



# A novel hybrid arithmetic optimization algorithm and salp swarm algorithm for data placement in cloud computing

Ahmed Awad Mohamed<sup>1</sup> · Ashraf D. Abdellatif<sup>2</sup> · Alhanouf Alburaikan<sup>3</sup> · Hamiden Abd El-Wahed Khalifa<sup>3,4</sup> · Mohamed Abd Elaziz<sup>5,6</sup> · Laith Abualigah<sup>7,8,9,10,11,12</sup> · Ahmed M. AbdelMouty<sup>13,14</sup>

Accepted: 28 December 2022 / Published online: 18 January 2023

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

## Abstract

In recent years, the Internet of Things (IoT) has led to the spread of cloud computing devices in all commercial, industrial and agricultural sectors. The use of cloud computing environment services is increasing exponentially with all technology applications based on IoT. Fog computing has led to addressing issues in cloud computing environments. Fog computing reduces load balancing, processing, bandwidth, and storage as data file replication from the cloud to the network closest to sensors in different geographic locations. There are three critical issues:—what data should be replicated?—when should the data be replicated? and—where the new replicas should be placed? These three main open questions must be tackled for data replication in the cloud environments. This strategy, the identification and mode of the data replication problem are designed as a multi-objective optimization with modern meta-heuristic optimization method. Therefore, a new hybrid method using Arithmetic Optimization Algorithm (AOA) and the salp swarm algorithm (SSA) is proposed in this paper. Firstly, a new hybrid metaheuristic method, using the Arithmetic Optimization Algorithm (AOA) and the salp swarm algorithm (SSA), is proposed to handle the problem of selection and placement data replication in fog computing. Secondly, the Floyd algorithm is used to strategy the least cost path, distance, and data transmission in different geographic locations. The performance of the designed AOASSA strategy to tackling the data replication problem is evaluated using different datasets of different sizes. To validate the AOASSA strategy, a set of experiments was carried out to validate the proposed strategy AOASSA. Experiment results show the superiority of AOASSA over its competitors in terms of performance measures, such as least cost path, distance, and bandwidth.

**Keywords** SBER · Data replication · SSA · AOA · Fog computing · IoT · Multi-objective optimization

## 1 Introduction

With the advancement of Internet of Things applications in all fields, the big data volumes increased, and the use of many applications in cloud computing environments (Nguyen et al. 2019; Ghasempour 2019; Fu et al. 2018). At the same time, with the increase in the amount of data and information from users via cloud computing, sensors increase exponentially, and data is stored in different geographic locations on fog computing (Lin et al. 2016; Liu et al. 2016; Abualigah and Diabat 2020). Cloud computing is one of the best solutions for placing and recalling data in different geographic locations and reducing user waiting time. Cloud computing has developed its resources

to enable users to access various services such as servers, storage, sensors, etc. (Wang et al. 2018; Yang et al. 2020). With the development of cloud computing with the Internet of things, huge amounts of different data are generated, which require processing and storage across different geographical locations. Cloud computing depends on load balance, bandwidth, and network performance in waiting time, the requirements of users based on the Internet of Things. Response time for users due to delays across nodes in cloud computing requires less bandwidth (John and Mirnalinee 2020; Pallewatta et al. 2022; Wang and Zhang 2020; Taghizadeh et al. 2021; Torabi et al. 2022; Jin et al. 2022).

Fog computing extends cloud services to the IoT network's edge, offering numerous benefits such as high performance, reduced response time, bandwidth, and load

Extended author information available on the last page of the article

balancing in fog computing (Liu et al. 2022; Li et al. 2022a; Yousif et al. 2022). Fog computing supports the Internet of Things and provides sensors in cloud environments. It also provides users with services from virtual machines and scalable storage spaces (Peake et al. 2022; Haris and Zubair 2021; Khemili et al. 2022). Cloud computing infrastructure is one model that manages cloud resources efficiently and effectively. It enables users to use the available resources as per usage and pay for usage. Cloud computing is widely used for big data processing and complex problems in different environments (Maheshwari et al. 2012; Long et al. 2014; Boru et al. 2015). Cloud computing that supports the Internet of Things offers a promising way to identify and establish replication across nodes in different geographic locations. Data replication is one of the most widely used methods for data availability, reliability, and distribution across nodes. It also works to process data and files and access file replication at the lowest cost and lowest path (Ebadi and Navimipour 2018; Chuang and Hsiang 2022). Data replication reduces data transfer, accessing data and files with less time and less distance, and reducing user waiting time (Failed 2013). Replication techniques consist of static and dynamic methods of defining, establishing, and placing replication across nodes. Static replication is set and established during system configuration and is not affected by momentary changes such as changing, deleting, or storing data within the database. At the same time, dynamic replication patterns are the access, storage, and deletion of data and its availability (Grami 2022). There are three critical issues: (1) what data should be replicated? (2) when should the data be replicated? and (3) where the new replicas should be placed? These three main open questions must be tackled for data replication in the cloud environments (Salem et al. 2020; Awad et al. 2021a, b).

The major contributions of the article are as follows:

- Improving a new hybrid swarm intelligent algorithm method AOASSA is the strategy for dynamic data replication of IoT applications tasks in fog computing. The integration of AOA and SSA with MOO to solve the least cost path in fog computing.
- To the best of our knowledge, there are introducing a few papers that handle data transmission and the least cost path for optimal placement file replication with MOO.
- To the best of our knowledge, our work is the first attempt to study a new hybrid AOA, SSA (AOASSA), solved to obtain low bandwidth, data transmission, and least cost path.
- The application of global optimization utilizing AOASSA gives better results when experimental results are compared with MOE, and MORM.
- AOASSA minimizes the computational complexity and it also works efficiently for transmission data, distance and placement data replication problems.
- The experimental results show the superiority of the AOASSA algorithm performance over other algorithms, such as cost, time, bandwidth, data transmission, and least cost path.

The rest of this paper is organized as follows. Section 2 presents related work. Section 3 presents the proposed architecture. Section 4 presents the proposed strategy. Section 5 presents the experimental results—finally, Sect. 6 presents the conclusion and future work.

