All-in-one: Toward hybrid data collection and energy saving mechanism in sensing-based IoT applications

Marwa Ibrahim, Hassan Harb, Ali Mansour, Abbass Nasser, Christophe Osswald

▶ **To cite this version:**

Marwa Ibrahim, Hassan Harb, Ali Mansour, Abbass Nasser, Christophe Osswald. All-in-one: Toward hybrid data collection and energy saving mechanism in sensing-based IoT applications. Peer-to-Peer Networking and Applications, 2021, 14 (3), pp.1154-1173. 10.1007/s12083-021-01095-5 . hal-03188878

HAL Id: hal-03188878

https://hal-ensta-bretagne.archives-ouvertes.fr/hal-03188878

Submitted on 5 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Peer-to-Peer Networking and Applications manuscript No.
(will be inserted by the editor)

All-in-One: Toward Hybrid Data Collection and Energy Saving Mechanism in Sensing-Based IoT Applications

Marwa Ibrahim* · **Hassan Harb** · **Ali Mansour** · **Abbass Nasser** · **Christophe Osswald**

* ***Corresponding author***

Received: date / Accepted: date

Abstract Big data collection and storage have become one of the most obvious challenge in this era. Indeed, much of that data is collected thanky to a huge number of connected devices in sensing-based IoT applications. Thus, in order to deal with data growth in such applications, researchers have focused on data reduction approach as an efficient solution for minimizing the amount of data collection and saving the limited sensor energy in such networks. Mainly, data reduction approach relies on various kinds of data processing techniques such that aggregation, compression, prediction, clustering, sensing frequency adaptation and spatial-temporal correlation. However, each of those techniques has its own advantages and disadvantages regarding sensor energy saving, data reduction ratio, data accuracy, complexity, etc. In this paper, we propose a hybrid data collection and energy saving mechanism, called All-in-One, for sensing-based IoT applications. The proposed mechanism takes advantages from existing data reduction techniques while optimizing various performance metrics. All-in-One relies on the cluster network architecture and works on three main phases: on-period, in-period and in-node. The first phase, e.g. on-period, allows each sensor node to search the similarity among its periodic collected data then to reduce its data transmission to the Cluster-Head (CH) by applying either data aggregation, compression or prediction technique. The second phase, e.g. in-period, allows each sensor to study the variation of the monitored condition then to reduce its data collection according to two techniques, on-off transmission or adapting sensing frequency. The last phase, e.g. in-node, is applied at the CH level and aims to remove the re-

M. IBRAHIM, H. HARB, A. MANSOUR, A. NASSER, C. OSSWALD
Lab-STICC, CNRS UMR 6285, ENSTA-Bretagne, Brest, France
E-mail: marwa.ibrahim@ensta-bretagne.org, E-mail: firstname.lastname@ensta-bretagne.fr

M. IBRAHIM, H. HARB, A. NASSER,
ICCS-Lab, American University of Culture and Education (AUCE), Beirut, Lebanon
E-mail: firstnamelastname@auce.edu.lb

dundancy among data collected by neighboring nodes, based on in-network correlation or data clustering techniques, before sending the data to the sink. We conducted simulations on real sensor data in order to evaluate the efficiency of our mechanism, in terms of several performance metrics, compared to other exiting techniques.

Keywords Sensing-based IoT applications · Cluster network architecture · Energy saving · Big data collection · Data reduction · Data redundancy study

1 Introduction

Nowadays, the number of connected sensor devices are widely increased and exceeding even the population number. In everyday life, one can find a huge number of deployed sensors in various applications collecting many kinds of data. Indeed, surveillance, data collection and sensing have been recently introduced in various applications such as military, agriculture, environments, industrial, home automation, transport, etc. [1–3]. Whilst data collected by such sensors can take values, images, audio or video types depending on the application requirements. Starting from the beginning of this decade, sensor devices have been more and more organized in networks under different communication protocols referred as Internet of Things (IoT). With IoT networks, we are able to monitor almost anywhere, anytime and anytime collected data to the sink for further analyzing and studying purposes.

Indeed, IoT applications are facing several challenges and problems caused by limited sensor resources and the densely deployment of the devices. However, one of the major challenges for researchers is how to deal, store and analyze a huge amount of data collected in such networks. Furthermore, the sensor devices are energy-constrained and recharging their batteries is not always an option and it may become a costly operation. In addition, data transmission is the higher energy cost in the sensor that quickly depletes its available power and lowers its lifetime. Hence, data reduction approach has taken a great attention from researchers in order to overcome the big data challenges imposed by IoT. The main objective of such approach is to minimize the data transmission at the sensors by removing on-node and in-node redundancy existing among the collected data. In the literature, a data reduction can be performed in several ways such as aggregation, compression, prediction, sensing rate adaptation, clustering, etc. However, the selection of a suitable technique is highly related to the targeted application and the desired performance metric (energy consumption, data accuracy, complexity, etc.) that must be optimized.

In this manuscript, we take advantages from all data reduction techniques and propose a hybrid and adaptive data collection mechanism, All-in-One, for energy saving in IoT applications. The idea behind our mechanism is to make the sensor self-reconfigurable by deciding about the most suitable data reduction technique to be applied according to several parameters, e.g. data redundancy ratio and remaining battery level. Basically, All-in-One works on

three phases; the first phase is called on-period and aims to reduce the amount of data transmitted from each sensor either by applying aggregation, compression or prediction techniques. The second phase is called in-period and allows to adapt the sensor data transmission according to the variation of the monitored condition; in-period is based on two data reduction techniques: on-off transmission and adapting the sensing frequency. The third phase is called in-node and seeks the data correlation among neighboring nodes based on in-network correlation and data clustering techniques.

The remainder of the manuscript is organized as follows. Section 2 outlines different data reduction and energy-efficient techniques proposed in sensing-based IoT applications. Section 3 presents the periodic clustering architecture used in our mechanism. Sections 4, 5 and 6 detail the three phases applied at sensor and CH levels. Simulation results are discussed in section 7. Finally, the conclusion and future work are highlighted in section 8.

2 Related Work

Data reduction in IoT is a challenging process as data are mostly correlated and contains a high level of redundancy. Thus, what to keep or discard becomes a crucial task affecting the accuracy of the collected data thus the decision made at the sink. In the literature, common techniques to perform data reduction in IoT can be developed by applying aggregation [4], compression [5], prediction [6], clustering [7] or adapting sensing frequency [8]. The idea behind all such techniques is to study the variation among the collected data and try to minimize the data transmission along the path to the sink.

The authors of [9–12] are targeting to minimize data transmission by applying aggregation techniques in IoT. In [9], the authors propose a multidimensional and multidirectional data aggregation (MMDA) technique in order to enhance the data communication and ensure the privacy of the data. MMDA allows each IoT device to organize the data into matrices then applying an aggregation process in two directions, e.g. rows and columns. The authors of [10] propose an entropy-driven data aggregation with a gradient distribution (EDAGD) technique that is relying on three algorithms. The first algorithm is called a multihop tree-based data aggregation and aims to reduce the transmission distance between the sensors and the sink by minimizing the number of hops required to reach the destination. The second algorithm is a tree-based aggregation scheme that uses the entropy and the Choquet integral that allows to monitor and detect abnormal events based on the sleep/active nodes strategy. The last aggregation method is a gradient deployment algorithm which aims to deal with the energy hole problem in IoT applications.

Some works such in [13–16] use data compression techniques in order to reduce the packet size sent to the sink. The authors of [13] propose a priority-based compressed data aggregation (PCDA) technique in order to reduce the amount of heath data transmitted. PCDA uses compressed sensing approach followed by a cryptographic hash algorithm at the biosensor level to save in-

formation accuracy before sending data for diagnosis. In [14], the authors propose a Sequential Lossless Entropy Compression (S-LEC) which organizes the alphabet of 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 [15], a coding provenance scheme (CPS) has been proposed. Compared to traditional compression techniques, CPS ensures a high provenance compression rate as well as it encodes and decodes incrementally the compression ratio at the base station depending on the condition observed.

Other works such as [17–20] are focused on data prediction approach that aims to build a predictive model to send to the sink instead of the whole collected data. The authors of [17] propose a hybrid prediction model based on two algorithms; a stagewise algorithm which is applied at sensor level and uses a set of data points to build a predictive model to reduce sensor data transmission. The other algorithm is used by the sink node and aims to reconstruct the raw data generated by the sensors. In [18], the authors propose an adapted version of dual prediction scheme (DPS) algorithm. The newest version uses a collection of models for data prediction during the past sequences of DPS algorithm, without updating classically the history data table. Indeed, the prediction model is computed at the sensors and sent to the sink or vice-versa. The authors of [19] propose an unsupervised machine learning algorithm, called Kohonen, for predicting data generated by the sensors. Kohonen introduces a self organizing map based on a predictive temporal model that makes sensor in standby mode to reduce its transmission.

In [21–24], the authors propose data clustering techniques in order to group similar data into clusters before eliminating the redundancy. In [21], the authors propose a layered adaptive compression design for efficient data collection (LACD-EDC) in industrial wireless sensor network (WSN). LACD-EDC is based on the clustering data scheme and it aims to search the spatio-temporal correlation within (e.g. intra) and among (e.g. inter) clusters. Then, a compression method is proposed at the sensor level followed by a recover technique at the sink in order to regenerate the raw data and achieve an approximate data collection. The authors of [22] propose a cluster-based data gathering algorithm for WSN called lifetime-enhancing cooperative data gathering and relaying (LCDGRA). Basically, LCDGRA works on three phases: the first phase aims to group the sensor nodes into clusters based on K-means clustering while applying a compression technique, e.g. Huffman coding algorithms, in each cluster. The second phase assigns a set of relay nodes to each CH in order to aggregate data before sending to the sink node. In the last phase, the aggregated data are coded based on random linear coding and then relayed to the base station.

The authors of [25–28] are dedicated to reduce the data collection and transmission in IoT by adapting the sensing frequency of each sensor according to the speed of the condition variation. In [25], the authors propose a data management framework for data collection and decision making in connected healthcare. The framework relies on three algorithms: first, an emergency de-

tection algorithm aims to send critical records directly to the coordinator; second, an adaptive sampling rate algorithm based on ANOVA (ANalysis Of VAriance) and Fisher tests in order to allow each sensor to adapt its sampling frequency to the variation of the patient situation; third, a data fusion and decision making model is proposed at the coordinator and it is based on a decision matrix and the fuzzy set theory. The authors of [26] propose two adaptive sampling techniques: exponential double smoothing adaptive sampling (EDSAS) and Wiener filter based adaptive sampling (WFAS). Both algorithms seek the correlation between current and previous collected data and aim to minimize the sensor sampling rate while a high level of data accuracy.

