Inverted Ant Colony Optimization Algorithm for Data Replication in Cloud Computing

Min YANG*
Chongqing City Vocational College, Chongqing; 402160, China

***Abstract*—Data replication is crucial in enhancing data availability and reducing access latency in cloud computing. This paper presents a dynamic duplicate management method for cloud storage systems based on the Inverted Ant Colony Optimization (IACO) algorithm and a fuzzy logic system. The proposed approach optimizes data replication decisions focusing on energy consumption, response time, and cost. Extensive simulations demonstrate that the IACO-based method outperforms existing techniques, achieving a remarkable 25% reduction in energy consumption, a significant 15% improvement in response time, and a substantial 20% cost reduction. By addressing the research gap concerning integrating IACO and fuzzy logic for data replication, our work contributes to advancing cloud computing solutions for large datasets. The proposed method offers a viable and efficient approach to improve resource utilization and system performance, benefiting various scientific fields.**

Keywords—Cloud computing; data replication; cloud data centers; reliability; energy efficiency; inverted ant colony optimization algorithm

I. Introduction

Cloud computing provides significant opportunities for progress in the information technology industry. As an emerging form of a distributed network, cloud computing entails providing hardware and software resources across a range of businesses and organizations [1]. Users are only charged for the services offered by the cloud provider. Several types of cloud computing services are available, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [2]. SaaS shares software applications between customers in the cloud through Internet-based service providers. Users do not need to install software on their personal computers. The PaaS model involves sharing a platform with consumers responsible for configuring, deploying, testing, and developing the application. Consumers do not maintain servers, hardware, and storage, although they may control applications. To run applications, the cloud service providers supply a pre-installed operating system. IaaS is a model for deploying consumer applications and systems on shared infrastructure [3].

Four major cloud computing deployment models exist: community, public, private, and hybrid clouds. An organization can use private clouds inside or outside of their offices. It is located within the virtualized data center's firewall. Users cannot control the services provided by the cloud service providers, as they provide the infrastructure. An organization that provides cloud services to the public or large industries operates a public cloud. Cloud service providers control the services and provide the infrastructure within the public cloud. Community clouds provide cloud services offered by multiple organizations to assist individuals with similar concerns. In hybrid clouds, numerous cloud models are combined, utilizing proprietary or standardized technology that permits the portability of data and applications [4, 5].

In the realm of cloud computing, the integration and utilization of various cutting-edge concepts are instrumental in driving innovation, efficiency, and intelligence. The Internet of Things (IoT) plays a pivotal role by connecting a myriad of devices and sensors, generating vast volumes of data [6, 7]. This data deluge, known as big data, necessitates advanced data processing and storage capabilities that cloud computing can readily provide [8, 9]. Artificial Intelligence (AI) and Machine Learning (ML) are the cornerstones of intelligent cloud applications, enabling systems to learn from data patterns, make informed decisions, and automate processes, thereby enhancing user experiences and optimizing operations [10-14]. Feature and channel selection are vital in cloud computing scenarios dealing with large and diverse data sources [15]. By identifying the most relevant features and data channels, cloud-based systems can reduce computational overhead, improve accuracy, and streamline data processing [16, 17]. Furthermore, meta-heuristic algorithms in the cloud facilitate efficient optimization of resource allocation, task scheduling, and load balancing, leading to optimal resource utilization and performance [18-21]. Association rule mining enables the discovery of valuable patterns and correlations within data, empowering businesses to make data-driven decisions and tailor services to individual preferences [22]. Additionally, game theory principles come into play in cloud computing, where cloud service providers and users engage in strategic interactions. Through game theory, fair and efficient pricing models can be devised, and resource allocation strategies can be optimized to ensure mutually beneficial outcomes [23].

Providing reliable services in conjunction with a high degree of availability and data performance constitute essential criteria. These requirements are ensured through the use of the replication concept. Data replication involves replicating the duplicate files at different locations to improve reliability, availability, and performance [24]. Data replication plays a vital role in data-intensive environments in which records must be synchronized between multiple sites located at diverse locations. The availability of the data is increased through replication. Requests may be fulfilled by other sites that host replicas of the requested file if the site responsible for hosting these files cannot serve them. Furthermore, replication enhances the overall performance of the system by allowing the user's requests to be satisfied at the nearest site that also holds replicas of the data file requested. In this manner, the system's response and access times are improved. As a final point, replication also enhances the reliability of the system. Despite the corruption or failure of any replica of a data file, user requests can be processed and handled by other non-corrupted servers that hold the required replica. In most cases, replication strategies cannot be adapted to cloud environments due to their intention to improve system performance by not considering replication costs. The creation of multiple replicas within cloud computing environments is not economically practical as it may lead to wasteful resource utilization and a reduction in service providers' profits. Thus, data replication approaches applied to cloud-based platforms must also meet the following objectives:

- Maintaining reliable Quality of Service (QoS) through the observance of Service Level Agreements (SLAs), which constitute legal contracts between cloud providers and their tenants, i.e., customers. Typically, an SLA consists of one or more tenant Service Level Objectives (SLOs), i.e., requirements.
- Based on the pay-as-you-go pricing model, the provider dynamically adjusts the resources it rents to its tenants.

Data replication, a critical technique for distributing data across multiple locations, remains challenging in cloud computing. The problem arises from the need to strike a balance between enhancing data availability and minimizing access latency for users. Replicating data to multiple locations can improve system availability and ensure users can access data from a nearby location. However, determining the optimal number and locations for data replicas is a complex optimization problem. This paper aims to address the data replication challenge in cloud computing by proposing a novel method of dynamic duplicate management. Our research draws inspiration from the Ant Colony Optimization (ACO) algorithm, which has shown promising results in solving combinatorial optimization problems. In particular, we introduce the Inverted ACO (IACO) algorithm, tailored to tackle the data replication issue in cloud storage systems. By leveraging the IACO algorithm and a fuzzy logic system, we seek to optimize data replication decisions regarding energy consumption, cost, and response time. The objective is to improve the overall performance of cloud data centers while maintaining data availability for end-users.

The rest of this paper is structured as follows: Section II presents the related work in data replication techniques for cloud computing. Section III provides a detailed description of the proposed IACO-based dynamic duplicate management approach. Section IV presents the results and analysis of extensive simulations, comparing the proposed method with existing techniques. Finally, Section V concludes the paper and outlines potential future research directions.

II. Related Work

In order to solve the data replication problem, Ebadi and Jafari Navimipour [25] proposed a hybrid metaheuristic algorithm. To achieve high-quality solutions, it combines the global search capabilities of the PSO algorithm with the local search capabilities of the Tabu search algorithm. Various test cases are used to demonstrate the efficiency of the method compared to PSO, TS, and ACO algorithms. Based on the results obtained, it appears that the method is more efficient in terms of energy consumption and cost.