## 2 Related work

Many related studies have researched data replication strategies in the cloud, as follows:

- Sarwar et al. (2022) proposed two cross-node replication privacy schemes for data protection, authentication, and reliability. Create a schema for cross-node select and placement data replication while maintaining privacy and confidentiality on fog computing. The proposed algorithm outperformed other algorithms regarding memory, cost, confidentiality, and privacy.
- Chen et al. (2021) proposed the first decentralized system to demonstrate data retrieval and repeatability—BOSSA, which is compatible with all parties on blockchain platforms. BOSSA also incorporates privacy-enhancing technologies to prevent decentralized peers (including blockchain nodes) from inferring private information about external data. The security analysis is presented in the context of integrity, privacy, and reliability, and we implement a BOSSA-based prototype to leverage smart nodes on the Ethereum blockchain. Our extensive beta reviews demonstrate the practicality of our proposal.
- Li et al. (2022b) proposed Suggested a Lagrange method for relaxation. This method considers the load balancing, storage, data dependency, data transmission, time, cost, and bandwidth to obtain minimum data transmission time between nodes. Aimed at the cloudlet and fault-tolerant task scheduling strategy is proposed. The strategy optimization of the task scheduling techniques by considering the task of time, cost, and energy method. The experiments proved the performance of the proposed strategy in transferring data through cloud computing and choosing the optimal location according to the proposed algorithm.
- Shiet et al. (2022) developed a new approach named Multi-Cloud Application deployment (MCApp). MCApp merges iterative mixed integer linear

programming with domain-tailored large node search to optimize data replication deployment and user requests. The experiments proved the performance of the proposed strategy using the real and datasets demonstrate that MCAApp significantly outperforms other algorithms.

- Majed et al. (2022) presented a hybrid strategy for peer-to-peer data replication in cloud computing environments. It chose the most suitable and optimal nodes in the network at a low cost. It also selects users' most common and accessible data files and puts them in the most appropriate placement. Experimental results showed improved network performance and reduced user waiting.
- Li et al. (2022c) Suggested an algorithm based on the Lagrangian relaxation method for optimum data replication across nodes in cloud computing. Consider balancing loads, bandwidth, and transmission time. Also, use the Floyd algorithm to reduce cost and bandwidth. The results showed the superiority of the proposed algorithm over other algorithms.
- Khelifa et al. (2022) Suggested a dynamic and periodic data replication strategy in cloud computing. The proposed strategy aims to reduce the time for users' requirements, achieve load balance, and reduce waiting for time and speed access to data. It also reduces the time to send data and transfer it through cloud computing. Also, a fuzzy logic algorithm of select and placement data replication across nodes was used. The proposed algorithm proved to be superior to other algorithms.
- Mohammadi et al. (2022) Suggested an algorithm for selecting and setting data replication across nodes in cloud computing. Use the hybrid fuzzy logic and ant colony optimization algorithm to reduce users' waiting time and discover the most suitable and optimal nodes for placement data replication. The proposed algorithm outperformed the other algorithms.

### 3 Suggested system and discussion

#### 3.1 Proposed system and structure

This section describes the proposed selection and placement of data replication across nodes in fog computing. The critical component of the proposed strategy is the Fog Broker, which is set in the fog nodes layer. Fog Broker consists of three stages: Task Manager, Resource Monitoring Service, and Task Scheduler. The fog computing system relies on our dynamic data replication strategy based on IoT in cloud computing. Select and placement

data replication cross node in cloud computing requires a set of configurations to move data across fog computing. We assume that our proposed strategy comprises a certain number of Fog nodes such as (DCs) data centers and IoT services. We organized the proposed strategy from different geographic locations to select and place data replication across nodes in fog computing. Services can be distributed on any DCs, Fog nodes, or IoT sensors. Use the AOA algorithm with the SSA algorithm to transfer data across nodes with the lowest path and lowest cost. MOO with a floyed algorithm was also used to reduce bandwidth across the network and balance in fog nodes. The system in Fig. 1 consists of a different set of geographically distributed node (G) and the structure is composed of many different locations.

#### 3.2 Arithmetic optimization algorithm (AOA)

##### 3.2.1 Initialization phase

According to Matrix, AOA, the optimization criteria apply to several produced random solutions (X). The best result obtained in each iteration is saved as being close to the current optimal result (Abualigah et al. 2021; Abdollahzadeh et al. 2021; Abdollahzadeh and Gharehchopogh 2022; Mahajan and Pandit 2021; Mahajan et al. 2022a, b).

$$X_i = \begin{bmatrix} x_1^1 & x_1^1 & \dots & x_{1,D}^1 \\ x_1^2 & x_2^2 & \dots & x_D^2 \\ \vdots & \ddots & \ddots & \vdots \\ x_1^N & x_D^N & \dots & x_D^N \end{bmatrix} \tag{1}$$

Based on the Math Optimizer Accelerated (MOA) function produced by utilizing Eq. (2), the search phase (exploration or exploitation) should be chosen for each iteration.

$$MOA(C\_Iter) = Min + C\_Iter \times \left( \frac{Max - Min}{M\_Iter} \right) \tag{2}$$

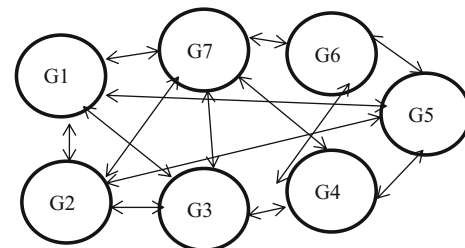


Fig. 1 Proposed strategy for data replication in fog computing

where  $MOA(C\_Iter)$  is the value of MOA at the current iteration.  $C\_Iter$  is the current iteration, which is between  $[1 M\_Iter]$ .  $M\_Iter$  is the maximum number of iterations.  $Min$  and  $Max$  are the minimum and maximum of the accelerated function values.