Lastly, some works such as [29–32] are targeting to reduce data collected by neighboring nodes using in-network data processing. The authors of [29] propose a prefix frequency filtering (PFF) technique based on clustering architecture of the network. Further to a local processing at the sensor node level, PFF uses Jaccard similarity function within aggregator nodes to identify similarities among near sensor nodes at each period and integrates their sensed data into one record. In [30], the authors propose a structure fidelity data collection (SFDC) technique dedicated to cluster-based periodic applications in WSNs. SFDC searches both spatial and temporal correlations between nodes, using distance functions and similarity metrics respectively. In [31], an energy-efficient communication method dedicated to periodic underwater sensor applications is proposed. On the basis of the proposed technique, each node cleans its collected data before transmitting to the appropriate CH. When receiving datasets, the CH applies K-means algorithm adopted to the ANOVA with statistical tests in order to eliminate inter-node correlations.

3 Network Design and Preliminaries

3.1 Network Design

Transmitting the raw data collected by the sensor nodes to the sink is a fundamental operation in IoT. Hence, the network architecture plays an important role in the performance of IoT applications. Subsequently, several metrics such as congestion, energy consumption, network overload, data loss, latency, etc. are highly affected by the selection of the network architecture. In this work, our mechanism relies on the cluster-based network architecture in which the data transmission between sensors and the sink is performed using two-hops communication.

Generally, the node clustering provides an efficient scheme to organize data traffic in the network, improve its scalability and reduce the energy consumption. Typically, the clustering approach divides the sensors in the network into clusters and assigns a CH for each cluster. Subsequently, the CH can be selected among the sensor nodes or defined prior to the network deployment with more resources than the normal nodes. Once selected, the CH is responsible of managing the cluster and can perform in-network processing over the sensor

data before sending toward the sink node. Fig. 1 illustrates a two-layer cluster architecture in which the communication among the sensors and their CHs or the CHs and the sink is performed according to a single-hop transmission.

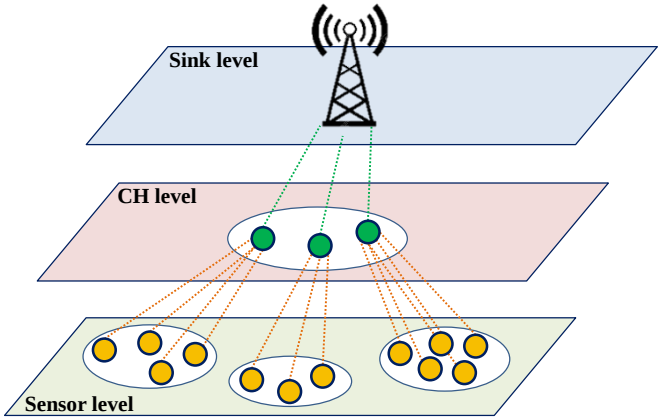


Fig. 1 Two-layers cluster-based architecture network.

3.2 Periodic Data Collection Model

After selecting the appropriate network architecture, the sensor nodes start sensing the surrounding and sending the data toward the sink. Indeed, we can distinguish among three types of data collection in IoT: query-based, event-based or periodic-based [33]. In this work, we focus on the last collection model which is used in a large number of applications that require a constant and continuous monitoring such as phenomena study, patient observation, habitat surveillance, traffic tracking, etc. In most of such applications, sensors collect data of interest and forward them to the sink at constant periodic time intervals for analysis and studying purposes.

Basically, in a periodic acquisition model, 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: $R_i^p = \{r_1, r_2, \ldots, r_{\mathcal{F}}\}$. After that, the sensor will send its data vector, e.g. R_i^p, to its appropriate CH. Fig. 2 shows an illustrative example for the periodic data collection model for a cluster of 3 member nodes and a CH. The period size $\mathcal{F}$ is fixed to 5 readings and each sensor node N_i, $i \in [1, 3]$, collects a set R_i^p during each period before sending to the CH at the end of the period.

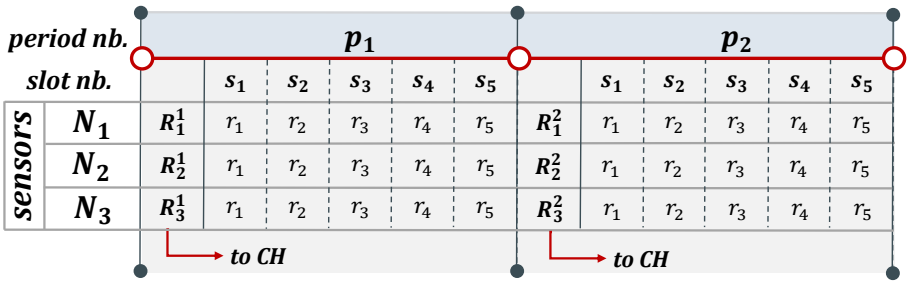


Fig. 2 Illustrative example of periodic data collection model.

3.3 Problem Formulation

Due to the huge amount of data collected, the periodic model provides a significant redundancy in IoT applications. This redundancy is mostly a consequence of the following situations: the on-period redundancy in which the readings collected by a sensor in each period, e.g. R_i^p, are redundant; this is usually happening due to the short time slot or the small size of the period size. The in-period redundancy in which the data collected by the same sensor in consecutive periods, e.g. R_i^p and R_i^{p+1}, are redundant; this can happen due to the slow variation of the monitored condition. The in-node redundancy in which the data collected by neighboring sensors, e.g. R_i^p and R_j^p, are redundant; this happens because of the spatial and/or temporal correlation between the nodes in the network. Unfortunately, the redundancy among data will lead, from one hand, to complicate the data analysis at the end user and, from the other hand, to deplete the limited energy of the sensors.

3.4 An Overview to All-in-One Mechanism

In this paper, we propose an All-in-One mechanism applying on sensors and CHs that allows eliminating the redundancy exiting in WSN. Fig. 3 shows the main phases of the proposed mechanism along with the process of redundancy elimination proposed at each phase. At the sensor level, our mechanism searches the redundancy among the data collected by a sensor at each period and round respectively. On one hand, we search the similarity among collected data by each sensor; then by using an on-period decision table based on the variation level and the sensor battery level, we select the most adequate data reduction method. Subsequently, we propose data reduction algorithms based on three concepts: prediction, compression, and aggregation. On the other hand, our mechanism searches the in-period redundancy by each sensor at each round. Therefore, an in-period decision table is introduced to consider the similarity between data in the round along with the sensor battery level to decide about the appropriate elimination method, e.g. Sensing Frequency Adaptation (SFA) or On-Off Transmission (OOT). At the CH level, the in-

node redundancy among neighboring nodes is investigated in order to reduce the periodic number of packets sent to the sink. Subsequently, the redundancy elimination process is based on the packet types. First, the compressed packets are grouped into clusters then sent the cluster centroids to the sink. Second, the aggregated packets are propagated using an in-network aggregation technique then sending the unsimilar data to the sink. Third, the predicted and off packets are directly forwarded to the sink without any elimination process.

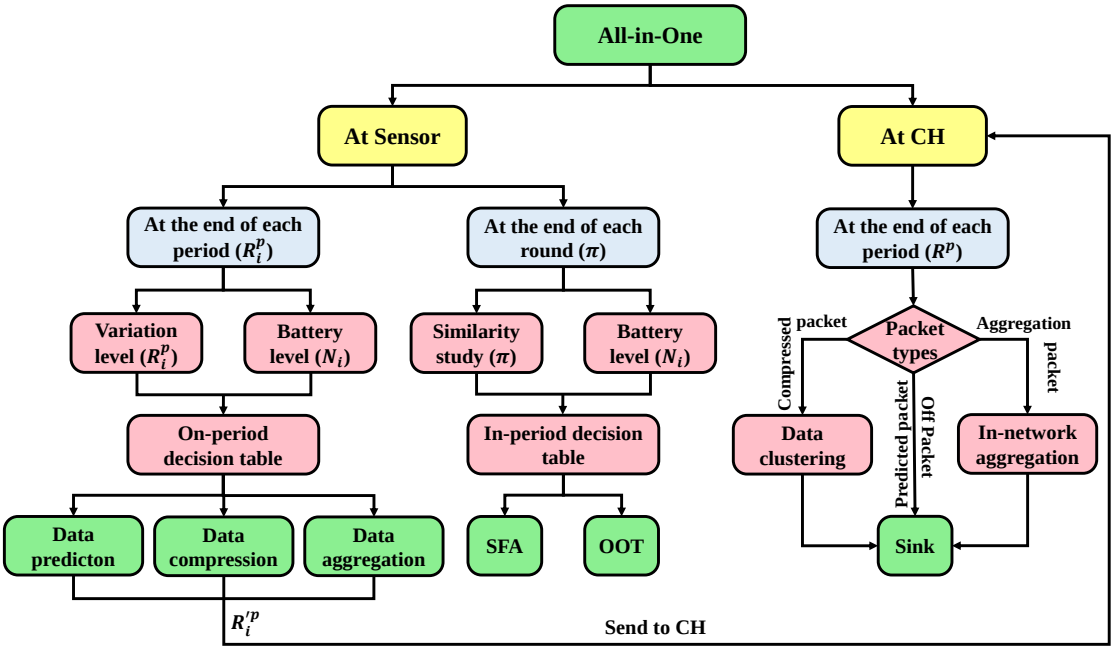


Fig. 3 Flow diagram of All-in-One mechanism.

4 On-Period Redundancy Elimination Model

In IoT, the periodic data collection is a fundamental operation in order to understand the behavior of the monitored environment and increase the reliability of the taken decision. However, this collection model produces a high redundancy level among the data that leads to send un-useful information to the sink and consumes the available energy in the sensor. In order to overcome these problems, researchers have focused on three main reduction approaches to eliminate in-period redundancy at each sensor: aggregation, compression and prediction. In this section, we introduce an efficient technique for each approach, then we propose a new hybrid model for removing the in-period redundancy.

4.1 Aggregation-Based Reduction Technique

The data aggregation seeks the similarities among the data collected in order to eliminate the existing redundancies and reduce the size of data transmission to the CH. Hence, we first define the *Aggregate* function that allows each sensor to search the similarities among the readings in R_i^p as follows:

Definition 1 $Aggregate(r_j, r_k)$. Assume r_j and r_k are two readings collected by the same sensor during a period p. Then, r_j and r_k are considered similar if and only if the difference between them is less than a defined threshold δ as follows:

$$Aggregate(r_j, r_k) = |r_j - r_k| \ \leq \ \delta \tag{1}$$

where δ is a user-defined threshold determined according to the application requirements.

Then, in order to maintain the accuracy of the aggregated data, we define the weight, called wgt, for each reading as follows:

Definition 2 $wgt(r_j)$. The weight of a reading r_j is defined as the number of similar readings to r_j in the same reading set R_i^p.

Based on the *Aggregate* and weight functions, the sensor searches the similarity among every pair of readings in R_i^p until no more redundancy exists (Algorithm 4.1). The algorithm takes as input the set of readings collected by a sensor during a period and returns, as output, the aggregated set of readings that will be sent to the CH at the end of the period. For each collected reading, the sensor searches for its similarity (according to the *Aggregate* function) with all readings in the set; if the two compared readings are similar (according to the similarity threshold) then the weight of the corresponding reading is added by one (lines 2-9). Then, the sensor calculates the weight for each reading and adds it to the aggregated set that will send to the CH (line 10).