By combining multiple objective optimizations and evaluating the cost distance using the knapsack problem, Salem, et al. [26] presented the shortest routes and the lowest costs in the cloud based on replication and data center placement. By using the Artificial Bee Colony (ABC) algorithm, the shortest route and lower-cost problems have been solved, identifying the best replication placement based on the distances of minimizing the costs and achieving the shortest routes that were achieved when the Knapsack algorithm was employed to deal with these issues. The proposed system can be optimized using the MOABC algorithm to achieve high efficiency and low costs. By using the Knapsack approach and optimizing replica distribution based on distances and data transmission rates, MOABC is capable of finding a suitable location to place data replicas. Based on measurements, MOABC is more effective and efficient than compared algorithms in regards to determining the best placement of replications.

Rambabu and Govardhan [27] proposed a new data replication approach derived from data mining principles. Data replication is accomplished by identifying widely used data patterns within the massive database of a node. A novel hybrid algorithm selects the best thresholds based on an optimization-assisted frequent pattern mining strategy. Greywolves Updated Exploration and Exploitation with Sealion Behavior (GUEES) is a hybrid algorithm that combines the concepts of Grey Wolf Optimization (GWO) and Sealion Optimization Model (SLnO). During the mining process, two criteria are defined: (i) Prioritization and (ii) Cost. The prioritization process can be divided into two categories: queueing both high and low-priority data, and the storage demand determines the cost. High-priority queues are optimized with GUEES. As a final step, a comparative validation is performed to evaluate the efficiency of the proposed model.

Mansouri, et al. [24] propose a hierarchical data replication strategy (HDRS) for dynamic replication. HDRS involves creating replicas capable of being dynamically increased by growing exponentially or depleting exponentially, placing replicas based on access loads and labeling techniques, and replacing replicas depending on future file values. CloudSim is used to evaluate different dynamic data replication methods. In comparison with other algorithms, HDRS has been demonstrated to reduce response time and bandwidth consumption. A popular file is replicated to the best site by the HDRS. By balancing the load of the sites, this method avoids unnecessary replications and reduces access latency.

Khelifa, et al. [28] suggested a periodic and dynamic data replication approach for cloud-based federated systems. The objective of this method is to ensure financial success for cloud service providers by meeting the demands of their users regarding availability and response time. In order to detect replicas, a periodic review of the users' tasks is conducted by employing the spectral clustering algorithm to determine relationships among remote data pertaining to SLA violations. Adapting correlations of this type can lead to a reduction in both the volume of data to be transferred and its transfer time. In the next step, a fuzzy inference system is used to select replicas from the owned or leased resources of other providers based on four main parameters. Additionally, replica numbers are adjusted when SLAs are met. In accordance with the results obtained, the proposed strategy reduces the number of SLA violations while maintaining the providers' monetary profits.

Despite the various existing techniques, there remains a research gap concerning the integration of the IACO algorithm and fuzzy logic for dynamic duplicate management in cloud storage systems. Previous studies have primarily focused on standard ACO algorithms or hybrid metaheuristics. Our research aims to bridge this gap by introducing the IACO algorithm, a novel adaptation with potential advantages in data replication. By combining IACO with a fuzzy logic system, our approach targets energy consumption, cost reduction, and response time improvement offering a comprehensive and efficient solution for data replication in cloud computing environments.

III. Proposed Method

Replication of data in a cloud environment reduces network delays, bandwidth usage, and costs. The replication of data to several sites improves data accessibility by allowing users to access data at their convenience. Cloud systems, however, face the challenge of determining the appropriate number and location of replicas. By increasing the number of replicas, accessibility is enhanced and access delay is reduced, but replication costs increase. In this regard, it is crucial to present a method that reduces data replication costs and balances accessibility and cost. In this study, a new approach based on the IACO algorithm and fuzzy logic system is proposed to improve the management of replication versions in cloud storage systems.

A. Problem Modeling and Formulation

Our model considers a cloud computing network with M data centers, each of which has its own storage media, memory, and processing power. si refers to the storage capacity of data center i (1≤i≤M) measured in a simple data unit (like blocks). The connection between two data centers, i and j (if any), has a positive integer number C (i, j) and another positive integer number E (i, j) related to it. Energy and communication costs are assigned to transfer a data unit among the data centers (communication costs can be expressed as delay, number of nodes, and so on). If there is no direct communication link between two data centers, the cost is calculated by the sum of all communication costs along the selected path between the data centers. Moreover, the energy is obtained by the sum of all communication energies along the selection path from data center i to j. The required assumptions for the proposed method are presented in the following:

$$C (i, j) = C (j, i) \text{ and } E (i, j) = E (j, i)$$

There are N data d1, d2, . . ., dN. The size of each data is shown by d_k(1≤k≤N) and is measured in a simple data unit. r_{ik} and w_{ik} are the total numbers of the read and written values during time T (the k index is used for data, and i and j indexes are used for data centers). r_{ik} includes some of the reads of local generation on i and some intermittent reads on their path to the objective via i.

Each data d_k has a main version on the network that is not re-transferable and removable. The main version's data center maintains d_k showing by cdbk. Each main data center cdbk considers the maintenance of information related to the replication marked by RSk for each d_k (a part of the data center in which each data k is repeated). RS = {RS1, RS2, . . ., RSN} is used to determine the replication plans for all data. Moreover, the main data center, cdbk, and the nearest data center $dcdb_{ik}$ store d_k. $dcdb_{ik}$is a data center that leads to the minimum cost and energy for read requests from i for d_k. Consider that if $dcdb_{ik} = cdb_k$, the main data center is close to the data center, and if $dcdb_{ik} = i$, the data center i maintains a replica of d_k. Hence, each request is served locally, as read for data (if there is a local replica of the data) or transferred to $dcdb_{ik}$, which can answer the request. To ease, we assume an update of d_k is performed by transmission of the updated replicas of the data to the main data center cdbk that is propagated to each data center, which has a replica of d_k (there are also other plans).

Based on the proposed system model in Fig. 1, the 1, 2, 3, ..., n locations are different, distributed, and connected by firmware, including different methods to access the replicas and their management. The replicas of file x are stored in locations 1, 2, 3, ..., n. Assuming that user 1 wants to access file x if distance and cost are important for the user, locations 1 and 3 are close to user 1, and their access cost is low. Different copies of file x make the file accessible if the information is lost in different locations. It results in more reliability and accessibility rate.

