On the performance of data-driven approaches for energy efficiency on WiFi and LoRa-based sensors: an experimental study

Gaby Bou Tayeh[a], Joseph Azar[a], Abdallah Makhoul[a], Christophe Guyeux[a], and Jacques Demerjian[b]
[a]*Femto-St Institute, UMR 6174 CNRS, Université de Bourgogne Franche-Comté, France*
Email: gaby.boutayeh@gmail.com; joseph.azar@univ-fcomte.fr; abdallah.makhoul@univ-fcomte.fr; christophe.guyeux@univ-fcomte.fr
[b]*LaRRIS, Faculty of Sciences, Lebanese University*
Email: jacques.demerjian@ul.edu.lb

***Abstract*—Most research on energy efficiency in wireless sensor networks has concluded that the communication subsystem consumes significantly more energy than the sensing and computing subsystems and that communication should be traded for computation. This paper does an experimental study with a Pysense sensor shield that utilizes WiFi and LoRa. It employs two data management strategies for energy conservation that have been demonstrated to be efficient through simulations and examines the obtained results on real hardware. The findings of this paper indicate that concentrating on lowering the energy consumption of the communication subsystem was advantageous while utilizing WiFi but was less effective and promising when using LoRa. Additionally, it demonstrates the importance of addressing many subsystems in order to extend the life of LoRa-based devices.**

***Index Terms*—Lora, LPWAN, Wireless sensor networks, energy efficiency, Internet of Things, power management**

I. Introduction

Over the last few years, the rapid expansion of sensor networks has intensified. Wireless sensor network (WSN) is a term that refers to a collection of spatially scattered and dedicated sensors that are used to monitor and record the physical environmental conditions and to organize the acquired data at a central point referred to as a Sink. WSN continues to attract the interest of academia, equipment, chip manufacturing sectors, and service providers capable of offering a variety of user-specific or multi-featured applications via WSN.

One of the critical issues in IoT is the network energy consumption [1], which is expanding at a fast pace as data rates increase, the number of Internet-enabled services increases, and the number of Internet-connected edge devices proliferates. The future Internet of Things will significantly increase network load and power consumption. Thus, green and energy-efficient technologies must be used to maximize the energy conservation of network devices. It has long been recognized in WSN research that the radio module is the primary component responsible for sensor node battery drain [2]. For instance, the authors in [3] reports that communication is more than 1000 times more energy-intensive than completing a basic aggregation process. On the other hand, the authors in [4] computed the energy cost of various operations, demonstrating that the sensor's sensing energy cost is comparable to the radio's cost. This, however, is only limited to the Extreme Scale Mote system. The authors demonstrated in [5] that the energy consumption of the sensing operation is comparable to or even more significant than that of the radio in several practical applications. They found that energy consumption for sensing is not always inexpensive, particularly for power-hungry sensors such as gas, flow control, or level sensors.

After reviewing the state-of-the-art of energy efficiency in WSN, one can conclude that, in general and for the majority of published articles, the emphasis on minimizing energy consumption associated with radio communication was the most critical factor in attaining energy efficiency. A recent survey on energy efficiency published in 2020 [6] reflects what the majority of research has established, namely that "most of the energy consumption is exhausted for transmitting." Numerous energy-saving strategies could be used to address the issue of battery-powered IoT devices consuming too much energy due to radio communication [7]. This paper is concerned with data reduction methods, a category of solutions that aims to reduce the amount of data delivered to the sink.

This paper is an experimental study in the field of energy efficiency for IoT, and the hypothesis it will attempt to demonstrate is that, while focusing on reducing energy consumption associated with radio communication when using WiFi has been shown to be effective in numerous research studies, this strategy may fail when using LoRa-based sensor networks.

To demonstrate the point made above, this paper will examine two data-driven energy management approaches that have been proposed for energy efficiency and transmission reduction and shown through simulations to increase the lifetime of a sensor node. It will then implement these approaches on real hardware and examine the energy consumption when using WiFi and LoRa. The experimental findings obtained will demonstrate that while utilizing LoRa, radio communication is not the primary and only module that matters. Thus, the assertion that "transmission reduction is the most energy-efficient strategy" is untrue.