### 3.2.2 Exploration phase

The simplest rule that can simulation the behaviour of mathematical operators was employed. The following location updating calculations for the exploratory part:

$$x_{i,j}(C\_Iter + 1) = \begin{cases} \text{best}(x_j) \div (MOP + \varepsilon) \times ((UB_j - LB_j) \times \mu + LB_j), & r2 < 0.5 \\ \text{best}(x_j) \times MOP \times ((UB_j - LB_j) \times \mu + LB_j), & \text{otherwise} \end{cases} \tag{3}$$

where  $x_i(C\_Iter + 1)$  indicates the  $i$ th solution location in the next repetition.  $x_{i,j}(C\_Iter)$  indicates the  $j$ th location of the  $i$ th solution.  $\text{best}(x_j)$  is the  $j$ th in the best-obtained location in the  $i$ th solution.  $\mu$  is an adjusting parameter used to select the search operation, which is set equal to 0.5.

$$MOP(C\_Iter) = 1 - \frac{C\_Iter^{\frac{1}{\alpha}}}{M\_Iter^{\frac{1}{\alpha}}} \tag{4}$$

where  $MOP$  is a coefficient.  $MOP(C\_Iter)$  indicates the value of the  $MOP$  at the  $i$ th iteration.  $C\_Iter$  indicates the current iteration.  $M\_Iter$  indicates the maximum number of iterations.  $\alpha$  is an adjusting parameter used to specify the exploitation accuracy. Throughout iterations, which is set equal to 5.

**Table 1** parameters data sets of the system

Cloud Entity	Parameter	Ranges
Nodes	Number of datacenter	[10, 100]
User	Number of users	[10, 1000]
Regions	Number of regions	[5, 25]
Geographical	Geographical capacity	[10, 128]
Bandwidth	Bandwidth	[2 Mbps, 128 Mbps]
Data sets	Data set size	[2G, 128G]
Data file	Number of file	[10, 5000]
Cost of file	Cost of data replica	[100, 6000]
Storage nodes	Storage capacity	[10, 128]
Transfer rate	Maximum transfer rate	[8, 256 MB/s]
Host	Number of host	[10, 200]
Processor	Processing elements	[12, 128]
MIPS	MIPS	[500, 2000]
Memory RAM	RAM	[2, 32G]
Virtual machine	Number of VM	[100, 800]
Processor	Processing elements	[8, 128]
MIPS	MIPS	[200, 2000]
Memory RAM	RAM	[2, 32G]
Cloudlet	Number of cloudlet	[1000, 5000]
Length of task	Length of task	[1000, 100000 MI]

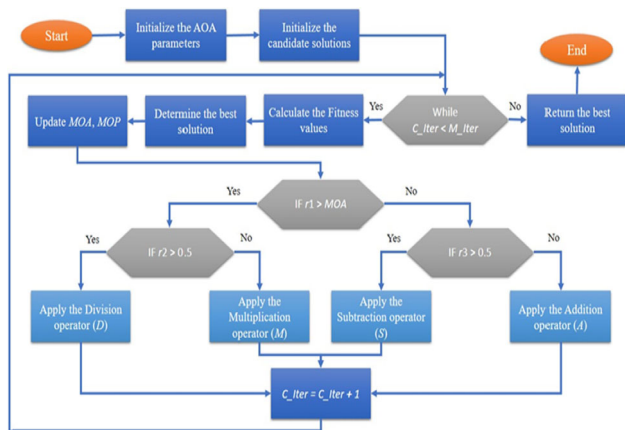
### 3.2.3 Exploitation phase

The MOA function value conditions this part of paper for the condition of  $r1$  is not greater than the current  $MOA(C\_Iter)$  value.

$$x_{i,j}(C\_Iter + 1) = \begin{cases} \text{best}(x_j) - MOP \times ((UB_j - LB_j) \times \mu + LB_j), & r3 < 0.5 \\ \text{best}(x_j) + MOP \times ((UB_j - LB_j) \times \mu + LB_j), & \text{otherwise} \end{cases} \tag{5}$$

### 3.3 Pseudo-code of AOA

The pseudo-code of the proposed AOA is reported in Algorithm 1 (Fig. 2).



**Fig. 2** Flowchart of the original AOA

## Algorithm 1 Pseudo-code of AOA

Initialize the Arithmetic Optimization Algorithm parameters  $N$ ,  $\alpha$ ,  $\mu$ , etc.  
Initialize the solutions' positions randomly. (Solutions:  $i=1 \dots N$ ).

```

WHILE (C_Iter < M_Iter)
  Calculate the Fitness Function for the given solutions.
  Find the best solution so far.
  Update the MOA parameter using Equation (2).
  Update the MOP parameter using Equation (4).
  FOR ( $i=1$  to Solutions)
    FOR ( $j=1$  to Positions)
      Generate random values between [0, 1] ( $r1$ ,  $r2$ , and  $r3$ )
      IF ( $r1 > MOA$ )
        IF ( $r2 > 0.5$ )           Exploration phase
          Update the  $i_{th}$  solutions' positions using the first part in Equation (3).
        ELSE
          Update the  $i_{th}$  solutions' positions using the second part in Equation (3).
        ENDIF
      ELSE
        IF ( $r3 > 0.5$ )           Exploitation phase
          Update the  $i_{th}$  solutions' positions using the first part in Equation (5).
        ELSE
          Update the  $i_{th}$  solutions' positions using the second part in Equation (5).
        ENDIF
      ENDIF
    ENDFOR
  ENDFOR
  C_Iter=C_Iter+1
ENDWHILE
Return the best solution (best_x).

```

## 4 Proposed swarm intelligence for data replication

This part explains the proposed strategy for selecting and placing data replication via a fog node. Based on IoT via fog computing for the proposed strategy, the shortest path, cost, bandwidth, time, cost, and distance were calculated. Use iFogSim to implement the proposed model.

### 4.1 Cost and time of replication

Cost is a primary factor in placement data replication close to the users according to the user's budget. The cost is different from user to user through the proposed strategy cost nodes are different from one fog node to another according to the placement near the users. The Equation can be represented as follows:

$$\text{cost}(DT^j) = \sum_{y=1}^n \text{cost}(dt_z^y) \quad (6)$$

where

$$\text{cost}(dt_z^x) = \sum_{z=1}^m x_z^y (p_z^y + \left(\frac{\text{size}(dt_z^y)}{b_z^y}\right) * tcost) \quad (7)$$

$DT^i$  Cost of data set.  $dt_z^y$  Data replica in region.  $x_z^y$  A binary decision variable  $q \in (1, 2, 3, \dots, l)$ .  $p_z^y$  Price of replica.  $b_z^y$  Bandwidth network between replicas in region.