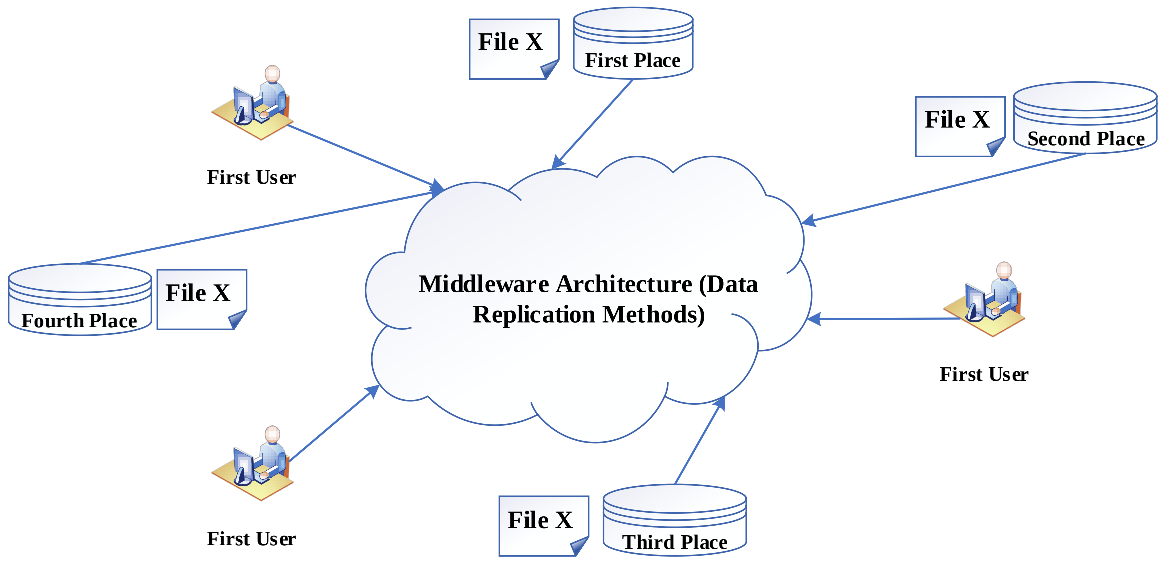

Fig. 1. Proposed model system.

B. *Proposed Hybrid Algorithm*

Different stages of the proposed method are explained in this section.

1) Determining the type of replicated file: The popular files are recognized, and the replication is performed. Based on it in this stage, it is investigated which file should be replicated. The proposed method performs this stage by recognizing the popular files. This stage uses the following three substages to select the files that should be replicated.

- Substage 1: Calculating the number of accesses to the files: Clustering is used to calculate the number of file accesses in this stage so that there are some cloud data centers in each cluster. A cluster head is considered in each cluster that its bandwidth is wider than other cloud data centers and can connect all the in-cluster cloud data centers. Since cloud data centers have some constraints like bandwidth limitation, the large size of the file, and the number of interactions among the cloud data centers, a method must reduce the interactions. Hence, first, the requests in each cluster are checked in each cluster to relocate the required files of that local cluster via the high bandwidth connection if possible. Transferring the high-volume files among the clusters can be avoided using clustering. This substage is explained as an algorithm in the following:
    - Calculating the count of the requests to file $\boldsymbol{f_i}$ in cluster C ($\boldsymbol{Num_{Req_c}(f_i)}$)
    - Decreasing the sort of all files based on $\boldsymbol{Num_{Req_c}(f_i)}$
    - Storing the sorted files in a list in each cluster
    - Determining the count of all the required files ($\boldsymbol{TF_c^n}$) of the data center c in period n (this stage is performed for all clusters).

- Substage 2: Calculating the amount of the file popularity: After determining the number of requests to the file in the previous stage, the file's popularity is calculated in this stage. To calculate the popularity of file f_i in period n, if $Num_{Req_c}(f_i) > 0$, equation (1) is used, otherwise, equation (2) is used. $Num_{Req_c}(f_i)$ is the count of the requests for file f_i from cluster c in period n. $size(f_i)$ is the file's size, and $TotalReq^n$ is the total number of requests in period n. p and q are constants, and their values directly affect the response time as one of the necessities for the proposed method's Quality of Service (QoS).

$$FP_c^n(f_i) = \frac{FP_c^{n-1}(f_i)+(Num_{Req_c}(f_i)\times p)}{size(f_i)} \times \frac{Num_{Req_c}(f_i)}{TotalReq^n} \quad (1)$$

$$FP_c^{n-1}(f_i) - q \quad (2)$$

- Substage 3: This substage involves determining the candidate files for replication. The count of the files, which should be replicated based on the two previous stages, is determined in this stage. Moreover, the files whose average popularity is calculated in the previous stages are sorted descendingly. In equation (3) $RepNum_i$ shows the number of files that must be copied and the proposed method selects the $RepNum_i$ files from begin of the sorted list to copy. ρ is a random number between 0 and 1. 30% of the repeatedly requested files are copied in the proposed method, meaning the x value is set to 0.7.

$$RepNum_i = TF_c^n \times (1-\rho) \quad (3)$$

2) Selecting the most suitable datacenter to save new replicas: One of the most important stages in the replication management process is finding the most suitable cloud data center to store the replicas. We use two indexes to find the most appropriate cloud data center to store the replicas. These indexes select the candidate data centers to store these replicas.

TABLE I. VARIABLES IN EQUATION (4)

Parameter	Definition
$\boldsymbol{Num_{Req_c}(f_i)}$	The number of requests for the file $\boldsymbol{f_i}$
$\boldsymbol{HNR_i}$	The data center with the highest number of requests for the $\boldsymbol{f_i}$ file
$\boldsymbol{FS^n}$	Cloud data center with free storage space
TS	Total storage space
$\boldsymbol{Dis^n}$	The distance between cloud data center n and other data centers
$\boldsymbol{HD_{is}}$	The cloud data center with the highest value for the Dis parameter

Index 1: The amount of the data center fitting: Fit_i^n can be calculated for each file f_i for a data center like n using equation (4). This equation explanation is presented in Table I.

$$\mathrm{Fit}_i^n = w_1 \times \frac{\mathrm{Num_{Req_c}(f_i)}}{\mathrm{HNR_i}} \times w_2 \times \frac{\mathrm{FS^n}}{\mathrm{TS}} + w_3 \times (1 - \frac{\mathrm{Dis^n}}{\mathrm{HD_{is}}})) \quad (4)$$

In equation (4), the coefficients w_1 , w_2 , and w_3 are weighted and between 0 and 1. The important note about these coefficients is that they can be valued based on the user's opinion, and their sum should be 1. Hence, the importance and priority of the user can be modeled based on this index. For example, if a user's priority is the data centers with the most repetition, $\frac{Num_{Req_c}(f_i)}{HNR_i}$, meaning that the w_1 coefficient should be increased, and w_2 and w_3 should be decreased. Generally, the amount of the tendency and the priority of a user can be modeled using the above equation.

