Egyptian Informatics Journal 24 (2023) 327–339

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

iFogRep: An intelligent consistent approach for replication and placement of IoT based on fog computing

Safa'a S. Saleh [a,*], Iman Alansari [b], Mounira Kezadri Hamiaz [b], Waleed Ead [c,e], Rana A. Tarabishi [b], Hatem Khater [d]

[a] Information Systems Department, Egyptian Institute of Alexandria Academy for Management and Accounting- EIA, Alexandria, Egypt
[b] Computer Science Department, College of Computer Science and Engineering, Taibah University, Madinah , Saudi Arabia
[c] Beni Suef University, Egypt
[d] Department of Electrical Engineering, Faculty of Engineering, Horus University, Egypt
[e] Egypt-Japan University of Science and Technology, (E-JUST), Egypt

ARTICLE INFO

Article history:
Received 24 November 2022
Revised 13 March 2023
Accepted 7 May 2023
Available online 13 May 2023

Keywords:
Replication
Consistency
Placement
Fog computing
IoT
Prediction
Optimization

ABSTRACT

Recently, the Internet of Things (IoT) has played key roles in every aspect of daily routine tasks. IoT applications suffer from higher latency and limited network bandwidth in the internet core. However, due to bandwidth and resource constraints, the resultant massive data traffic triggers storage overhead, high latency, and network bottleneck problems. Therefore, cloud computing and fog computing are introduced as popular IoT assistance. In fact, using a fog layer with such a highly dynamic environment raises a placement problem that needs to identify the optimal fog node that can be assigned to place the sensed data there. However, using a single copy of the datum in a single node can increase the concurrent access and maximize the network latency. Therefore, replication is considered a known technique to improve availability by publishing many replicas to many sites. However, using replication causes a degree of temporal data inconsistency and requires the application to be tolerant. This work proposes a novel intelligent solution for dynamic consistent replication and optimal placement of IoT using fog computing. The results show the ability of the current work to reduce the system latency and cost of replica access and increase the data availability.

1. Introduction

Recently, the Internet of Things (IoT) has played key roles in every aspect of daily routine tasks. It uses the continuous communication of the internet to link different computing devices (things) to permit users to do their jobs [1]. The IoT is based on sensing devices that monitor, capture, and share data using a wireless network (WSN) [2]. These sensing devices are sometimes referred to as data producers, and the IoT applications are referred to in the current work as data consumers. The IoT is characterized by a large data flow, so it is known as a data-intensive domain [1].

However, due to bandwidth and resource constraints, the rapid increase in the number of IoT users and the resultant immense data traffic trigger numerous problems including capacity overhead, high latency, and network bottlenecks. Therefore, the introduction of novel models that face these problems to make data processing and storage nearer to consumers has become a vital need.

Fog computing was invented by Cisco [3] to extend cloud computing and has become a popular feature of the IoT. It helps to deliver massive data between numerous smart instruments [4]. It is usually added as a network layer to receive data from IoT producers in real time and deliver them to the IoT application layer. Fog nodes are equipped for storage, processing, and communication tasks. In fact, using a fog layer with such a highly dynamic environment raises another placement problem. The chief function of the placement technique is to identify the optimal fog node that can be assigned to every sensing device. The current work is motivated to address this need using intelligent optimization and machine learning techniques.

However, a single copy of the datum in a single node can increase concurrent access and maximize the network latency. In fact, transmission delays are significantly known with several IoT applications and can easily grow into a bottleneck. Thus, data replication is seen here as a suitable solution to minimize the network

* Corresponding author.
E-mail address: safaa34@gmail.com (S.S. Saleh).

https://doi.org/10.1016/j.eij.2023.05.003
1110-8665/

latency [5]. Replication is one of the most widely utilized techniques that involves storing many copies (replicas) of information on numerous destinations in a distributed framework [6]. It is a common method to develop the availability and persistency of large-scale distributed systems [7]. Consequently, it is necessary to specify the ideal number of each datum's replicas and their locations to improve the system functioning and reduce network latency [7]. Consequently, it is necessary to specify the best number of each datum's replicas and their locations to improve the scheme functioning and reduce the network latency. Therefore, the current work is motivated to intelligently address this need using an ocean mining model. To achieve a rapid response in order to meet the time requirement and fault tolerance, the replication tasks are planned to be performed by IoT fog nodes. In this way, the computational tasks are moved to the network's edge, that is, they become closer to IoT devices.

However, in many cases, meeting time constraints comes at the expense of consistency [8]. Replication is one case that results in a sort of temporal data discrepancy that requires the application's comprehension. Although using state transfer instead of operation transfer can improve the level of consistency by decreasing the inconsistency duration [6], there is still a critical need to minimize the system latency of such systems. Controlling data consistency between different replicas of the same replicated datum represents a large challenge in such systems because of the large amount of information and the large number of nodes with heterogenous frameworks. The authors of [1] stated that the data items of the IoT can be stored at different sites with different consistency levels for different applications.

The current work plans to reduce the energy and latency costs for enhancing data availability and system performance for all IoT applications. The sensed data and their traffic conditions are assessed and estimated using an intelligent approach to identify the accurate data need of each fog node. This can achieve efficient network traffic that minimizes the collision between the transmitted packets. The authors of this work proposed a novel intelligent solution for dynamic consistent replication and placement of IoT using fog computing. This proposed approach is called iFogRep and makes the following three contributions. To save and assess the information in the fog layer using iFogRep, each sensing device must transfer its data to the predefined owner to decrease the latency, energy, and network consumption.

The **first** contribution of this work is a new dynamic placement mechanism that allocates each sensing device to the optimal owner node. It uses a machine learning technique with an optimization function to estimate the real latency between the sources and destinations. The dynamic placement algorithm also defines the optimal capacity of each owner. The optimal owner is identified based on the lowest latency and energy. The proposed placement algorithm reduces the consumption of the whole system latency.

To reduce the quantity of transmitted data and minimize the access latency of applications, the **second** contribution proposes a novel intelligent replication protocol. The proposed partial detached replication algorithm uses an adaptive prediction technique. The adaptive prediction technique predicts the ideal number of copies of each datum and determines their locations dynamically. Additionally, it uses an efficient machine learning mechanism to estimate the lowest latency path of each location.

The paper's **third** contribution is to determine the consistency level of each replica of each datum dynamically based on the optimistic replication to support IoT in defeating the temporal inconsistency problem. The proposed consistency method can adaptively change the consistency level of each replica according to the remaining network resources. It permits numerous nodes to overhaul their data simultaneously without any need for dispersed synchronization.

The rest of the paper is organized as follows: Section 2 addresses the previous work that was concerned with IoT problems of latency, availability, and inconsistency. Section 3 presents the suggested approach and algorithms to improve the placement, replication, and consistency of the IoT. Section 4 shows an experimental study that evaluates the proposed approach. Finally, the conclusion is presented in Section 5.

2. Related work

This section reviews the efforts introduced to enhance the data placement, replication, and consistency using the edge/fog environment. These related works are categorized briefly into three domains in light of the contribution of the current work.

2.1. Placement

Various placement approaches and techniques are introduced to reduce latency cost using fog, edge, or cloud environments. In [9], the authors suggested a graph portioning replica placement approach to identify the optimal fog node for storing replicas to secure a rise in data availability. [10] proposed a dynamic approach for information arrangement in fog computing depending on the node's serviceability and predicting the production of data replicas. [11] used a concurrent data placement method to minimize the system latency and reduce the consumed network bandwidth. [12] developed a dynamic data placement protocol for data recovery and balancing the workload during the process of replication. [13] introduced a block replica placement management approach that selects and categorizes the edge nodes into various categories based on the needed capacity and abilities. The work by [14] introduced a strategy for copy arrangement and information reliability using fog computation. It splits the fog nodes into failure clusters and then selects the "N" nearest fogs to each producer to put "N" copies. However, this approach neither takes the access pattern of consumers nor the delay of replica synchronization into account when placing replicas. In [13], both latency and data overhead were reduced during the data placement retrieval service using edge computing. They utilize hashing information identifiers that are recognized by employing a string of characters. Then, they concatenate the serial number of replicas with the identifier to create a novel string. The new hashing string determines the placement of the replica. However, this strategy does not consider the data access latency during replica placement identification and does not manage data consistency. A work by [15] proposed two placement strategies: one for latency-aware various copies using a fog layer and a machine learning strategy. Their results reveal a reduction. However, this work did not address either the data consistency or the synchronization latency of replication. [16] presented an optimization conceptual fog system that demonstrates the benefit situation by employing a genetic algorithm as a heuristic determination. Their outcomes display a decrease in network communication postponements with the genetic algorithm and an improved use of resources with the optimization method.

