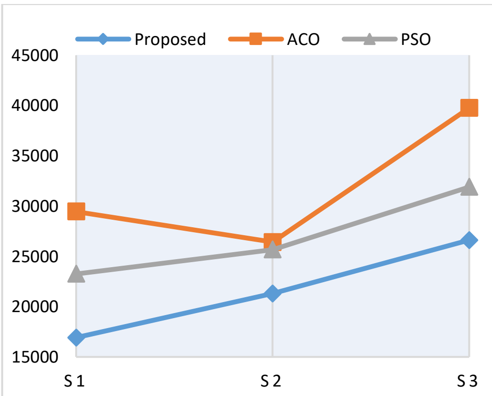


Fig. 8. Number of VM migration.

Finally, the solutions' SLA violations are displayed in Fig 9. The Service Level Agreement (SLA) is relied upon as the yardstick by which to measure actual performance. Whether the service quality is satisfactory is determined by the QoS parameters in the SLA. The primary goal of this contract is to set forth the legally binding parameters for the quality, availability, or cost of the services to be supplied. As can be seen in Fig. 9, the proposed solution makes the servers more reliable than the other two solutions, which in turn creates higher accessibility in the cloud infrastructure as well; in other words, this solution's SLA violation is lower than that of the two other ones. Load balancing, which has been accomplished with the aid of the resource fuzzy selector and predicter, and the VMs migration, which has been accomplished with the help of the fuzzy selector, are used to achieve this.

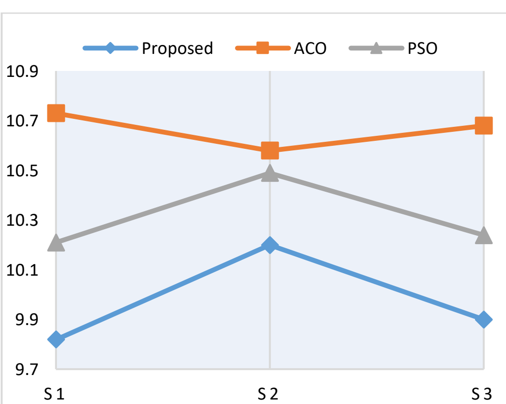


Fig. 9. Violation of the SLA.

A. *Evaluation of the Cost of Consumption*

In order to implement the cost of the proposed solution, we have considered a three-layer cloud program, where each layer has different characteristics from the other layers. Table VII shows the general characteristics of the different layers of the three-layer cloud architecture.

TABLE VII. SPECIFICATIONS OF DIFFERENT LAYERS OF THE CLOUD PROGRAM

Layer	Avg. Request Arrival Rate	VM Type	Number of Startup VMs
Web	100%	Large	5
Application	70%	Medium	5
Database	40%	Small	5

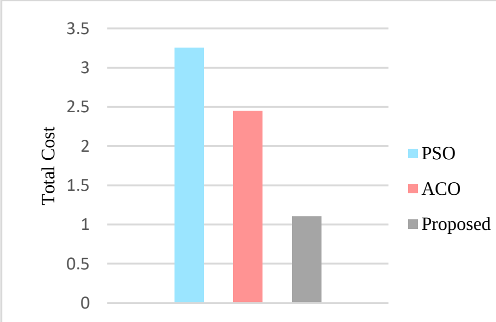
The rate of entering requests in different layers is different. More precisely, the number of requests entered in the first layer is according to what is read from the load, but the number of requests in the second and third layers is, on average equal to 70 and 40% of the requests in the first layer, respectively.

The cost is the sum of the cost of the virtual machines and the penalty cost. That the penalty cost is the total amount of the penalty cost for all customer requests, and the total cost of virtual machines is for setting up and running all virtual machines. The total cost is calculated by equation (5).

