

RESEARCH

Open Access



LOGO: an efficient local and global data collection mechanism for remote underwater monitoring

Hussein Baalbaki¹, Hassan Harb^{1*} , Ameer Sardar Kwekha Rashid², Ali Jaber¹, Chady Abou Jaoude³, Chamseddine Zaki⁴ and Kifah Tout¹

*Correspondence:
hassan.harb.1@ul.edu.lb
¹ Computer Science
Department, Faculty
of Sciences, Lebanese
University, Beirut, Lebanon
Full list of author information
is available at the end of the
article

Abstract

The oceans play an important role in our daily life and they form the lungs of our planet. Subsequently, the world ocean provides so many benefits for humans and the planet including oxygen production, climate regulation, transportation, recreation, food, medicine, economic, etc. However, the oceans suffer nowadays from several challenges ranging from pollution to climate change and destruction of underwater habitat. Hence, the use of remote sensing technologies, like sensor networks and IoT, is becoming essential in order to continuously monitor the wide underwater areas and oceans. Unfortunately, the limited battery power constitutes one of the major challenges and limitations of such technologies. In this paper, we propose an efficient LOcal and GLObal data collection mechanism, called LOGO, that aims to conserve the energy in remote sensing applications. LOGO is based on the cluster scheme and works on two network stages: local and global. The local stage is at the sensor node and aims to reduce its data transmission by eliminating on-period and in-period data redundancies. The global stage is at the autonomous underwater vehicle (AUV) level and aims to minimize the data redundancy among neighboring nodes based on a spatial-temporal node correlation and Kempe's graph techniques. The simulation results on real underwater data confirm that LOGO mechanism is less energy consumption with high data accuracy than the existing techniques.

Keywords: Remote underwater monitoring, Data collection, Energy conservation, Spatial-Temporal node correlation, Kempe's graph coloring, Argo data

1 Introduction

Since the old, the oceans have been taken a lot of attention from human as they cover more than the three fourth of the earth surface. According to the United Nations report for oceans [1], 37% of the global population are living in coastal areas, between USA \$ 3–6 trillion/year is the estimated ocean-economy and 2900 million tonnes of oil are transported every year by sea. Unfortunately, over the last two decades, the marine life has facing an increasing number of challenges including marine debris, oil spills, loss of biodiversity, ice melting in polar regions, sea level rise, extreme weather events, displacement, etc. Hence, the importance of monitoring the ocean activities, such as water

sampling and oceanographic data collection, appears. This monitoring will allow experts to better understand the marine life, help in preserving the natural resources by tracking the pollution and early notify of marine disasters.

1.1 Problem statement

Recently, the ocean monitoring has taken a great attention from researchers and communities thanks to the rapid development in remote sensing technologies. Indeed, such technologies mainly consist of acoustic sensor networks and IoT, mostly referred as acoustic underwater IoT (AUIoT), that allow users to collect detailed information about the oceans in a real-time manner. Basically, AUIoT consists of a set of acoustic sensors and vehicles that are deployed in a wide underwater areas and collecting data about salinity, pressure, temperature, speed of current flow, etc. Then, the collected data are sent to a sink, mostly a navigator, located on the water surface which, in its turn, forward the data to the offshore station for a later analysis and decision making. Subsequently, the selection of the acoustic communication is due to the multi-path propagation and the strong signal attenuation in underwater environments. However, the AUIoT networks provide much more challenges compared to terrestrial ones: (1) the densely deployment of sensors because of the wide ocean surfaces. (2) The big data collection resulted from the periodic monitoring of the oceans. (3) The energy consumption of acoustic communications is very high and it proportionally increases with the increase in the amount of data transmission and the distance to the sink. (4) The acoustic transmission has a small bandwidth with less reliability and quality of data. Therefore, integrating new data reduction and redundancy elimination techniques becomes an essential task for researchers in order to save the AUIoT energy and extend the network lifetime.

1.2 Our contribution

In order to overcome the acoustic challenges, we propose a new data collection mechanism called LOGO that aims to minimize the data transmission in AUIoT networks and enhance their lifetime. LOGO relies on the cluster architecture and consists in two stages: local and global. Thus, the contribution of this paper is described as follows:

- At the local stage, LOGO allows each sensor to eliminate the redundancy between its data collected within each period, e.g. on-period, and among successive periods, e.g. in-period. Subsequently, LOGO proposes a state-based model that searches the similarities among collected readings then removes on-period data redundancy. While, the in-period data redundancy is eliminated through an adaptive sensing model that reduces the data collection at each sensor according to the variation of the monitored condition.
- The global stage allows the AUV to search the similarities among neighboring nodes in order to reduce the data transmission to the sink. This stage is based on two main algorithms: spatial-temporal node correlation and Kempe's graph coloring.
- To assess the efficiency of LOGO, we conducted a set of simulations based on real underwater data collected by Argo project and the results are compared to other existing techniques.

1.3 Paper structure

The rest of the paper is organized as follows. Section 2 reviews the state of the art on the data reduction techniques in AUIoT. Section 3 describes the cluster based underwater architecture considered in our mechanism. Sections 3.3 and 3.4 detail the data reduction models proposed at the local and global stages, respectively, in our mechanism. Section 4 reports the performance indicators and the obtained results. Finally, we conclude the paper and provide directions for future work in Sect. 5.

2 Related works

Over the last two decades, underwater monitoring takes great attention from both industries and the community. On one hand, the industries try to integrate new technologies to help in the ocean monitoring and discovering the significant wealth lies underwater. On the other hand, researchers aim to propose energy-efficient data collection techniques as well as try to overcome the challenges provided in AUIoT. In [2, 3], the authors give an overview on the remote sensing devices and IoT components fabricated by the industries and dedicated to underwater applications. While, the authors of [4, 5] summarize the data reduction and energy conservation algorithms proposed by the research community for the AUIoT applications.

Some works such as [6–11] are dedicated to conserve the network energy by proposing efficient cluster schemes and routing protocols. In [6], an energy-efficient adaptive clustering routing algorithm (ACUN) for AUIoT has been proposed. ACUN aims to optimize the lifetime of the cluster-heads by integrating a selection method based on the distance between CHs and the sink, the residual energy of the CHs and the size of the competitive radius. Accordingly, ACUN adopts a set of routing rules, either single-hop or multi-hop, in order to balance the energies of the nodes. The authors of [7] introduce a fuzzy clustering scheme based on particle swarm optimization (PSO) that increases the AUIoT network lifetime. The proposed scheme designs a fitness function to select the CHs of clusters based on the remaining node energies and the communication range between nodes-CHs and CHs-sink. In [8], the authors propose a routing protocol called DVOR, e.g. distance-vector opportunistic routing, in order to reduce the data communication in AUIoT networks. Based on the distance vectors established by a query mechanism, DVOR achieves opportunistic forwarding by selecting the smallest hop counts towards the sink while avoiding communication void and long detour. The authors of [12] propose an energy-efficient cluster technique called FCMMFO that is based on fuzzy C-means (FCM) and moth-flame optimization method (MFO). Initially, FCMMFO determines the optima number of clusters according to elbow method, then it divides the networks into clusters based on FCM. Furthermore, an objective function based on MFO is defined in order to find the optimal locations of the CHs.

Some works such as [13–15] propose efficient data collection mechanisms using autonomous underwater vehicle (AUV). In [13], the authors propose an AUV-aided solution called a prediction-based delay optimization data collection (PDO-DC) algorithm aiming to reduce the data collection delay in AUIoT. First, PDO-DC uses a machine learning technique called Kernel Ridge Regression (KRR) in order to build and update the prediction to fit the collected data. Then, it proposes an AUV path

planning strategy based on the competition coefficient in order to reduce the number of visited nodes when collecting the data and thus, reduce the collection delay and avoid the packet loss. The authors of [14] propose a district partition-based data collection algorithm with event dynamic competition in AUIoT. The proposed algorithm defines a metric called value of information (VoI) that determines the priority of the packet transmitted from each node. Then, the whole network is divided into subregions and an Q learning algorithm of reinforcement learning is proposed in order to determine the path of the AUV in each subregion. The authors of [16] propose a deployment-based system called AUV-OMN, that combines AUV and ocean monitoring network (OMN), for underwater data collection. First, AUV-OMN models sink nodes and AUV as a mixed integer optimization problem. Accordingly, a deployment scheme is designed to solve the formulated problem with the help of time division and NOMA access schemes.

The authors of [17–22] propose data collection mechanisms based on the statistical models and data reduction approaches. In [17], the authors propose two level transmission efficient mechanism dedicated to AUIoT based on clustering scheme. The first level is executed at the sensors and aims to filter its periodic data collected using a data aggregation method. At the second level, an enhanced version of Kmeans adapted to ANalysis Of VAriance (ANOVA) with three statistical tests has been proposed. The authors of [18] propose a Sequential Lossless Entropy Compression (S-LEC) which organizes the alphabet of integer residues obtained from differential predictor into increased size groups. S-LEC codeword consists of two parts: the entropy code specifying the group and the binary code representing the index in the group. In [19], the authors are dedicated to reduce the data transmission at the CH under a cluster-based underwater network. The proposed technique uses two distance functions, e.g. Euclidean and Cosine, in order to search the data correlation among neighboring nodes, thus removing the data redundancy, before sending the data to the sink. The authors of [23] propose a structure fidelity data collection (SFDC) technique dedicated to the cluster-based periodic applications in sensor networks. SFDC searches both spatial and temporal correlation between nodes, using distance functions and similarity metrics, respectively. Then, it exploits the dependencies to reduce the number of nodes required to work for sampling and data transmission and prove that such reduction is bound to save energy.