The rest of this paper is as follows. Section II discusses the two data-driven approaches for energy management used in this paper. Section III presents the hardware used for the experiments. Sections IV and V respectively discuss the experiments done using WiFi and LoRa and the conclusion is presented in section VI.

II. Related work

Numerous optimization approaches, such as the data-driven approach, could be considered for resolving the energy efficiency problem. Data compression, aggregation, adaptive sampling, and data prediction are some of the methods used in this approach (Figure 1) [8]–[11]. The algorithms considered for this paper are the Fault-Tolerant Data Transmission Reduction (FTDTR) method [12] and the Adaptive Sampling Transmission Reduction (ASTR) method [13]. The former falls within the data prediction category, while the latter combines adaptive sampling with data prediction.

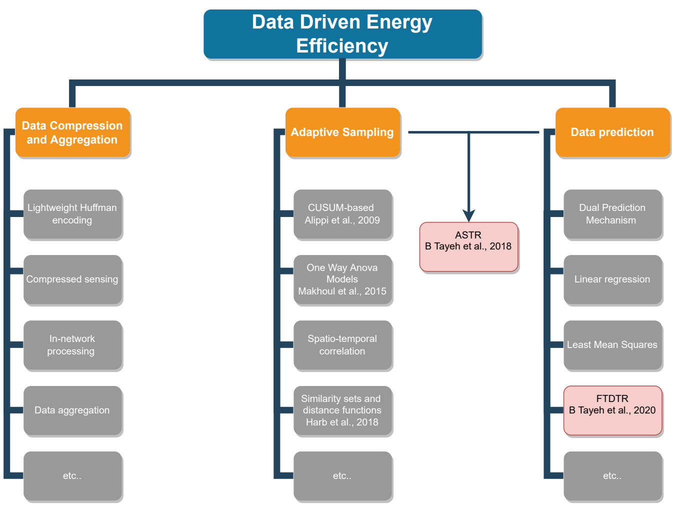

Fig. 1: Some of the data-driven methods for energy efficiency. The red boxes are the methods considered for this paper.

The FTDTR approach is based on the assumption of a periodic sensor network in which each sensor collects and transmits data every specified number of seconds. The primary goal of this method is to develop a prediction model that the sensor and sink can use. After activation, the sensor transmits the first two measurements obtained to the sink. The sensor and sink then compute the temporal correlation between recently acquired and upcoming measurements. Both the sensor and the sink can update the prediction model as needed by keeping the most recently communicated data. The efficient aspect of this method is that the sensor is no longer necessary to transmit any collected measurements to the sink as long as the sink can predict them within a defined error boundary.

The ASTR method enhances the FTDTR method by using an adaptive sampling technique. Adaptive sampling is an approach that enables the sensor node to adjust its sampling rate in response to changes in the data collected over a specified time period. If no significant change is detected, the sensor node's sampling rate may gradually be reduced, and it may sleep for an extended period of time. The authors in [13] employed the Kruskal-Wallis statistic model to determine whether there is a difference between sampled data points from period P_t and sampled data points from period P_{t-1}. The advantage of ASTR over FTDTR is that it takes into account not only the radio communication module but also the sensing module and the wake-up time when evaluating energy consumption.

The simulations run on real data sets in [12] and [13] will be tested and validated on real hardware in the subsequent sections of this paper.

III. Experimental setup

The energy management methods described in Section II will be implemented on the Pysense sensor shield[1] shown in Figure 2. In addition to the overall energy consumption of each method, the energy consumption of each activity (transmission, sensing, processing, and idle) is measured independently. To accomplish this, the USB current tester "UM24C" was utilized. This device is capable of measuring currents ranging from 0 to 5000 Amperes. Additionally, it monitors current variations in real-time and automatically estimates the consumed current in mAh and energy in mWh. The data collected is transferred via Bluetooth to a laptop/mobile device, where it is visualized. The UM24C is depicted in Figure 3, together with the mobile and desktop applications. Note that the MicroPython implementation of the Python 3 programming language was used to implement the FTDTR and ASTR methods. To test the implemented methods with WiFi and LoRa, the LoPy4 Micropython-programmable quadruple bearer board has been used[2].