Algorithm 1 Data Aggregation Algorithm.

Require: a sensor node: N_i; a period: p; a set of readings: R_i^p; similarity threshold: δ.
Ensure: Aggregated set of readings: $R_i^{'p}$.

1: $R_i^{'p} \leftarrow \emptyset$
2: **for** each reading $r_j \in R_i^p$ **do**
3: $wgt(r_j) = 1$
4: **for** each reading $r_k \in R_i^p$ where $k > j$ **do**
5: **if** $Aggregate(r_j, r_k) \leq \delta$ **then**
6: increment $wgt(r_j)$
7: delete r_k from R_i^p
8: **end if**
9: **end for**

10: $R_i^{'p} \leftarrow R_i^{'p} \cup \{(r_j, wgt(r_j))\}$
11: **end for**
12: **return** ${R'}_i^p$

4.2 Compression-Based Reduction Technique

By definition, the compression is the process of combining redundant readings into a reduced set of records. Indeed, in order to determine the data redundancy, the correlation among the readings should be studied. In this paper, we focus on the Pearson correlation coefficient (PCC) as one of the metrics that is most used to measure the correlation degree among data sets. Pearson coefficient gives a value between -1 and $+1$ where $+1$ (respectively -1) indicates a perfect (respectively negative perfect) correlation among the datasets. Mathematically, the Pearson correlation coefficient between two data sets R_i and R_j is given by to the following equation:

$$Pearson(R_i, R_j) = \frac{\mathcal{F}\sum_{k=1}^{\mathcal{F}} r_{i_k} r_{j_k} - \sum_{k=1}^{\mathcal{F}} r_{i_k} \sum_{k=1}^{\mathcal{F}} r_{j_k}}{\sqrt{\mathcal{F}\sum_{k=1}^{\mathcal{F}} r_{i_k}^2 - (\sum_{k=1}^{\mathcal{F}} r_{i_k})^2}\sqrt{\mathcal{F}\sum_{k=1}^{\mathcal{F}} r_{j_k}^2 - (\sum_{k=1}^{\mathcal{F}} r_{j_k})^2}} \quad (2)$$

where $r_{i_k} \in R_i$, $r_{j_k} \in R_j$ and $\mathcal{F}$ is the number of readings in R_i or R_j.

Therefore, R_i and R_j are considered to be highly correlated (e.g. redundant) if and only if:

$$|Pearson(R_i, R_j)| > \varepsilon \quad (3)$$

where ε is the Pearson's threshold.

Algorithm 4.2 shows the compression technique applied over the data collected by each sensor during a period, based on the Pearson coefficient metric. First, all the readings are assumed correlated and R_i is assigned to a temporary set of reading subsets, e.g. S (line 2). Then, the correlation among the readings is calculated by dividing them into two equal subsets using the function *Partition* (line 4). Thus, if the correlation exceeds the Pearson threshold then the readings are considered redundant and, consequently, the average of the readings is computed (e.g. $\overline{r}$) and added with its weight (e.g. $wgt(\overline{r})$) to the final reading set that will sent to the CH (lines 5-10). Otherwise, e.g. the correlation coefficient does not exceed the Pearson threshold, the readings are considered unsimilar and we repeat the process over each subset until all readings within each subset become redundant. Therefore, at the end of each period, each sensor will send the compressed set of readings $R_i^{'p}$ to the CH.

Algorithm 2 Data Compression Algorithm.

Require: a sensor node: N_i; a period: p; a set of readings: R_i^p; Pearson threshold: ε.
Ensure: Compressed set of readings: $R_i^{'p}$.

$R_i^{'p} \leftarrow \emptyset$
$S \leftarrow R_i$
for each set $R_k \in S$ **do**
  $(R_{k_l}, R_{k_r}) = Partition(R_k)$
  if $Pearson\,(R_{k_l}, R_{k_r}) \leq \varepsilon$ **then**
    $\overline{r} = Mean(R_k)$
    $wgt(\overline{r}) = |R_k|$
    // $|R_k|$ is the total number of elements in R_k
    $R_i^{'p} \leftarrow R_i^{'p} \cup \{(\overline{r},\, wgt(\overline{r}))\}$
    remove R_k from S
  else
    $S \leftarrow S \cup \{R_{k_l}, R_{k_r}\}$
    remove R_k from S
  end if
end for
return $R_i^{'p}$

4.3 Prediction-Based Reduction Technique

In IoT, the data prediction allows each sensor to build, based on the collected data, a predictive model in order to send to the sink which, in its turn, regenerates the raw data. In this work, our prediction model is based on the Newton Forward Differences (NFD) method that takes the periodic data collected by a sensor, e.g. R_i^p, and finds the polynomial coefficient set, e.g. E_i^p, to send to the CH. Mathematically, given a set of readings $R_i^p = \{(s_1, r_1), (s_2, r_2), \ldots, (s_\mathcal{F}, r_\mathcal{F})\}$, where s_i represents the time slot in which the reading r_i is taken during the period, then the NFD first set up the forward difference table as:

s	r	Δr	$\Delta^2 r$	$\ldots$	$\Delta^{c-1} r$	$\Delta^c r$
s_1	r_1					
s_2	r_2	Δr_1	$\Delta^2 r_1$			
$\vdots$	$\vdots$	Δr_2	$\vdots$			
		$\vdots$			$\Delta^{c-1} r_1$	
$s_{\mathcal{F}-2}$	$r_{\mathcal{F}-2}$		$\Delta^2 r_{\mathcal{F}-3}$	$\ddots$		
$s_{\mathcal{F}-1}$	$r_{\mathcal{F}-1}$	$\Delta r_{\mathcal{F}-2}$	$\Delta^2 r_{k-2}$			
$s_\mathcal{F}$	$r_\mathcal{F}$	$\Delta r_{\mathcal{F}-1}$				

Table 1 Forward Difference table.

where $\Delta r_j = r_{j+1} - r_j$, $j \in [0, \mathcal{F} - 1]$.

Then, the NFD method uses the Newton forward formula in order to find the value for any reading r_i taken at the slot time s_i as follows:

$$r_i \approx f(s_1 + hu) = r_1 + u\Delta r_1 + \frac{u\,(u-1)}{2!}\Delta^2 y_1 + \ldots + \frac{u\,(u-1)\,(u-2)\ldots(u-c+1)}{c!}\Delta^c r_1 \quad (4)$$

where h is called the interval of difference ($h = s_2 - s_1$) and $u = (r_i - r_1)/h$.

Finally, in order to allow the sink to regnerate the readings in R_i^p, the sensor must send the set of variables needed in the Newton forward formula to calculate the r_k values of all readings, e.g. $E_i^p = \{s_1, s_2, r_1, \Delta r_1, \Delta^2 r_1, \ldots, \Delta^c r_1\}$.

4.4 Performance Discussion of On-Period Techniques

This section gives further considerations of the three introduced on-period techniques by studying the thresholds' selection, the accuracy, the complexity, and the energy consumption.

4.4.1 Selection of Thresholds' Values

Obviously, the efficiency of the aggregation, compression, and prediction techniques are highly related to the selection of the thresholds c, δ, and ε respectively. Subsequently, increasing or decreasing the threshold values may change the performance of several metrics in WSN, such as: the accuracy, the data latency, the data transmission ratio, and the energy consumption. Hence, selecting the appropriate values of thresholds are critical in the first stage of our mechanism. Therefore, we consider that the thresholds' values should be determined by the decision makers or experts depending on the application requirements. For instance, in health monitoring applications, the thresholds should optimize the accuracy of the collected data more than other metrics, while, in the environmental applications, the energy conservation gets the highest priority compared to other metrics. Thus, these parameters are based on the application criticality and the studied phenomenon.

After selecting their values, the decision makers assign the thresholds accordingly into all sensors nodes prior to deployment or they can adjust it online in function of the application requirement.

4.4.2 Accuracy Study

In compression-based and prediction-based techniques, the increase of the values of thresholds (e.g. c and ε) will proportionally increase the amount of data sent, thus the accuracy of the information sent, and vice versa. While, in the aggregation-based technique, the accuracy of the sent data will increase

with the decrease of similarity threshold, e.g. δ. However, in our mechanism, the *wgt* function defined in the aggregation and compression algorithms (e.g. Algorithms 1 and 2) will maintain the full accuracy of the sent data.

4.4.3 Complexity Study

The complexity is an important metric in WSN due to the limited sensor resources, especially processing and storage. From one hand, the processing complexity of any proposed technique may affect the system latency which is crucial in many WSN applications, especially in healthcare or military. On the other hand, sensors are characterized by relatively small memory size; hence, any technique should satisfy the memory constraint. The complexity of the three proposed techniques can be studied as follows:

- The aggregation technique: each sensor node N_i forms a set R_i^p of $\mathcal{F}$ readings in each period. Due to the *Aggregate* function, the size of this set can be reduced from $\mathcal{F}$ to $|{R'}_i{}^p|$. Therefore, this technique has at most $O(|{R'}_i{}^p|^2)$ as a computation complexity at the sensor and saves at most $2 \times |{R'}_i{}^p|$ values, e.g. readings with their weights, at each period in its memory.
- The compression technique: the sensor N_i recursively divides the reading set collected in every period into two equal partitions before calculating their correlation according to Pearson coefficient. Hence, the computation complexity of the compression algorithm will be of $O(log(|R_i^p|))$ while the memory storage will be equal, at most, to $2 \times |{R'}_i{}^p|$ (e.g. mean values with their weights), similarly to that of the aggregation technique.
- The prediction technique: according to the NFD method, the sensor sends the set of coefficients, e.g. E_i^p, calculated at each period to the CH. Thus, the computation complexity of the prediction algorithm should be of $O(c \times log(|R_i^p|))$ while the memory storage is limited to the length of the NFD coefficient set, e.g. $|E_i^p|$.

Based on the above study, we clearly show that the complexities of all techniques are suitable for the case of sensor nodes.

4.4.4 Energy Consumption Study

In WSN, the data transmission operation consumes most of the sensor energy compared to other operations, e.g. sensing and processing [41]. Thus, minimizing the periodic data transmitted from sensors and CHs is mandatory to save the energies. Hence, the three proposed on-period techniques can be considered as good solutions for conserving the node energies and extending their lifetime. This is due to the redundancy elimination process introduced in each one of them that allows to reduce the amount of transmitted data and only send the useful information towards the sink. Furthermore, as mentioned before, the elimination process ensures the accuracy of the sent data and limits the effect on the decision made by the end user.

4.5 Hybrid-Based On-Period Reduction Technique

Indeed, the selection among the data reduction approaches (aggregation, compression or prediction) is a crucial decision for the sensor since it affects several performance metrics. For instance, the data prediction technique can highly save the sensor's energy because it reduces the data transmission more than aggregation and compression techniques. However, the prediction technique can negatively affect the accuracy of the transmitted data. Hence, we propose a hybrid-based on-period reduction model that takes advantages from several reduction techniques while optimizing several performance metrics. The proposed model is based on two main parameters, e.g. the condition variation and the remaining sensor battery, in order to decide the reduction technique that should be used in each period. Subsequently, the condition variation is calculated according to the ANOVA and a statistical test, e.g. Bartlett test.