Table 1 shows a summarization for each proposed technique in terms of the used technique, the implemented features, and the objective of the proposal. Indeed, despite the existing approaches offer good solutions for AUIoT, they mostly suffer from several disadvantages: 1) Most of them are fairly complex, and difficult to implement efficiently due to the limited computational resources of most sensors. 2) Some of them compromise some aspects of sensed data such as temporal information for the sake of energy saving. 3) Most focus on handling data at only one level of the network (e.g. sensor or CH). In this paper, we tackle a new trend with decreasing data transmission and saving sensor energy in the AUIoT and remote sensing technology. Our proposed mechanism consists of a set of low complexity techniques that eliminate data redundancy on two levels of the network and on several phases: on-period, in-period and in-node.

Table 1 Summarize of state-of-the-art techniques

| Reference | Year | Technique | Features | Architecture | Objective |
|------------------|------|-----------------------|---|---------------------|--|
| [19] | 2013 | Data transmission | Distance functions | Clustering | Data reduction |
| [18] | 2014 | S-LEC | Compression | Distributed | Data reduction |
| [23] | 2015 | SFDC | Distance and similarity functions | Clustering | Data reduction |
| [17] | 2015 | Data transmission | Aggregation, Kmeans ANOVA | Clustering | Energy conservation, information integrity |
| [6] | 2019 | ACUN | Routing set | Clustering | Increase network lifetime |
| [7] | 2019 | Fuzzy clustering, PSO | CH selection | Clustering | Increase network lifetime |
| [8] | 2019 | DVOR | Query mechanism | Distributed routing | Reduce data communication |
| [13] | 2019 | PDO-DC | KRR, path planning | AUV-based | Reduce collection delay, avoid packet loss |
| [14] | 2019 | Data collection | Vol, path planning | Subregions | Determining best AUV path |
| [12] | 2020 | FCMMFO | FCM, MFO | Clustering | Increase network lifetime |
| [16] | 2020 | UAV-OMN | Optimization | AUV-based | Increase network lifetime |
| <i>This work</i> | 2021 | LOGO | Data similarity, sensing adaptation, node correlation | AUV + Clustering | Data reduction, energy conservation, information integrity |

3 Methods

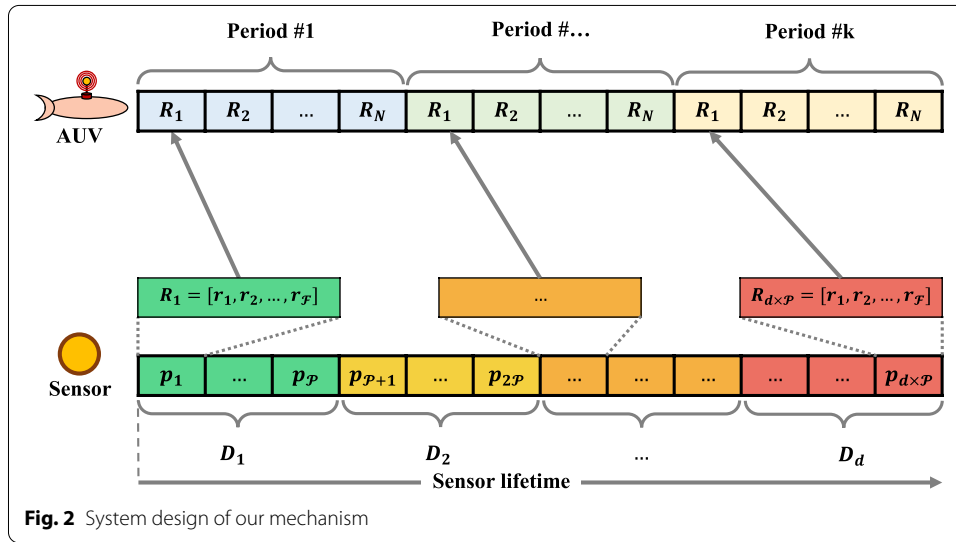
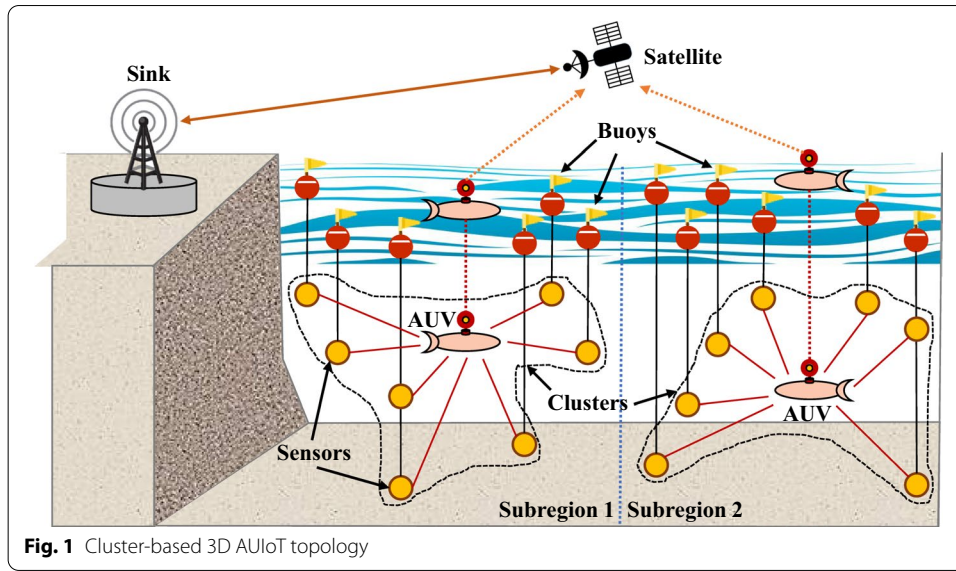
In this section, we first present the network topology and the sensing collection model that we adapted in our mechanism. Then, we detail each of the local and global data collection phases in our mechanism applied respectively at sensor and AUV nodes.

3.1 Cluster-based AUIoT topology

In this paper, we consider a 3D cluster-based AUIoT topology in order to apply our mechanism (Fig. 1). In such topology, the sensors are scattered in the ocean in a 3D plane where each sensor has a fixed position and depth after its deployment. Then, the monitored zone is divided into subregions in which the sensors in each subregion form a cluster. Accordingly, an AUV is assigned for each cluster and is responsible to periodically collect the data from the sensors before moving up to the ocean surface and forward the data towards the sink, through a satellite communication. For the sake of simplicity, the sensors use the single-hop communication over acoustic channels during the data collection while the AUV uses the radio frequency during the transmission of data to the satellite.

3.2 Periodic sensing monitoring

After their deployment and localization, the sensors start to monitor the target zone and update the sink with the desired information. Unfortunately, the data transmission is a highly cost operation in term of energy consumption [24, 25]. Thus, taking its limited energy power, the lifetime of the sensor will drastically decrease if all the collected data are sent to the AUV. Hence, periodic sensing acquisition model has been



introduced in AUIoT applications with the aim to reduce the amount of collected and transmitted data from the sensors. Basically, in a periodic sensing monitoring, data are collected on a periodic basis where each period p is partitioned into time slots. At each slot t , each sensor node N_i captures a new reading r_i then it forms, at the end of p , a vector of \mathcal{F} readings as follows: $R_i = \{r_1, r_2, \dots, r_{\mathcal{F}}\}$. After that, the sensor will send its vector of data, e.g. R_i , to its appropriate AUV. For analysis purposes, we assume, in our mechanism, that the lifetime of each sensor is divided into a set of rounds, $\mathcal{D} = [D_1, D_2, \dots, D_d]$, where each round $D_i \in \mathcal{D}$ consists of a set of \mathcal{P} periods. Figure 2 shows the system design of our mechanism in which the lifetime of each sensor is divided into rounds and the AUV receives a set of N reading sets coming from all its cluster members at the end of each period p .

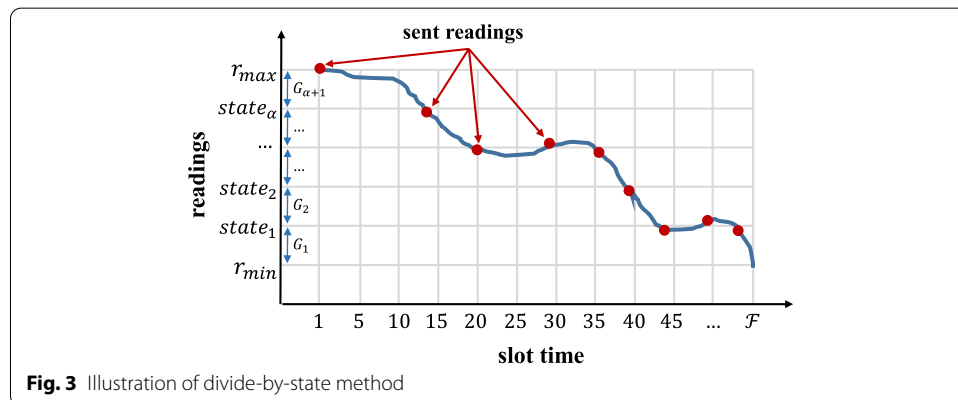
3.3 Local data collection at sensor

Obviously, the use of periodic monitoring model is a fundamental operation in AUIoT in order to ensure a reliable data collection and allow a better understanding of the underwater phenomena. However, this model produces a huge amount of data collection that consume most of the sensor energy during their transmission. Furthermore, replacing or recharging the sensor batteries is a very complicated mission and it is of high cost. Therefore, in order to conserve its energy and prolong its lifetime, the amount of data transmission from each sensor should be reduced without a loss of the collected information. This can be performed by eliminating the redundancy among the collected data either within the same period, e.g. on-period, or among periods in the same round, e.g. in-period. At the local stage, we propose two energy-efficient methods that allow to remove the data redundancy in both on-period and in-period on each sensor.