The IoT device is set to follow a cycle in which it wakes up every minute, performs a particular task, and then sleeps again. This process is repeated for one hour. The UM24C tester is then used to retrieve the real-time current values that have been collected. The energy consumption for this task is calculated by averaging the peak current consumption for each wake-up. Knowing that the device consumes between 2 and 8×10^{-6} Amps during deep sleep, we ignored these numbers and assumed that each peak begins when the current exceeds 8×10^{-6} Amp and stops when the current drops below 8×10^{-6} Amp. In the following experiments, we were interested in the current consumed by sensing, processing, transmission, and idle activities.

IV. Experiments with WiFi

The first experiment evaluates the FTDTR and ASTR algorithms when used with WiFi. This experiment enables us to investigate the current consumption of sensing, processing, transmission, and idle activities, as well as the ASTR, FTDTR, and Naive (wake, collect, transmit, sleep) approaches.

Figure 4 shows the current consumption in Amp of one of the many peaks recorded for the different activities and algorithms. The idle, sensing and processing peaks are the shortest. This is because these operations do not require a significant

[1]https://pycom.io/product/pysense-2-0-x/

[2]https://pycom.io/product/lopy4/

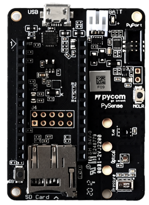

Fig. 2: The Pysense sensor used in this paper's experiments.

Fig. 3: The energy measurement device used to estimate the current consumption.

amount of time to complete, and hence the microcontroller does not remain awake for an extended period of time. It's critical to highlight that the sensors utilized in this experiment are temperature and humidity sensors, which collect samples quickly. Transmission and the Naive approach have the most prolonged peaks. The FTDTR and ASTR techniques produce a peak that is intermediate in length. These findings validate that WiFi transmission does have a significant impact on overall energy consumption.

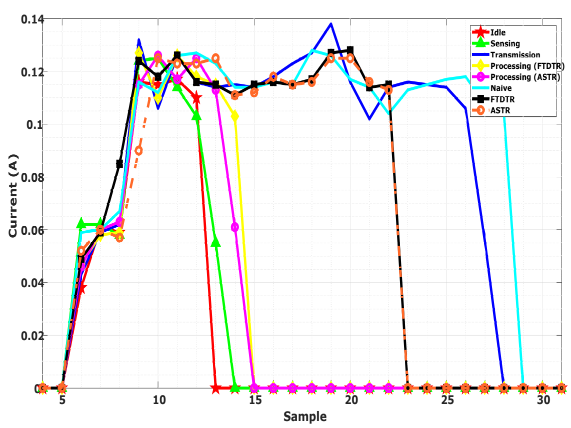

Fig. 4: Peak current consumption comparison when using WiFi.

First, we measure the "E_{Idle}" value. This value represents the amount of current consumed by the microcontroller when waking up. We obtain an average consumption of 0.373 Amp by averaging the recorded current values during the idle phase, i.e. when the sensor wakes up and does nothing.

"$E_{Sensing}$" denotes the energy consumed by the sensing activity. To obtain an approximation of "$E_{Sensing}$", we subtract "E_{Idle}" from "$E'_{Sensing}$" as specified in Equation 1. "$E'_{Sensing}$" represents the energy consumed when waking up and sensing a measurement. Given that the microcontroller must wake up to collect temperature/humidity measurements, we can estimate "$E_{Sensing}$" by disregarding the energy consumed by waking up and doing nothing. "$E_{Sensing}$" has an average of 0.07425 Amp, as determined by the results.