4.5.1 ANOVA Model and Bartlett Test

ANOVA is a well-known statistical method that is used to test the variance among a group of data sets if it is significant or not. First, ANOVA computes a T-statistic value, according to a statistical test, then the data sets are considered redundant (or have low variance) if the calculated T is less than a critical value T_α for some false-rejection probability α; more the value of T_α is decreased, more the redundancy among the data sets is.

On the other hand, Bartlett test [34] checks if a group of data sets have an equal variance. Thus, Bartlett test verifies the null hypothesis that variances are equal across data sets comparing to the alternative hypothesis that the variances are significant. In our case, the objective is to calculate the variance among readings collected by a sensor during a period (e.g. R_i^p). Hence, we first divide R_i^p into d equal divisions (or subsets) where each division D_j, $j \in [1, d]$, contains $\mathcal{F}/d$ readings. Then, the Bartlett test can be applied over R_i^p as follows:

$$T = \frac{(\mathcal{F} - d)\ln(\sigma_p^2) - \left(\frac{\mathcal{F}}{d} - 1\right)\sum_{j=1}^{d}\ln\left(\sigma_j^2\right)}{\lambda} \tag{5}$$

where :

$$\lambda = 1 + \frac{1}{3(d-1)}\left(\frac{d}{\frac{\mathcal{F}}{d} - 1} - \frac{1}{\mathcal{F} - d}\right)$$

and σ_p^2 is the pooled variance that is defined as:

$$\sigma_p^2 = \frac{1}{\mathcal{F} - d}\sum_{j=1}^{d}\sigma_j^2$$

Therefore, in order to test the variance T among the readings in R_i^p, we select two critical values for T_α, e.g. T_{α_0} and T_{α_1} where $\alpha_0 < \alpha_1$. Then, the condition variation is based on:

- $T \leq T_{d-1,\alpha_0}$ or *low variation*: the variance among the divisions is not significant and the readings in R_i^p are considered similar.
- $T_{d-1,\alpha_0} < T \leq T_{d-1,\alpha_1}$ or *medium variation*: the variance among the divisions is a bit significant and the readings in R_i^p are considered redundant.
- $T > T_{d-1,\alpha_1}$ or *high variation*: the variance among the divisions is significant.

4.5.2 Sensor Battery Level

The lifetime of the IoT networks is heavily related to the sensor battery level which, in its turn, can be quickly consumed when the amount of data transmission increases. Hence, in addition to condition variation level, we propose to take into account the remaining energy of the sensor in order to adapt the periodic data transmission to the CH. The idea is that when the sensor battery level becomes crucial, e.g. less than a defined threshold, its data transmission must be more and more reduced but without highly affecting the data integrity.

Let assume that the initial energy of the sensor is E_i and the remaining one during the current period p is E_r. Then, we define a critical threshold E_c where the sensor energy becomes crucial if it reaches this threshold. Therefore, the decision about the sensor battery level during p can be made as follows:

- if $E_i \geq E_c$ then *high battery level*.
- otherwise, *low battery level*.

4.5.3 On-Period Data Decision

At the end of each period, the hybrid-based reduction technique allows each sensor to decide about the reduction approaches (aggregation, compression and prediction) that should be applied over the collected data. Table 2 shows the decision made by the sensor based on the calculated variation and battery levels. Subsequently, the selection of the reduction approaches inside the on-period data decision table is motivated by the following reasons:

- if the variation and battery levels are low then the data prediction must be used. This will reduce the data transmission to the minimum (thus save the sensor energy) but without losing the information collected by the sensor.
- if the variation is high then the data aggregation is preferably to be used. This is because the aggregation will decrease the similarity between the transmitted data without ensuring a high level of data accuracy.
- otherwise, the data compression constitutes an ideal technique that compromises between data reduction and data accuracy.

Variation level \ Battery level	*Low*	*High*
Low	Data prediction	Data compression
Medium	Data compression	Data compression
High	Data aggregation	Data aggregation

Table 2 On-period data decision table.

5 In-Period Redundancy Elimination Model

Mostly, the data collected by each sensor during successive periods are highly correlated depending on the variation of the monitored condition. Particularly, the slowdown of the environment leads to increase the redundancy among the sensed data which results in sending useless information to the sink and consuming the sensor energy. Hence, eliminating the in-period data redundancy becomes an essential technique to achieve fair data reduction rates and conserve the limited energy resources of IoT. In the next section, we introduce two mechanisms in order to search, then eliminate, the redundancy existing among periods: on-off transmission and sensing frequency adaptation.

5.1 Sensing Frequency Adaptation (SFA) Mechanism

In the periodic collection model, the selection of the appropriate sensing frequency of each sensor is a very important decision before deploying the network. Consequently, a high sensing frequency can lead to increase the redundancy among the collected data and consume the sensor energy while the decreasing of the sensing frequency can affect the accuracy of the transmitted data. Hence, adapting the sensing frequency to the environment variation is thereby resulting in data reduction and saving sensor energy.

Mathematically, let assume a round π consisting of P period in which a sensor node N_i will collect a set of readings sets as follows: $R_i = \{R_i^1, R_i^2, \ldots, R_i^P\}$. Therefore, in order to study the condition variation, ANOVA and Bartlett test are applied again over the data sets in R_i. Thus, the condition is "slow down" if the calculated variation T is less than a certain threshold $T_{P-1,\beta}$ for some false rejection probability (risk β). Consequently, the sensor must adapt its sensing frequency according to the *Adapting* function based on the Bezier curve [35]:

$$Adapting(T, T_{P-1,\beta}, C_r, \mathcal{F}) =$$

$$\begin{cases} \frac{(\mathcal{F}-2b_y)}{4b_x^2}T^2 + \frac{b_y}{b_x}T & if\ (T_{P-1,\beta} - 2b_x = 0) \\ (\mathcal{F} - 2b_y)(\propto (T))^2 + 2b_y \propto (T), & if\ (T_{P-1,\beta} - 2b_x \neq 0) \end{cases}$$

where

$$\propto (T) = \frac{-b_x + \sqrt{{b_x}^2 - 2b_x \times T + T_{P-1,\beta} \times T}}{T_{P-1,\beta} - 2b_x} \ \wedge \begin{cases} 0 \leq b_x \leq T_{P-1,\beta} \\ 0 \leq T \leq T_{P-1,\beta} \\ T_{P-1,\beta} > 0 \end{cases}$$

and $b_x = -T_{P-1,\beta} \times C_r + T_{P-1,\beta}$ while $b_y = \mathcal{F} \times C_r$.

Subsequently, the *Adapting* function takes four variables as input: the variance between readings in a round (T), the variance threshold ($T_{P-1,\beta}$), the criticality of the monitored application (C_r) and the original period size ($\mathcal{F}$). Indeed, the application criticality (C_r) is a value between 0 and 1 that is assigned by the expert depending on the monitored application and that should be taken into account when adapting the sensor frequency. For instance, C_r must take a value near to 1 in high critical applications (i.e. healthcare and military) and near to 0 in low critical applications (i.e. weather and environment monitoring). Therefore, the *Adapting* function calculates the new sensing frequency of the sensor in the next round.

5.2 On-Off Transmission (OOT) Mechanism

The objective of this technique is to avoid sending similar data in successive periods from each sensor to the CH. Thus, the sensor will update the CH about the condition variation only if a noticed difference is detected compared to the mast sent data. This will decrease the number of packets sent from each sensor, save its energy and reduce the congestion in the network. Indeed, one can find several functions that allows to search the similarity among data sets such as Jaccard, Dice, Cosine, etc. In this paper, we focus on the Jaccard similarity as one of most used and well adapted functions to several domains. For the sake of simplicity, let assume a round consisting of two periods, e.g. $R_i = \{R_i^1, R_i^2\}$, thus reading sets in R_i are considered similar according to the Jaccard function if:

$$Jaccard(R_i^1, R_i^2) = \frac{|R_i^1 \cap R_i^2|}{|R_i^1 \cup R_i^2|} \geq t_J \tag{6}$$

where t_J is the Jaccard threshold in $[0, 1]$ where 0 indicates that the readings are totally different and 1 that are totally equal.

Algorithm 5.2 shows the on-off transmission mechanism applied at each sensor during a round. Indeed, we define two types of packets that will send

by the sensor: On_Packet which contains the identification (id) of the sensor with its readings collected during the current period; Off_Packet which only contains the id of the sensor informing the CH that the current collected readings are removed due to the similarity with the previous ones. Thus, the sensor sends the reading set collected during the first period to the CH in a On_Packet while saving it in its memory at the same time (lines 1-2). Then, for every new reading set collected in the next period, the sensor searches its similarity with the set saved in the memory based on the Jaccard function; if the new set is similar to the saved one, then the sensor removes the new one, while sending a Off_Packet to the CH (lines 4-6). Otherwise, e.g. the new one is not similar to the saved one, the sensor sends the new reading set to the CH while replacing the saved set by the new reading set (lines 7-10).

Algorithm 3 On-Off Transmission Algorithm.

Require: a sensor node: N_i; a round: π; set of reading sets: $R_i = \{R_i^1, R_i^2, \ldots, R_i^P\}$; Jaccard similarity threshold: t_J.
Ensure: Saved reading set: R_i^j.

1: $R_i^j \leftarrow R_i^1$
2: $On\_Packet(i, R_i^j)$
3: **for** each set $R_i^k \in R_i$ where $k \geq 2$ **do**
4: **if** $Jaccard\ (R_i^k, R_i^j) \geq t_J$ **then**
5: ignore R_i^k
6: $Off\_Packet(i)$
7: **else**
8: $R_i^j \leftarrow R_i^k$
9: $On\_Packet(i, R_i^j)$
10: **end if**
11: **end for**
12: **return** R_i^j

5.3 Hybrid-Based In-Period Reduction Technique

Obviously, SFA and OOT can both minimize the in-period data redundancy and save the sensor energy. However, SFA can reduce the data transmission to the CH more than OOT because it minimizes its data collection even all readings collected in successive periods are similar. Otherwise, OOT can ensure more data accuracy than SFA because data collected are not sent to the CH only if they are very similar. Hence, in order to make a trade-off between energy saving and data accuracy, we propose a hybrid-based in-period model that allows each sensor to select between SFA and OOT at the end of each round. The proposed model takes into account the in-period similarity among the collected data and the remaining sensor battery then it decides about the

suitable technique to apply at the end of each round. Subsequently, the sensor battery level usage is similar to the situation proposed in subsection 4.5.2 while the in-period similarity study is described on the next section.