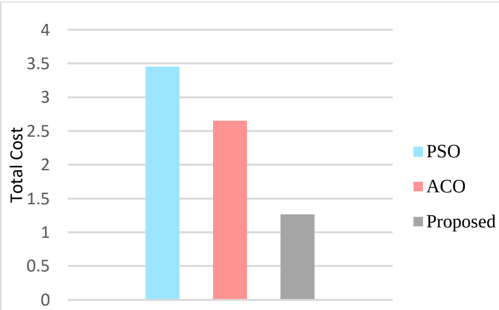
$$\text{Total Cost} = \text{VM Cost} + \text{Penalty Cost} \quad (5)$$

Fig. 10 show the average cost for different layers. The values displayed in these figures are the average of all three data sets. In Fig. 10(a), the average cost for the application layer is displayed; as can be observed, in this layer, the suggested method can uniformly provide the elasticity of cloud computing compared to other methods. In addition, the proposed method in this research reduces the total cost more than other methods. One should know about health data; due to the huge amount of data and the importance of the subject, the amount of cost is very important. Fig. 10(b) shows the cost in the web layer. Based on the distribution of the workload entered into cloud computing, the compared methods have different behaviours with the cost criterion. Any amount of workload entry has an unbalanced distribution; the complexity of its analysis and decision-making by the examined frameworks also increases. Therefore, it can be seen that in most cases and on average, the proposed method in this research has a better performance in the application layer. Fig. 10(c) examines the cost metric in the database layer for the proposed methods. This layer faces a large number of parallel transactions, so the use of dynamic methods that provide elasticity for this layer is one of the requirements that the service provider should pay great attention to. Considering the three layers of the proposed architecture for big data in the field of health, it is necessary to examine the cost criteria in each of the three layers.

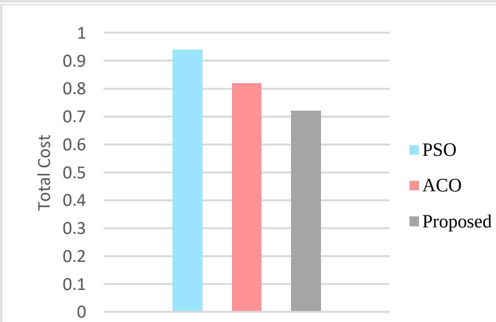
According to Fig. 10(a), (b), and (c), it can be seen that the proposed method imposes a lower average cost on the supplier.



(a) Application layer



(b) Web layer



(c) Database layer

Fig. 10. Average cost comparison for the compared methods.

B. *Quantification of Computational Complexity*

The high number of rejected requests will reduce the profit, and on the other hand, it may also increase the cost. Therefore, the proposed method should be able to reject fewer requests. In this way, it increases the profit and keeps the quality of the services within the agreed limits [38]. In this paper, the number of rejected requests is counted to quantify the computational complexity based on the timeline. According to Fig. 11 and considering the number of user requests based on the timetable, as can be seen, for example, at time point 49, the number of rejected requests for the ACO algorithm equals 145, and for the PSO algorithm, it equals has been 240. In contrast, at this point in time, the number of rejected requests for the proposed method was around 50. Therefore, according to the calculation quantity, it can be concluded that the proposed method can have a higher efficiency than other methods.

C. *Estimating the amount of Rejected Requests*

The high number of rejected requests will reduce the profit and on the other hand, it may also increase the cost. Therefore, the proposed method should be able to reject fewer requests. In this way, it increases the profit and keeps the quality of the services within the agreed limits. Fig. 12 shows the rejected requests in the application layer for the test data set. As can be seen, the proposed method in this research has a better performance than other methods.

Fig. 13 also shows the rejected requests in the web layer. Fig. 14 also shows the number of rejected requests in the database layer. From these graphs, it can be seen that the proposed method in this research has a better performance than the compared methods.

Table VIII shows that, in accordance with the behavior of the graphs, the average length of the schedule has dropped by 5.36% of the minimum execution time for the number of 300 jobs and by 6.14% when compared to the PSO method. Similarly, Table IX shows that with 600 tasks, the ratio of successful execution increases by 6.81% for the ACO schedule and by 3.92% for the lowest execution time.

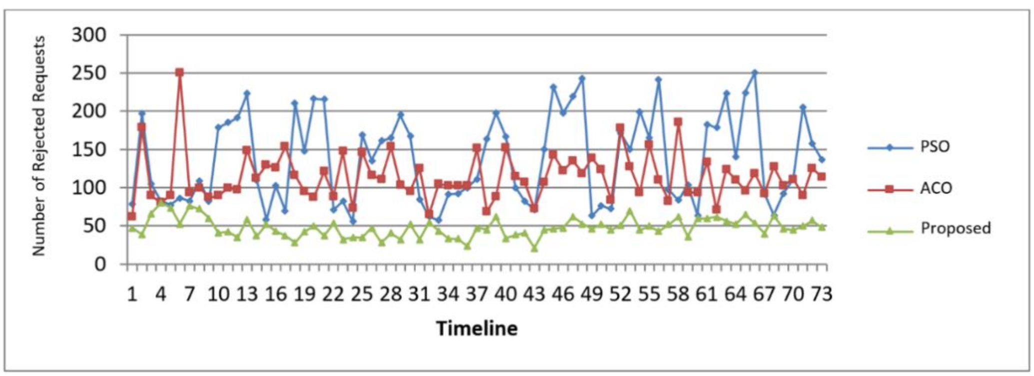


Fig. 11. Computational complexity comparison.