3.3.1 On-period data reduction method

Mostly, on-period redundancy happens due to the slow variation of the monitored condition or when a small value is assigned to the slot time. This leads to increase the similarity among readings in R_i and, consequently, it increases the redundancy among the data transmitted from the sensor. Thus, in order to overcome the on-period redundancy problem, we propose a divide-by-state method that allows to decrease the size of R_i by searching the similarities among successive readings.

Basically, the divide-by-state method is illustrated in Fig. 3. The blue curve represents the set of readings, e.g. R_i , collected during a period of size \mathcal{F} ; r_{min} and r_{max} indicate the minimum and maximum reading values in R_i . Then, we define a threshold α that allows to divide the readings in R_i into a set of reading ranges. Subsequently, the number of ranges equals to $(\alpha + 1)$ and the set of ranges is indicated as $\mathcal{G} = \{G_1, G_2, \dots, G_{\alpha+1}\}$; the lower bound of the first range G_1 is r_{min} while the upper bound of the last range $G_{\alpha+1}$ is r_{max} . In our method, the readings belong to the same range are considered redundant. Thus, if two successive readings r_i and r_j belong to a range G_k then the first reading is only sent while the last one is removed. However, in order to maintain the integrity of the information, we define a variable called occurrence, termed as $\mathcal{O}(r_i)$, that counts the number of similar readings according to a reading r_i . Hence, our method will transform the initial reading set R_i to a reduced set R'_i in the form $\{(r_1, \mathcal{O}(r_1)), (r_2, \mathcal{O}(r_2)), \dots, (r_k, \mathcal{O}(r_k))\}$; r_i represents the first reading in each range and $\mathcal{O}(r_i)$ is its occurrence to the next reading



state. It is important to notice that the accuracy of R'_i can be regulated by the expert according the value assigned to α which can be in $[1, \mathcal{F}]$; 1 means that all readings are belonging to the same range $[r_{min}, r_{max}]$ while \mathcal{F} indicates that all the collected data will send to the sink. Thus, more the value of α is, more the accuracy of sent data is.

Mathematically, Algorithm 1 shows the process of divide-by-state method which is applied locally on each sensor. The algorithm takes, as input, the on-period readings of a sensor along with the number of states and it returns the reduced set of readings R'_i . The process starts by adding the first reading, e.g. current reading, with occurrence of 1 to the set of final readings (lines 1-2). Then, for the next readings in R_i , they are added to R'_i if and only if their range is different from the current one, e.g. G_c (lines 4-9). Otherwise, the readings are considered to the previous one and they removed from R_i while adding the occurrence of the current one by 1 (lines 10–12).

Algorithm 1 Divide-by-State Algorithm.

Require: Set of readings: $R_i = \{r_1, r_2, \dots, r_{\mathcal{F}}\}$, Number of states: α .

Ensure: Set of sent readings: R'_i .

```

1:  $r_c = r_1$  //  $r_c$  is the current reading
2:  $R'_i \leftarrow \{(r_c, 1)\}$ 
3: consider  $r_c$  belongs to range  $G_c$ 
4: for each reading  $r_k \in R_i$  where  $k \geq 2$  do
5:   assume  $G_k$  is the corresponding range of  $r_k$ 
6:   if  $G_k \neq G_c$  then
7:      $R'_i \leftarrow R'_i \cup \{(r_k, 1)\}$ 
8:      $r_c = r_k$ 
9:      $G_c = G_k$ 
10:  else
11:     $\mathcal{O}(r_c) = \mathcal{O}(r_c) + 1$ 
12:  end if
13: end for
14: return  $R'_i$ 

```

3.3.2 In-period data reduction method

Obviously, the redundancy level among the collected data is highly dependent on the variation of the monitored condition. For instance, the monitoring of salinity or temperature of an ocean will produce a high redundancy level because such conditions are slowly varying during the progress of periods. Thus, in-period data redundancy should be also eliminated in order to further conserve the sensor energy. In this section, we propose an adaptive sensing model that studies the data collected by a sensor during a round (Fig. 2) then adapts its sensing frequency according to the condition variation and its remaining battery level.

3.3.2.1 Condition Variation Study

At the end of each round, the sensor searches for the variation between readings in order to determine the dynamicity of the condition. Subsequently, if the variation is low then the condition is stable and the sensor must decrease its sampling frequency to avoid collecting redundant data, and vice versa. The condition variation is calculated according to the number of state change during the whole round; more the number of state change is more the monitored condition is dynamic. Obviously, the number of the state change will be less than the sum of the state change on each

period of the round because the readings collected during the end of a period and the begging of the next one can belong to the same range. In our method, the condition dynamicity, $\mathcal{C}(D_i)$, during a round $D_i \in \mathcal{D}$ is calculated according to the following equation:

$$\mathcal{C}(D_i) = \frac{|\text{Divide-by-State}(D_i, \alpha)|}{\mathcal{P} \times \mathcal{F}} \times 100 \quad (1)$$

where $|\text{Divide-by-State}(D_i, \alpha)|$ is the number of state change obtained by applying Divide-by-State algorithm on the whole round. $\mathcal{C}(D_i)$ will take a value in $[1\%, 100\%]$ where 1% and 100% indicates no and full state change, respectively. Therefore, in order to assess the value of $\mathcal{C}(D_i)$, we first define two thresholds \mathcal{C}_{low} and \mathcal{C}_{high} , where $\mathcal{C}_{low} < \mathcal{C}_{high}$, then we distinguish between three levels of condition variation:

- $\mathcal{C}(D_i) \leq \mathcal{C}_{low}$ or *low variation*: this indicates that the monitored condition is slowly changing over the round which results in a high similarity among the collected data.
- $\mathcal{C}_{low} < \mathcal{C}(D_i) \leq \mathcal{C}_{high}$ or *medium variation*: this indicates that the monitored condition is constantly changing over the time which leads to a certain level of redundancy among the collected data.
- $\mathcal{C}(D_i) > \mathcal{C}_{high}$ or *high variation*: in which the monitored condition is quickly changing over the round.

3.3.2.2 Sensor Decision-Making for Sampling Adaptive

In addition to the condition dynamicity, the sensor takes into account another metric to adapt its sampling rate, e.g. the battery level. The idea is that when the battery level depletes less than a defined threshold then, the sensor must decrease its sampling rate in order to save its available energy but without affecting the accuracy of collected data. More formally, let consider that the initial energy of the sensor N_i prior to its deployment is E_i and its remaining energy at the end of a round D_i is E_r . Then, we define an energy threshold \mathcal{E} where the sensor energy becomes crucial if it reaches this threshold.

Table 2 shows the decision made by the sensor to adapt its sampling rate based on the condition variation and the battery level. Subsequently, the new sampling rate is determined according to two thresholds: $\mathcal{L}_{k \in L, M, H}$ and $\mathcal{H}_{k \in L, M, H}$ indicating low (e.g. $E_i \leq \mathcal{E}$) and high (e.g. $E_i > \mathcal{E}$) battery levels respectively. The letters L , M and H represent the condition variation if it is low, medium or high respectively, where $\mathcal{L}_L < \mathcal{L}_M < \mathcal{L}_H$ and $\mathcal{H}_L < \mathcal{H}_M < \mathcal{H}_H$. Indeed, the new sampling rates in Table 2 are customized by the experts and depending on the application requirements. However, the table customization should respect the following rules:

Table 2 Sampling rate decision table

| Battery level | | |
|---------------------|------------------------|---------------------|
| Condition variation | $E_i \leq \mathcal{E}$ | $E_i > \mathcal{E}$ |
| Low | \mathcal{L}_L | \mathcal{H}_L |
| Medium | \mathcal{L}_M | \mathcal{H}_M |
| High | \mathcal{L}_H | \mathcal{H}_H |

- The sensor must decrease its sampling rate when the condition variation and the battery are in low levels. This can reduce the redundancy among the collected data and save the sensor energy without degrading the accuracy of the information.
- The sensor must increase its sampling rate when a high condition variation is detected. This leads to increase the reliability of the collected information.