### 4.2 Shortest paths problem (SPP) between nodes based on the Floyd algorithm

The Floyd algorithm obtains the shortest path between the node in fog computing. Generally, when implementing the Floyd algorithm to obtain the weighted length between the

shortest path among the DCs in fog computing. This paper aims to solve the issue of selecting and placing dynamic data replication across a geographically distributed node to the shortest and optimal path in data transmission and bandwidth (Cheng et al. 2014). The equations can be represented as follows:

$$\text{first weighted adjacency matrix } A = [a_{i,j}]_{m \times m} \quad (8)$$

$a_{i,j}$  is a path from node  $i$  to node  $j$  matrix  $m$ .

The state transition equation is as follows Eq.

$$\text{map } [I, J] := \min \{ \text{map } [I, k] + \text{map } [K, J], \text{map } [I, J] \} \quad (9)$$

Map  $[I, J]$  demonstrste the shortest distance from  $I$  to  $j$ .  $K$  is the breakpoint of exhausting  $I$  and  $j$ .

### 4.3 Popularity degree of the data file

The popularity of a file is determined by frequent access from users, especially in recent use. The file that is very popular with users in recent times is the file that has been identified, replicated, and placed between DCs. The Equation can be represented as:

$$PD_i = an_i * w_i \quad (10)$$

Each file’s replication factor (RF<sub>*i*</sub>) is calculated based on the popularity degree as in Eq. (11).

$$RF_i = \frac{PD_i}{RN_i * FS_i} \quad (11)$$

The dynamic threshold (TH) value is calculated as in Eq. (12).

$$DH = \min \left( (1 - \alpha) * RF_{\text{system}} \max \left( \forall_{k \in [1, 2, \dots, l]} RF_k \right) \right), \alpha \in [0, 1] \quad (12)$$

$PD_i$  popularity degree,  $an_i$  number of access,  $w_i$  time based forgetting factor.  $RF_i$  replica factor,  $RN_i$  number of replica,  $FS_i$  size of data file

#### 4.3.1 System-level availability

SBER is the system’s overall high availability. Users should be able to access all files via tasks for data replication. Due to frequent user access, access to the most popular files. SBER keeps the file accessible and well-liked throughout the entire system. The following is a representation of the Equation:

$$SBER = \frac{\sum_{i=1}^s (an_k x (\sum_{j=1}^{n_k} bs_j) x P(FA_k))}{\sum_{i=1}^s (an_k x (\sum_{j=1}^{n_k} bs_j))} \quad (13)$$

### 4.4 Placement of new replicas

Placement dynamic data replica between nodes for optimal selection of minimum distances. When placement data replication between DCs, it considers the optimal minimum path and low cost for users. It can also be represented as:

$$br_k(dc_i) = \left\lfloor \frac{RF_k(dc_i)}{\sum_{i=1}^s RF_k(dc_i)} x_{br_k(\text{add})} \right\rfloor \quad (14)$$

### 4.5 Salp swarm algorithm

Recently, the SSA (Mirjalili et al. 2017) has been used and inspired by the conduct of a swarm of salps in oceans. Algorithm SSA is inspired by swarm intelligence, simulating the integration between SSA and nature. It consists of two main groups: leaders and followers. First, the leaders do optimal fitness. The rest is followers. Second, the fitness value is calculated for each algorithm’s optimal solutions. The optimal solution found is called a leader. The equations can be represented as follows:

1) *Leader Phase:* The leader location is modernized using the following Equation:

$$X_j^1 = \begin{cases} X_{bj} + c_1 ((ub_j - lb_j) c_2 + l) & \text{if } c_3 > 0.5 \\ X_{bj} - c_1 ((ub_j - lb_j) c_2 + l) & \text{otherwise} \end{cases} \quad (15)$$

$c_1$  decreases through the iterations as.

Where

$$c_1 = 2e^{-\left(\frac{\#}{\#}\right)^2} \quad (16)$$

$X_j^1$  and  $X_{bj}$  represent new placement.  $c_2$  and  $c_3$  random variable from 0 to 1.  $ub_j$  and  $lb_j$  refer the domin of serarch at dimintion  $j$

2) *Followers Phase:* To modernize the followers’ locations,

Newton’s law of motion is used, which defined as

$$X_j^i = \frac{1}{2} g t^2 + \omega_0 t, i \geq 2 \quad (17)$$

So, modernizing the procedure of followers can be formulated as

$$X_j^i = \frac{1}{2} (X_j^i + X_j^{i-1}) \quad (18)$$

$t$  iteration.  $\omega_0 = 0$  and  $g$  velocity and the acceleration.

### 4.6 Pseudo-code of SSA

The pseudo-code of the proposed SSA algorithm is reported in Algorithm 2.



**Algorithm 2** Pseudo-code of SSA algorithm

```

Initialize the population's n=1, 2, 3, 4 ..... m. and the total number of generations (tmax).
Construct the initial set of N solutions X.
while (final condition is not satisfied) do
For each Xi Calculate the fitness of each search and determine Xb.
update c1 in Eq. (16)
  for each i to n do
    if (i==1)
      Update the placement of the leading in Eq. (15)
    else
      Update the placement of the follower in Eq. (18)
    end
  end
  update upper and lower bounds of variables
end
return Xb.
    
```

**4.7 Mean service time (MST)**

MST describes the ability of a system to speed up responses to users. When selecting the more popular files, waiting for users reduces load balancing and minimum bandwidth. The mean service time of file fi can be calculated by:

$$stf_i = \sum_{j=1}^m \left( stf(i,j) * \frac{A(i,j)}{A(i)} \right) \tag{19}$$

The mean service time of the system can be defined as follows:

$$mst = \frac{1}{n} * \sum_{i=1}^n \sum_{j=1}^m \left( \varnothing(i,j) * \frac{s_i}{tp_j} * \frac{A(i,j)}{A(i)} \right) \tag{20}$$

stf(i,j) expected service time of file in data node. A(i,j) access rate of read requests from data node. A(i) mean access rate. si size of file. tpj transfer rate of data node.