Index 2: The amount of the total cost of the cloud data center: Generally, in addition to the amount of fitting the cloud data center, another index should be used to calculate the total cost of the data center and select the most suitable data center to store the replicas based on the movement pattern of the inverse ants. In this stage, QoS parameters determine the total cost of the data center. These parameters are used in Fuzzy logic (a fitting function for each data center) to evaluate the candidate centers for the replica's storage. After calculating the two indexes to select the most suitable cloud data center to store the replicas, IACO algorithm and fuzzy logic are used to traverse the cloud data center and evaluate each center based on these two indexes.

The IACO algorithm is an improvement of the ACO algorithm in that the pheromone of the path affects the ants' movement inversely, and it has a hatred effect on the ants instead of attract effect. At first, a graph is considered, including nodes and the edges connecting them. Each node in the proposed method is a cloud data center, including different data that a cloud's data may be repeated in another cloud. A user's requested data and the objective data to answer the user's request are determined. Then, the beginning node is selected randomly, and all the nodes start their movement from the beginning node. The ants traverse the graph, and each ant selects the next ant using equation (5), including pheromone, heuristic, and the condition to check its existence. The heuristic matrix in the proposed inverse ants' algorithm is calculated and valued using fuzzy logic. This fuzzy system's stages are presented in the following.

At first, because there is no pheromone on the path, it is not required to have a primary traverse to create the pheromone on the graph, the first ant has a traverse, and the pheromone on the traversed path is updated using equation (6) after its satisfaction. The next ants may not select the traversed path by the first ant because the pheromone's smell is not desired for the next ants. The second ant chooses the graph's best node (the node with the highest probability) to traverse and moves to the node. If all the paths are traversed, and there are some ants, the remaining ants copy the traversed paths by the first, second, and other ants.

A tolerance is selected for the graph's nodes because the objective data may be in the beginning node. If there is no tolerance, all ants select the first node, which leads to the inefficiency of the proposed method to provide workload balance among the nodes. The considered tolerance value is one so that if the utilized data is from a cloud data center, this center's tolerance is zero, and its data cannot be used unless the tolerances of other cloud centers with the objective data are also zero. If the node's tolerance is zero, we can pass over the node but cannot use its data. A complete graph tolerance is considered only for the first node. But it is considered for all the cloud centers in a non-complete graph.

If the objective data is not in the beginning node and the graph is complete, the selected node to traverse by the first ant is the node with the appropriate data. But if the graph is non-complete, a specific policy is used to choose the suitable data. If the objective data is in the beginning node, the first ant should use that cloud center's data, and the second ant should select the next node to traverse. Hence, the beginning node is compared with the traversed node by the second ant to find the objective data. If we are on the beginning node, and the objective data is not in its adjacent nodes, the nodes are selected randomly, and their adjacent nodes are also checked. Finally, the nodes, including the objective data, are compared based on the heuristic values, and the better node, including the objective data, is selected. These heuristic values are generated using the proposed Fuzzy system.

Each data center has some parameters called QoS parameters. QoS parameters are non-operational features of the system, which can be provided by the service providers, evaluated by their performance, and monitored by the users. QoS parameters can be divided into two categories: positive and negative. Increasing values of the positive QoS parameters can be helpful for the users, like accessibility and reliability. On the other hand, decreasing values of the negative QoS parameters can be useful for users, like cost and execution time. Each cloud data center is equal to an ant in the inverse ants' algorithm, and the cloud data centers are evaluated based on the QoS parameters. These parameters are the inputs to the proposed fuzzy model, and an output value is generated after defuzzification. This fuzzy generated value is used as a heuristic function value, and the ants traverse the network based on these values. A list of the positive and negative parameters with their units is defined in the following.

Value: It shows the amount of service reliability (in percent).

Accessibility: It shows the probability of service accessibility (in percent).

Cost: It is the cost that the service requester should pay to the service provider to use the services (in dollars).

Response time: It is the execution time between the reception of the service and the response to the service (in milliseconds).

Consumption energy: We used the energy model introduced in [29]. Energy consumption is being increased by the number of replicas which is relied by the given work resources so that for decreasing the level in energy amount.

Equation (5) is used to select the cloud data center for graph traversal. The first ant, which starts the traverse, chooses the optimal data, and the optimal data is in the optimal cloud data center with more probability for selection. After selecting the optimal data and traversing the path related to the data center, the path's pheromone is updated, and its smell makes hatred for the following ants. Hence, the path traversed by the previous ant may be ignored by the following ants. It continues by the next ant while all the paths to reach the objective data are traversed. Then if there is any ant, the traversed paths by the previous ants are repeated in order.

$$p_{kj}^{i}(t) = \frac{[\tau_{kj}(t)]^{\alpha}[\eta_{kj}(t)]^{\beta}[D]^{\gamma}}{\sum_{l=1}^{n} p_{kl}^{i}(t)[\tau_{kl}(t)]^{\alpha}[\eta_{kl}(t)]^{\beta}[D]^{\gamma}} \qquad (5)$$

In this equation, t is the current replica, $\tau_{kj}(t)$ is the existing pheromone in an edge (e_{kj}), and D is the condition of checking data existence. If the objective data is on the cloud data center, D=1, else D=0. η_{kj} refers to the heuristic value generated by the proposed fuzzy logic system. α, β, and γ are the effects on the pheromone, the heuristic, and the condition of checking the existing data that all three values are considered 1 in this method. Equation (6) shows the pheromone update in each repetition that $\tau(t)$ is the value of the previous pheromone and is considered one before traversing a node by the ants. The k denotes the number of traversed nodes.

$$\tau(t+1) = \tau(t) \times \frac{1}{k} \qquad (6)$$

The fuzzy system output is used for the quantification of the heuristic matrix. Selecting the precise fuzzy rules helps the algorithm to traverse the cloud data centers precisely.

Stage 1: Defining the variables and linguistic words (input information fuzzification): This method begins by fuzzification of the information related to the QoS of each cloud data center. This information includes parameters of positive and negative service quality. Indeed, these values are applied to this stage of the Fuzzy system as inputs. Information on the QoS is constructed based on Fuzzy Inference System (FIS). The Fuzzy function is determined using input parameters. FIS generates a replica value (RV).