2.2. Replication

The work by [17] offered an adaptive combined distance method that depends on spatiotemporal data, the distance from the user of data, and a self-ruled battery-powered node. The results show decreased communication time due to the replicas being placed closer to consumers. A work by [18] introduced a dis-

tributed placement and replication architecture that uses fog nodes. [5] introduced a middleware framework to support data replication using a fog layer and a data distribution component. [19] established a heuristic technique to categorize IoT data for defining the type of data (latency-sensitive or computationally sensitive). Similarly, they presented a graded information placement framework by means of cloud and fog layers. Their data replication is dependent on the most prominent centrality score, as the data replication is performed on disconnected fog nodes utilizing two parts. [15] improved a greedy-based replication approach to decrease the system latency. It minimizes the search space by dismantling the unfitting solutions by experimental rules in polynomial time. [7] proposed a simple technique for random replication in portable edge instruments. The solution is then analyzed compared to different portable instrument specifications. The results show a significant reduction in network latency and bandwidth usage. [13] suggested a metaheuristic replication technique. Their strategy may be a multi-objective optimization that utilizes a genetic algorithm. Their essential objective is to find the optimal target node using minimum time and network bandwidth. [20] proposed a dynamic technique for information copies in edge and fog foundations depending on a metaheuristic ant colony. Their technique evaluates the sum of information replicas dynamically earlier, recognizing the ideal capacity instrument to put in the fog-edge situation. [21] offered a limit approach to optimize the edge and fogging structure by dynamically assessing the number of replicas, and the optimal storage goal to store them relies on the frequency of data blocks. [15] suggested a delay-adaptive recuperation strategy for replication. It dynamically matches and equalizes the number of copies. [4] developed a proximity-aware activity steering framework for replication. This suggested strategy allows the administrators to control the trade-off between diminishing the latencies and adjusting the workload between copies. Additionally, [22] presented a novel solution based on data mining for dynamic replica management using fog computing. This work uses the maximum frequent pattern to enhance replication and reduce the cost of data management. The results showed decreased total latency issues compared to their related works, which points to the good optimization for Fog-enabled IoT. The authors of [3] presented a metaheuristic-based method that is based on a genetic algorithm for fog-based IoT applications. They provided an automatic replica transmissions method that organizes them in a fog cloud situation.

The current work plans to use an optimized method to estimate the lowest delay and energy path during the propagating phase. Some related works in this domain are also explored. In fact, there have been many efforts that try to obtain the path optimization from similar networks. A work by [23] used the particle optimization approach for directing WSNs with a moving sink. This approach discovers a tree of lower energy and latency by distribution routing data. However, this work ignores utilizing an accurate optimization method to obtain optimal values, so the obtained values are not optimal. Another work by [24] suggested a Markov algorithm with a multimodal client portability pattern forecast approach that relies on real-world GPS positions for forecasting routes. The authors of [25] proposed an attention method into the Markov technique to expect upcoming positions depending on a GPS dataset that has operator directs. A work by [26] supported the fast establishment of the transmission paths in mobile networks for reducing the network latency and energy utilization of the network. It introduced various objective optimization methods and a prediction protocol for distributed routing using the Markov chain technique to predict the ideal communication method between a sink node and source node. They also introduced a deep learning method to forecast the upcoming distances of nodes to determine whether the next travels of nodes can lead to interruptions in communication. The authors of [27] introduced a dynamic data placement and replication approach. The graph partitioning (GP) data placement method uses an edge server for efficient storage to minimize the latency. The proposed replication method implements a support vector machine as a machine learning technique to classify the data center and forecast the workload.

2.3. Consistency

Consistency is one of the main topics when working with replicated data in distributed systems [6]. Maintaining temporal or even permanent consistency is a main goal in distributed time-constrained systems because of the ease of consistency at the expense of availability. Several resolutions with different techniques have been presented to address the inconsistency issues of such distributed systems. A work by [14] introduced a strategy for copy placement and information consistency using fog nodes. For the data consistency administration, this work identified many consistency levels based on the number of copies that must reply to all read requirements. Then, it assigned each read request to the right consistency level according to the user location. A work by [6] proposed a consistent replication protocol based on an adaptive procedure to dynamically identify pattern access. Because time is important in such environments, the authors considered the timing characteristics of both network and data to address the temporal inconsistency troubles by eliminating useless processes. This approach lets several nodes access data concurrently. Their results showed a reduced time delay that consumed and maintained the consistency. A work by [1] introduced an approach for data placement, replication, and consistency. Their methodology depends on exploring the best storage location of replicas to decrease the latency of synchronizing replicas by consumer applications. The authors also found the proper replica number of each datum and their location while respecting the required consistency to lower the access latency. They suggested two solutions for replica allocation by means of integer programming approaches and geographic-area-based heuristics. The heuristic method divides the fog nodes into a number of subsections and implements replication in these reduced parts. To propose a comprehensive solution, all subsections of the solution are aggregated.

Studies by [1,27,22,3] are considered the most related works to the introduced approach. The proposed replication approach is similar to these works in several aspects:

1) Similar to the current work, all related replication and placement approaches are based on a fog or edge layer to minimize the whole system latency.
2) Additionally, all of these related works involve the enhancement of replica placement between the fog/edge nodes to reduce the latency of placing and transmitting replicas.
3) Similar to the proposed protocol, some of them dynamically retrieve the access pattern to determine the consumers of replicas to achieve fully automatic tuning.

One of the key differences between these related works and the current work is that the current work uniquely studies the reduction of the delay caused by transmission from the producer layer to the fog layer.

Additionally, the current work and the work by [1] are the only ones that take consistency and optimal routing into account, but they use different techniques. Although the results of the placement technique were used to allocate replicas between fog nodes by [27], this work decided to use support vector regression (SVR) as a mining model instead of the ARIMA model based on the results of a comparative study of [28].

Table 1
Summary of most related works.

Authors	Initial Placement	Rep_ placement	Replica #	Shortest path	Consistency
Dhande, 2020 [22]	✗	LP	Pattern Mining	✗	✗
Guerrero et al., 2020 [9]	✗	LP	GA	✗	
Naas et al., 2021 [1]	✗	mathematically		P- median	✓
Shankar et al., 2022 [27]	✗	SVM	Optimized GP	✗	✗
Taghizadeh et al., 2022 [3]	✗	ARIMA model	GA	✗	✗
iFogRep	Optimized SVR	FP-MFIA Mining model		Deep learning	✓

LP: Liner Programming- GA: Genetic Algorithm - GP: Graph Partitioning- SVR: Support Vector Regression – SVM: Support Vector Machine.

A summary of the differences and commonalities between the present work and these interrelated works is offered in Table 1. To better understand this table, it is crucial to consider the following observations:

- Initial placement mentions the placement technique used to associate each sensing device with the optimal fog node.
- Rep_ placement refers to the placement technique used to allocate replicas to fog nodes.
- Replica # refers to the method utilized to detect the number of copies of every datum.
- The shortest path refers to the method used to determine the optimal path.
- Consistency means whether a consistency mechanism is used during prediction operations.

Accordingly, this proposed approach introduces four kinds of novelties compared with the related approaches.

(i) Rather than placing data into many fog nodes, the current approach identifies one optimal fog node to be the owner of each datum for quicker access through reduced responding time.

(ii) A dynamic optimized placement method is introduced with reduced latency and minimum cost to fulfill the dynamic nature of the SN.

(iii) The intelligent prediction algorithm using the FP-MFIA Mining model is used to estimate the optimal owner dynamically of each storage to reduce system latency.

(iv) The proposed protocol uses a novel consistency model inspired by the mapping between the timing requirement of applications and the latency cost among the dispersed nodes.

3. Proposed approach

The sensing devices of the IoT cannot hold many data copies for retransmissions because of the limited energy and limited storing capacity, which may result in an availability problem. In addition, the limited bandwidth of such systems can cause a problem in meeting the critical time requirement (system latency). Therefore, fog computing is used here for holding data to face these problems while keeping the system efficient [27].