$$E_{Sensing} = E'_{Sensing} - E_{Idle} \tag{1}$$

Similarly, the transmission energy consumption "$E_{Transmission}$" is estimated using "E_{Idle}". Rather than waking up and collecting data, the microcontroller wakes up and transmits a random temperature measurement. To obtain an approximation of "$E_{Transmission}$", we subtract "E_{Idle}" from "$E'_{Transmission}$" as specified in Equation 2. "$E'_{Transmission}$" represents the energy consumed when waking up and sending a random measurement to the sink. "$E_{Transmission}$" has an average of 1.751 Amp.

$$E_{Transmission} = E'_{Transmission} - E_{Idle} \tag{2}$$

The processing energy consumption of the FTDTR technique may be calculated by subtracting "$E'_{Sensing}$" from "$E'_{P_{FTDTR}}$" as denoted in Equation 3, where "$E'_{P_{FTDTR}}$" is the energy consumed when the device is woken up and the FTDTR algorithm is executed. Notably, transmission energy is not considered here because we consider the FTDTR makes a correct prediction and hence there is no need to transmit the data to the sink. The estimated value of "$E_{P_{FTDTR}}$" is 0.1474 Amp.

$$E_{P_{FTDTR}} = E'_{FTDTR} - E'_{Sensing} \tag{3}$$

Similar to "$E_{P_{FTDTR}}$" calculation, the ASTR method's processing energy consumption is estimated as denoted in Equation 4. The estimated value of "$E_{P_{ASTR}}$" is 0.115 Amp.

$$E_{P_{ASTR}} = E'_{ASTR} - E'_{Sensing} \tag{4}$$

We ran ASTR, FTDTR, and the Naive approach on the microcontroller and averaged the peak current consumption for each execution. The estimated values of E_{Naive}, E_{FTDTR}, and E_{ASTR} are 2.489, 0.4832, 0.4888 respectively. The obtained results are illustrated in Figure 5. As can be seen, transmission consumes the most energy compared to other activities, while sensing consumes the least. It's worth noting that temperature and humidity sensors are generally not energy-intensive. If alternative types of sensors were utilized, the results might be different. The idle mode consumes the second most energy, whereas the processing action consumes the third most. Figure 5 also shows that the Naive approach consumes the most energy on average, followed by ASTR and FTDTR. The impact on transmission efficiency is already evident since both ASTR and FTDTR significantly lowered average current consumption compared to the Naive approach. However, we observe that ASTR utilized more energy than FTDTR despite incorporating a mechanism that minimizes sensing and transmission by increasing the deep-sleep duration. This is because we calculated the current consumption only while the sensor was awake and not during deep sleep.

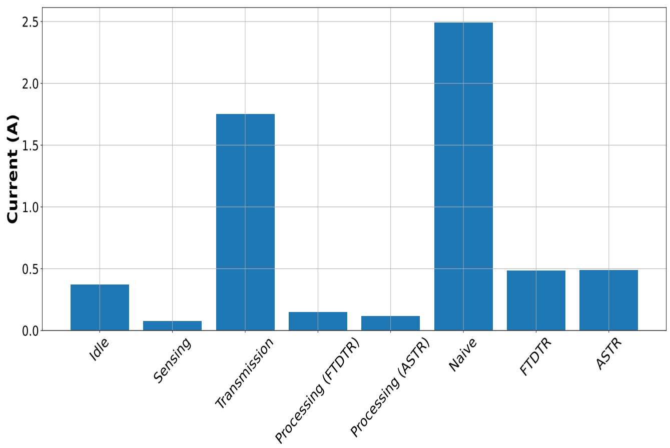

Fig. 5: Average current consumption comparison for the peaks when using WiFi.

Using the average current consumption of each algorithm and considering a 2000 mAh Lipo battery, one may estimate the microcontroller's operational lifetime in days. The lifetime of the microcontroller is estimated in Figure 6 using the Naive, FTDTR, and ASTR approaches. The Naive approach limits the device's life to 4.456 days, and the FTDTR approach enables a 357 percent (20.38 days) improvement in operational lifespan. In contrast to what is displayed in Figure 5, the ASTR was the most efficient and showed a 1535 percent (72.77 days) gain in performance since it reduces wake-ups and keeps the microcontroller in a deep sleep for a longer time. As a result, the results obtained from real implementation are consistent with those obtained from simulations in [12] and [13].