**4.8 Computational complexity**

Calculate the time complexity of the proposed strategy AOASSA from tasks for m number of data repetitions. Calculate the number of num\_DCs and AOA with SSA. Suppose N represents the size of the population, D

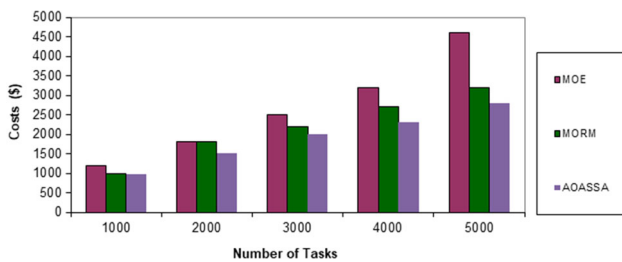


Fig. 3 Cost number of tasks

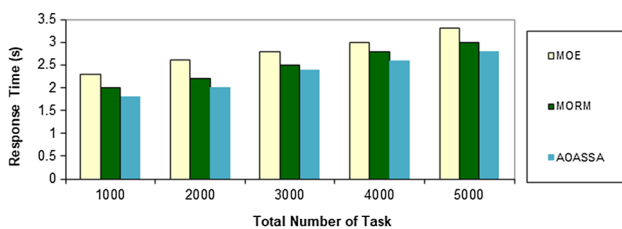


Fig. 4 Mean service of time for tasks

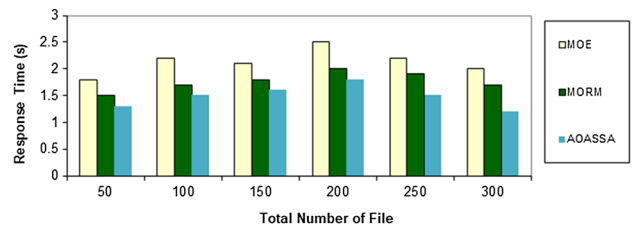


Fig. 5 Mean service of time for file

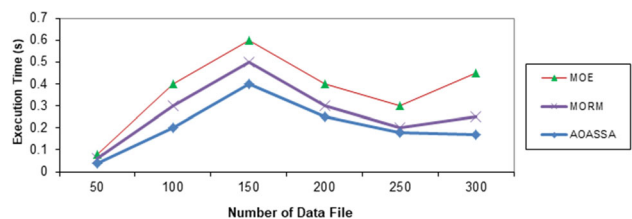
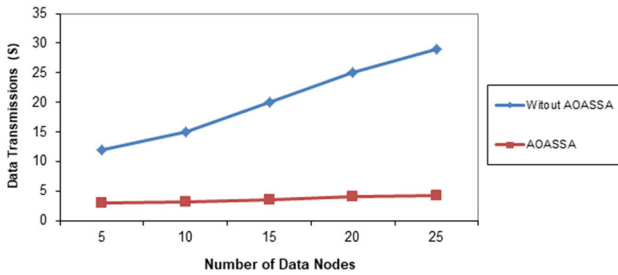
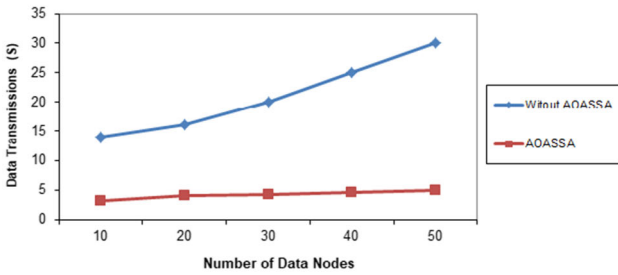


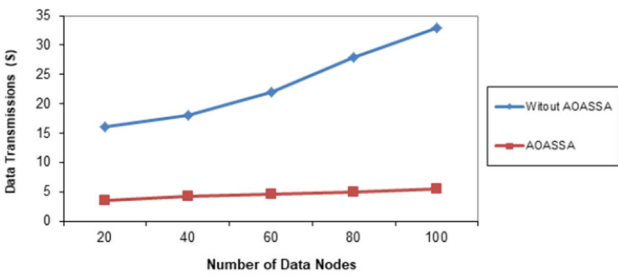
Fig. 6 Execution time for file



(a) data transmission in different nodes = 25 nodes



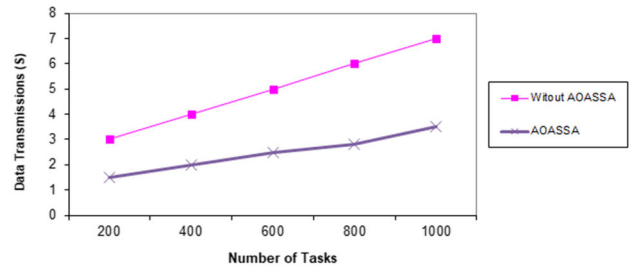
(b) data transmission in different nodes = 50 nodes



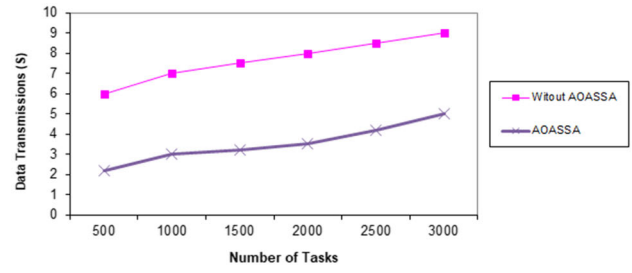
(c) data transmission in different nodes = 100 nodes

Fig. 7 Impact of data replication on that transmission nodes

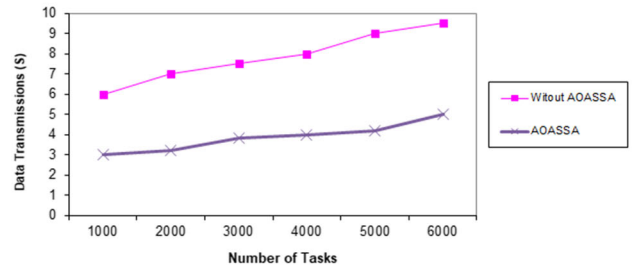
represents the number of objectives,  $T$  represents the number of iterations, and  $CoF$  represents the cost of function. The SSA algorithm has a calculated complexity of  $O(T(D * N + CoF * N))$ . Based on the algorithm phase AOASSA strategy, the time complexity is  $O(N)$ . Hence, the AOASSA total time complexity is  $O(N * T * C)$ , so it is  $O(N)$ .