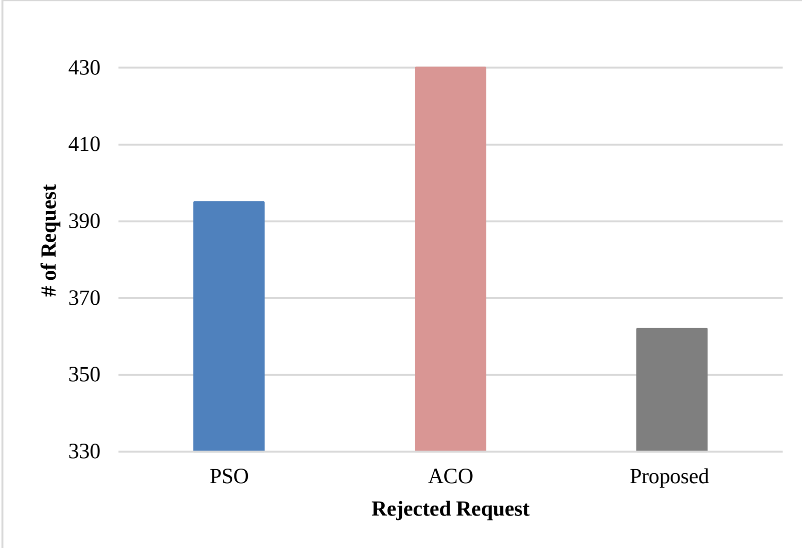


Fig. 12. Comparison of the investigated methods in the number of rejected requests in the application layer.

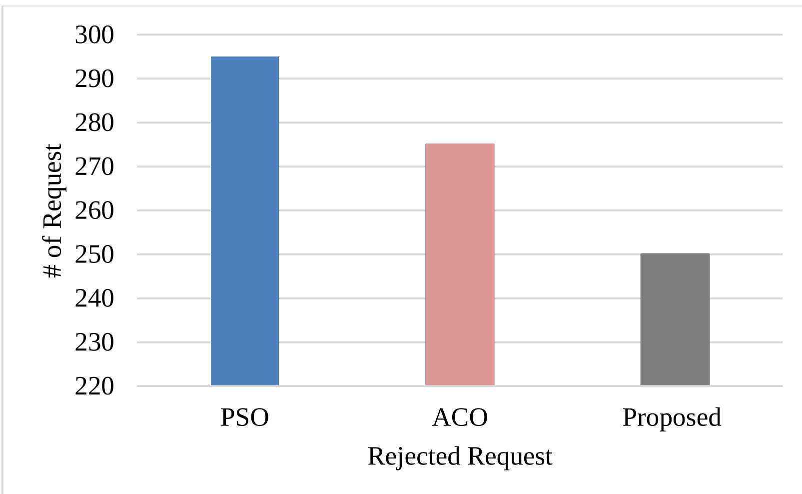


Fig. 13. Comparison of the investigated methods in the number of rejected requests in the web application layer.

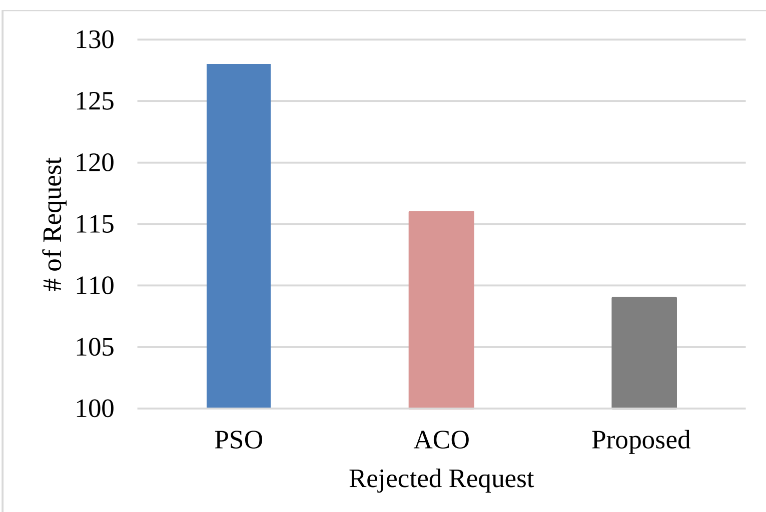


Fig. 14. Comparison of the investigated methods in the number of rejected requests in the database application layer.