On the other hand, using one copy of the datum at one site permits the concurrent reads, which increases the system latency and impacts the ability of the system to meet the time requirements. In fact, latency is considered a critical problem of the IoT. To minimize the latency while using fog computing, replication techniques are implemented to provide many copies of datums at different sites. This allows many applications to be run concurrently and enhances the system performance while meeting the time constraints and minimizing the latency [7].

However, using replication in general raises the cost of storage capacity, energy consumption, and network traffic, especially when disseminating replicas. Full replication consumes system resources, as all updates must be published to all nodes regardless of whether these replicas are used in all sites there [6]. Therefore, the challenge of using replication with such systems is to avoid full replication, which can consume all system resources. In fact, the distribution data needed in such systems are extremely skewed, as a very small part of the data is accessed frequently by all sites, while the remaining large part is rarely required there [17]. In other words, all data items are not usually required at all sites; that is, each site requires only a very small part of the data. Considering this need can help minimize network traffic and ensure low response times. One fundamental requirement here is to predict the actual data needed at each site.

Replication is usually added to achieve high data availability, but adding replication introduces a key need to maintain data consistency among replicas. Data consistency is defined as the capability of the platform to synchronize the data copies to the similar last form [1]. Indeed, replica synchronization adds a latency above to the platform based on the quantity of copies and their locations.

Organizing data consistency in dispersed platforms is a recognized problem, especially given the large quantity of data and the large number of sites [1]. The trade-off between information consistency and meeting time necessities offers the ultimate challenge to some work in the replication field, especially with time-constrained systems. In fact, different applications may need the same data copies with different consistency requirements [1]. For example, security programs may need strong consistent data (latest data version) that aid in fast danger defense decisions. Other applications can tolerate some lateness to make the necessary decisions (can use nonupdated information) [1].

The current work uses three levels of consistency for each group of data at each fog node (see Fig. 1). The data items in the first consistency level are transferred from the sensing devices to the corresponding owner, which acts as a publisher of these data items the first level, data items are transferred automatically once they are collected to their owners. The nodes that are listed in the strong consumers of datum are in the second level of consistency due to their higher priority to receive the last version of data as much as they are received. Finally, the third level is the regular consistency level, where data items are synchronized regularly at low workload hours.

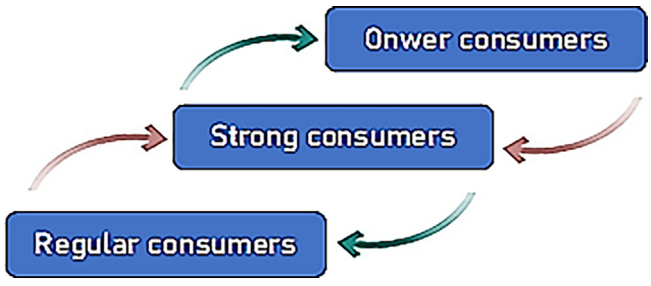


Fig. 1. Consistency levels of iFogRep.

This research suggests a new tolerant optimistic replication protocol for IoT referred to as **iFogRep** to enhance consistency while meeting the time requirements. **iFogRep** is based on a machine learning mechanism to predict and identify the owner based on the optimized parameters. Additionally, **iFogRep** acts to detect the optimal number of replicas of each datum at each site and their destinations to reach the lowest system latency. To do this, **iFogRep** uses a customized pattern discovery machine learning technique to identify the strong consumers of each datum according to recent access patterns. The proposed adaptive consistency technique is used to determine the appropriate consistency level of each replica of each datum at each site to reduce the whole system latency.

3.1. iFogRep

This section introduces the structure of the suggested **iFogRep** system that supports the consistency and latency of the IoT system. The basic structure of the **iFogRep** system is proposed with the main components required to set the adaptive replication with an intelligent placement protocol. Each fog node in such a structure is a peer node, and the important operational components that are utilized by every node are introduced in Fig. 2.

The sensor devices represent the data **producer component** and are represented by the sensing devices that are responsible for collecting data from the surrounding environment and forwarding them to the corresponding fog nodes via the placement manager component. This **placement component** is responsible for allocating each sensing device to the optimal fog node with minimum latency based on physical properties such as the required communication power and time costs between the sensing node and fog node. The data **consumer component** represents the applications from different domains that require accessing the different nodes in the fog layer to read the recent version of the stored datum. The information of each access of these applications is recorded in a special history log file, which is used by the consumer miner component to predict the consumers of each datum. The **replication manager component** (Rep. Mgr.) consists of two mining modules: the path miner and the consumer miner.

The resulting access history file is continuously monitored by consumers miner to discover and extract the pattern access and uses the extracted pattern to predict the strong consumers of each datum.

The proposed algorithm for discovering and extracting the pattern access is modified from the work by [8,6,22] to predict the consumer matrix. The consumer estimator module uses a machine learning technique to predict the consumer list of each datum. This list is used by each fog node to publish the most recent version of its own data items once received.

The consumer list is applied by increasing or eliminating nodes according to any change in the pattern access. The path miner subcomponent is responsible for using the information about the physical properties of the network to determine and estimate the optimized shortest path between all fog nodes. The current path miner uses an algorithm introduced by [26].

Finally, the fog nodes are responsible for propagating the recent version of each datum once it receives to the consumer using both the consumers list and the optimized path list.

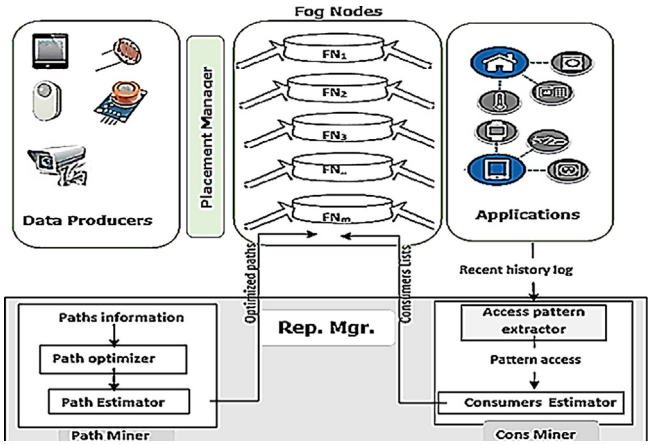


Fig. 2. iFogRep System Components.

3.2. Assumptions

This work considers the following assumptions for all undermentioned algorithms and protocols:

The following physical properties of the communication are known or can be measured:

a. The latencies between fog nodes and each other and between fog nodes and sensing nodes.
b. The required energy cost between fog nodes and between sensing nodes and fog nodes.

The collected data cannot be modified at fog nodes, so they can be requested for reading only.

Each datum is coded by the identifier of its sensing device, i.e., sensor si is responsible for collecting the datum di.

Each sensing device is assigned uniquely to its optimal fog node, and every fog node j is in charge of receiving a subset of data items. Therefore, the fog node is referred to as the owner of those items. In other words, each datum di is allocated to only one specific fog node that is referred to as the owner of that item.

3.3. Mathematical model formulation

Table 2 summarizes the notations used in this section. The fog nodes in **iFogRep** are represented as FN= {fn_1, fn_2..., fn_m}, and the data items are represented as D = {d_1, d_2..., d_n}. Each FN is associated with a subset of data items, which are represented as D_k = {d_1, d_2..., d_k}, where k < n. The candidate consumers of d_i reflect the number of replicas of d_i, and they are represented as con_i = {fn_1, fn_2..., fn_y}, where y < m.

The fog network in iFogRep can also be seen as an undirected graph G = {FN, E}, where the vertices represent the fog node set and the edge set E = {e_1, e_2..., e_z} represent the path between nodes. The latency between a sensing device i and a fog node j is

Table 2
Description of notation.

Notation	Description
SD	The set of sensing devices
FN	The set of Fog Nodes
fni	The Fog Node i
D	The set of Data items
di	The datum i coded by the identifier of its sensing device si
MyFNi	The set of candidate owners of the datum i
Lij	The latency between sensing device i and a fog node j
Conij	The datum i is needed by the Fog Node j
Ri	The communication range of the sensing device for the datum di
REi	The residual energy of node i
QE_{ij}	The required energy between node i and node j

L_{ij}, which is the cost of the shortest path between the sensing device i and the node j. The shortest path is that with the lowest latency cost.