5.3.1 In-Period Similarity Study

Indeed, similarity functions are one of the most accurate approaches to search the redundancy among the data compared to other approaches, particularly ANOVA and distance functions. Therefore, we propose to use the Jaccard similarity function in order to determine the similarity level among data collected in successive periods. Once the data similarity level is calculated, the sensor decides about the in-period technique that must be used according to the in-period decision table (see next section). Given a round π consisting of two periods, e.g. R_i^1 and R_i^2, the Jaccard similarity between both periods can be calculated according to the equation 6. Then, in our model, we distinguish between three levels of similarities among data collected in π:

- $0 \leq Jaccard(R_i^1, R_i^2) \leq 0.5$ or *low similarity*: this indicates that the monitored condition is rapidly changing over the periods.
- $0.5 < Jaccard(R_i^1, R_i^2) \leq 0.75$ or *medium similarity*: this indicates that the monitored condition is slowly changing over the time which leads to a certain level of redundancy among the collected data.
- $0.75 < Jaccard(R_i^1, R_i^2) \leq 1$ or *high similarity*: in which the monitored condition is not significantly changing which results in a high similarity among the collected data.

5.3.2 In-Period Decision Table

This table shows the decision made by the sensor at the end of each round based on the data similarity and the battery levels (Table 3). Subsequently, the sensor selects the in-period reduction technique according to the following criteria:

- the sensor must decrease its sensing frequency when the similarity level increases, either with low or high battery level. This will reduce the redundancy among the collected data.
- by fixing to the similarity level to low, medium or high, the sensor must decrease it sensing frequency with the decreasing level of its battery. This will save the sensor energy and avoid a rapid depletion of its battery.
- if a high data similarity level is detected, the sensor will not send the current collected data to the CH (e.g. apply OOT) and will adapt its sensing frequency to the minimum.

Similarity level \ Battery level	*Low*	*High*
Low	$\mathcal{F}^{'} = 40\%$ of $\mathcal{F}$	$\mathcal{F}^{'} = \mathcal{F}$
Medium	$\mathcal{F}^{'} = 30\%$ of $\mathcal{F}$	$\mathcal{F}^{'} = 60\%$ of $\mathcal{F}$
High	OOT + $\mathcal{F}^{'} = 20\%$ of $\mathcal{F}$	OOT + $\mathcal{F}^{'} = 40\%$ of $\mathcal{F}$

Table 3 In-period data decision table.

6 In-Node Redundancy Elimination Model

At the end of each period, the CH receives all data sets coming from its sensors. Indeed, such data are mostly redundant due to the spatial and temporal correlation among the sensors. Therefore, the CH can remove this redundancy in order to reduce the number of packets sent to the sink (thus saves its own energy) and provide only a useful information to the end user. In this section, we introduce two approaches to eliminate in-node (e.g. between nodes) redundancy at the CH: in-network aggregation and data clustering. Subsequently, in order to apply each of the proposed approaches, the CH must recalculate the raw data, e.g. R_i^p, of each received data set, e.g. $R_i^{'p}$, according to the applied in-period approaches.

6.1 In-Network Aggregation Approach

This approach aims to eliminate redundant data sets generated by pairs of neighboring sensors before sending to the sink. Pairs of redundant sets are determined by using distance functions that compute the dissimilarities between two data sets. Thus, two data sets are considering duplicate if the distance between them is less than a predefined threshold. Once all duplicated pairs are found, the CH selects a subset of data to send to the sink while eliminating the other ones. Therefore, the in-network aggregation approach is divided into two steps:

– Pairs generation: In this step, the CH searches all pairs of redundant data sets based on the distance functions. In this paper, we use the Euclidean distance as one of the most distance functions used in the literature. Given two sets of data R_i^p and R_j^p collected by two sensors at the same period p, then the Euclidean distance E_d between both sets is:

$$E_d(R_i^p, R_j^p) = \sqrt{\sum_{k=1}^{\mathcal{F}} (r_{i_k} - r_{j_k})^2} \tag{7}$$

where $r_{i_k} \in R_i^p$ and $r_{j_k} \in R_j^p$. Then, R_i^p and R_j^p are considered redundant if the Euclidean distance between them is less than a threshold, t_E:

$$E_d(R_i^p, R_j^p) \leq t_E \tag{8}$$

– Pairs selection: After determining all redundant pairs, the CH tries to reduce the number of data sets to the sink by selecting a subset among them instead of sending the whole data sets (Algorithm 6.1). For each generated pair, the CH selects the received set having the highest number of elements, e.g. $|R_j^{'p}|$, then it adds it to the final list of data sets that will send to the sink (line $2-4$). Simultaneously, the CH removes all pairs that contain $R_i^{'p}$ or $R_j^{'p}$ from the set of generated pairs (line 5).

Algorithm 4 In-Network Aggregation Algorithm.

Require: List of generated pairs: $A = \{(R_i^p, R_j^p)$ such that $E_d(R_i^p, R_j^p) \leq t_E$ and $i \neq j\}$.
Ensure: List of sent data sets: L.
1: $L \leftarrow \emptyset$
2: **for** each pair $(R_i^p, R_j^p) \in A$ **do**
3: Consider $|R_i^{'p}| \geq |R_j^{'p}|$
4: $L \leftarrow L \cup \{R_i^{'p}\}$
5: Remove all pairs containing R_i^p or R_j^p
6: **end for**
7: **return** L

6.2 Data Clustering Approach

Generally, clustering is a data exploratory task that aims to group data into a set of K clusters in a way that the similarity among data in the same cluster is high and that among clusters is low. Thus, data clustering can be an efficient solution to reduce the data transmission from the CH by sending only one information, e.g. the centroids of the clusters, from each cluster to the sink. Researchers have proposed a lot of clustering techniques for various types of data. One of the most popular algorithms in data clustering is K-means [36]; it is flexible, simple, already adapted to huge number of applications and used with various kinds of data [37–39].

Typically, the K-means is an iterative algorithm in which the process starts by randomly selecting an initial centroid for each cluster. Then, each data set is assigned to the nearest centroid, according to the Euclidean distance (see equation 7), and the first round of cluster formation is performed. After that, the cluster centroids are updated and the process is repeated until the convergence of the criterion function (Algorithm 6.2).

Algorithm 5 K-means Algorithm.

Require: Set of reading sets: $R^p = \{R_1^p, R_2^p, \ldots, R_n^p\}$, Cluster number: K.
Ensure: Set of clusters $C = \{C_1, C_2, \ldots, C_K\}$.
1: **for** $j \leftarrow 1$ to K **do**
2: randomly choose centroid c_j among R^p belongs to C_j
3: **end for**
4: **repeat**
5: **for** each data set $R_i^p \in R^p$ **do**
6: Assign R_i^p to the cluster $C_j$with nearest c_j (i.e., $E_d(R_i^p, R_{j*}^p) \leq E_d(R_i^p, R_j^p)$; $j \in \{1, \ldots K\}$)
7: **end for**
8: **for** each cluster $C_j, where\, j \in \{1, \ldots K\}$ **do**
9: Update the centroid c_i to be the centroid of all data readings currently in C_j
10: **end for**
11: **until** no change in the cluster memberships
12: return C

6.3 Hybrid-Based In-Node Reduction Technique

Obviously, in-network aggregation and data clustering approaches are quite different from the redundancy elimination point of view. Thus, they have different impacts regarding various performance metrics, especially number of periodic packets sent and data accuracy. Since the first approach searches the redundant data sets in pairs instead of groups in the second one, it saves the data integrity more than the other one. However, the data clustering saves the sensor energy more than the in-network aggregation because it limits the number of transmitted packets to the cluster centroids. Thus, in order to ensure a trade-off between both metrics, we propose a hybrid in-node reduction approach to apply over the data sets received by the CH at each period.

Let first recall the four types of packets received by a CH during a period: 1) Off_Packet indicating that the data set collected at the current period is similar to that sent in the previous one; 2) $Aggregate\_Packet$ containing the data aggregated according to the Algorithm 1; 3) $Compressed\_Packet$ containing the data compressed according to the Algorithm 2; 4) $Predicted\_Packet$ containing the coefficient set calculated based on the Newton forward formula 4. Therefore, the forwarded packets from the CH to the sink can be shown according to the in-node decision algorithm (Algorithm 6.3). First, all packets of types Off_Packet and $Predicted\_Packet$ will be added to the final list of sets sent to the sink, e.g. I (lines 4-7). Indeed, such types of packets do not consume the energy of CH because they contain no data (e.g. Off_Packet) or a few data values (coefficient set in $Predicted\_Packet$). Then, for the sen-

sors sending aggregated packets, the CH applies the in-network aggrgeation approach in order to remove the redundancy among them and reduce the number of packets sent to the sink. Finally, the CH applies the K-means algorithm to the data sets compressed by the sensors (lines 10-12 and 16).

Algorithm 6 In-Node Reduction Algorithm.

Require: set of reading sets: $R^p = \{R_1^p, R_2^p, \ldots, R_n^p\}$, cluster number: K, Euclidean distance threshold: t_E.
Ensure: Final list of sent packets: I.
1: $I \leftarrow \emptyset$
2: $A \leftarrow \emptyset$
3: $C \leftarrow \emptyset$
4: **for** each $R_i^p \in R^p$ **do**
5: **if** R_i^p is of type Off_Packet or $Predicted\_Packet$ **then**
6: $I \leftarrow I \cup \{R_i^p\}$
7: **else**
8: **if** R_i^p is of type $Aggregate\_Packet$ **then**
9: $A \leftarrow A \cup \{R_i^p\}$
10: **else**
11: $C \leftarrow C \cup \{R_i^p\}$
12: **end if**
13: **end if**
14: **end for**
15: $I \leftarrow I \cup$ In-Network_Aggregation(A, t_E)
16: $I \leftarrow I \cup$ Data_Clustering(C, K)
17: **return** I

7 Simulation Results

In order to evaluate the performance of our mechanism, we used real sensor data collected from Intel Berkeley Research Lab [40]. This data contains readings for 46 sensors recording environmental condition including temperature, humidity, light and voltage. Every 31 seconds, the sensor collects new reading for each feature then it sends toward the sink for archive purpose. In our simulation, we used a file that includes a log of about 50000 readings for each sensor. We assume that each sensor reads the data from its corresponding file for a period of time, then it sends them toward a CH placed at the center of the lab after applying our mechanism. We implemented the algorithms used in our mechanism based on Java simulator and we compared the obtained results to those obtained in the PFF [19] and S-LEC [14].

Table 4 summarizes the parameters used in our simulation with their tested values.

Parameter	Symbol	Values
Aggregate threshold	δ	0.05, 0.1, 0.2
Pearson threshold	ε	0.4, 0.5, 0.6, 0.7
Prediction threshold	c	4, 5, 6
Period size	$\mathcal{F}$	50, 100, 250
ANOVA thresholds	α_0, α_1	0.01, 0.05
Initial sensor energy	E_i	5 mJ
Critical energy threshold	E_c	$\frac{E_i}{2}$
Round size	π	2 periods
Jaccard threshold	t_J	0.7
Eulidean distance threshold	t_E	0.4
Clusters number	K	4, 6, 8

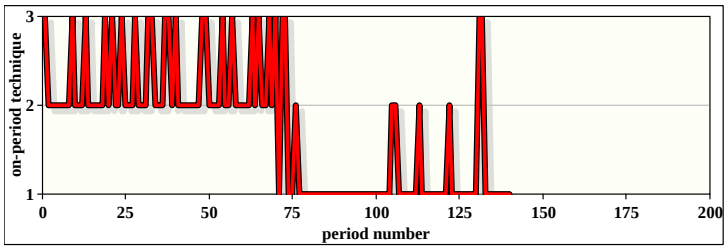
Table 4 Simulation environment.