TABLE VIII. THE PROPOSED METHOD COMPARED TO SIMILAR METHODS FOR AVERAGE PROGRAM LENGTH

Iteration number	Number of tasks	PSO	ACO	Proposed Method
1	100	343	336	323
2	200	1025	1007	985
3	300	1765	1729	1742
4	400	2516	2369	2236
5	500	3129	3102	3024
6	600	3024	3035	2954

TABLE IX. THE PROPOSED METHOD COMPARED WITH SIMILAR METHODS FOR THE RATIO OF SUCCESSFUL EXECUTION

Iteration number	Number of tasks	PSO	ACO	Proposed Method
1	100	0.48	0.50	0.51
2	200	0.68	0.70	0.73
3	300	0.75	0.77	0.78
4	400	0.79	0.80	0.81
5	500	0.77	0.79	0.80
6	600	0.79	0.80	0.79

D. *Real Time Case Study based Discussion*

Existing processes for collecting and processing patients' vital information require a large amount of labor and calculations. These processes are usually error-prone and have large delays that prevent correct information from being available in real time. Cloud computing as a new technology, with internet infrastructure and new solutions, has brought significant advantages in providing medical services electronically. Along with that, through the rapid development of processing and storage technologies as well as the success of the Internet, computing resources have become cheaper, stronger and more accessible than before, and government organizations have started using cloud architecture, platforms and programs to provide services and meet the needs of their subordinates. But in between, there are many challenges, the most important of which is the optimal resource management and how much processing resources each data needs.

V. CONCLUSION

Each patient's data is added to hospital systems in such large quantities that they become the metadata that must be processed by such a large number of facilities, which practically results in high costs to the hospital, additionally to execution times as well as accuracy issues. Because of this, the use of cloud computing's processing capacity for metadata has become more widespread. Accurate resource provision is a crucial element of cloud computing. User satisfaction rises proportionally as the accuracy of the resources grows, and the number of violations of the services decreases. In order to handle healthcare metadata in a cloud computing environment, a better resource management approach has been provided in the current research. It was assumed that the proposed architecture would have each IOTDC have a Broker, who would first receive and analyse requests from users referred to that IOTDC before executing the algorithm and returning the result to the users of that IOTDC if the resources required to execute those requests were available. However, the DEMATEL fuzzy algorithm should be used by the Broker to select the appropriate IOTDC from the available associated IOTDCs if it is unable to fulfil user requests using the available IOTDC resources. The proposed solution was then tested and evaluated using the PSO and ACO algorithms based on the various factors, including SLA and execution time, developed and simulated in the ClouSim program. The findings indicate that the solution performed better in all tests run. Among the things that can be done in the future in the continuation of the research are: combining time series to predict resources optimally, combining the reinforcement learning method with fuzzy logic for automatic scaling, and using the combination of reinforcement learning and neural network in methods based on service level agreements, which can significantly improve the proposed solution.

FUNDING

This work was supported by the Natural Science Foundation of the Anhui Wenda University of Information Engineering： Research on the Prevention and Control System of Sudden Diseases on Campus in Universities (No: XZR2020A09).

REFERENCES

[1] S. S. Gill et al., "Holistic resource management for sustainable and reliable cloud computing: An innovative solution to global challenge," Journal of Systems and Software, vol. 155, pp. 104–129, 2019.

[2] F. Nzanywayingoma and Y. Yang, "Efficient resource management techniques in cloud computing environment: a review and discussion," International Journal of Computers and Applications, vol. 41, no. 3, pp. 165–182, 2019.

[3] P. Kumar and R. Kumar, "Issues and challenges of load balancing techniques in cloud computing: A survey," ACM Computing Surveys (CSUR), vol. 51, no. 6, pp. 1–35, 2019.