The consumers of each data item are predicted by the consumer miner component of the replication manager. Each list is presented as a row in the data consumers matrix Con in which rows represent data items and columns represent nodes, and it is represented as

$$Con_{ij} = \begin{cases} 1 & datum\ i\ is\ needed\ by\ node\ j \\ 0 & otherwise \end{cases} \quad (1)$$

This matrix is updated constantly if any change occurs in the access pattern, such as when a new datum is added (new sensor is joined). The existing datum is removed (existing sensor die) or a fog node is added or removed.

The total time cost can be evaluated as the sum time needed by every FN to publish its own data to its consumers:

$$Total\ time\ cost = \sum_{i=1}^{n} \sum_{j=1}^{k} L_{ij} \quad (2)$$

Each sensing device that identifies the data item must be assigned to only one fog node. The current work seeks to allocate each datum to the optimal fog node. The optimal fog node here is the fog node that achieves the minimum latency and energy consumption. By the process of iFogRep, each sensing device of datum d_i has a set of candidate owners ($MyFN_i$) that are in its communication range (R_i) with a distance $\leq$ threshold (d_0). The d_0 is the distance that can conserve the energy of the sensing device. The number of elements of MyFN can be used as an indicator of the density around each cluster header (CH) [29]. For sensing devices, the set of candidate owners is defined by Eq. (3), where FN_j is any fog node in the communication range of the sensing device of d_i.

$$\text{MyFNi} = \{fn_j | \mathbf{d}(\mathbf{i}, \mathbf{j}) < \boldsymbol{R_i}\} \quad (3)$$

The first level of the optimization process is achieved based on the ability of the current sensing device (S_i) to connect the fog node (fn_j) with minimal energy (minimum required energy QE) to maximize its residual energy (RE_i). This level of optimization acts to identify all possible suitable fog nodes for the sensing device.

$$f1(i,j) = \begin{cases} -1 & if\ wene_{ij} > \ threshold \\ 1 & otherwise \end{cases} \quad (4)$$

where

$$wene_{ij} = RE_i - QE_{ij} \quad (5)$$

To define the set of accepted owners for each d_i, a decision function is used. It is represented using the binary function $\boldsymbol{f}1 \in \{-1, 1\}$, which means the sensing device of d_i can be allotted to the fog node fn_j according to QE.

The second level of the optimization process is then performed to weight each candidate owner for each d_i using the cost of the latency factor using Eq. (6). The fog node is accepted as an owner if the latency is less than the acceptable delay $l_{\varnothing}$. The delay cost (latency) of the data placement is computed here based on the basic latency between sensing nodes and fog nodes. The value of $\boldsymbol{f}2$ decreases the chance of raising the latency between data producers and owners.

$$\boldsymbol{f}2(\boldsymbol{i}) = \min_{j \leq l} \left(\frac{\mathbf{L_{i,j}}}{l_{\varnothing}} \right) \text{ Where } l = |\text{MyFN}_i| \quad (6)$$

The placement optimization process ensures that each sensing device is allocated to the lowest-delay fog node using minimized energy according to Equation (7). This expression tries to obtain an optimal solution which combines two requirements of latency and the suitability of power consumption. Otherwise, considering only the delay function (f2) with the main target of the present work to minimize the latency without considering the power consumption (f1), is very a big drawback especially in case of WSN with its known limited energy resources that need to be conserved as much as possible. Likewise, considering the power consumption function (f1), the solution will not consider the presence of long delays, which can result in a negative effect on time-sensitive applications.

$$\boldsymbol{F} = \min \sum W_2 * f_2(f_1(\boldsymbol{x}_1) * w_1 + x_2 \quad (7)$$

Equation (7) as a general objective function consists of two objective functions (f1) and (f2) for the power consumption cost then on the latency cost. The constraints of the placement optimization process are to ensure that each sensing device (SDi) is assigned to one fog node (Equation (8), and the highest number of sensing devices that can be allocated to each fog node is limited based on the available resources (Equation (9).

Subject to:

$$\sum_{j=1}^{\boldsymbol{MyFN}} fn_j = 1 \forall SD_i \in n \quad (8)$$

$$(\sum_{i=1}^{SD} SD_i) < \text{Max}_0 \forall fn \in MyFN \quad (9)$$

This work also proposes a two-phase machine learning approach to predict the best owner for data of each sensing device. The first learning phase acts to estimate the future energy cost as formulated as an SVM hyperplane in Equation (10). The hyperplane is used to divide the data into two classes that are: (1) positive sample: fn_j is a possible candidate powered owner for the datum d_i, and (2) negative sample: fn_j is not a possible candidate powered owner for the datum d_i. The training data of this first learning stage is defined as a vector $\times$ where each component contains the information about the distance between every fn_j and a sensing device of datum d_i, the communication range of the sensing device, the minimum required energy, and the residual energy.

$$\omega * x + b = 0 \quad (10)$$

Where ω is the weighting vector and b is a threshold. The second learning phase is used to estimate latency cost of the accepted nodes at first level as expressed in Equation (8). Formally, the real-valued function is also approximated with support vector regression where the two classes are: (1) a candidate owner fnj is a suitable owner of the datum di, and (2) a candidate owner fnj is not a suitable owner of the datum di. For this stage, the training data contains the results of the previous learning stage in addition to the expected delay between the owner of datum and its producer.

3.4. Data placement algorithm

The main task of the placement manager is to assign the sensed data dynamically to the optimal fog node. The placement technique depends on the optimization functions in Eq. (4) - Eq. (8). The optimal fog node here is the node that needs the lowest energy cost and communicates the sensing device with a minimal delay. Algorithm 1 introduces the operating procedure of the placement manager. The function Capacity(C) is a Boolean function that returns true if the total number of assigned sensing devices is still accepted, i.e., lower than the predefined optimal capacity. The optimal capacity is defined based on imitation from the work introduced by [29,31].

Algorithm 1: Optimal SVR for Dynamic placement

Inputs: FN vector {fn_1, fn_2, ..., fn_m}, sensing devices set {S_1, S_2, ..., S_r}, delay cost matrix d[m][r], power cost matrix (for each FN/S) E[m][r].

Output: Optimal fog list

```
Step 1:  Foreach data Si
Step 2:    initialize the accepted fog vector fv[] with no length
Step 3:    foreach fnj in FV
Step 4:      Get all accepted FN using the 1st competition Eq. (4)
Step 5:      Crossover and mutation and selection
Step 6:      If fnj is optimal solution then, add to fv[]
Step 7:    end for
Step 8:    For each C ∈ fv[]
Step 9:      Estimate the latency
Step 10:     Get all optimal latency using the 2nd competition Eq. (6)
Step 11:     Select next C with lowest latency Eq. (7)
Step 12:   If Capacity(C) then owner= C
           Else go to step 11
Step 13:   end for
Step 14    Return owner of Si
Step 15  End for
```

3.5. Replica allocation and prediction algorithm

The dynamic replica allocation technique is dependent on the significance measuring of the last access data. To decrease the latency produced due to data access, it is important to consider the maximal frequent pattern. The main function here is to convert history log file access into a two-dimensional matrix (FA[datum] [fog node]) of applications that are indicated by the owners of their frequently accessed data items.

The following dynamic mining allocation method uses the maximal frequent factor to estimate the correlated pattern based on the work of [22] for fog computing. This algorithm that receives the historical log file in real time (real-time traces) as input, implements a predefined mining function. This can reduce the overall latency by forecasting the requesting nodes (consumers) using the real-time traces.

FP-MFIA is a maximum frequent itemset mining function introduced by [32,33]. It is based on the FP-tree as a tree architecture for frequent patterns. This function starts by constructing a one-way FP-tree architecture.

Algorithm 2: Number of replicas and their allocations

Inputs: Historical Log File (HLF)

Processing

Initialize: frequent patterns vector P and replica placement nodes vector R

(1) Initialize: + replica vector for datum I (RV_i)
(2) Initialize the suggested consumers matrix SC[][]
(3) Investigate the access history file (HLF)

```
For each raw of SC (datum)
    For each col of SC (Node)
        retrieve fog node in (FNi) vector.
        remove duplication.
    next
next
```

(4) Validate the suggested consumers matrix SC[][]
(5) Convert HLF into integer value in FA[][] as a maximum frequent access to get groups.

```
For each raw of HLF
  If FN is not data owner, then
    SC[ID][FN] ++
  End if
next
```

(6) Run FP-MFIA () function to mining & discover the strong consumers.
(7) Sort the frequent consumers descending for each datum.
(8) Explore the P highest consumers of each datum for dynamic replication.