3.4 Global data collection at AUV

In AUIoT, the sensors are mostly scattered in a random way through aircraft or rocket due to the harsh or inaccessibility of most zones of oceans. This leads to a high level of spatial-temporal correlations between sensor nodes. Thus, when receiving the data sets from all sensors at the end of each period, the AUV can benefit from such correlations in order to eliminate the data redundancy among neighboring sensors, e.g. in-node data redundancy, before sending them towards the sink. Therefore, the periodic data transmitted by the AUV will be reduced which will save its energy and facilitate the data analysis task of the end user. Let first assume that the AUV will periodically receive a set of data sets $\mathcal{R}' = \{R'_1, R'_2, \dots, R'_n\}$, where n is the total number of sensors. Then, we propose an energy-efficient data reduction mechanism that allows the AUV to : first, search the spatial-temporal correlation among neighboring sensors; second, select a set of data sets based on the Kempe's graph to send towards the sink instead of the whole datasets.

3.4.1 Spatial node correlation

Spatial node correlation indicates that two or more sensors are geographically close in which there is an overlapping between their sensing ranges. This is mostly leads to a certain level of redundancy among their collected data. First, we use the Monte-Carlo method to determine the overlapping between two sensors followed by the Jaccard coefficient to check if these sensors are spatial correlated or not.

3.4.1.1 Monte-Carlo (MC) Method

By definition, MC [26] is a special kind of computational algorithms that use the process of repeated random sampling to make numerical estimations of unknown parameters. In the literature, one can find a lot of applications for MC in various domains including finance and business, physical sciences, engineering, computer graphics, computational biology, etc.[27–29]. In order to find the spatial correlation among nodes, let assume that each sensor is defined by its position in the 3D plane and its sensing range as follows: $(P_{x,y,z}, S_r)$. Then, MC considers that the zone of interest is dividing into a fixed number of points distributed in equitable way in the plane with distance Δ between every two points. Every point has its coordinates and it is covering by a sensor if the distance between them is less than the sensing range of the sensor. Thus, we define the set \mathcal{V}_{N_i} containing all the points covered by the sensor N_i while $|\mathcal{V}_{N_i}|$ indicates the size of the set, e.g. the number of points.

Figure 4 shows an illustration of the MC method adapted in our mechanism to calculate the spatial correlation between nodes. Based on the set \mathcal{V}_{N_i} , we define the overlap sensing set, termed as \mathcal{V}_{N_i, N_j} , between two nodes N_i and N_j as the set of points in which their distance to both sensors is less than the sensing range S_r . Similarly, we assume that $|\mathcal{V}_{N_i, N_j}|$ is the size of the point set, e.g. the number of points in \mathcal{V}_{N_i, N_j} .

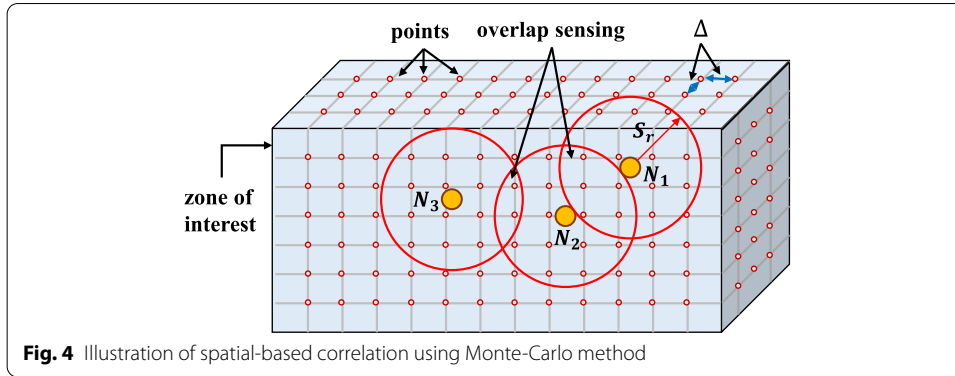


Fig. 4 Illustration of spatial-based correlation using Monte-Carlo method

Furthermore, more the size of point set \mathcal{V}_{N_i, N_j} is more the spatial correlation between nodes becomes. For instance, Fig. 4 shows that the spatial correlation between N_1 and N_2 is more than that between N_2 and N_3 because $|\mathcal{V}_{N_1, N_2}| = 5 > |\mathcal{V}_{N_2, N_3}| = 2$.

3.4.1.2 Jaccard Coefficient

After calculating the overlap sensing between every pair of nodes, the AUV uses the Jaccard function in order to determine the set of pairs having strong spatial correlation among others. Typically, the Jaccard coefficient is used for gauging the similarity and diversity between sample sets. Thus, it has been used in a wide range of application including community detection in social networks [30], document and web pages plagiarism [31], attack detection [32], market analysis [33], etc. In this paper, we use the Jaccard coefficient to measure the degree of spatial correlation, e.g. \mathcal{A} , between nodes. Thus, two nodes N_i and N_j are considered spatial correlated according to the Jaccard coefficient if and only if:

$$\mathcal{A}(N_i, N_j) = \frac{|\mathcal{V}_{N_i, N_j}|}{|\mathcal{V}_{N_i}| + |\mathcal{V}_{N_j}| - |\mathcal{V}_{N_i, N_j}|} \geq \mathcal{J} \quad (2)$$

where $\mathcal{J} \in [0, 1]$ is the Jaccard threshold defined by the application itself; 0 indicates that the overlap sensing does not contain any point while 1 means that all points covered by the first sensor are also covered by the second one.

3.4.2 Temporal node correlation

The aim of the temporal correlation is to measure the similarity among data collected by neighboring nodes. Thus, by exploiting such correlation, the AUV can reduce the periodic number of sets in \mathcal{R}' sent to the sink and avoid sending repeated data to the end user. However, the received sets have different lengths which make the calculation of similarities between data is not a trivial task. In our mechanism, we focus on the Dynamic Time Wrapping (DTW) to determine the pairs of temporal correlated nodes. By definition, DTW is a measure of distance functions that evaluate the difference between two time series. Thus, less the value of DTW is obtained more the similarity between the sets is noticed. Furthermore, DTW has been proven as a good indicator to measure the difference between data collected in various domains [30, 34, 35].

Indeed, our mechanism uses DTW in two steps in order to calculate the temporal node correlation between a pair of data sets R'_i and R'_j coming from the sensors N_i and N_j :

- *Cost matrix, \mathcal{M} , calculation*: the aim of this matrix is to reduce the overhead of the DTW distance calculation in the next step. \mathcal{M} has a dimension of *row* and *col* where *row* equals to the length of R'_i and *col* equals to the length of R'_j . Subsequently, $\mathcal{M}[r][c]$ represents the element at the row index $r \in [0, \text{row} - 1]$ and column index $c \in [0, \text{col} - 1]$. Then, the elements in \mathcal{M} can be calculated as follows:

$$\mathcal{M}[r][c] = \begin{cases} |r_{r+1_i} - r_{c+1_j}| & \text{if } r \neq 0 \text{ and } c \neq 0 \\ |r_{r+1_i} - r_{c+1_j}| + |r_{r+1_i} - r_{c_j}| & \text{if } r = 0 \text{ and } c \neq 0 \\ |r_{r+1_i} - r_{c+1_j}| + |r_{r_i} - r_{c+1_j}| & \text{if } r \neq 0 \text{ and } c = 0 \\ |r_{r+1_i} - r_{c+1_j}| + \min(|r_{r_i} - r_{c+1_j}|, |r_{r+1_i} - r_{c_j}|, |r_{r_i} - r_{c_j}|) & \text{otherwise} \end{cases} \quad (3)$$

where r_{k_i} (respectively, r_{k_j}) is the k^{th} reading in the set R'_i (respectively R'_j).

- *DTW distance computation*: after calculating the cost matrix, the AUV computes the DTW distance, referred as $DTWD(R'_i, R'_j)$, between both sets. Algorithm 2 shows the process of the distance computation by taking, as input, both reading sets with their calculated cost matrix. First, we indicate indexes on the last row and column of the cost matrix and the initial distance is set to the element in such indexes (lines 1-3). In addition, we define a variable called *iteration* that indicates the number of steps needed to reach the first element in the matrix. Then, while the first element is not reached, the algorithm finds the minimum value of the three elements preceding the current element and adds to the distance (lines 6-7). Subsequently, the current row and column are set to those of the minimum element while incrementing the iteration number (lines 8-10). Finally, the average distance in all steps is calculated and returned by the algorithm.

Algorithm 2 DTWD Distance Computation Algorithm.

Require: Reading set of N_i : $R'_i = \{r_1, r_2, \dots, r_k\}$, Reading set of N_j : $R'_j = \{r_1, r_2, \dots, r_q\}$,
Cost matrix: \mathcal{M} .

Ensure: DTW distance.

```

1: row =  $k - 1$ 
2: col =  $q - 1$ 
3: distance =  $\mathcal{M}[r][c]$ 
4: iteration = 1
5: while (row  $\neq 0$  and col  $\neq 0$ ) do
6:   element =  $\min(\mathcal{M}[\text{row}][\text{col} - 1], \mathcal{M}[\text{row} - 1][\text{col}], \mathcal{M}[\text{row} - 1][\text{col} - 1])$ 
7:   distance = distance + element
8:   row = row index of element
9:   col = column index of element
10:  iteration = iteration + 1
11: end while
12: distance = distance / iteration
13: return distance

```

Thus, based on the Algorithm 2, the AUV considers that both sensors are temporal correlated if the DTWD between them is less than a defined threshold \mathcal{W} as follows:

$$DTWD(R'_i, R'_j) = DTWD_Distance_Computation(R'_i, R'_j, \mathcal{M}) \leq \mathcal{W} \quad (4)$$

where \mathcal{W} is a threshold defined by the application.