(a) data transmission in different tasks = 1000 tasks



(b) data transmission in different tasks = 3000 tasks



(c) data transmission in different tasks = 6000 tasks

Fig. 8 Impact of data replication on that transmission tasks

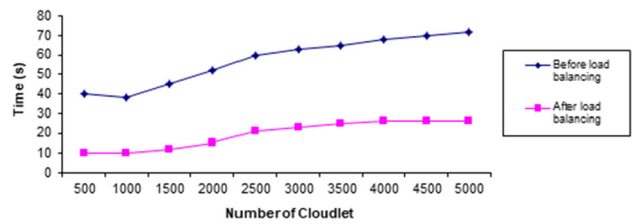


Fig. 9 Degree of imbalance

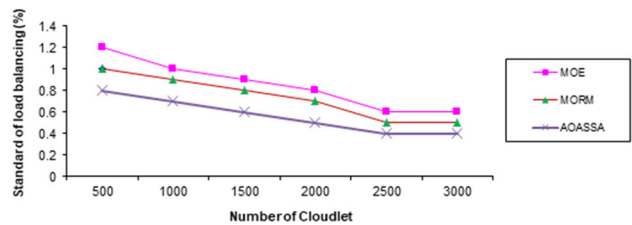


Fig. 10 Standard of load balancing



**Algorithm 3: the proposed algorithm AOASSA**

**Input:** Regions, Datacenters, Data availability, Minimum distance between Regions, cost, time, MSE, SBER, Fog Nodes, popularity data file, max\_iter, population size, number of IOT tasks.

**Output:** select and placement data file replica optimal trade-off between carbon and energy consumption optimal solution.

**Begin**

Initialize no. of IOT tasks  
 Initialize of the population;  
 Initialize of the population using the fitness function  
 Initialize availability and unavailability probabilities  
 Initialize replicas according to costs and time  
 Initialize distance between regions  
 Initialize popularity data file  
 Initialize data replication costs and time  
 Initialize optimal best data replica placement in DC solution  
 Initialize least cost path  
 Initialize SBER  
 Initialize RF  
 Initialize budget  
 Initialize MST

**repeat**

Initialize the Arithmetic Optimization Algorithm parameters  $N$ ,  $\alpha$ ,  $\mu$ , etc.  
 Initialize the solutions' positions randomly. (Solutions:  $i=1 \dots N$ ).

WHILE ( $C\_Iter < M\_Iter$ )

    Calculate the Fitness Function for the given solutions.

    Find the best solution so far.

    Update the MOA parameter using Equation (2).

    Update the MOP parameter using Equation (4).

    FOR ( $i=1$  to Solutions)

        FOR ( $j=1$  to Positions)

            Generate random values between  $[0, 1]$  ( $r1$ ,  $r2$ , and  $r3$ )

            IF ( $r1 > MOA$ )

                IF ( $r2 > 0.5$ )           Exploration phase

                    Update the  $i$ th solutions' positions using the first part in Equation (3).

                ELSE

                    Update the  $i$ th solutions' positions using the second part in Equation (3).

                ENDIF

            ELSE

                IF ( $r3 > 0.5$ )           Exploitation phase

                    Update the  $i$ th solutions' positions using the first part in Equation (5).

                ELSE

                    Update the  $i$ th solutions' positions using the second part in Equation (5).

                ENDIF

            ENDIF

        ENDFOR

    ENDFOR

$C\_Iter = C\_Iter + 1$

ENDWHILE

Return the best solution (best\_x).

Apply carbon in Eq. ()

Apply energy aware in Eq. ()

Apply Salp Swarm Algorithm to EO

$t++$

End while

Calculate the RF

Calculate distance between regions

Calculate SBER

Calculate the cost and time

Calculate MST

**Return** the optimal minimum data replica placement in the region.

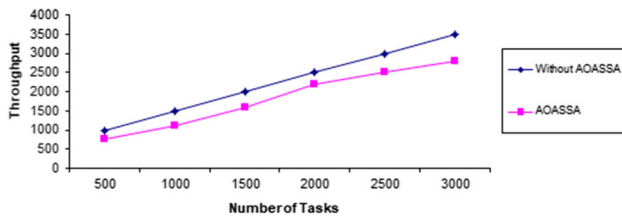


Fig. 11 Throughput

## 5 Experimental evaluation

### 5.1 Configuration details

AOASSA select and placement dynamic data replication between nodes. The proposed strategy has been implemented on iFogSim. In this section, we will discuss the setting, experimental result and configuration of fog computing for the proposed strategy in detail (Table 1).

### 5.2 Results and discussion

#### 5.2.1 Selecting the optimal data file

In Fig. 3, 100 to 5000 cloudlets are created on data replication crosses nodes and access to it. The proposed algorithm proved to be the least cost path than the proposed algorithm in terms of accessing nodes through the least distance of fog cloud.

Figure 4 creates from 1000 to 5000 tasks to reach data replication from the minimum path and low cost. The proposed strategy proved to be speed response to users than other algorithms.

Figure 5 creates from 50 to 300 data replications to placement from the minimum distance and cost. The average service time in select and placement data replication across nodes is less than the algorithms. The proposed strategy proved to be low mean service time to users than other algorithms.

In Fig. 6, create from 50 to 300 data replication to speed up file access across nodes in cloud computing. The proposed algorithm proves faster access to files across nodes than other algorithms. The proposed strategy proved to be faster for data replication than other algorithms.

Figure 7, shows the rate of data file transfer across nodes. As shown in Figures a, b, and c, the number of nodes varies to transfer files by the lowest path. The proposed strategy proved to be superior to other algorithms.

Figure 8 shows the rate of data file transfer across nodes. As shown in Figures a, b, and c, the number of tasks varies

to transfer files by the least cost path. The proposed strategy proved to be superior to other algorithms.

### 5.3 Performance evaluation

#### 5.3.1 Degree of balancing

Figure 9 shows the degree of imbalance over the network nodes to perform several different cloudlets at different times. The proposed strategy minimizes the degree of imbalance to a low level.

In Fig. 10 shows the standard of load balancing over the fog nodes to perform several different tasks at different times. The proposed strategy minimizes the standard of load balancing to a low level.