[4] M. Elhoseny, A. Abdelaziz, A. S. Salama, A. M. Riad, K. Muhammad, and A. K. Sangaiah, "A hybrid model of internet of things and cloud computing to manage big data in health services applications," Future generation computer systems, vol. 86, pp. 1383–1394, 2018.

[5] M. Trik, A. M. N. G. Molk, F. Ghasemi, and P. Pouryeganeh, “A hybrid selection strategy based on traffic analysis for improving performance in networks on chip,” J Sens, vol. 2022, 2022.

[6] G. Manogaran, C. Thota, and D. Lopez, “Human-computer interaction with big data analytics,” in Research Anthology on Big Data Analytics, Architectures, and Applications, IGI global, 2022, pp. 1578–1596.

[7] A. Gaurav and K. T. Chui, “Advancement of Cloud Computing and Big Data Analytics in Healthcare Sector Security,” Data Science Insights Magazine, Insights2Techinfo, vol. 1, pp. 12–15, 2022.

[8] Chen Cao, Jianhua Wang, Devin Kwok, Zilong Zhang, Feifei Cui, Da Zhao, Mulin Jun Li, Quan Zou. webTWAS: a resource for disease candidate susceptibility genes identified by transcriptome-wide association study. Nucleic Acids Research.2022, 50(D1): D1123-D1130.

[9] A. G. Sreedevi, T. N. Harshitha, V. Sugumaran, and P. Shankar, “Application of cognitive computing in healthcare, cybersecurity, big data and IoT: A literature review,” Inf Process Manag, vol. 59, no. 2, p. 102888, 2022.

[10] Ning Xu, Zhongyu Chen, Ben Niu, and Xudong Zhao. Event-Triggered Distributed Consensus Tracking for Nonlinear Multi-Agent Systems: A Minimal Approximation Approach, IEEE Journal on Emerging and Selected Topics in Circuits and Systems, DOI: 10.1109/JETCAS.2023.3277544, 2023.

[11] M. Trik, S. P. Mozaffari, and A. M. Bidgoli, “Providing an adaptive routing along with a hybrid selection strategy to increase efficiency in NoC-based neuromorphic systems,” Comput Intell Neurosci, vol. 2021, 2021.

[12] Haoyu Zhang, Quan Zou, Ying Ju, Chenggang Song, Dong Chen. Distance-based Support Vector Machine to Predict DNA N6-methyladine Modification. Current Bioinformatics. 2022, 17(5): 473-482.

[13] R. Weerabathiran and K. A. Srinath, “Application of the extent analysis method on fuzzy AHP,” International Journal of Engineering Science and Technology, vol. 4, no. 7, pp. 3472–3480, 2012.

[14] M. Zakarya and L. Gillam, “Modelling resource heterogeneities in cloud simulations and quantifying their accuracy,” Simul Model Pract Theory, vol. 94, pp. 43–65, 2019.

[15] Khezri, E., Zeinali, E., & Sargolzaey, H. (2022). A novel highway routing protocol in vehicular ad hoc networks using VMaSC-LTE and DBA-MAC protocols. Wireless Communications and Mobile Computing, 2022.

[16] E. Barbierato, M. Gribaudo, M. Iacono, and A. Jakobik, “Exploiting CloudSim in a multiformalism modeling approach for cloud based systems,” Simul Model Pract Theory, vol. 93, pp. 133–147, 2019.

[17] K. Park and V. S. Pai, “CoMon: a mostly-scalable monitoring system for PlanetLab,” ACM SIGOPS Operating Systems Review, vol. 40, no. 1, pp. 65–74, 2006.

[18] B. Jana, M. Chakraborty, and T. Mandal, “A task scheduling technique based on particle swarm optimization algorithm in cloud environment,” in Soft Computing: Theories and Applications: Proceedings of SoCTA 2017, Springer, 2019, pp. 525–536.

[19] M. Vahidi Farashah, A. Etebarian, R. Azmi, and R. Ebrahimzadeh Dastjerdi, “A hybrid recommender system based-on link prediction for movie baskets analysis,” J Big Data, vol. 8, pp. 1–24, 2021.

[20] W. Li et al., “A comprehensive survey on machine learning-based big data analytics for IoT-enabled smart healthcare system,” Mobile networks and applications, vol. 26, pp. 234–252, 2021.

[21] M. Trik, S. Pour Mozafari, and A. M. Bidgoli, “An adaptive routing strategy to reduce energy consumption in network on chip,” Journal of Advances in Computer Research, vol. 12, no. 3, pp. 13–26, 2021.