7.1 On-Period Decision Study

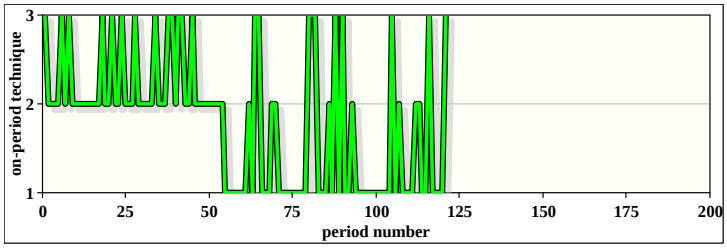
Fig. 4 shows which on-period technique has been selected by a sensor at the end of each period based on the on-period decision table. In each subfigure (4(a), 4(b) and 4(c)) represents prediction, compression and aggregation techniques respectively. The obtained results confirm the behavior of our proposed technique as follows: 1) when its remaining energy is high, the sensor selects between compression and aggregation in order to ensure a high data accuracy along with the reduced amount of data transmission; 2) when its remaining energy becomes low, the sensor applies the prediction technique, except if the data redundancy is low, in order to reduce to the minimum its data transmission while saving the information integrity. We can also observe that the lifetime of the sensor is more extended with the light condition compared to temperature and humidity; this indicates that the light readings are highly redundant compared to other ones thus the sensor can more reduce its data transmission by applying either compression or prediction techniques.

7.2 In-Period Decision Study

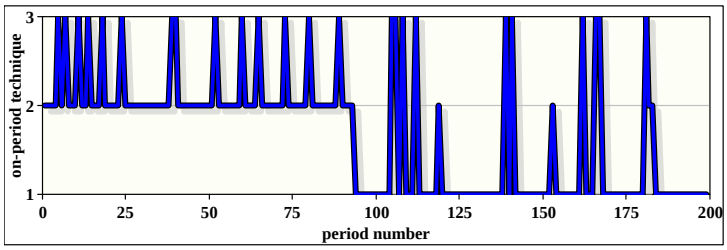
Fig. 5 shows the decision made by the sensor at the end of each round according to the in-period decision table. Subsequently, the numbers in the y-axes are describing as follows: 1, 3 and 5 indicate a low battery level with low, medium and high data similarity respectively; 2, 4, and 6 indicate a high battery level with low, medium and high data similarity respectively. The obtained results reveal several observations: 1) the sensing frequency of the sensor is dynamically adapted after each round in each of the three conditions (temperature, humidity and light). 2) By analyzing the new sensing frequencies of the sensors, we observe that the light condition reduces its data collection more than the other conditions because the light readings are more similar compared to other ones. Hence, we observe that the light sensor, mostly, selects between the fifth and sixth in-period techniques depending on its battery level, e.g. low



(a) temperature



(b) humidity



(c) light

Fig. 4 Variation of the on-period technique selected by the sensor at each period, $\mathcal{F} = 50$, $\delta = 0.1$, $\varepsilon = 0.5$, $c = 5$.

or high. Otherwise, the data similarity level of temperature readings almost varies between low and medium, thus its sensing frequency varies between 1 and 4, while the humidity readings are more redundant than temperature and it varies between 1 and 5.

Based on the selected in-period technique, Fig. 6 shows the new sensing frequencies of a sensor after adapting its sampling rate after each round. Because the light readings are very similar, the light sensor adapts its sensing frequencies to the minimum in order to avoid collecting redundant data, e.g. 40% when its battery level is low and 20% when its battery level is high. On the other hand, the temperature and humidity readings are less similar than those of light, thus they adapt their sensing frequencies less than the light sensor, e.g. mostly between 20% and 50% for the temperature and between 10% and 50% for the humidity.

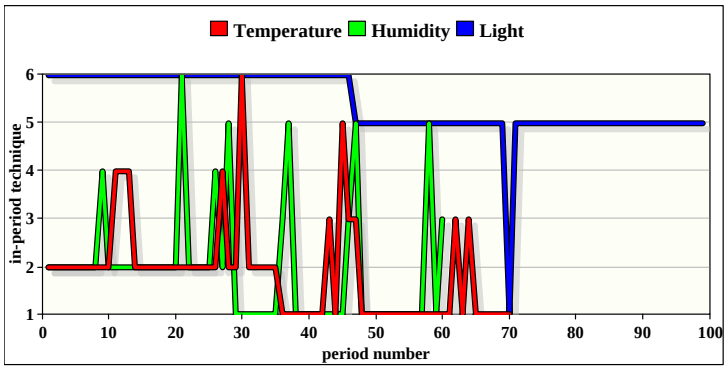


Fig. 5 Variation of the in-period decision made by the sensor at each round, $\mathcal{F} = 50$, $\delta = 0.1$, $\varepsilon = 0.5$, $c = 5$.

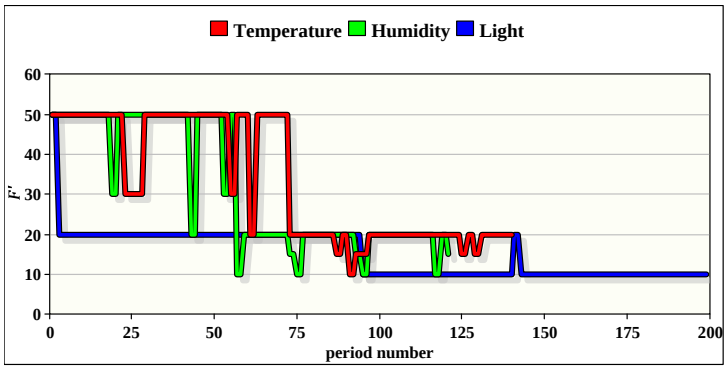
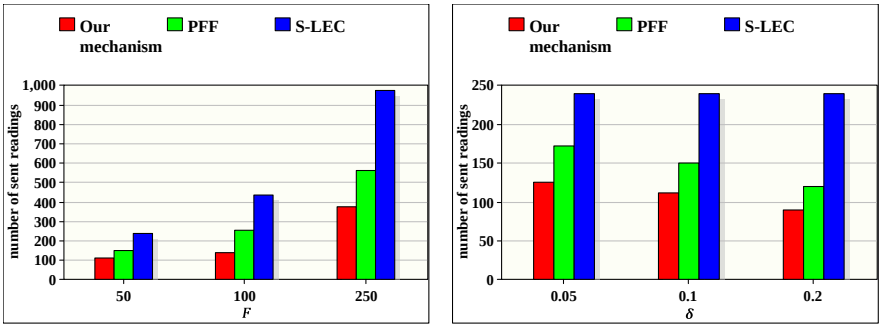


Fig. 6 Variation of the sensing frequency of a sensor during periods, $\mathcal{F} = 50$, $\delta = 0.1$, $\varepsilon = 0.5$, $c = 5$.

7.3 Data Transmission Ratio at Sensor

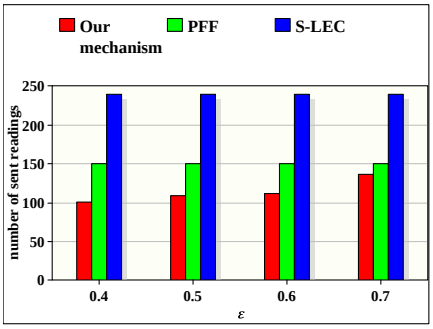
Fig. 7 shows the number of readings sent from each sensor to the CH after applying both on-period and in-period techniques, for 15 periods of simulations. The results are dependent on the period size (Fig. 7(a)), the aggregate threshold (Fig. 7(b)), the compression threshold (Fig. 7(c)) and the prediction polynomial degree (Fig. 7(d)). We observe that our mechanism can reduce the data transmission to the CH more than the PFF and S-LEC in all cases. Subsequently, it allows each sensor to send 9% to 45% of data less than PFF and 28% to 67% of data less than S-LEC. Furthermore, the obtained results show that: 1) the data transmission from the sensor, using our mechanism, increases with the increasing values of the period size (Fig. 7(a)) and the compression threshold (Fig. 7(c)). This is because, from one hand, the variance among the data calculated using ANOVA increases when the period size increases and, from the other hand, the collected readings become less redundant when the compression threshold increases. 2) The sensor sends, using our mechanism,

less data to the CH when the aggregated threshold increases (Fig. 7(b)). This is due to the similarity among the collected readings, which increases with the increasing of the aggregate threshold. 3) The data transmission will not be highly affected when varying the predicted polynomial degree.

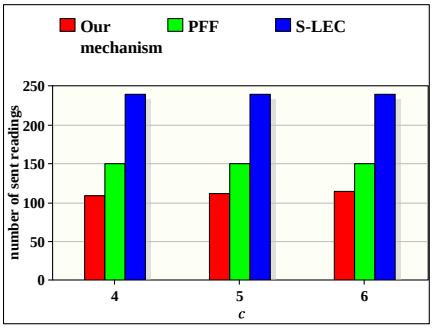


(a) $\delta = 0.1$, $\varepsilon = 0.5$, $c = 5$

(b) $\mathcal{F} = 50$, $\varepsilon = 0.5$, $c = 5$



(c) $\mathcal{F} = 50$, $\delta = 0.1$, $c = 5$



(d) $\mathcal{F} = 50$, $\delta = 0.1$, $\varepsilon = 0.5$

Fig. 7 Number of readings sent from each sensor to the CH.

7.4 Energy Consumption in Sensor

As previously mentioned, the energy consumed in the sensor node is highly related to the amount of its transmitted data. Fig. 8 shows the remaining energy of temperature, humidity and light sensors in function of the period progress. In our simulations, we implemented the Heinzelman model proposed in [41] as one of the most models used to evaluate the energy consumption in sensor networks. Accordingly to this model, the energy consumption highly depends on the transmission and receiving operations while negligating the other factors (sensing and processing). Thus, the energy consumption of a sensor for transmitting its set of data $R_i^{'p}$ with size $|R_i^{'p}|$ to the CH located at distance d is:

$$E_{TX} = E_{elec} \times |R_i^{'p}| \times 64 + \beta_{amp} \times |R_i^{'p}| \times 64 \times d^2 \quad (9)$$

where 64 indicates the bit representation of each value, and E_{elec} is the energy consumption of a sensor in its electronic circuitry (usually $E_{elec} = 50\,nJ/bit$), and β_{amp} represents the energy consumption in RF amplifiers to compensate the loss (usually $\beta_{amp} = 100\,pJ/bit$).

Obviously, the remaining energy in each sensor proportionally decreases depending on the amount of data transmitted, with the progress of the period number. Subsequently, more the amount of data is reduced at each period, e.g. using on-period, and more the sensing frequency of the sensor is minimized at each round then less the available energy will be depleted. This supports the extension of the light sensor lifetime compared to those of other sensors due to the high redundancy level existing among light readings.