Output: Matrix of strong consumers for each datum

Table 3
Complexity analysis of IReIDe.

The Algorithm	Time Complexity	Component
Algorithm 1	O(*S*CF*)	Placement
Algorithm 2	O(*S*N*)	Rep. Mgr

3.6. Complexity analysis

To evaluate the time complexity of IFogRep, we consider the two main algorithms (Algorithm 1, and Algorithm 2) that are presented in sections 3.4 and 3.5. Algorithm 1 is responsible for identifying optimal fog node for each sensing device. The time complexities of Algorithm 1 and Algorithm 2 as shown in Table 3 are O(S* CF) and O(S*N) respectively where S is the total number of sensing devices, CF the number of candidate owners, and N is the number of fog nodes. Fortunately, these two algorithms are used centrally once at the beginning of system only [34].

3.7. Replication model

The current study addresses the troubles of data placement and replication of IoT data using fog nodes. Each fog node acts as a primary data actor for a subset of data (its own data items) that are stored in its first data directory (DC1). It is responsible for publishing the last versions of these datums once receiving. At the same time, the same fog node is defined as a consumer of another subset of data (its target data items) and plays here as a secondary data actor for this subset of data. It is responsible for receiving the last propagated versions of the data that they interest in and storing them in its secondary data directory (DC2) (**see** Fig. 3). The regular data directory (DC3) is created to keep the last version of the replicated remaining subset of data, which is replicated by the replication manager (Rep. Mgr.) in regular periods during offline hours or low-workload hours. Each fog node is responsible for updating its main database using the receiving any new versions of (DC1), (DC2), or (DC3). Each replica is identified in each directory by three parameters: {owner, value, TS}. Data represents the value that is sent by the corresponding sensing device, and TS represents the timestamp of each collected value. The number of replicas of each datum for DC1 is restricted to (x) replicas based on the available storage capacity, and the newest version overwrites the oldest if the number of replicas $\geq$ x.

Each application can read from the primary DC1 of any fog node or DC2 of another node according to the lower communication latency. The fog node responds to data access requests with the newest version of the requested datum from the three directories using timestamp (TS_d). The global recent timestamp (TS_g) is continuously updated by each fog node. The fog node asks and the replication manager for the newest replica if the difference between TS_g and TS_d is greater than the threshold. In general, the global timestamp is the timestamp of the newest sensing operation, and it is tracked by the replication manager that broadcasts any update of it. The proposed publish-subscribing replication model of iFogRep is as follows (**see** Fig. 4):

- The sensing device collects the new value and sends it to its owner. (Algorithm 1 identifies this owner.)
- The owner receives the new version of the datum and stores it in DC1. This triggers the publishing module.

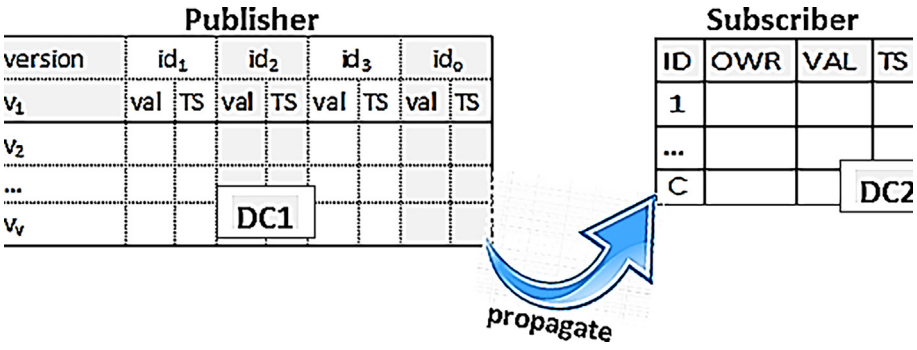


Fig. 3. Publishing-subscribing Replication Model.

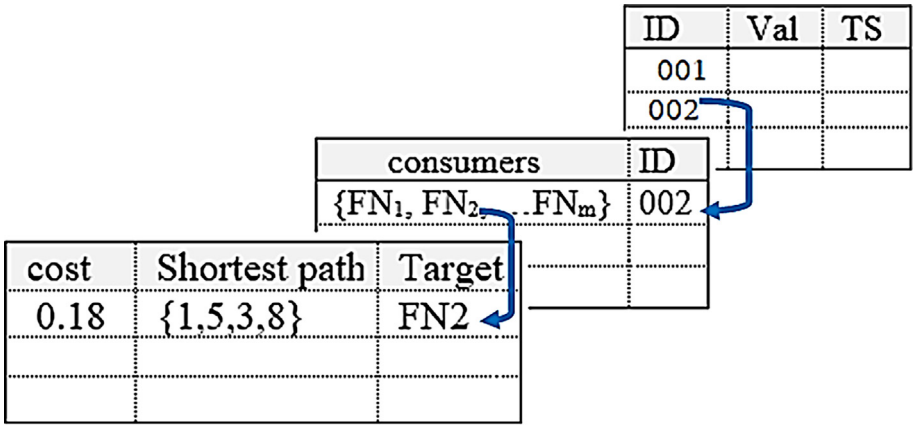
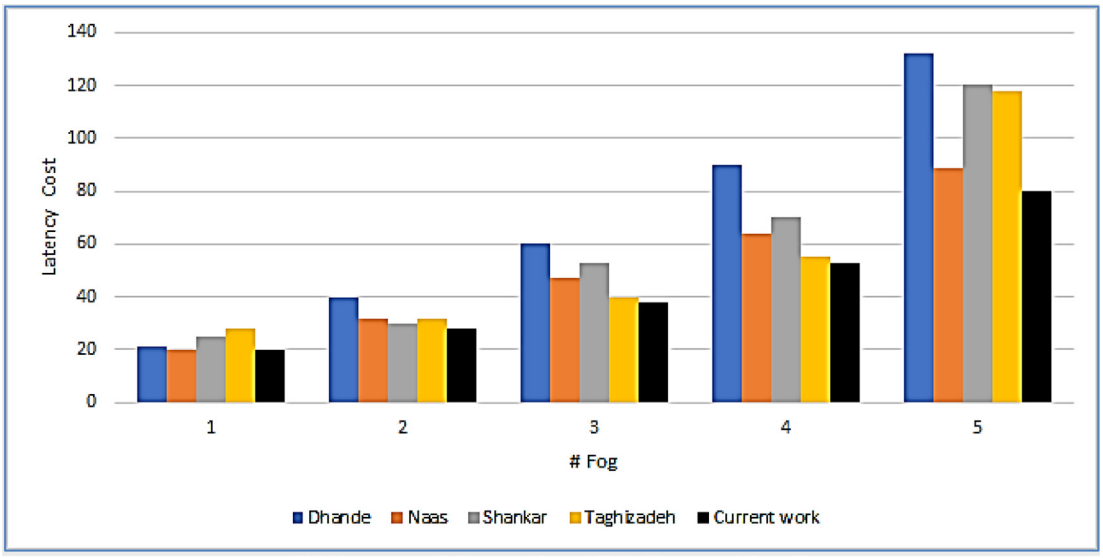
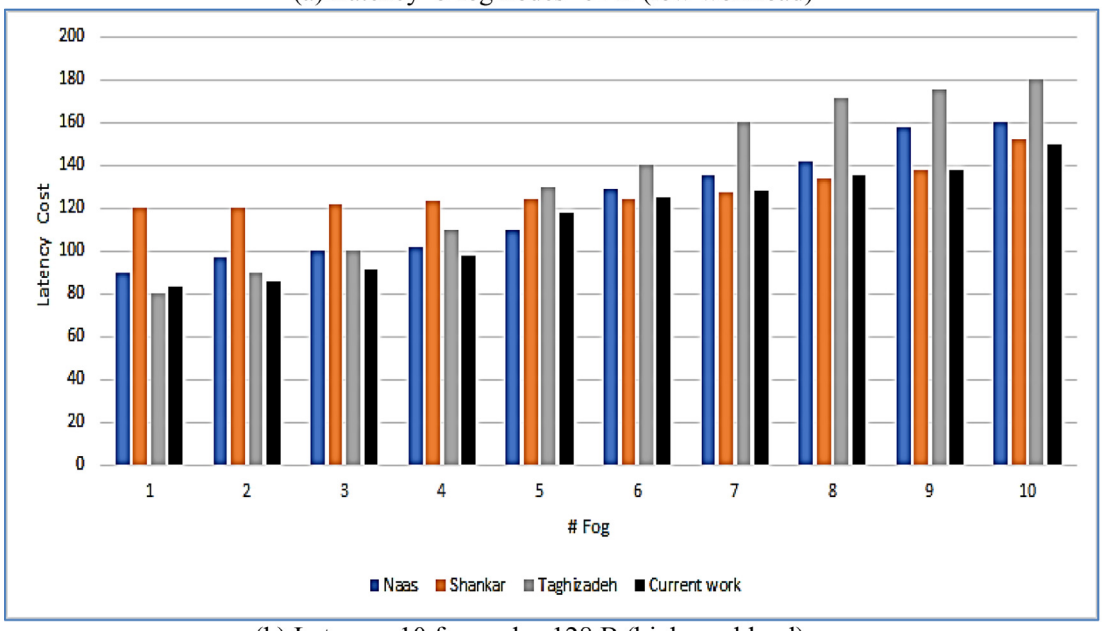