[22] A. Alelaiwi, “A collaborative resource management for big IoT data processing in Cloud,” Cluster Comput, vol. 20, pp. 1791–1799, 2017.

[23] N. Saraei, M. Khanal, and M. Tizghadam, Removing Acidic Yellow Dye from Wastewater Using Moringa Peregrina, Computational Research Progress in Applied Science & Engineering 8 (2022) 1–8, doi: 10.52547/crpase.8.3.2483.

[24] M. Hadji and D. Zeghlache, “Mathematical programming approach for revenue maximization in cloud federations,” IEEE transactions on cloud computing, vol. 5, no. 1, pp. 99–111, 2015.

[25] M. Vahidi Farashah, A. Etebarian, R. Azmi, and R. Ebrahimzadeh Dastjerdi, “An analytics model for TelecoVAS customers’ basket clustering using ensemble learning approach,” J Big Data, vol. 8, pp. 1–24, 2021.

[26] Haoyan Zhang, Xudong Zhao, Huangqing Wang, Ben Niu, Ning Xu, Adaptive Tracking Control for Output-Constrained Switched MIMO Pure-Feedback Nonlinear Systems with Input Saturation, Journal of systems science & complexity, 36: 960–984, 2023.

[27] G. Agapito and M. Cannataro, “An Overview on the Challenges and Limitations Using Cloud Computing in Healthcare Corporations,” Big Data and Cognitive Computing, vol. 7, no. 2, p. 68, 2023.

[28] S. J. Miah, E. Camilleri, and H. Q. Vu, “Big Data in healthcare research: a survey study,” Journal of Computer Information Systems, vol. 62, no. 3, pp. 480–492, 2022.

[29] Khezri, E., Zeinali, E., & Sargolzaey, H. (2023). SGHRP: Secure Greedy Highway Routing Protocol with authentication and increased privacy in vehicular ad hoc networks. Plos one, 18(4), e0282031.

[30] M. Trik, H. Akhavan, A. M. Bidgoli, A. M. N. G. Molk, H. Vashani, and S. P. Mozaffari, “A new adaptive selection strategy for reducing latency in networks on chip,” Integration, vol. 89, pp. 9–24, 2023.

[31] G. Dhiman et al., “Federated learning approach to protect healthcare data over big data scenario,” Sustainability, vol. 14, no. 5, p. 2500, 2022.

[32] A. T. Lo’ai and G. Saldamli, “Reconsidering big data security and privacy in cloud and mobile cloud systems,” Journal of King Saud University-Computer and Information Sciences, vol. 33, no. 7, pp. 810–819, 2021.

[33] Khezri, E., & Zeinali, E. (2021). A review on highway routing protocols in vehicular ad hoc networks. SN Computer Science, 2, 1-22.

[34] M. Karatas, L. Eriskin, M. Deveci, D. Pamucar, and H. Garg, “Big Data for Healthcare Industry 4.0: Applications, challenges and future perspectives,” Expert Syst Appl, p. 116912, 2022.

[35] J. Wang, C. Xu, J. Zhang, and R. Zhong, “Big data analytics for intelligent manufacturing systems: A review,” J Manuf Syst, vol. 62, pp. 738–752, 2022.

[36] Heng Zhao, Huanqing Wang, Ben Niu, Xudong Zhao, K. H. Alharbi, Event-Triggered Fault-Tolerant Control for Input-Constrained Nonlinear Systems With Mismatched Disturbances via Adaptive Dynamic Programming, Neural Networks, 164: 508-520, 2023.

[37] A. Selvakumar and G. Gunasekaran, “A novel approach of load balancing and task scheduling using ant colony optimization algorithm,” International Journal of Software Innovation (IJSI), vol. 7, no. 2, pp. 9–20, 2019.

[38] J. Sun, Y. Zhang, and M. Trik, “PBPHS: a profile-based predictive handover strategy for 5G networks,” Cybern Syst, pp. 1–22, 2022.

[39] Zhongwen Cao; Ben Niu; Guangdeng Zong; Xudong Zhao; Adil M. Ahmad, "Active Disturbance Rejection-Based Event-Triggered Bipartite Consensus Control for Nonaffine Nonlinear Multiagent Systems", International Journal of Robust and Nonlinear Control, DOI:10.1002/rnc.6746.