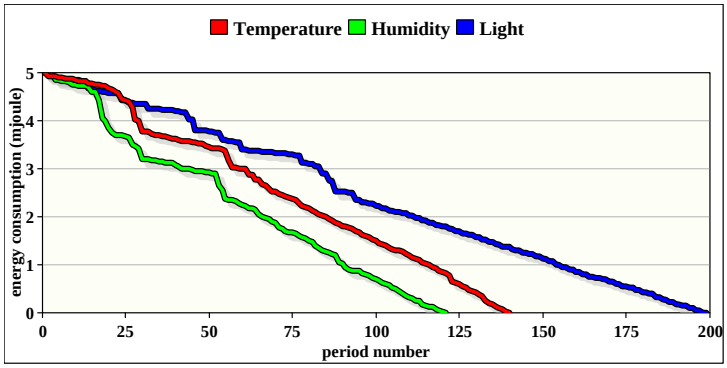


Fig. 8 Remaining energy in a sensor in function of the period progress, $\mathcal{F} = 50$, $\delta = 0.1$, $\varepsilon = 0.5$, $c = 5$.

7.5 Packet Types Study at CH

In Fig. 9, we study the types of packets (Off_Packet, $Aggregate\_Packet$, $Compressed\_Packet$ and $Predicted\_Packet$) received by the CH at each period. The obtained results show that the number of packets for each type can differ from one period to another for the same sensor (e.g. temperature, humidity or light) or they can differ for the different sensors at the same period. We can also observe that most of the received packets are of type $Compressed\_Packet$ followed by the $Aggregate\_Packet$, $Predicted\_Packet$ and Off_Packet respectively, for various kind of sensors and for all periods. This is because compression is a compromised decision between aggregation and prediction approaches for energy saving and data accuracy at the same time. Furthermore, the results shows that the CH is receiving more packets

of type *Predicted_Packet* starting from the period number 27 for the temperature and humidity readings, and from the period number 36 for the light readings; this indicated that the energy of the sensors becomes low starting from such periods and the sensors have to reduce their data transmission in order to conserve their power supply. Finally, we observe that some sensors are delivering Off_Packet to the CH indicating that the readings collected in successive periods are similar.

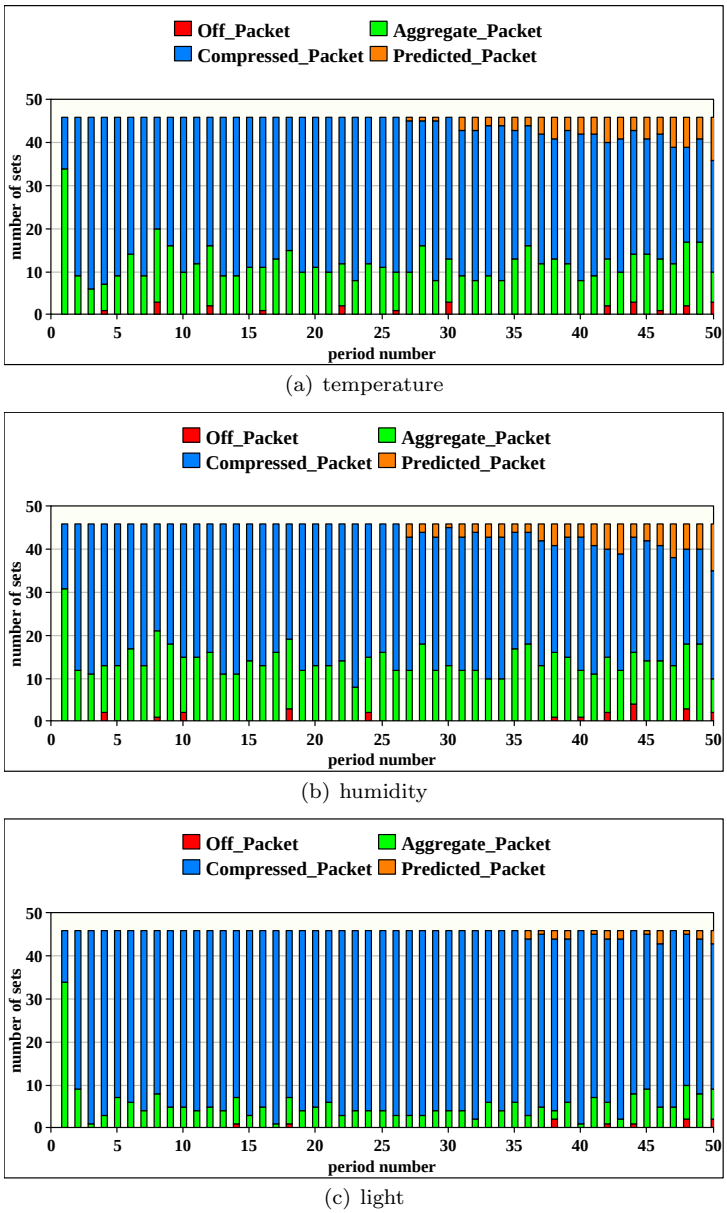
7.6 In-Node Decision Study

Fig. 10 shows the number of sets periodically sent from the CH to the sink after applying the in-node reduction algorithm (Algorithm 6). In addition to the Off_Packet and *Predicted_Packet*, the CH sends a subset of the *Aggregate_Packet*, after removing the redundancy among them (Algorithm 4), and a subset of the compressed packets, after making them in clusters (Algorithm 5), to the sink. Thus, the obtained results are dependent on the period size ($\mathcal{F}$), the aggregation threshold (δ) and the number of clusters (K) (Fig. 10(a) to 10(c)) while they are not affected by the changing of the predicted polynomial degree (Fig. 10(d)). Subsequently, we observe, using our mechanism, that the periodic number of sent sets decreases when the values of $\mathcal{F}$ or K decrease, or the value of δ increases. This is because when $\mathcal{F}$ decreases or δ increases the similarity among the sensor sets will increase thus the CH will send less sets to the sink in order to avoid sending redundant data sets. Whilst, the decreasing of the cluster number leads to decrease the number of cluster centroids send to the sink. Furthermore, we observe that our mechanism outperforms PFF from 20% to 40% and S-LEC from 56% to 73% in terms of reducing the number of packets sent to the sink.

In Fig. 11, we show an illustrative example of the packet types received by the CH during a period and after applying K-means over the *Compressed_Packet*. During this period, we observe that the CH receives 2 packets of type Off_Packet, 3 packets of type *Predicted_Packet*, 15 packets of type *Aggregate_Packet* and 26 packets of type *Compressed_Packet*. Thus, after dividing the *Compressed_Packet* into 4 clusters, the following observations are eminent: 1) the sets are unequally distributed to the clusters; this is due to the random selection of the cluster centroids and the convergence function used in K-means. 2) The sensors in the same cluster are not necessary spatially correlated. 3) The temporal correlation among sensors can happen even they are not spatially correlated.

8 Conclusion and Future Work

Data reduction will remain one of the main concerns for researchers in order to extend the sensing-based IoT applications and deliver a useful data for the end user. In this paper, we proposed a hybrid-based data collection mechanism, called All-in-One, with the aim to reduce the data transmission at several



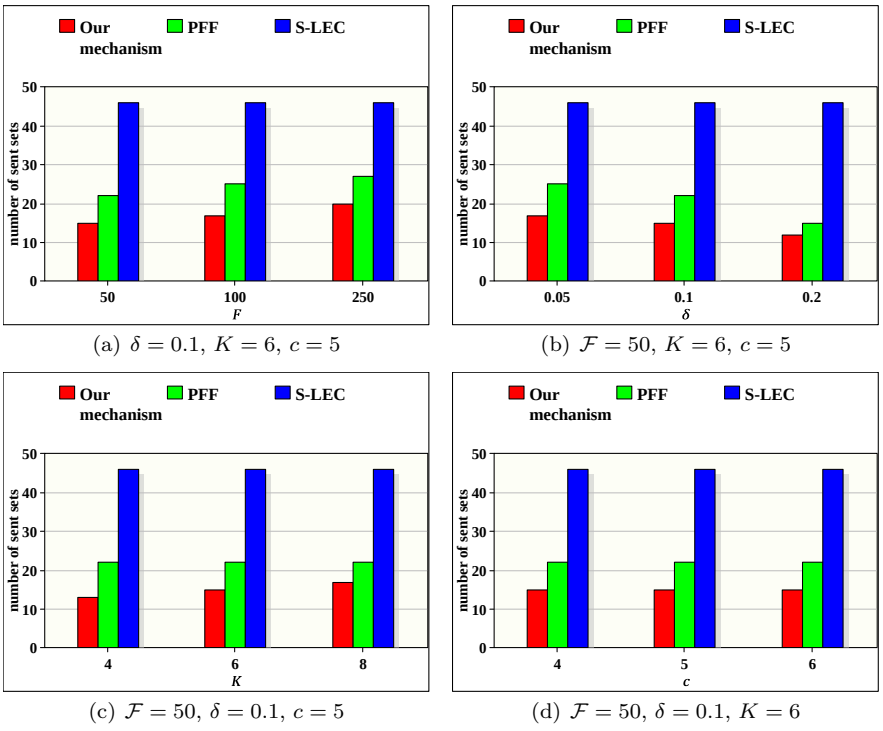
(a) temperature

(b) humidity

(c) light

Fig. 9 Variation of periodic packet types received by the CH.

stages in the network. The proposed mechanism allows to remove the redundancy existing among the collected data on on-period, in-period and in-node levels. Furthermore, on each level, we introduced several data reduction techniques while proposing hybrid-based approaches in order to optimize several



(a) $\delta = 0.1$, $K = 6$, $c = 5$ (b) $\mathcal{F} = 50$, $K = 6$, $c = 5$

(c) $\mathcal{F} = 50$, $\delta = 0.1$, $c = 5$ (d) $\mathcal{F} = 50$, $\delta = 0.1$, $K = 6$

Fig. 10 Number of sets sent periodicaly from the CH to the sink.

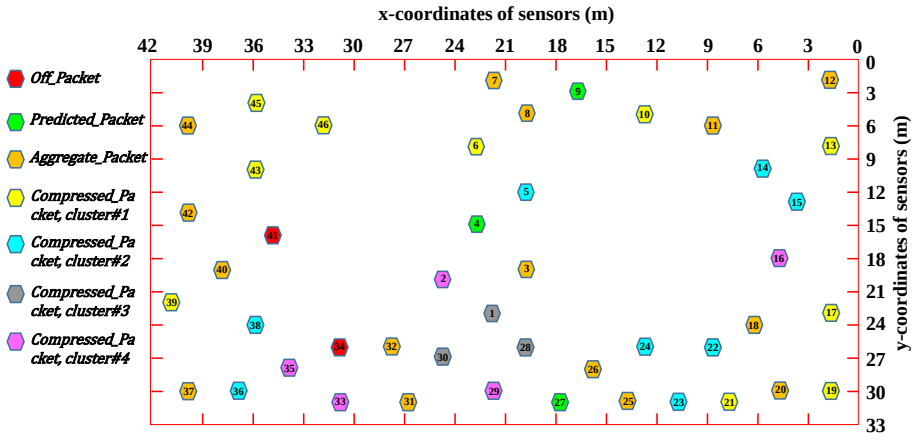


Fig. 11 Illustrative example of packet types received by the CH during a period and after applying K-means over the compressed packets, $K = 4$.

performance metrics of the network. We conducted extensive simulations on

real sensor data in order to evaluate the efficiency of our mechanism compared to other exiting techniques.