Stage 2: Defining the fuzzy rules: Fuzzy rules based on different input parameters states are written at this stage. FIS generates an output value based on input different states by the FIS. The fuzzy rules are written using nested If-then. The Mamdani inference algorithm is one of the known algorithms for fuzzy inference. These systems can be used extensively in decision support systems because of their visual and interpretative nature. They also have a high power of expression and can be implemented in both forms of Multiple Input Multiple Output (MIMO) and Multiple Input Single Output (MISO). Mamdany controller is used for ease. Input and output parameters are expressed using linguistic parameters. The triangular membership function is used for input variables' linguistic values of Very-Low (VL), Low (L), Medium (M), High (H), and Very-High (VH). Output variables have five triangular membership functions for the VL to VH values, and the fuzzy system uses 25 rules shown in Fig. 2. The FIS input parameters' membership functions are presented in Fig. 3 and 4. As mentioned in the previous section, some QoS parameters are positive, and some are negative. In other words, the minimizer and the maximizer parameters can be expressed as positive and negative. Because the lower values for the positive parameters (close to zero) result in the optimal output value, and higher values for the negative parameters (close to one) lead to the optimal output value. Hence, we should find a trade-off value between these values to select the best answer that is performed using the Fuzzy rules set. Thus, we define PQ and NQ as positive and negative parameters, respectively, that are obtained by the sum of the normalized values of the quality parameters.

Fuzzy rules
1. If (PQ is VL) and (NQ is VH) then (Fitness is VH)
2. If (PQ is VL) and (NQ is H) then (Fitness is H)
3. If (PQ is VL) and (NQ is M) then (Fitness is M)
4. If (PQ is VL) and (NQ is L) then (Fitness is L)
5. If (PQ is VL) and (NQ is VL) then (Fitness is VL)
6. If (PQ is L) and (NQ is VH) then (Fitness is VH)
7. If (PQ is L) and (NQ is H) then (Fitness is H)
8. If (PQ is L) and (NQ is M) then (Fitness is M)
9. If (PQ is L) and (NQ is L) then (Fitness is L)
10. If (PQ is L) and (NQ is VL) then (Fitness is VL)
11. If (PQ is M) and (NQ is VH) then (Fitness is M)
12. If (PQ is M) and (NQ is H) then (Fitness is M)
13. If (PQ is M) and (NQ is M) then (Fitness is L)
14. If (PQ is M) and (NQ is L) then (Fitness is VL)
15. If (PQ is M) and (NQ is VL) then (Fitness is VL)
16. If (PQ is H) and (NQ is VH) then (Fitness is L)
17. If (PQ is H) and (NQ is H) then (Fitness is VL)
18. If (PQ is H) and (NQ is M) then (Fitness is VL)
19. If (PQ is H) and (NQ is L) then (Fitness is VL)
20. If (PQ is H) and (NQ is VL) then (Fitness is VL)
21. If (PQ is VH) and (NQ is L) then (Fitness is VL)
22. If (PQ is VH) and (NQ is H) then (Fitness is VL)
23. If (PQ is VH) and (NQ is M) then (Fitness is VL)
24. If (PQ is VH) and (NQ is L) then (Fitness is VL)
25. If (PQ is VH) and (NQ is VL) then (Fitness is VL)

Fig. 2. The proposed fuzzy rules.

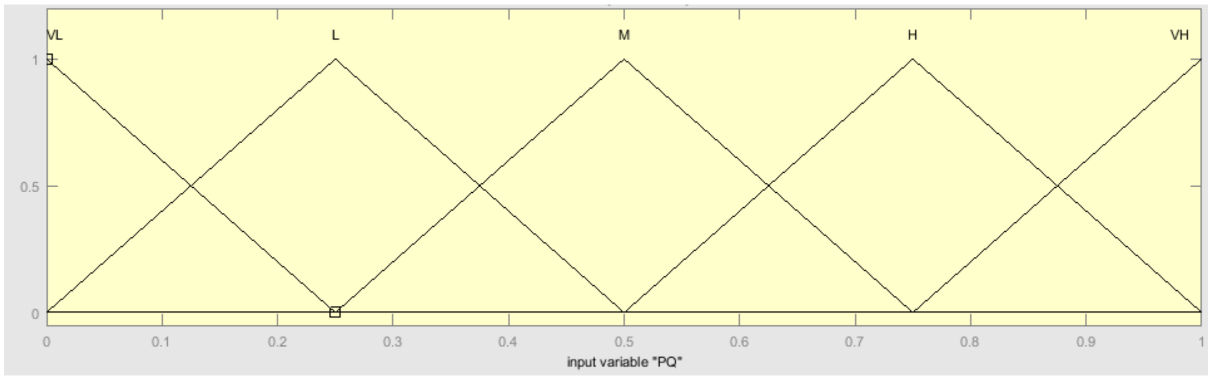

Fig. 3. Membership functions of the input variable (positive QoS parameter).

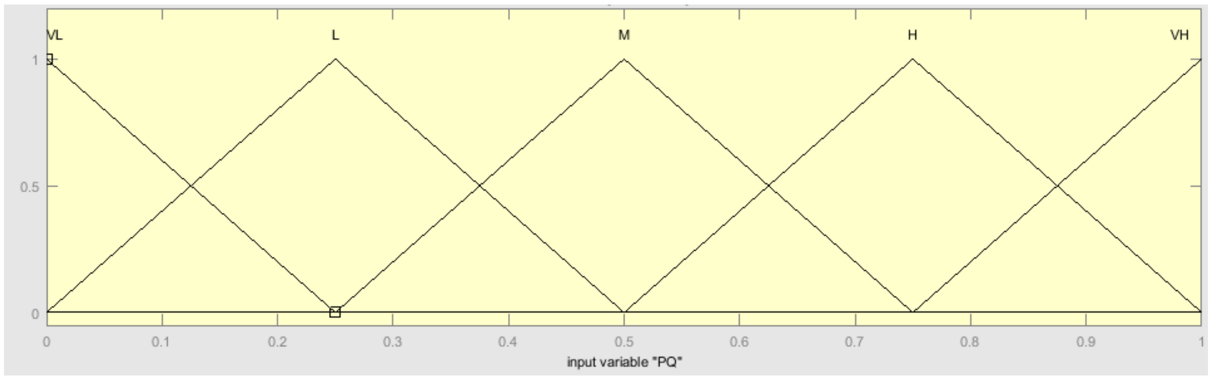

Fig. 4. Membership functions of the input variable (negative QoS parameter).

Stage 3: Defuzzification and analysis of the outputs: Defuzzification of the fuzzy values is performed in this stage after producing multiple fuzzy values according to the fuzzy rules and the input values. In other words, an interval in the fitting function is defined for and mapped to each of the fuzzy outputs in this stage. Fig. 5 illustrates the membership function of the fuzzy system's output variables.