Fig. 4. Publishing-subscribing Replication Model.



(a) Latency- 5 fog nodes- 64 B (low workload)



(b) Latency- 10 fog nodes-128 B (high workload)

Fig. 5. System latency by iFogRep and related works.

- The publishing module propagates the recently received version of the datum to all of its consumers, which are defined by (Algorithm #2) using the lowest-cost path.
- The consumer fog node receives the published replica and stores it in DC2, overwriting the previous version of the datum.

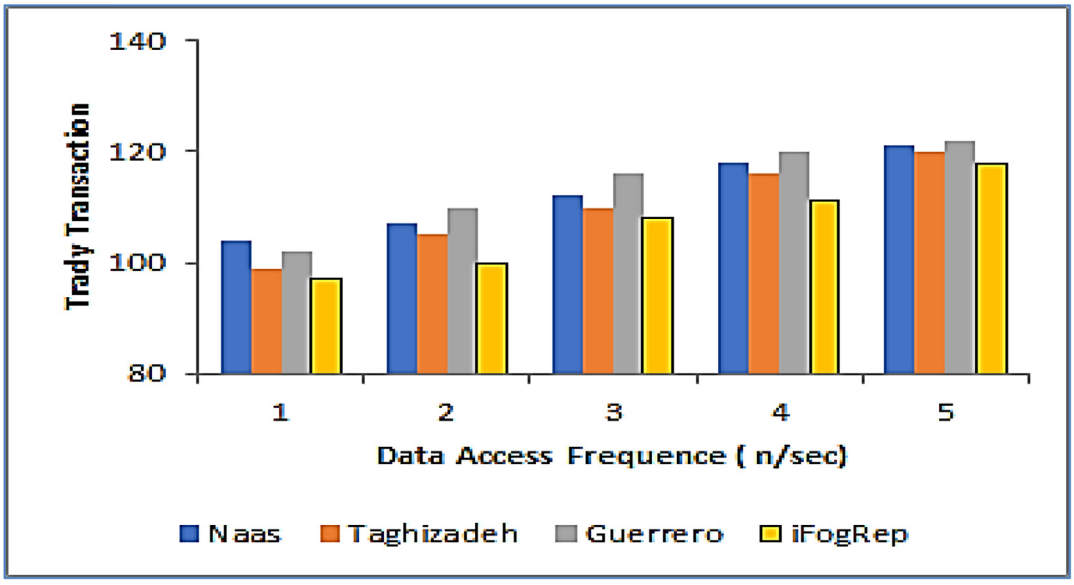
Table 4
Complexity analysis of IReIDe.

Parameter	Set value
Monitoring area	100 m^2 – 200 m^2
# Sensing devices	100–200
Packet size	4000 bit
# Fog Nodes	10–50
Storage capacity of FNs	10 TB
Initial delay SD-FN	10 ms
Initial delay FN-FN	5 ms

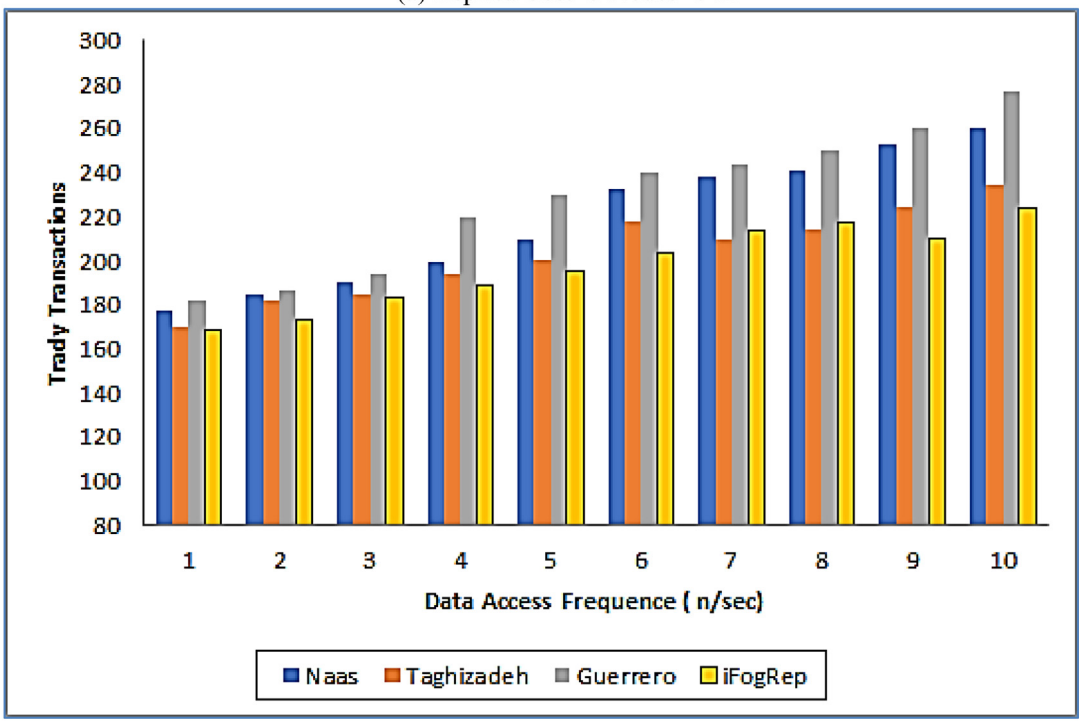
3.8. Consistency model

iFogRep implements three consistency modes. Every application in the platform utilizes a specified mode. The implemented modes are as follows:

- The application with a need for strong consistency of a certain datum is assigned to its DC1 as a first choice and then to the DC2 with the lowest latency as a spare choice.
- The application with a lower consistency needs of a certain datum can be assigned to the lowest-latency DC1 or DC2.
- The application with a need for relaxed consistency of any datum can be assigned to fog node with the lowest latency to read DC1, DC2, or even as a first choice, then to DC2 as a spare choice.



(a) Replica Cost- 5 Nodes- 64B



(b) Replica Cost- 10 Nodes – 128 B

Fig. 6. Replica Cost by iFogRep and related works.

- Requesting any replica from DC2 automatically calls the consistency_check() function that triggers replication requests if the difference between the current timestamp and global timestamp > the threshold.

4. Experiments and results

Because of the deficiency of experimental tools in this framework, an extended version of iFogSim [35] is modified using a Java editor to include the proposed algorithms. This extension is used to evaluate the dynamic replication iFogRep approach. The iFogSim2 simulator is introduced as an extension platform from iFogSim [36] to support machine learning, microservices, and mobility of different fog-based IoT frameworks. In fact, iFogSim is simulated in many scenarios where it stresses the power cost, latency, and network traffic load as metrics of the system performance.

The assessment experiments are performed using a Windows 10 64-bit machine with an Intel Core i7 2.4 GHz, 1 TB storage space, and 16 GB RAM. Table 4 summarizes the experimental settings that are similar to those of related works.

Initial delay indicates the minimum latency between fog nodes (FN-FN) and between fog nodes and sensing devices (SD-FN).