3.4.3 Spatial-temporal node correlation

In this section, we integrate both spatial and temporal correlation at the AUV over the data sets received at each period. Thus, in order to remove the in-node data redundancy, the AUV finds the set of spatial-temporal correlated nodes for each sensor. Subsequently, two sensors N_i and N_j are considered spatial-temporal correlated if they are geographically close and they generate redundant data at the same time according to the following condition:

$$\mathcal{A}(N_i, N_j) \geq \mathcal{J} \quad \text{and} \quad DTWD(R'_i, R'_j) \leq \mathcal{W} \quad (5)$$

3.4.4 Final sets selection based on Kempe's method

After determining all pairs of spatial-temporal nodes, the AUV selects a set of sensors to send their data to the sink instead of sending the whole data sets. We use the Kempe's method which is a graph coloring strategy developed by Kenneth Appel and Wolfgang Haken in 1976. Kempe's method states that any graph, defined by its set of edges connected through a set of vertices, requires no more than a few colors—mostly four or five—to color its edges in a way that none of connected edges have the same colors. Thus, this method found its way into various fields such as graph theory (i.e. graph coloring), chemistry (i.e. connection between molecules), social network (i.e. user friendships), etc. [36, 37]. In this work, the sensor nodes represent the edges of the graph and the spatial-temporal correlations among nodes indicate the vertices; two sensors are connected only if they are spatial-temporal correlated. Therefore, the selection of the final sent sets based on the Kempe's method could be done according to the following steps:

- *Stack order*: this step pushes every node having less than β connected edges in a stack \mathcal{K} then removes the node from the graph. β is a value determined by the experts depending on the number of nodes in the graph. Then, this step will be repeated until no more nodes are on the graph.
- *Node coloring*: in this step, we pop out the nodes back to the graph one by one. Whenever a node is popped out, it is colored by a new color that is not used in its connected nodes. At the end of this step, all the graph nodes are coloring using the minimum number of colors while having no connected nodes carrying the same color.
- *Sets selection*: the AUV selects the set of nodes having the following characteristics to send their data sets to the sink: 1) they have the same color; 2) their number is greater than the other nodes with similar colors; 3) they have connections to all

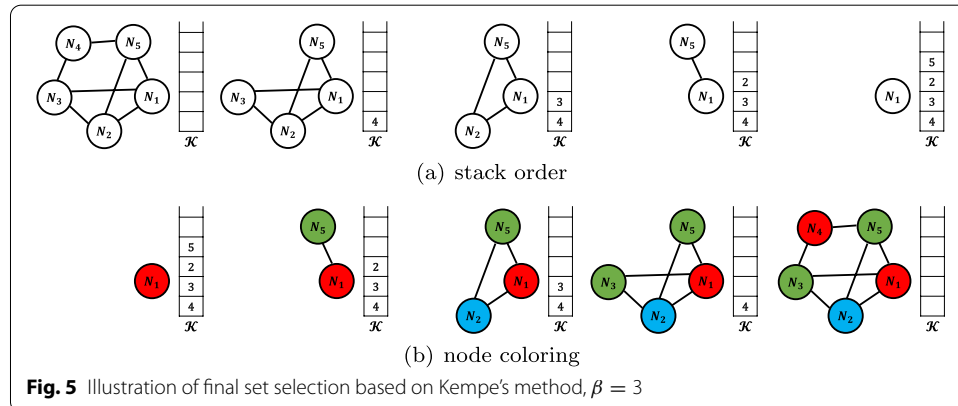
other nodes on the graph. This selection can help in reduce the in-node redundancy among neighboring nodes while keeping a high level of data accuracy.

Figure 5 shows an illustration example for the selection of the final sets based on the Kempe's method. We assume a graph of five connected nodes according to the spatial-temporal correlations between them and we set the Kempe's threshold β to 3. The stack order phase starts by selecting the node having less than 3 connections, e.g. N_4 , and push it on the stack while remove it from the graph. Similarly, N_3 is the next node to remove and push to the stack and this process continue until no more nodes are on the graph (last subfigure in Fig. 5a). After that, the node coloring phase starts by coloring the node N_1 in a random color, e.g. red. Then, the node N_5 popped out from the stack and it is colored by another color different from red, i.e. green. The process is repeated with the other nodes until no more nodes are on the stack. At the end of this phase, we obtain three sets of nodes with the same colors $\{N_1, N_4\}$, $\{N_3, N_5\}$ and $\{N_2\}$. Finally, the AUV randomly selects one of the first

and second set to send towards the sink because both meet the conditions of the sets selection phase.

4 Results and discussion

We used real underwater data collected from the Argo project [38] in order to evaluate the performance of our mechanism. The Argo project deploys more than 3600 nodes over the global oceans. Each node collects salinity, temperature and velocity readings in the upper 2000 meters of depth. In this work, we are interested in data for 120 nodes distributed in the Indian ocean over an area of $5000 \times 5000 \times 5000 m^3$. Then, the area is divided into two clusters: the first cluster contains 40 sensors with AUV₁ and the second cluster consists of 80 sensors with AUV₂. For the sake of simplicity, we focus, in our simulations, on the salinity readings collected by each node. We implemented our mechanism based on Java simulator and we compared the results to those obtained with the techniques proposed in [19], referred as EuDi, and SFDC in [23] used in AUIoT. Subsequently, we used a HPE laptop machine with a processor of 64-bit 8-core Intel i7-4800MQ CPU running at 2.7 GHz. In addition, the used RAM is 16 GB and the storage capacity is 512 GB HDD. The machine runs Windows 10. Table 3 shows the configuration of the parameters adapted in our simulation.



4.1 On-period data reduction study

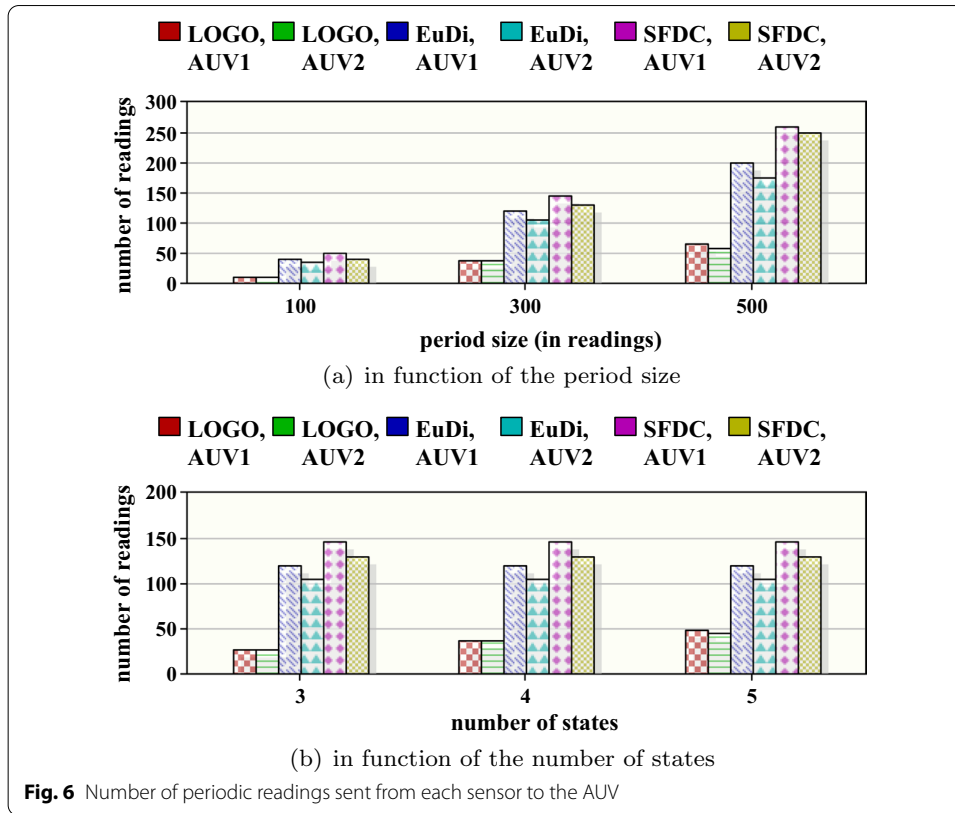
Figure 6 shows the relevance of on-period stage proposed in our mechanism in terms of reducing the number of periodic readings sent from each sensor to its AUV, compared to EuDi and SDFC. The obtained results are dependent on the period size (Fig. 6a) and the number of states (Fig. 6b). First, we show that each technique gives similar results in both clusters, AUV1 and AUV2, because they apply distributed reduction techniques at the sensor level. Indeed, LOGO outperforms EuDi and SDFC in reducing the amount of data transmission to the AUVs in all cases. For instance, LOGO reduce up to 78% and 73% of transmitted data in AUV1 and AUV2, respectively, compared to EuDi and reduce up to 81% and 78% of those in AUV1 and AUV2 compared to SFDC. Furthermore, the results of LOGO show the following observations: 1) the sensor sends less data when the number of states decreases. This is because the similarities among readings increase with the decrease in α . 2) The data transmission ratio from the sensor increases with the increase in the period size. This is because the transitions between states increase when \mathcal{F} increases thus the number of readings sent to the AUV will increase.