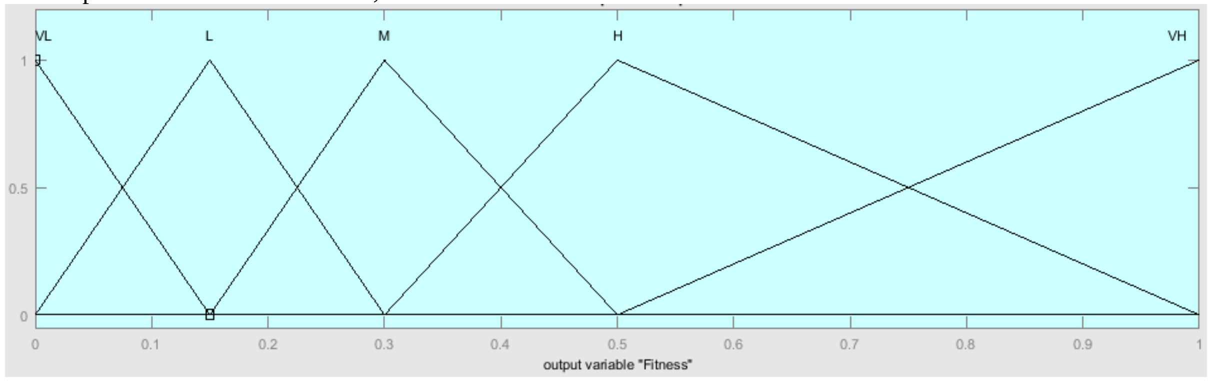

Fig. 5. Membership functions of the output variable

IV. Experimental Results

Based on the experimental setup, results demonstrate a clear improvement strategy for data elimination and data ambiguity. The significance of the system is demonstrated by conducting experiments using the MATLAB simulator.

First experiment: In this experiment we used the [11] data for evaluating the performance of the proposed method. Based on this experiment result, with increasing replication numbers, replication costs also increase. The value of the main and super data centers and their respective probabilities of file availability will be maintained in accordance with the probability of file availability and number of replicas in each data center. In each data center, the value of the super and main data centers and the corresponding probability of file availability will be determined based on the probability of file availability and the number of replicas. According to the Fig. 6-8, when the proportion parameters are set to the same value, the proposed method performs better than the alternative methods. Also, the proposed method demonstrates the medium level of cost efficiency that has been incorporated into the file system. Experimental results demonstrate that our proposed algorithm is capable of dynamically adapting to the preferences of the user.

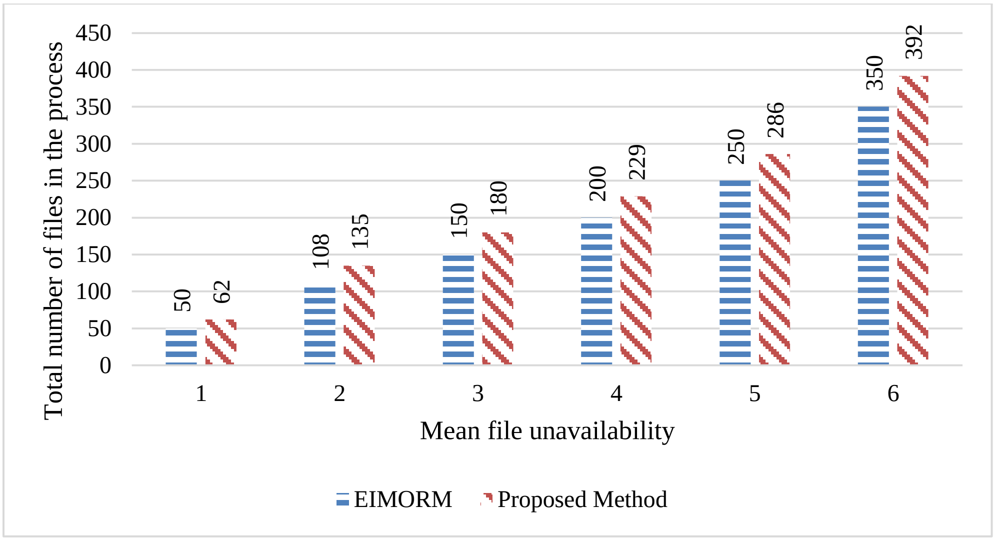

Fig. 6. Mean file unavailability.

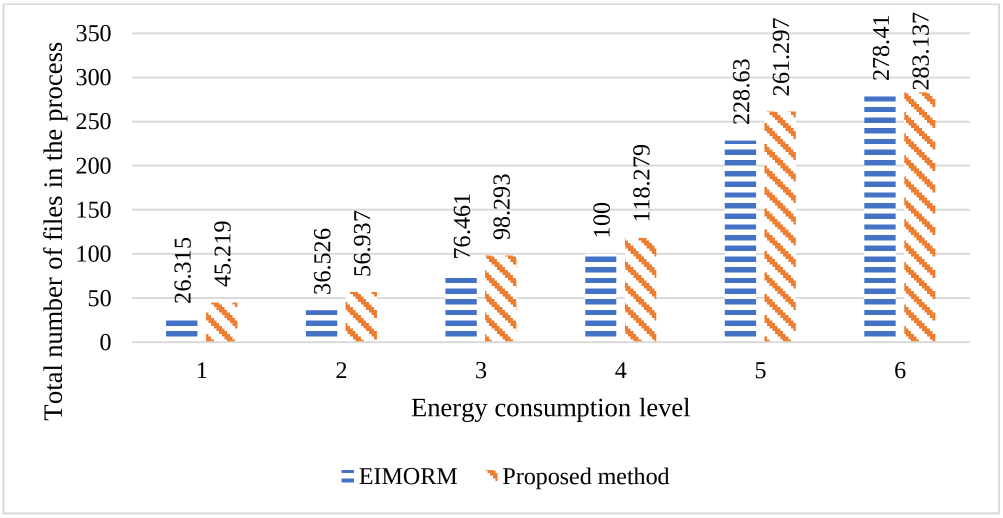

Fig. 7. Energy consumption level.

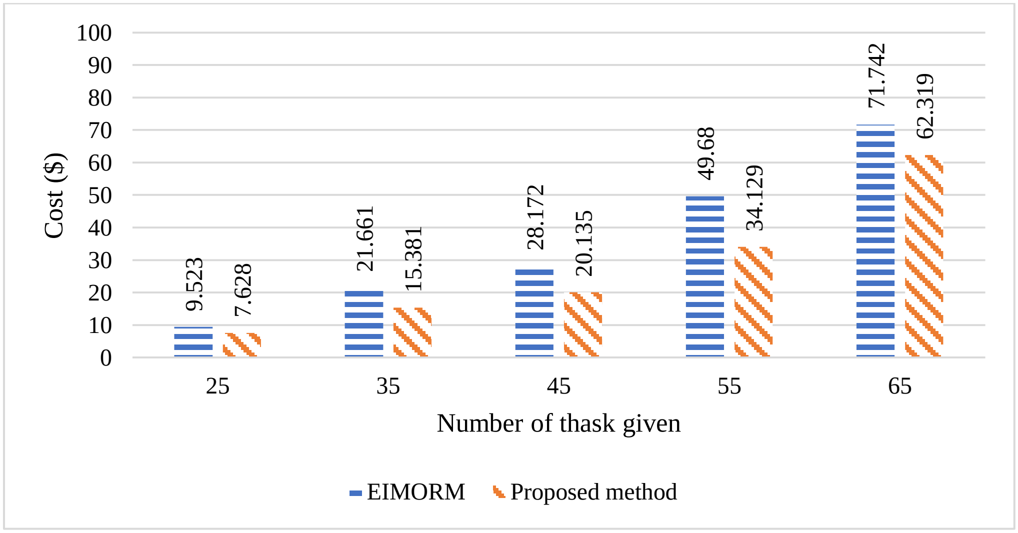

Fig. 8. Number of tasks given.