4.1. Performance metrics

Influenced by the related works, this study uses the following metrics to evaluate and validate the proposed approach:

4.1.1. Average latency cost

This metric is defined as the average generated latency (delay) periods of accessing data (downloading and uploading replica) between the sender and destination. In fact, this is the most important metric in this work because reduction of the system latency is the ultimate goal of this work and its related works. Thus, by this metric and for each strategy, the average of generated latencies for accessing all data in the system (VL) is calculated according to Eq. (9) where *Total time cost* is expressed by Eq. (2).

$$VL = (\sum_{i=1}^{n} Total\ time\ cost)/n \tag{9}$$

4.1.2. Replica cost

This metric is defined as the average number of replicas generated by using each approach. This metric can be considered a good indicator of the other three metrics: network traffic, utilized storage cost, and power consumption of each evaluated work. These are all correlated to the number of replicas generated. The data storage cost (ST) of accessing a data replica was calculated by [3] according to Eq. (10), where tn is the cost of location access, and size (d_i) is the size of the datum replica.

$$ST = \sum_{i=1}^{n} (tn \times size\,(d_i)) \tag{10}$$

The communication cost that is needed to transfer the replica or replica communication cost (RCC) was calculated by [3] according to Eq. (11), where size (di) is the size of the datum replica, and B_{up} is the bandwidth in time that is data transferred.

$$RCC = \sum_{i=1}^{n} \frac{size\,(d_i)}{B_{up}} \times tn \tag{11}$$

The consumed energy (EC) is calculated by [29,30] according to Eq. (12), where $e_i^c(k)$ and $e_i^g(k)$ are the power consumption by collecting and aggregating data of size k, respectively, while $e_i^s(k)$ denotes the energy consumption by sending data with size k between the sensing node and fog node or between the fog node and fog node.

$$EC_i = \sum_{i=1}^{n} 2 \times (e_i^c(size\,(d_i)) + e_i^g(size\,(d_i))\ e_i^s(size\,(d_i)) \tag{12}$$

To calculate the total data access cost of DAC, Eq. 13 is used according to Eq. (10), Eq. (11), and Eq. (12) and the data transfer cost equation. Thus, we have the following equation:

$$DAC = ST + RCC + EC \tag{13}$$

4.1.3. Availability of datums

This metric is used by the current work to measure the ability of applications to satisfy and reach the requested datum replica on fog nodes. The percentage of unsatisfied requests can be used as an indicator of the availability of data, which also depends on the reduced number of replicas. Therefore, this work adds a deadline property to all request transactions and counts the deadline that they miss without completing. Equation (14) calculates the availability rate:

$$DataAvailabilty = \sum_{i=1}^{n} \frac{x}{n} \tag{14}$$

where × denotes the number of missing deadline transactions (read requests), and n is the total number of read transactions.

4.2. Results and discussion

4.2.1. Average latency cost

Fig. 5 (a) and (b) display the system latency of iFogRep compared to the mentioned related works (Table 3). The configurations include 5 and 10 fog nodes with relaxed consistency and different workloads. It is observed that as access frequencies increase, the latency increases for all works, and the outperforming of iFogRep is clearly noticed with high access frequencies. iFogRep can reduce the system latency by 16% compared to the work by [3] and by 22% compared to work by [27], 12% from the work by [1], and 13% from [22]. By increasing the number of fog nodes, as presented in Fig. 5 (b), it can be observed that the system latency remains approximately unchanged. In fact, the outperformance of iFogRep is due to the allocating of sensing devices to their optimal fog node with

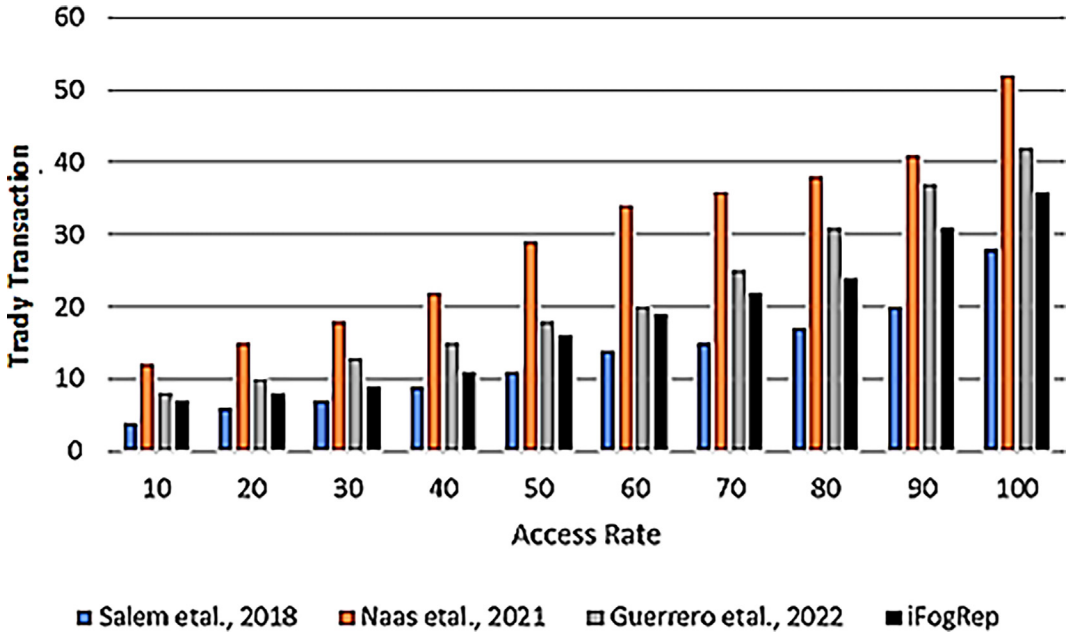


Fig. 7. Availability of iFogRep and related works.

the lowest latency cost. Also, this work aims to reduce the transmission data based on the access pattern which result in minimizing the transmission delay cost.

4.2.2. Replica cost

iFogRep obtains the lowest number of replicas with both workloads. The results depicted in Fig. 6 show that iFogRep outperforms their solutions. As presented in Fig. 6 (b), the replica cost of data increases with an increasing number of fog nodes and workloads. Selecting the optimal fog nodes for each sensing device in iFogRep reduces the energy consumption in addition to the data transfer costs. Additionally, identifying the consumers of each datum has a significant impact on reducing the number of replicas, consequently lowering the overall cost of replicas.

4.2.3. Availability of datums

Fig. 7 depicts the availability in terms of the missing or unsatisfied read requests to the total read requests. As stated by [1], using different workloads has no considerable impact on the number of tardy transactions; therefore, the results are presented with one workload. using Fig. 7. It is observed that:

The current work value of the missing transactions comes from all works except that by [6], which achieves the best availability results. This is because the work by [6] worked with a critical real-time system that concerns avoiding tardy transactions. The proportion of unsatisfied requests grows higher by increasing the access rate due to the increasing the rate of data locking. Reducing the access latency and transmission latency by iFogRep results in reducing the temporal consistency periods and consequently better data availability results.

5. Conclusion and future work

Replication is a technique that is used by many distributed systems to improve the availability where the replicas of updates propagated to many sites. However, this technique triggers a degree of temporal inconsistency that needs to be faced by the applications. Managing data replication and consistency in a Fog infrastructure for enhancing the availability and performance can be done by reducing the energy and latency costs. This work proposes a novel intelligent solution for dynamic consistent replication and optimal placement of IoT using fog computing. The proposed approach depends on three components, the **first** is for dynamic placement that is based on allocating each sensing device to the optimal fog node. It uses a machine learning technique with an optimization function to estimate the real latency between the sources and destinations. The second partial detached replication component acts to reduce the quantity of transmitted data and minimize the access latency. It uses an adaptive prediction technique to expect the ideal number of replicas of each datum and determines their locations dynamically. The role of consistency component is to determine the consistency level of each replica dynamically. It has the ability to adaptively change the consistency level of each replica according to the remaining network resources. It enables many nodes to pass their data simultaneously without any need for dispersed synchronization. The results show the ability of the current work to reduce the system latency and cost of replica access and increase the data availability. We plan to extend this work to identify the locations and actions that cause the unexpected undesired delay using the event mining technique to support the process model of this work support the data availability and consistency tolerance.