4.2 In-period data reduction study

Figure 7 shows the performance of in-period stage proposed in LOGO in terms of adapting the sampling rate of the sensor after each round. The results show that LOGO allows a sensor to dynamically adapt its sensing frequency according to the period size (Fig. 7a) and the number of states (Fig. 7b). Subsequently, the following observations are eminent: 1) the salinity readings in the Indian ocean are slowly varying and contains a high redundancy level; this can be clearly seen on the figures when the sensor adapts its sampling rate to the minimum compared to the original sensing frequency, e.g. period size. 2) During the last periods of its lifetime, the sensor collects less data than those during the first ones; this is due to the low battery level reached by the sensor thus, it decreases its sensing frequency in order to save more energy. 3) The sensor decreases its

Table 3 Simulation environment

| Parameter | Symbol | Values |
|----------------------------------|--|---|
| Period size | \mathcal{F} | 100, 300, 500 readings |
| Number of periods per rounds | \mathcal{P} | fixed to 2 |
| Number of states | α | $3\% \times \mathcal{F}$, $4\% \times \mathcal{F}$, $5\% \times \mathcal{F}$ |
| Condition variation thresholds | $\mathcal{C}_{low}, \mathcal{C}_{high}$ | 15%, 25% |
| Initial sensor energy | E_i | 5 mJ |
| Critical sensor energy threshold | \mathcal{E} | $E_i/2$ |
| Decision table thresholds | $\mathcal{L}_L, \mathcal{L}_M, \mathcal{L}_H$ $\mathcal{H}_L, \mathcal{H}_M, \mathcal{H}_H$ | 20%, 40%, 60% 40%, 60%, 100% |
| Sensor sensing range | S_r | 500, 700, 900 meters |
| MC point distance | Δ | 10 meters |
| Jaccard threshold | \mathcal{J} | 0.3, 0.4, 0.5 |
| DTWD threshold | \mathcal{W} | 20 |
| Kempe threshold | β | fixed to 4 |

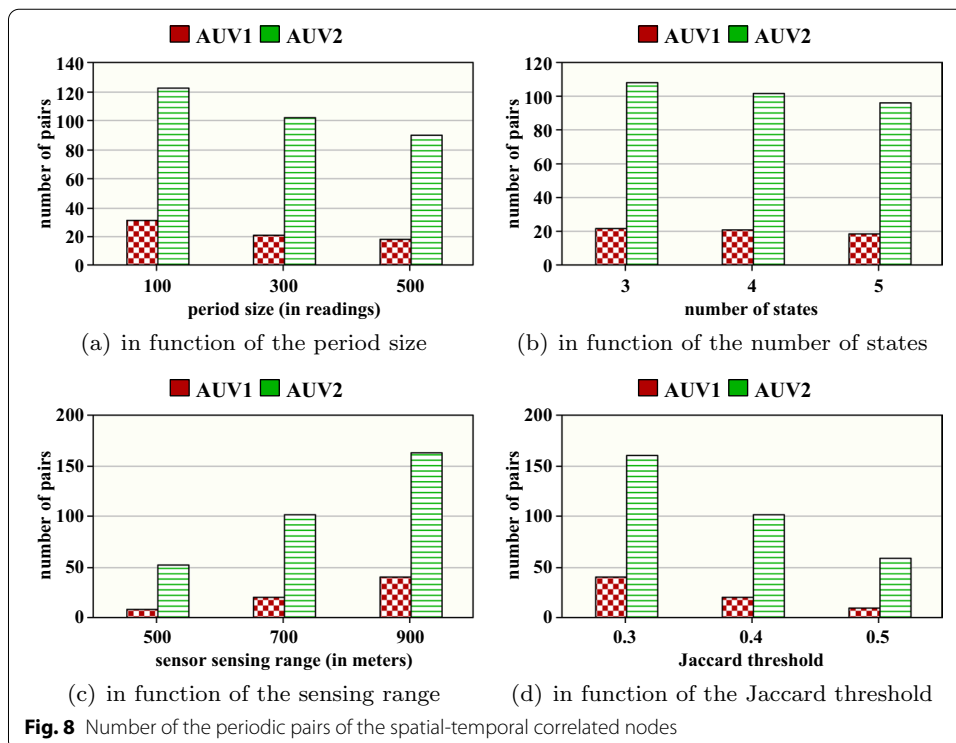
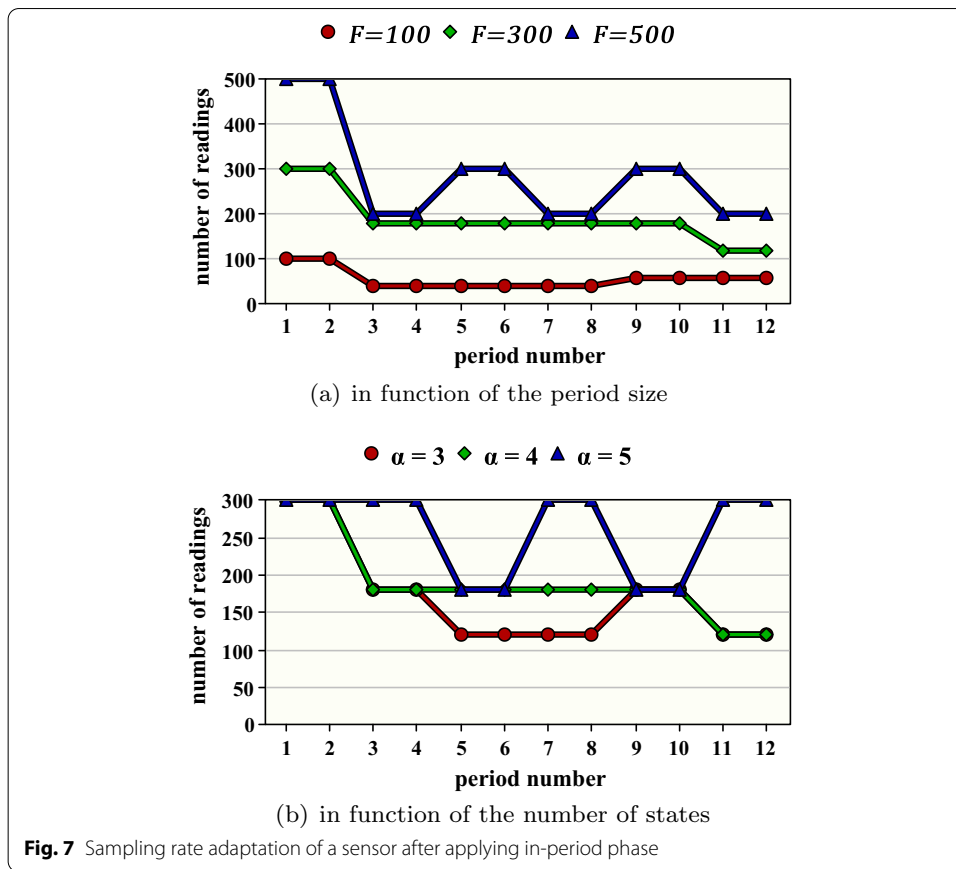


sampling rate when the period size or number of states decreases (for the same reasons mentioned in Fig. 6).

4.3 Spatial-temporal node correlation study

In this section, we study the performance of the spatial-temporal correlation method proposed in LOGO in terms of various values of tested parameters. Figure 8 shows the average number of spatial-temporal correlated pairs obtained at each period after applying our mechanism in each cluster. The obtained results confirm the behaviour of our mechanism in finding the correlated nodes which dependent on the number of nodes in the cluster. From on hand, the results show a significant number of obtained pairs in both clusters explaining the high level of redundancy existing between the collected data. On the other hand, the number of pairs increases with the increasing number of the cluster size. Furthermore, the results reveal the following observations:

- The number of correlated pairs increases when the size of the compared data sets decreases. This will happen if the period size (Fig. 8a) or the number of states (Fig. 8b) decrease.
- The number of pairs increases by increasing the spatial correlation between nodes. This can be done in two manners: first, by increasing the sensing range of the sensors (Fig. 8c), which increases the sensing overlap between nodes; second, by decreasing the Jaccard threshold (Fig. 8d) that makes more flexible the spatial correlation condition between sensors (see equation 4).