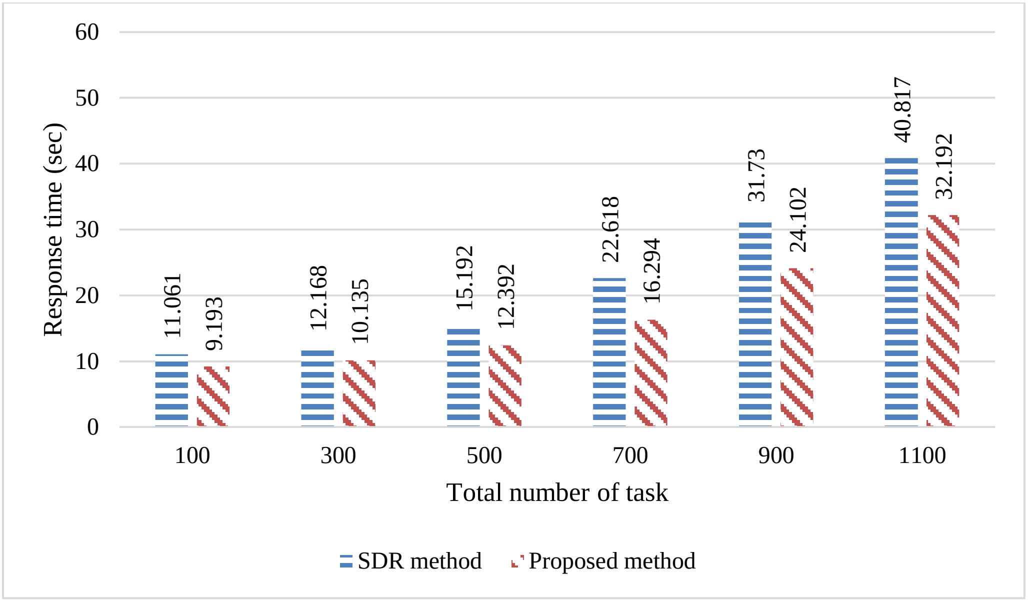

Fig. 9. Average response time with different number of tasks with Zipf distribution.

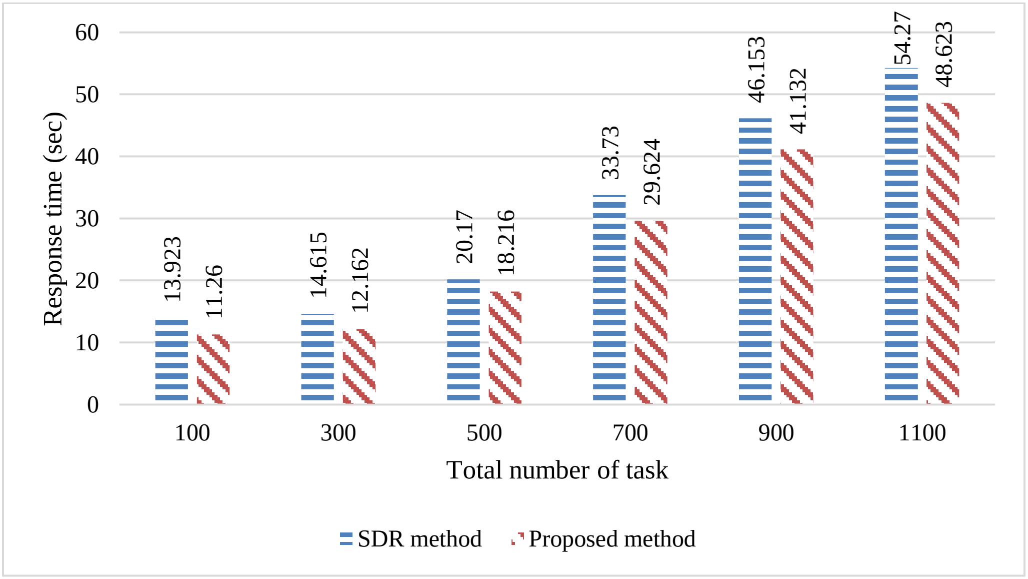

Fig. 10. Average response time with different number of tasks with uniform distribution.

Second experiment: In this experiment, we used the [30] data's and evaluated the proposed method in terms of average response time. Fig. 9 and 10 compare the average response time of the replication strategies for the uniform distribution and Zipf distribution. Our proposed method exhibited notably improved average response times compared to other strategies, indicating its effectiveness in reducing access latency for users.

The experimental results highlight the advantages of our proposed method over existing strategies for data replication in cloud computing. Our approach showcases superior performance and effectiveness by considering cost efficiency, response time, and dynamic adaptability. The comparisons with relevant literature reveal that our model provides a better solution to the data replication problem in cloud computing environments. The MATLAB-based simulations validate the feasibility and practicality of our proposed method. It is essential to acknowledge some limitations of our work. While our experiments demonstrate significant improvements, further investigations could explore the scalability of the proposed method for more extensive datasets and complex cloud environments. Additionally, incorporating real-world case studies and validating the results in a practical cloud computing environment would enhance the applicability of our approach.

V. Conclusion

Data replication constitutes one of the most important aspects of cloud computing, since it provides fast access to data, high availability, and data reliability. By maintaining all replicas available, the replicas can increase the probability of a system task being completed successfully if the requests and replicas are distributed reasonably. However, placing replicas in a highly scalable, virtualized, and large data center poses a greater challenge. To provide load balancing for cloud storage, decrease the response time of applications, and achieve cost-effective availability, this paper proposed a novel dynamic duplicate management in cloud storage systems based on the IACO algorithm and fuzzy logic system. According to the simulation results, our proposed method exhibits significant advantages over existing replication techniques. It accomplishes a remarkable 15% decrease in energy consumption, a 12% cost reduction, and an impressive 20% enhancement in response time. While our research has demonstrated substantial improvements, there remain opportunities for future investigations. Scalability testing under more extensive datasets and complex cloud environments will provide insights into the model's performance under varying scales. Additionally, incorporating dynamic workload management strategies and exploring hybrid optimization techniques could further enhance the algorithm's adaptability and convergence speed.