In Fig. 11 shows the throughput over the tasks nodes to perform several different tasks at different times. The proposed strategy minimizes the throughput to a low level.

## 6 Conclusion and future work

This article proposes a novel hybrid metaheuristic algorithm based on IoT in fog computing for select and placement data replication. AOASSA is a hybrid AOA with SSA to improve data transmission and the least cost path between nodes in fog computing. Our proposed strategy was compared with other algorithms in fog computing. The proposed strategy enhancement the selection and placement of data replication and choosing the least cost path, throughput and standard of load balancing. The results proved the superiority of the AOASSA strategy over other algorithms according to data transmission, load balancing, and distance. In future work, work on improving our strategy through modern algorithms from metaheuristics. In the future, to reduce the least cost path, bandwidth, and cost of data replication. The evaluation of other priority strategies, such as bandwidth, fault tolerance, and enhancement QoS for the new algorithm, will be studied.

**Acknowledgements** The researchers would like to thank the Deanship of Scientific Research, Qassim University for funding the publication of this project.

**Funding** The authors have not disclosed any funding.

**Data availability** Data are available from the authors upon reasonable request.

### Declarations

**Conflict of interest** The authors declare that there is no conflict of interest regarding the publication of this paper.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

**Informed consent** Informed consent was obtained from all individual participants included in the study.

## References

- Abdollahzadeh B, Gharehchopogh F (2022) A multi-objective optimization algorithm for feature selection problems. *Eng Comput* 38(3):1845–1863
- Abdollahzadeh B, Gharehchopogh F, Mirjalili S (2021) Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. *Int J Intell Syst* 36(10):5887–5958
- Abualigah L, Diabat A (2020) A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments. *Cluster Comput* 24(1):205–223
- Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi A (2021) The arithmetic optimization algorithm. *Comput Methods Appl Mech Engrg* 376:113609
- Awad A, Salem R, Abdelkader H, Abdsalam M (2021a) A novel intelligent approach for dynamic data replication in cloud environment. *IEEE Access* 9:40240–40254
- Awad A, Salem R, Abdelkader H, Abdsalam M (2021b) A swarm intelligence-based approach for dynamic data replication in a cloud environment. *Int J Intell Eng Syst*. <https://doi.org/10.22266/ijies2021.0430.24>
- Boru D, Kliazovich D, Granelli F, Bouvry P, Zomaya A (2015) Energy-efficient data replication in cloud computing datacenters. *Cluster Comput* 18(1):385–402
- Chen D, Yuan H, Hu S, Wang Q, Wang C (2021) BOSSA: a decentralized system for proofs of data retrievability and replication. *IEEE Trans Parallel Distrib Syst* 32(4):786–798
- Cheng L, Liu C, Yan B (2014) Improved hierarchical A-star algorithm for optimal parking path planning of the large parking lot. In: 2014 IEEE international conference on information and automation (ICIA)
- Chuang Y, Hsiang C (2022) A popularity-aware and energy-efficient offloading mechanism in fog computing. *J Supercomput* 78(18):19435–19458
- Ebadi Y, Navimipour N (2018) An energy-aware method for data replication in the cloud environments using a Tabu search and particle swarm optimization algorithm. *Concurrency Computat Pract Exper* 31(1):e4757
- Boru D, Kliazovich D, Granelli F, Bouvry P, Zomaya A (2013) Energy-efficient data replication in cloud computing datacenters. *IEEE*
- Fu J, Liu Y, Chao H, Bhargava B, Zhang Z (2018) Secure data storage and searching for industrial IoT by integrating fog computing and cloud computing. *IEEE Trans Ind Inf* 14(10):4519–4528
- Ghasempour A (2019) Internet of things in smart grid: architecture, applications, services, key technologies, and challenges. *Inventions*
- Grami M (2022) An energy aware scheduling of dynamic workflows using big data similarity statistical analysis in cloud computing. *J Supercomput* 78(3):4261–4289
- Haris M, Zubair S (2021) Mantaray modified multi-objective Harris hawk optimization algorithm expedites optimal load balancing in cloud computing. *J King Saud Univ Comput Inf Sci* 34(10):9696–9709
- Jin W, Lim S, Woo S, Park C, Kim D (2022) Decision-making of IoT device operation based on intelligent-task offloading for improving environmental optimization. *Complex Intell Syst*. <https://doi.org/10.1007/s40747-022-00659-z>
- John S, Minalinee T (2020) A novel dynamic data replication strategy to improve access efficiency of cloud storage. *IseB* 18(3):405–426
- Khelifa A, Mokadem R, Hamrouni T, Charrada F (2022) Data correlation and fuzzy inference system-based data replication in federated cloud systems. *Simul Model Pract Theory*. <https://doi.org/10.1016/j.simpat.2021.102428>
- Khemili W, Hajlaoui J, Omri M (2022) Energy aware fuzzy approach for placement and consolidation in cloud data centers. *J Parallel Distrib Comput* 161:130–142
- Li J, Shang Y, Qin M, Yang Q, Cheng N, Gao W, Kwak K (2022a) Multiobjective oriented task scheduling in heterogeneous mobile edge computing networks. *IEEE Trans Veh Technol* 71(8):8955–8966
- Li C, Liu J, Wang M, Luo Y (2022) Fault-tolerant scheduling and data placement for scientific workflow processing in geo-distributed clouds. *J Syst Softw* 187:111227
- Li C, Cai Q, Youlong L (2022c) Optimal data placement strategy considering capacity limitation and load balancing in geographically distributed cloud. *Future Gener Comput Syst*
- Lin B, Guo W, Xiong N, Chen G, Vasilakos A, Zhang H (2016) A pretreatment workflow scheduling approach for big data applications in multicloud environments. *IEEE Trans* 13(3):581–594
- Liu X-F, Zhan Z-H, Deng JD, Li Y, Gu T, Zhang J (2016) An energy efficient ant colony system for virtual machine placement in cloud computing. *IEEE Trans Evol Comput* 22(1):113–128
- Liu C, Wang J, Zhou L, Rezaeipannah A (2022) Solving the multi-objective problem of IoT service placement in fog computing using cuckoo search algorithm. *Neural Process Lett* 54(3):1823–1854
- Long S, Zhao Y, Chen W (2014) MORM: a multi-objective Optimized Replication Management strategy for cloud storage cluster. *J Syst Architect* 60(2):234–244
- Mahajan S, Pandit A (2021) Hybrid method to supervise feature selection using signal processing and complex algebra techniques. *Multimedia Tools Appl*. <https://doi.org/10.1007/s11042-021-11474-y>
- Mahajan S, Abualigah L, Pandit A, Altalhi M (2022a) Hybrid Aquila optimizer with arithmetic optimization algorithm for global optimization tasks. *Soft Comput* 26(10):4863–4881
- Mahajan S, Abualigah L, Pandit A, Altalhi M (2022b) Hybrid arithmetic optimization algorithm with hunger games search for global optimization. *Multimedia Tools Appl*. <https://doi.org/10.1007/s11042-022-12922-z>
- Maheshwari N, Nanduri R, Varma V (2012) Dynamic energy efficient data placement and cluster reconfiguration algorithm for MapReduce framework. *Futur Gener Comput Syst* 28(1):119–127
- Majed A, Raji F, Miri A (2022) Replication management in peer-to-peer cloud storage systems. *Clust Comput* 25(1):401–416
- Mirjalili S, Gandomi A, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) SALP swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 114:163–191
- Mohammadi B, Navimipour N (2022) A fuzzy logic-based method for replica placement in the peer to peer cloud using an optimization algorithm. *Wireless Pers Commun* 122(2):981–1005
- Nguyen B, Binh H, Son B (2019) Evolutionary algorithms to optimize task scheduling problem for the IoT based bag-of-tasks application in cloud-fog computing environment. *Appl Sci* 9(9):1730
- Pallewatta S, Kostakos V, Buyya R (2022) QoS-aware placement of microservices-based IoT applications in Fog computing environments. *Future Gener Comput Syst* 131:121–136