As future work, we have several directions to optimize our mechanism at both sensor and CH nodes. From one hand, it is important to add a shift phase between successive transmissions in order to avoid collision in the network. In addition, we seek to add other parameters when adapting the sensing frequencies of the sensors like correlation with other nodes. At the CH level, we plan to add a scheduling strategy in order to switch the correlated nodes into sleep/active modes. Thus, the overall network energy will be improved and the collision will be reduced.

9 Conflict of Interest

The authors declare that they have no conflict of interest.

References

1. Parvaneh Asghari, Amir Masoud Rahmani and Hamid Haj Seyyed Javadi, *Internet of Things applications: A systematic review*, Computer Networks, Vol. 148, pp. 241-261, 2019.
2. Alireza Souri and Monire Norouzi, *A state-of-the-art survey on formal verification of the internet of things applications*, Journal of Service Science Research, Vol. 11, Iss. 1, pp. 47-67, 2019.
3. WZ Khan, MH Rehman, HM Zangoti, MK Afzal, N. Armi and K. Salah, *Industrial internet of things: Recent advances, enabling technologies and open challenges*, Computers & Electrical Engineering, Vol. 81, pp. 106522, 2020.
4. Soroush Abbasian Dehkordi, Kamran Farajzadeh, Javad Rezazadeh, Reza Farahbakhsh, Kumbesan Sandrasegaran and Masih Abbasian Dehkordi, *A survey on data aggregation techniques in IoT sensor networks*, Wireless Networks, Vol. 26, Iss. 2, pp. 1243-1263, 2020.
5. S Pushpalatha and KS Shivaprakasha, *Energy-Efficient Communication Using Data Aggregation and Data Compression Techniques in Wireless Sensor Networks: A Survey*, Advances in Communication, Signal Processing, VLSI, and Embedded Systems, pp. 161-179, 2020.
6. Gabriel Martins Dias, Boris Bellalta and Simon Oechsner, *A survey about prediction-based data reduction in wireless sensor networks*, ACM Computing Surveys (CSUR), Vol. 49, Iss. 3, pp. 1-35, 2016.
7. Sivadi Balakrishna and M Thirumaran, *Semantics and Clustering Techniques for IoT Sensor Data Analysis: A Comprehensive Survey*, Principles of Internet of Things (IoT) Ecosystem: Insight Paradigm, pp. 103-125, 2020.
8. Hassan Harb and Abdallah Makhoul, *Energy-efficient sensor data collection approach for industrial process monitoring*, IEEE Transactions on Industrial Informatics, Vol. 14, Iss. 2, pp. 661-672, 2017.
9. Peng Zeng, Bofeng Pan, Kim-Kwang Raymond Choo and Hong Liu, *MMDA: Multidimensional and Multidirectional Data Aggregation for Edge Computing-Enhanced IoT*, Journal of Systems Architecture, pp. 101713, 2020.
10. Jing Zhang, Zhiwei Lin, Pei-Wei Tsai and Li Xu, *Entropy-driven data aggregation method for energy-efficient wireless sensor networks*, Information Fusion, Vol. 56, pp. 103-113, 2020.
11. Osama M Bushnaq, Abdulkadir Celik, Hesham ElSawy, Mohamed-Slim Alouini and Tareq Y Al-Naffouri, *Aeronautical Data Aggregation and Field Estimation in IoT Networks: Hovering and Traveling Time Dilemma of UAVs*, IEEE Transactions on Wireless Communications, Vol. 18, Iss. 10, pp. 4620-4635, 2019.

12. Ata Ullah, Ghawar Said, Muhammad Sher and Huansheng Ning, *Fog-assisted secure healthcare data aggregation scheme in IoT-enabled WSN*, Peer-to-Peer Networking and Applications, Vol. 13, Iss. 1, pp. 163-174, 2020.
13. Ben Othman Soufiene, Abdullah Ali Bahattab, Abdelbasset Trad and Habib Youssef, *Lightweight and confidential data aggregation in healthcare wireless sensor networks*, Transactions on Emerging Telecommunications Technologies, Vol. 27, Iss. 4, pp. 576-588, 2016.
14. Yao Liang and Yimei Li, *An efficient and robust data compression algorithm in wireless sensor networks*, IEEE Communications Letters, Vol. 18, Iss. 3, pp. 439-442, 2014.
15. Qinbao Xu, Rizwan Akhtar, Xing Zhang and Changda Wang, *Cluster-based arithmetic coding for data provenance compression in wireless sensor networks*, Wireless Communications and Mobile Computing, Vol. 2018, 2018.
16. Chacko John Deepu, Chun-Huat Heng and Yong Lian, *A hybrid data compression scheme for power reduction in wireless sensors for IoT*, IEEE transactions on biomedical circuits and systems, Vol. 11, Iss. 2, pp. 245-254, 2016.
17. Xiaobin Xu and Guangwei Zhang, *A hybrid model for data prediction in real-world wireless sensor networks*, IEEE Communications Letters, 2017.
18. Hidaya Liazid, Mohamed Lehsaini and Abdelkrim Liazid, *An improved adaptive dual prediction scheme for reducing data transmission in wireless sensor networks*, Wireless Networks, Vol. 25, Iss. 6, pp. 3545-3555, 2019.
19. Adrien Russo, François Verdier and Benoît Miramond, *Energy saving in a wireless sensor network by data prediction by using self-organized maps*, Procedia computer science, Vol. 130, pp. 1090-1095, 2018.
20. Metehan Guzel, Ibrahim Kok, Diyar Akay and Suat Ozdemir, *ANFIS and Deep Learning based missing sensor data prediction in IoT*, Concurrency and Computation: Practice and Experience, Vol. 32, Iss. 2, pp. e5400, 2020.
21. Siguang Chen, Shujun Zhang, Xiaoyao Zheng and Xiukai Ruan, *Layered adaptive compression design for efficient data collection in industrial wireless sensor networks*, Journal of Network and Computer Applications, Vol. 129, pp. 37-45, 2019.
22. G Pius Agbulu, G Joselin Retna Kumar and A Vimala Juliet, *A lifetime-enhancing cooperative data gathering and relaying algorithm for cluster-based wireless sensor networks*, International Journal of Distributed Sensor Networks, Vol. 16, Iss. 2, pp. 1550147719900111, 2020.
23. Kashif Naseer Qureshi, Muhammad Umair Bashir, Jaime Lloret and Antonio Leon, *Optimized Cluster-Based Dynamic Energy-Aware Routing Protocol for Wireless Sensor Networks in Agriculture Precision*, Journal of Sensors, Vol. 2020, 2020.
24. Dinesh Kumar Kotary and Satyasai Jagannath Nanda, *Distributed robust data clustering in wireless sensor networks using diffusion moth flame optimization*, Engineering Applications of Artificial Intelligence, Vol. 87, pp. 103342, 2020.
25. Carol Habib, Abdallah Makhoul, Rony Darazi and Christian Salim, *Self-adaptive data collection and fusion for health monitoring based on body sensor networks*, IEEE transactions on Industrial Informatics, Vol. 12, Iss. 6, pp. 2342-2352, 2016.
26. Mehmet Başaran, Stephan Schlupkothen and Gerd Ascheid, *Adaptive Sampling Techniques for Autonomous Agents in Wireless Sensor Networks*, 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1-6, 2019.
27. Sabah Abdul-Wahab, Yassine Charabi, Selma Osman, Kaan Yetilmezsoy, and Isra Ibrahim Osman, *Prediction of optimum sampling rates of air quality monitoring stations using hierarchical fuzzy logic control system*, Atmospheric Pollution Research, Vol. 10, Iss. 6, pp. 1931-1943, 2019.
28. Yuan Rao, Gang Zhao, Wen Wang, Jingyao Zhang, Zhaohui Jiang and Ruchuan Wang, *Adaptive Data Acquisition with Energy Efficiency and Critical-Sensing Guarantee for Wireless Sensor Networks*, Sensors, Vol. 19, Iss. 12, pp. 2654, 2019.
29. Jacques Bahi, Abdallah Makhoul and Maguy Medlej, *A two tiers data aggregation scheme for periodic sensor networks*, Ad Hoc & Sensor Wireless Networks, Vol. 21, Iss. (1-2), pp. 77-100, 2014.
30. Mou Wu, Liansheng Tan and Naixue Xiong, *A structure fidelity approach for big data collection in wireless sensor networks*, Sensors, Vol. 15, Iss. 1, pp. 248-273, 2015.

31. Hassan Harb, Abdallah Makhoul and Raphaël Couturier, *An enhanced K-means and ANOVA-based clustering approach for similarity aggregation in underwater wireless sensor networks*, IEEE Sensors Journal, Vol. 15, Iss. 10, pp. 5483-5493, 2015.
32. Yifan Yin, Boyi Xu, Hongming Cai and Han Yu, *A Novel Temporal and Spatial Panorama Stream Processing Engine on IOT Applications*, Journal of Industrial Information Integration, pp. 100143, 2020.
33. Hassan Harb, Abdallah Makhoul, Rami Tawil and Ali Jaber, *A suffix-based enhanced technique for data aggregation in periodic sensor networks*, 2014 international wireless communications and mobile computing conference (IWCMC), pp. 494-499, 2014.
34. George W Snedecor and William G Cochran, *Statistical Methods, eight edition*, Iowa state University press, Ames, Iowa, 1989.
35. Abdallah Makhoul, Hassan Harb and David Laiymani, *Residual energy-based adaptive data collection approach for periodic sensor networks*, Ad Hoc Networks, Vol. 35, pp. 149-160, 2015.
36. James MacQueen and others, *Some methods for classification and analysis of multivariate observations*, Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1, Iss. 14, pp. 281-297, 1967.
37. P Govender and V Sivakumar, *Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019)*, Atmospheric Pollution Research, Vol. 11, Iss. 1, pp. 40-56, 2020.
38. K Lavanya, Rani Kashyap, S Anjana and Sumaiya Thasneen, *An Enhanced K-Means MSOINN Based Clustering Over Neo4j with an Application to Weather Analysis*, International Conference on Intelligent Computing and Smart Communication 2019, pp. 451-461, 2020.
39. Guiqing Zhang, Yong Li, and Xiaoping Deng, *K-Means Clustering-Based Electrical Equipment Identification for Smart Building Application*, Information, Vol. 11, Iss. 1, pp. 27, 2020.
40. Samuel Madden, *Intel lab data*, http://db.csail.mit.edu/labdata/labdata.html, 2004.
41. Wendi Beth Heinzelman, *Application-specific protocol architectures for wireless networks*, Thesis at Massachusetts Institute of Technology, 20000.