CRediT authorship contribution statement

Safa'a S. Saleh: Conceptualization, Validation, Formal analysis, Resources, Writing – review & editing, Visualization, Supervision, Project administration. **Iman Alansari:** Methodology, Software, Writing – review & editing, Supervision. **Mounira Kezadri Hamiaz:** Software, Validation, Formal analysis, Investigation, Data curation, Writing – review & editing, Visualization. **Waleed Ead:** Formal analysis, Investigation, Writing – original draft, Visualization. **Rana A. Tarabishi:** Methodology, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Hatem Khater:** Conceptualization, Validation, Investigation, Resources, Data curation, Writing – review & editing, Visualization, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Naas MI, Lemarchand L, Raipin P, Boukhobza J. IoT Data Replication and Consistency Management in Fog Computing. J Grid Computing 2021;19:33.
[2] Olga C, Chukhno N, Araniti G, Campolo C, Iera A, et al. Optimal placement of social digital twins in edge IoT networks. Sensors 2020;20(21):61–81.
[3] Taghizadeh J, Ghobaei-Arani M, Shahidinejad A. A metaheuristic-based data replica placement approach for data-intensive IoT applications in the fog computing environment. Softw: Pract Exper 2022;52(2):482–505. doi: https://doi.org/10.1002/spe.3032.
[4] Fahs AJ and Pierre G. Proximity-aware traffic routing in distributed fog computing platforms. Proceedings of the 2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID); May 2019:478-487; IEEE.
[5] Hasenburg J, Grambow M, Bermbach D Fbase: a replication service for data-intensive fog applications; 2019. arXiv preprint arXiv:1912.03107.
[6] Salem R, Saleh S, Abdul-Kader H. Intelligent Replication for Distributed Active Real-Time Databases Systems. The International Arab Journal of Information Technology May 2018;15(3).
[7] Saranya N, Geetha K, Rajan C. Data replication in mobile edge computing systems to reduce latency in Internet of Things. Wirel Pers Commun 2020;112(4):2643–62.
[8] Salem R, Saleh SS, Abdul-Kader H. Scalable Data-Oriented Replication with Flexible Consistency in Real-Time Data Systems. Data Sci J 2016;15(4):1–15. doi: https://doi.org/10.5334/dsj-2016-004.
[9] Guerrero C, Lera I, Juiz C. Optimization policy for file replica placement in fog domains. Concurr Comput Pract Exper 2020;32(21).
[10] Li C, Tang J, Luo Y. Scalable replica selection based on node service capability for improving data access performance in the edge computing environment. A. J Supercomput 2019;75(11).
[11] Li C, Bai J, Tang J. Joint optimization of data placement and scheduling for improving user experience in edge computing. J Parallel Distrib Comput B 2019;125:93–105.
[12] Guo J, Li C, Luo Y. Fast replica recovery and adaptive consistency preservation for edge cloud system. Soft Comput 2020;24:14943–64. doi: https://doi.org/10.1007/s00500-020-04847-2.
[13] Xie, J., Qian, C., Guo, D., Li, X., Shi, S., Chen, H.: Efficient data placement and retrieval services in edge computing. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pp. 1029–1039. https://doi.org/10.1109/ICDCS.2019.00106 (2019).
[14] Mayer, R., Gupta, H., Saurez, E., Ramachandran, U.: Fogstore:Toward a distributed data store for fog computing. arXiv:1709.07558 (2017).
[15] Huang T, Lin W, Li Y, He L, Peng S. A latency-aware multiple data replicas placement strategy for fog computing. J Signal Process Syst 2019;91(10):1191–204.
[16] Skarlat O, Nardelli M, Schulte S, Borkowski M, Leitner P. Optimized IoT service placement in the fog. SOCA 2017;11(4):427–43.
[17] Vales R, Moura J, Marinheiro R. Energy-aware and adaptive fog storage mechanism with data replication ruled by spatio-temporal content popularity. J Netw Comput Appl 2019;135:84–96.
[18] Monga SK, Ramachandra SK, Simmhan Y. ElfStore: a resilient data storage service for federated edge and fog resources. Proceedings of the 2019 IEEE International onference on Web Services (ICWS); July 2019:336-345; IEEE.
[19] Karatas F, Korpeoglu I. Fog-Based Data Distribution Service (F-DAD) for Internet of Things (IoT) applications. Futur Gener Comput Syst 2019;93. doi: https://doi.org/10.1016/j.future.2018.10.039.

[20] Aral A, Ovatman T. A decentralized replica placement algorithm for edge computing. IEEE Trans Netw Serv Manag 2018;15(2):516–29. doi: https://doi.org/10.1109/TNSM.2017. 2788945.
[21] Shao Y, Li C, Tang H. A data replica placement strategy for IoT workflows in collaborative edge and cloud environments. Comput Netw 2019;148:46–59.
[22] Dhande, Rahul Dynamic Replica Management in Fog-enabled IoT using Enhanced Data Mining Technique. Masters thesis, Dublin, National College of Ireland, 2020.
[23] Youse_ankalareh A, Najari A, Hosseynzadeh M, "Tree-based routing protocol in wireless sensor networks using optimization algorithm batch particles with a mobile sink," in Proc. IEEE 17th Int. Conf. Smart Communities, Improving Qual. Life ICT, IoT AI, Dec. 2020, pp. 1_5, doi: 10.1109/HONET50430.2020.9322844.
[24] Ding J, Liu H, Yang LT, Yao T, Zuo W. Multiuser multivariate multiorder Markov-based multimodal user mobility pattern prediction. IEEE Internet Things J 2020;7(5):4519–31.
[25] Wang H, Li Y, Jin D, Han Z. Attentional Markov model for human mobility prediction. IEEE J Sel Areas Commun Jul. 2021;39(7). doi: https://doi.org/10.1109/JSAC.2021.3078499.
[26] Montoya GA, Lozano-Garzon C, Donoso Y. Energy-Efficient and Delay Sensitive Routing Paths Using Mobility Prediction in Mobile WSN: Mathematical Optimization, Markov Chains, and Deep Learning Approaches. IEEE Access 2021;9:153382–400. doi: https://doi.org/10.1109/ACCESS.2021.3124737.
[27] Shankar BP, Chitra S. Optimal Data Placement and Replication Approach for SIoT with Edge. Comput Syst Sci Eng 2022;41:661–76.
[28] Amirkhalili YS, Aghsami A, Jolai F. Comparison of Time Series ARIMA Model and Support Vector Regression. Int J Hybrid Inform Technol, GV Press 2020;13 (1):7–18.
[29] Saleh SS, Mabrouk TF, Tarabishi RA, An improved energy-efficient head election protocol for clustering techniques of wireless sensor network, Egyptian Informatics Journal 22 (2021) 439–445, ScienceDirect.
[30] Khater HA, Baith Mohamed A, Kamel SM. A Proposed Technique for Software Development Risks Identification by using FTA Model, International Journal of Computer and Information Engineering, World Academy of Science. Eng Technol 2013;73(1):105–11.
[31] EL-Geneedy M, Moustafa H-D, Khalifa F, Khater H, AbdElhalim E. An MRI-based deep learning approach for accurate detection of Alzheimer's disease. Alex Eng J 2023;63:211–21.
[32] Mostafa MZ, Khater HA, Rizk MR, Bahasan AM. GPS/DVL/MEMS-INS smartphone sensors integrated method to enhance USV navigation system based on adaptive DSFCF. IET Radar Sonar Navig 2019;13(10):1616–27.
[33] Mi X, Li Q. The Mining Algorithm of Maximum Frequent Itemsets Based on Frequent Pattern Tree. Comput Intell Neurosci 2022;2022:1–11.
[34] Abdelmoez W, Khater H, El-shoafy N, "Comparing maintainability evolution of object-oriented and aspect-oriented software product lines," 2012 8th International Conference on Informatics and Systems (INFOS), Giza, Egypt, 2012, pp. SE-53-SE-60.
[35] Mahmud R, Pallewatta S, Goudarzi M, Buyya R. iFogSim2: An extended iFogSim simulator for mobility, clustering, and microservice management in edge and fog computing environments. J Syst Softw 2022;190:111351.
[36] Gupta H, Vahid A, Dastjerdi, Ghosh SK, Buyya R. iFogSim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments. Software: Practice Experience 2017;47(9):1275–96.