- Peake J, Amos M, Costen N, Masala G, Lloyd H (2022) PACO-VMP: parallel ant colony optimization for virtual machine placement. *Future Gener Comput Syst* 129:174–186
- Salem R, Abdsalam M, Abdelkader H, Awad A (2020) An artificial bee colony algorithm for data replication optimization in cloud environments. *IEEE Access* 8:51841–51852
- Sarwar K, Yong S, Yua J, Rehman S (2022) Efficient privacy-preserving data replication in fog-enabled IoT. *Futur Gener Comput Syst* 128:538–551
- Shi T, Ma H, Chen G, Hartmann S (2022) Cost-effective web application replication and deployment in multi-cloud environment. *IEEE Trans Parallel Distrib Syst* 33(8):1982–1995
- Taghizadeh J, Arani M, Shahidinejad A (2021) A metaheuristic-based data replica placement approach for data-intensive IoT applications in the fog computing environment. *Softw Pract Exper* 52(2):482–505
- Torabi E, Arani M, Shahidinejad A (2022) Data replica placement approaches in fog computing: a review. *Cluster Comput*. <https://doi.org/10.1007/s10586-022-03575-6>
- Wang M, Zhang Q (2020) Optimized data storage algorithm of IoT based on cloud computing in distributed system. *Comput Commun* 157:124–131
- Wang Y, Guo C, Yu J (2018) Immune scheduling network based method for task scheduling in decentralized fog computing. *Wirel Commun Mobile Comput*. <https://doi.org/10.1155/2018/2734219>
- Yang M, Ma H, Wei S, Zeng Y, Chen Y, Hu Y (2020) A multi-objective task scheduling method for fog computing in cyber-physical-social services. *IEEE Access* 8:65085–65095
- Yousif A, Alqhtani S, Bashir M, Ali A, Hamza R, Hassan A, Tawfeeg T (2022) Greedy firefly algorithm for optimizing job scheduling in IoT grid computing. *Sensors* 22(3):850

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

## Authors and Affiliations

Ahmed Awad Mohamed<sup>1</sup>  · Ashraf D. Abdellatif<sup>2</sup> · Alhanouf Alburaikan<sup>3</sup> · Hamiden Abd El-Wahed Khalifa<sup>3,4</sup> · Mohamed Abd Elaziz<sup>5,6</sup> · Laith Abualigah<sup>7,8,9,10,11,12</sup> · Ahmed M. AbdelMouty<sup>13,14</sup>

✉ Alhanouf Alburaikan  
a.albrikan@qu.edu.sa

✉ Laith Abualigah  
aligah@ammanu.edu.jo

Ahmed Awad Mohamed  
Ahmedawadmohamed2000@gmail.com

Ashraf D. Abdellatif  
ashraf\_abdellatif@yahoo.com

Hamiden Abd El-Wahed Khalifa  
hamiden@cu.edu.eg

Mohamed Abd Elaziz  
abd\_el\_aziz\_m@yahoo.com

Ahmed M. AbdelMouty  
a\_abdelmouty@zu.edu.eg

<sup>1</sup> Information System Department, Cairo Higher Institute for Languages and Simultaneous Interpretation, and Administrative Science, Cairo, Egypt

<sup>2</sup> Department of Technological Management and Information, Higher Technological Institute, 10th of Ramadan, Egypt

<sup>3</sup> Department of Mathematics, College of Science and Arts, Qassim University, Al-Badaya, Saudi Arabia

<sup>4</sup> Department of Operations Research, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza, Egypt

<sup>5</sup> Faculty of Computer Science & Engineering, Galala University, Suez, Egypt

<sup>6</sup> Department of Mathematics, Faculty of Science, Zagazig University, Zagazig, Egypt

<sup>7</sup> Center for Engineering Application & Technology Solutions, Ho Chi Minh City Open University, Ho Chi Minh, Vietnam

<sup>8</sup> Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan

<sup>9</sup> Computer Science Department, Prince Hussein Bin Abdullah Faculty for Information Technology, Al Al-Bayt University, Mafraq 25113, Jordan

<sup>10</sup> Faculty of Information Technology, Middle East University, Amman 11831, Jordan

<sup>11</sup> Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan

<sup>12</sup> School of Computer Sciences, Universiti Sains Malaysia, 11800 Gelugor, Pulau Pinang, Malaysia

<sup>13</sup> Information System Department, Faculty of Computers and Information, Zagazig University, Zagazig, Egypt

<sup>14</sup> Vice Governor of EL-Sharqia, Zagazig, Egypt