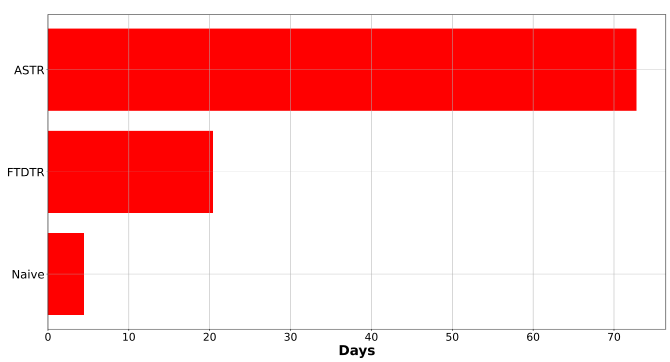

Fig. 6: Lifetime estimate in days when using WiFi.

V. Experiments with LoRa

This section will examine the impact of the approaches presented in Section IV on a microcontroller equipped with a LoRa radio module, as LoRa transmissions consume significantly less power than WiFi transmissions. In this experiment, the WiFi radio model was disabled and only the LoRa radio model was used. The same methodology has been used to calculate peak current consumption, average peak current consumption, and average operational lifespan in days.

Figure 7 compares one of the numerous current consumption peaks across various activities and algorithms. We note that the current peaks for the different activities are shorter than in Figure 4, implying that less current is consumed, especially for the transmission activity.

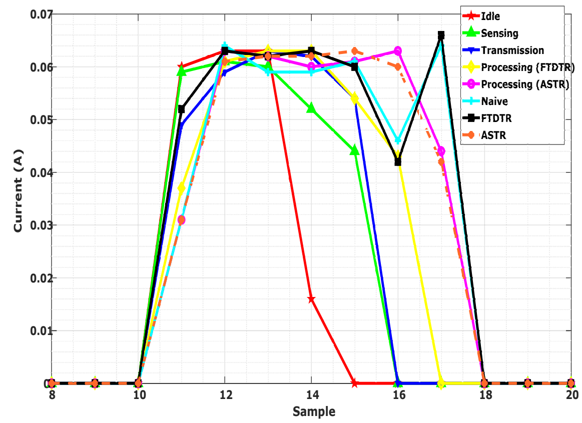

Fig. 7: Peak current consumption comparison when using LoRa.

The difference in current consumption between the Naive and FTDTR methods has been significantly reduced as shown in Figure 8, as the transmission no longer consumes as much current. Additionally, ASTR consumes more power than the Naive approach when the microcontroller exits the sleep mode.

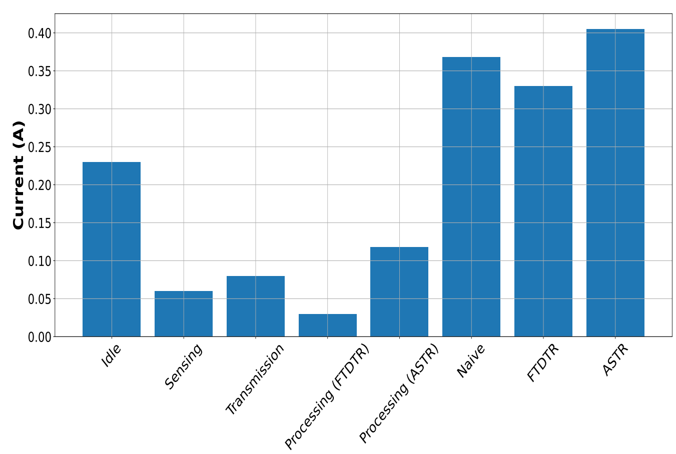

Fig. 8: Peak current consumption comparison when using LoRa.