REFERENCES

[1] B. Pourghebleh, A. A. Anvigh, A. R. Ramtin, and B. Mohammadi, "The importance of nature-inspired meta-heuristic algorithms for solving virtual machine consolidation problem in cloud environments," Cluster Computing, pp. 1-24, 2021.

[2] V. Hayyolalam, B. Pourghebleh, M. R. Chehrehzad, and A. A. Pourhaji Kazem, "Single-objective service composition methods in cloud manufacturing systems: Recent techniques, classification, and future trends," Concurrency and Computation: Practice and Experience, vol. 34, no. 5, p. e6698, 2022.

[3] 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.

[4] X. Liu and J. Liu, "A truthful online mechanism for virtual machine provisioning and allocation in clouds," Cluster Computing, vol. 25, no. 2, pp. 1095-1109, 2022.

[5] Y. Xu and K. Abnoosian, "A new metaheuristic-based method for solving the virtual machines migration problem in the green cloud computing," Concurrency and Computation: Practice and Experience, vol. 34, no. 3, p. e6579, 2022.

[6] S. Habib, S. Aghakhani, M. G. Nejati, M. Azimian, Y. Jia, and E. M. Ahmed, "Energy management of an intelligent parking lot equipped with hydrogen storage systems and renewable energy sources using the stochastic p-robust optimization approach," Energy, p. 127844, 2023.

[7] B. Pourghebleh and N. J. Navimipour, "Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research," Journal of Network and Computer Applications, vol. 97, pp. 23-34, 2017.

[8] A. Peivandizadeh and B. Molavi, "Compatible authentication and key agreement protocol for low power and lossy network in IoT environment," Available at SSRN 4194715, 2022.

[9] M. Najarian, Z. Sarmast, S. Ghasemi, and S. Sarmadi, "Evolutionary vertical size reduction: A novel approach for big data computing," International Journal of Mathematics and its Applications, vol. 6, no. 3, pp. 215-225, 2018.

[10] M. Sarbaz, M. Manthouri, and I. Zamani, "Rough neural network and adaptive feedback linearization control based on Lyapunov function," in 2021 7th International Conference on Control, Instrumentation and Automation (ICCIA), 2021: IEEE, pp. 1-5.

[11] M. Bagheri et al., "Data conditioning and forecasting methodology using machine learning on production data for a well pad," in Offshore Technology Conference, 2020: OTC, p. D031S037R002.

[12] R. Soleimani and E. Lobaton, "Enhancing Inference on Physiological and Kinematic Periodic Signals via Phase-Based Interpretability and Multi-Task Learning," Information, vol. 13, no. 7, p. 326, 2022.

[13] S. R. Abdul Samad et al., "Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection," Electronics, vol. 12, no. 7, p. 1642, 2023.

[14] W. Anupong et al., "Deep learning algorithms were used to generate photovoltaic renewable energy in saline water analysis via an oxidation process," Water Reuse, vol. 13, no. 1, pp. 68-81, 2023.

[15] M. Javidan, M. Yazdchi, Z. Baharlouei, and A. Mahnam, "Feature and channel selection for designing a regression-based continuous-variable emotion recognition system with two EEG channels," Biomedical Signal Processing and Control, vol. 70, p. 102979, 2021.

[16] S. Yumusak, S. Layazali, K. Oztoprak, and R. Hassanpour, "Low-diameter topic-based pub/sub overlay network construction with minimum–maximum node degree," PeerJ Computer Science, vol. 7, p. e538, 2021.

[17] C. Han and X. Fu, "Challenge and Opportunity: Deep Learning-Based Stock Price Prediction by Using Bi-Directional LSTM Model," Frontiers in Business, Economics and Management, vol. 8, no. 2, pp. 51-54, 2023.

[18] H. Kashgarani and L. Kotthoff, "Is algorithm selection worth it? Comparing selecting single algorithms and parallel execution," in AAAI Workshop on Meta-Learning and MetaDL Challenge, 2021: PMLR, pp. 58-64.

[19] S. Aghakhani, A. Larijani, F. Sadeghi, D. Martín, and A. A. Shahrakht, "A Novel Hybrid Artificial Bee Colony-Based Deep Convolutional Neural Network to Improve the Detection Performance of Backscatter Communication Systems," Electronics, vol. 12, no. 10, p. 2263, 2023.

[20] S. Mahmoudinazlou and C. Kwon, "A Hybrid Genetic Algorithm for the min-max Multiple Traveling Salesman Problem," arXiv preprint arXiv:2307.07120, 2023.

[21] S. Mahmoudinazlou and C. Kwon, "A Hybrid Genetic Algorithm with Type-Aware Chromosomes for Traveling Salesman Problems with Drone," arXiv preprint arXiv:2303.00614, 2023.

[22] M. Shahin et al., "Cluster-based association rule mining for an intersection accident dataset," in 2021 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube), 2021: IEEE, pp. 1-6.

[23] V. Ashrafimoghari and J. W. Suchow, "A game-theoretic model of the consumer behavior under pay-what-you-want pricing strategy," arXiv preprint arXiv:2207.08923, 2022.

[24] N. Mansouri, M. M. Javidi, and B. M. H. Zade, "Hierarchical data replication strategy to improve performance in cloud computing," Frontiers of Computer Science, vol. 15, pp. 1-17, 2021.

[25] Y. Ebadi and N. Jafari Navimipour, "An energy-aware method for data replication in the cloud environments using a tabu search and particle swarm optimization algorithm," Concurrency and Computation: Practice and Experience, vol. 31, no. 1, p. e4757, 2019.

[26] R. Salem, M. A. Salam, H. Abdelkader, and A. A. Mohamed, "An artificial bee colony algorithm for data replication optimization in cloud environments," IEEE Access, vol. 8, pp. 51841-51852, 2019.

[27] D. Rambabu and A. Govardhan, "Optimization assisted frequent pattern mining for data replication in cloud: Combining sealion and grey wolf algorithm," Advances in Engineering Software, p. 103401, 2023.

[28] A. Khelifa, R. Mokadem, T. Hamrouni, and F. B. Charrada, "Data correlation and fuzzy inference system-based data replication in federated cloud systems," Simulation Modelling Practice and Theory, vol. 115, p. 102428, 2022.

[29] E. B. Edwin, P. Umamaheswari, and M. R. Thanka, "An efficient and improved multi-objective optimized replication management with dynamic and cost aware strategies in cloud computing data center," Cluster Computing, vol. 22, pp. 11119-11128, 2019.

[30] N. Mansouri, M. M. Javidi, and B. Mohammad Hasani Zade, "A CSO-based approach for secure data replication in cloud computing environment," The Journal of Supercomputing, vol. 77, pp. 5882-5933, 2021.