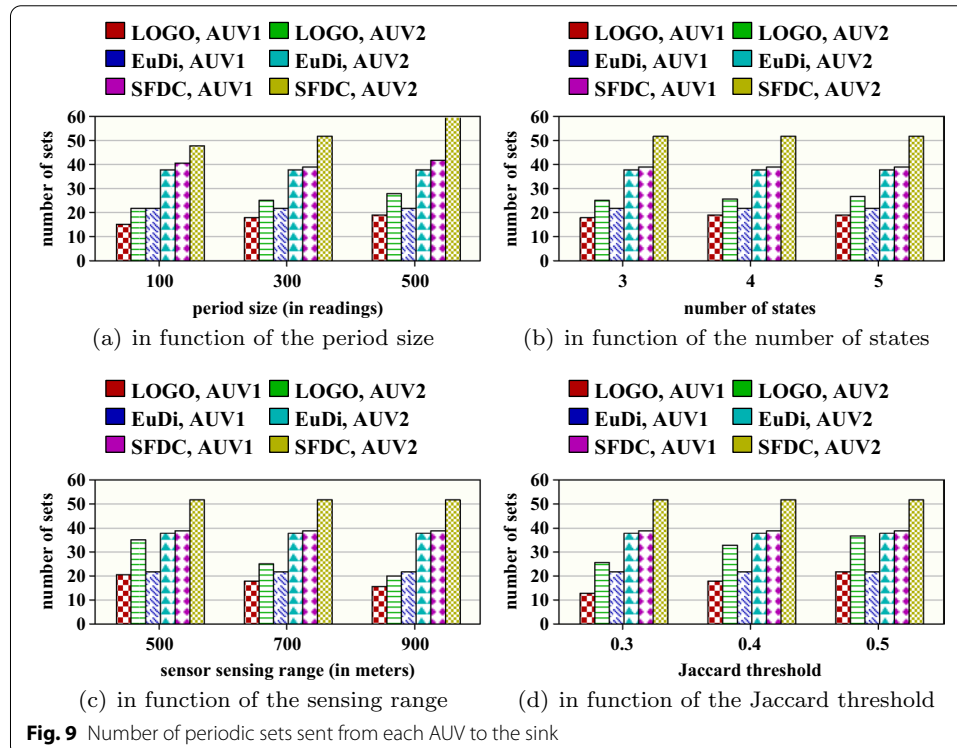


4.4 In-node data reduction study

In this section, we study the performance of the in-node phase proposed in LOGO in terms of reducing the number of sets sent to the sink after eliminating the correlated ones. Subsequently, Fig. 9 shows the number of periodic transmitted data sets after applying Kempe's graph method in our mechanism in comparison with EuDi and SFDC techniques. The results show that LOGO outperforms both EuDi and SFDC in terms of reducing the number of sets sent to the sink in all cases. Subsequently, it reduces up to 31% and 42% less sets in AUV1 and AUV2 respectively compared to EuDi and up to 37% and 46% less sets in AUV1 and AUV2 compared to SFDC. We can also show that LOGO remove more data sets when the number of correlated nodes increases. This confirms the relevance of our mechanism in eliminating more the redundancy when the similarity between data increases. Based on the results on Fig. 8, the following observations can be shown: 1) the AUV sends less number of data sets when the period size or the number of states decreases (Fig. 9a, b). 2) the AUV removes more redundant sets when the sensing range increases or the Jaccard threshold decreases (Fig. 9c, d).

4.5 Data accuracy study

In this section, our objective is to study the accuracy of the three phases proposed in LOGO: on-period, in-period and in-node. Figure 10 shows an illustration example of the on-period phase applied over 100 readings collected during a period from a random sensor with a state number fixed to 3. The green curve represents the raw



data collected by the sensor while the yellow curve shows the selected readings sent to the AUV. First, we observe that the readings are highly redundant where they range between $r_{min} = 35.63$ and $r_{max} = 36.29$. Then, thanks to on-period phase, the sensor selects a few set of readings indicating the change on the reading states to send to the AUV. This can highly preserves the integrity of the sent information while reducing the data transmission at the same time.

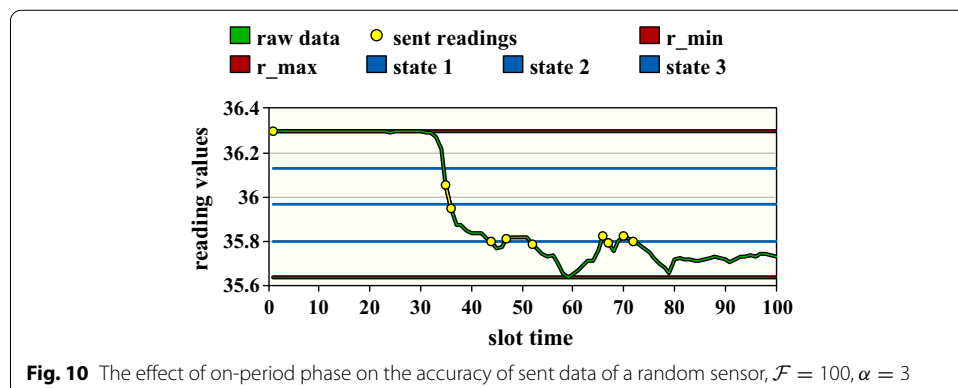
Figure 11 shows the accuracy the in-node phase applied at the AUV level after eliminating the redundant data sets, for a set of 12 periods. The data accuracy is calculated by dividing the number of missing readings over the whole readings received by the AUV during a period; a reading is indicated as missed reading if it appears in the sets received by the AUV but not in those sent to the sink. The results show that the missing readings range between 1% and 3.8% for AUV1 and between 1.9% and 5.2% for AUV2 which are negligible compared to the readings received by the sink. Thus, this percentage of missing readings will not affect the decision made by the experts.

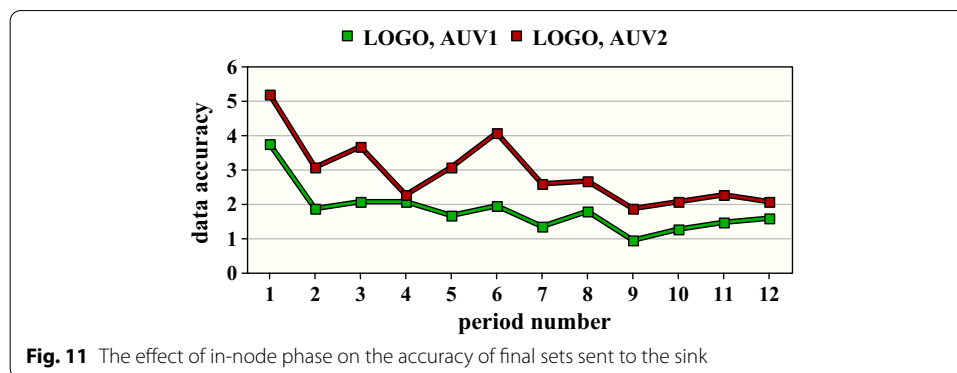
4.6 Further discussions

In this section, we give further consideration to our proposed mechanism while summarizing the obtained results of LOGO and analyzing its performance regarding several metrics and under various conditions.

From the data transmission reduction point of view, LOGO mechanism outperforms other techniques (especially EuDi and SFDC) in terms of reducing the amount of data collection and transmission (see Fig. 6). Subsequently, LOGO can reduce from 56% to 78% compared to EuDi and from 64% to 81% compared to SFDC, depending on the period size and the number of states. Consequently, when the priority for the application is the reduction of the amount of data transmission in order to make less complex the analysis of data at the sink side, the LOGO mechanism becomes more suitable.

From the energy consumption point of view, divide-by-state and sampling rate adaptation algorithms proposed at the node level along with the set selection process proposed at the AUV level can largely reduce the energy consumption due the huge reduction in the data collection and transmission compared to other techniques (see Figs. 6 and 9). Consequently, LOGO can extend the sensor lifetime and ensure a long





time monitoring of the observed condition, which is a must requirement in underwater monitoring.

From the data accuracy point of view, LOGO ensures a high level of data integrity at both sensor and AUV levels, without any loss of information. Subsequently, the entire data loss in LOGO does not exceed 5.2% which is negligible compared to the amount of data received by the sink. This is because LOGO allows, on one hand, the sensor to provide real-time information to the sink whenever the monitored condition is changed, e.g. new state is detected, as well as it allows CH to only eliminate useless data and sending useful information to the sink.

Indeed, the selection of the threshold values and the processing complexity of LOGO at the AUV level are two main challenges facing our mechanism. On one hand, selecting the appropriate values of thresholds is very essential in our mechanism which highly affects the results. Indeed, we believe that threshold values should be determined by the decision makers or experts depending on the monitored features (e.g. water conditions). On the other hand, the processing complexity of LOGO at the AUV level is highly dependent on the distance computation of DTWD (e.g. Algorithm 2) that affects the latency of data sent to the sink. Thus, it becomes essential to reduce the complexity of Algorithm 2 especially for critical monitored condition that require a fast delivery packet to the end-user. In order to overcome this problem, we believe that a pruning-based method should be inserted into Algorithm 2 thus, the number of pairs comparison will reduce (see Fig. 8) and the algorithm complexity as well.

5 Conclusion

Until nowadays and although the great advancements in technologies, we know very little about the oceans, estimated less than 5%. Therefore, ocean exploration will take more and more attention from researchers and communities aiming to make a deep understanding of biological, chemical, physical, geological and archaeological aspects of the ocean. This explains the huge investments made in the AUIoT as one of the most important technologies to discover the oceans. In this paper, we proposed a data collection mechanism called LOGO that aims to reduce the data transmission in AUIoT networks and enhance their lifetime. LOGO works in two levels, e.g. sensors and AUV,

and allows to remove the data redundancy existing in on-period, in-period and in-node phases based on a set of data reduction algorithms. To assess the efficiency of LOGO, we conducted a set of simulations based on real underwater data collected by Argo project and we compared the results to other existing techniques.

In the future work, we plan to enhance and extend LOGO in several ways. First, we plan to test our mechanism in real-case scenarios in order to validate its performance. Second, we seek to adapt our mechanism to take into account various types of underwater data like images for target detection, video for discovering operations, etc. Finally, we plan to extend our mechanism to multivariate data collection where each sensor can monitor several conditions (like salinity, temperature, pressure, etc.) at the same time.

Abbreviations

IoT: Internet of Things; AUV: Autonomous Underwater Vehicle; AUIoT: Acoustic Underwater IoT; ANOVA: ANalysis Of VAriance; MC: Monte-Carlo; DTW: Dynamic Time Wrapping.

Author contributions

HB carried out experiments, data analysis and drafted the manuscript. HH designed, coordinated and supervised this research. ASR participated in the formal analysis of this manuscript. AJ validated and visualized this manuscript. CAJ participated in the writing-review and editing this manuscript. CZ helped in the conceptualization and the methodology editing of this manuscript. KT validated the software and resources used in this manuscript. All authors read and approved the final manuscript.

Funding

The authors declare that they did not receive any type of funding for this work.

Availability of data and materials

The datasets generated and/or analysed during the current study are available in the ARGO repository, <https://argo.ucsd.edu/>.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

¹Computer Science Department, Faculty of Sciences, Lebanese University, Beirut, Lebanon. ²Business Information Technology, College of Administration and Economics, University of Sulaimani, Sulaimaniya, Iraq. ³TICKET Lab, Faculty of Engineering, Antonine University, Baabda, Lebanon. ⁴College of Engineering and Technology, American University of the Middle East, Egaila, Kuwait.

Received: 3 September 2021 Accepted: 27 December 2021

Published online: 15 January 2022

References

1. U. Nations, The ocean conference factsheet. *oceanconference.un.org*, 1–7, (2017)
2. M. Muzzammil, N. Ahmed, G. Qiao, I. Ullah, L. Wan, Fundamentals and advancements of magnetic field communication for underwater wireless sensor networks. *IEEE Trans. Antennas Propag.* 1–16 (2020)
3. R. Khalil, M. Babar, T. Jan, N. Saeed, Towards the internet of underwater things: recent developments and future challenges. *IEEE Consum. Electron. Mag.* 1–6 (2020)
4. N. Goyal, M. Dave, A.K. Verma, Data aggregation in underwater wireless sensor network: recent approaches and issues. *J. King Saud Univ.-Comput. Inf. Sci.* **31**(3), 275–286 (2019)
5. G. Khan, K.K. Gola, M. Dhinra, Efficient techniques for data aggregation in underwater sensor networks. *J. Electr. Syst.* **16**(1), 105–119 (2020)
6. Z. Wan, S. Liu, W. Ni, Z. Xu, An energy-efficient multi-level adaptive clustering routing algorithm for underwater wireless sensor networks. *Cluster Comput.* **22**(6), 14 651–14 660 (2019)
7. V. Krishnaswamy, S.S. Manvi, Fuzzy and PSO based clustering scheme in underwater acoustic sensor networks using energy and distance parameters. *Wirel. Pers. Commun.* **108**(3), 1529–1546 (2019)
8. Q. Guan, F. Ji, Y. Liu, H. Yu, W. Chen, Distance-vector-based opportunistic routing for underwater acoustic sensor networks. *IEEE Internet Things J.* **6**(2), 3831–3839 (2019)
9. E. Jiang, L. Wang, J. Wang, Decomposition-based multi-objective optimization for energy-aware distributed hybrid flow shop scheduling with multiprocessor tasks. *Tsinghua Sci. Technol.* **26**(5), 646–663 (2021)

10. R. Bi, Q. Liu, J. Ren, G. Tan, Utility aware offloading for mobile-edge computing. *Tsinghua Sci. Technol.* **26**(2), 239–250 (2020)
11. M. Mortada, A. Makhoul, C. Abou Jaoude, H. Harb, D. Laiymani, A distributed processing technique for sensor data applied to underwater sensor networks, in: 15th International Wireless Communications & Mobile Computing Conference (IWCMC). IEEE **2019**, 979–984 (2019)
12. W. Fei, B. Hexiang, L. Deyu, W. Jianjun, Energy-efficient clustering algorithm in underwater sensor networks based on fuzzy c means and moth-flame optimization method. *IEEE Access* **8**, 97 474–97 484 (2020)
13. G. Han, S. Shen, H. Wang, J. Jiang, M. Guizani, Prediction-based delay optimization data collection algorithm for underwater acoustic sensor networks. *IEEE Trans. Veh. Technol.* **68**(7), 6926–6936 (2019)
14. G. Han, Z. Tang, Y. He, J. Jiang, J.A. Ansere, District partition-based data collection algorithm with event dynamic competition in underwater acoustic sensor networks. *IEEE Trans. Ind. Inf.* **15**(10), 5755–5764 (2019)
15. X. Zhuo, M. Liu, Y. Wei, G. Yu, F. Qu, R. Sun, Auv-aided energy-efficient data collection in underwater acoustic sensor networks, *IEEE Internet Things J.* (2020)
16. R. Ma, R. Wang, G. Liu, H.-H. Chen, Z. Qin, Uav-assisted data collection for ocean monitoring networks. *IEEE Network* **34**(6), 250–258 (2020)
17. H. Harb, A. Makhoul, R. Couturier, An enhanced k-means and anova-based clustering approach for similarity aggregation in underwater wireless sensor networks. *IEEE Sens. J.* **15**(10), 5483–5493 (2015)
18. Y. Liang, Y. Li, An efficient and robust data compression algorithm in wireless sensor networks. *IEEE Commun. Lett.* **18**(3), 439–442 (2014)
19. K.T.-M. Tran, S.-H. Oh, J.-Y. Byun, Well-suited similarity functions for data aggregation in cluster-based underwater wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **9**(8), 645243 (2013)
20. M. Ibrahim, H. Harb, A. Nasser, A. Mansour, C. Osswald, Adaptive strategy and decision making model for sensing-based network applications, in: 19th International Symposium on Communications and Information Technologies (ISCIT). IEEE **2019**, 96–101 (2019)
21. H. Harb, A. Makhoul, A. Jaber, S. Tawbi, Energy efficient data collection in periodic sensor networks using spatio-temporal node correlation. *Int. J. Sensor Netw.* **29**(1), 1–15 (2019)
22. J. Mabrouki, M. Azrou, G. Fattah, D. Dhiba, S. El Hajjaji, Intelligent monitoring system for biogas detection based on the internet of things: Mohammedia, morocco city landfill case. *Big Data Mining Anal.* **4**(1), 10–17 (2021)
23. M. Wu, L. Tan, N. Xiong, A structure fidelity approach for big data collection in wireless sensor networks. *Sensors* **15**(1), 248–273 (2015)
24. M. Ibrahim, H. Harb, A. Nasser, A. Mansour, C. Osswald, On-in: an on-node and in-node based mechanism for big data collection in large-scale sensor networks, in: 27th European Signal Processing Conference (EUSIPCO). IEEE **2019**, 1–5 (2019)
25. J. Mabrouki, M. Azrou, D. Dhiba, Y. Farhaoui, S. El Hajjaji, Iot-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts. *Big Data Mining Anal.* **4**(1), 25–32 (2021)
26. D. Luengo, L. Martino, M. Bugallo, V. Elvira, S. Särkkä, A survey of monte carlo methods for parameter estimation. *EURASIP J. Adv. Signal Process.* **2020**(1), 1–62 (2020)
27. A. Barbu, S.-C. Zhu, *Monte Carlo Methods* (Springer, 2020)
28. D. Grana, L. Azevedo, M. Liu, A comparison of deep machine learning and Monte Carlo methods for facies classification from seismic data. *Geophysics* **85**(4), WA41–WA52 (2020)
29. G. Sin, A. Espuña, Applications of Monte Carlo method in chemical, biochemical and environmental engineering. *Front. Energy Res.* **8**, 1–2 (2020)
30. K. Guo, L. He, Y. Chen, W. Guo, J. Zheng, A local community detection algorithm based on internal force between nodes. *Appl. Intell.* **50**(2), 328–340 (2020)
31. N.E. Diana, I.H. Ulfa, Measuring performance of n-gram and Jaccard-similarity metrics in document plagiarism application, in: *Journal of Physics: Conference Series*, IOP Publishing. 1196(1):012069 (2019)
32. B. Li, M. Gao, L. Ma, Y. Liang, G. Chen, Web application-layer DDOS attack detection based on generalized Jaccard similarity and information entropy, in: *International Conference on Artificial Intelligence and Security*. Springer, 576–585 (2019)
33. R. Moodley, F. Chiclana, F. Caraffini, J. Carter, Application of uninorms to market basket analysis. *Int. J. Intell. Syst.* **34**(1), 39–49 (2019)
34. P. Parthasarathy, S. Vivekanandan, A typical iot architecture-based regular monitoring of arthritis disease using time wrapping algorithm. *Int. J. Comput. Appl.* **42**(3), 222–232 (2020)
35. H. Zhu, X. Wang, X. Chen, L. Zhang, Similarity search and performance prediction of shield tunnels in operation through time series data mining. *Autom. Constr.* **114**, 103178 (2020)
36. J.A. Tilley, The a-graph coloring problem. *Discret. Appl. Math.* **217**, 304–317 (2017)
37. Y. Cao, G. Chen, G. Jing, M. Stiebitz, B. Toft, Graph edge coloring: a survey. *Graphs Comb.* **35**(1), 33–66 (2019)
38. ARGO, “Argo project,” <http://www.argo.ucsd.edu/index.html>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.