The estimated lifetime of a microcontroller using the Naive, FTDTR, and ASTR techniques is shown in Figure 9. While Figure 8 indicates that the ASTR's peak current consumption is greater than the other peaks, the ASTR method remains the most efficient in terms of total energy efficiency due to the device remaining in sleep mode for a longer period of time than the other ways. The microcontroller can operate for 26.68 days using the Naive method. When the FTDTR method was used, the operating lifetime increased by 10.79 percent (29.56 days), compared to 357 percent when WiFi was used. When the ASTR technique was used, the operational lifetime increased by 247 percent (92.74 days), compared to 1535 percent when WiFi was used.

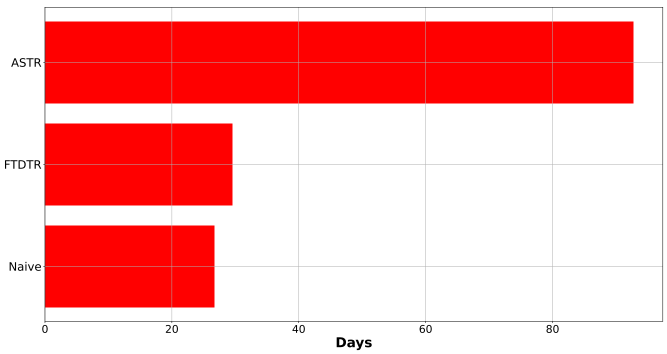

Fig. 9: Lifetime estimate in days when using LoRa.

This experiment using LoRa demonstrated that the proposed data-driven energy management algorithms are still efficient when used with LoRa-based sensors but not as efficiently as when used with WiFi-based sensors. Due to the nature of the data used in this experiment, the FTDTR strategy outperformed the Naive approach. Temperature and humidity time series are often smooth, stable, and free of large fluctuations, simplifying the prediction task. With more rapidly changing data, we may find that the Naive method outperforms the FTDTR. As a result, the data's nature is critical when working with LoRa-based devices. Another essential aspect to take away from this experiment is the crucial importance of targeting the sensing and processing modules for energy efficiency, considering that LoRa does not require as much power as WiFi does. This explains why the ASTR technique achieved the best performance by increasing the amount of time spent in deep sleep and minimizing the frequent use of the sensing module.

VI. Conclusion

The purpose of this paper is to conduct an experimental study on energy efficiency to determine whether transmission reduction is the most efficient energy efficiency strategy. Two proposed methods for data reduction have been implemented and validated using WiFi and LoRa. While the results acquired on real devices match those obtained through simulations, it was noted that when LoRa was used, the results were less optimistic. It was demonstrated that the approach based on transmission reduction did indeed slightly increase the device's lifetime when using LoRa, although only little when compared to WiFi. Additionally, it is necessary to emphasize that the data types used in this paper's experiments are easily predictable (temperature and humidity). If the data are more difficult to predict, a transmission reduction strategy may have performed worse than that presented in this paper. On the other hand, the solution that combines transmission reduction with adaptive sampling and allows the microcontroller to wake up less frequently did extend the device's lifespan, although not as efficiently as when WiFi was used. As a result, the assertion that "radio communication is most closely related to energy consumption" cannot be generalized. The approaches that have proven efficient with one communication protocol may not have the same efficiency with another. Also, when proposing a solution for high energy consumption in IoT, it is advisable to target many modules (processing, sensor, and radio).

Acknowledgment

This work is partially funded by the EIPHI Graduate School (contract "ANR-17-EURE-0002"), and the Hubert Curien CEDRE project n° 46543ZD.

References

[1] S. Balaji, K. Nathani, and R. Santhakumar, "Iot technology, applications and challenges: a contemporary survey," *Wireless personal communications*, vol. 108, no. 1, pp. 363–388, 2019.

[2] X. Yin, "Research on the key technologies of energy-efficient clustering routing algorithm for wireless sensor networks," Ph.D. dissertation, Ph. D. Dissertation. East China University of Science and Technology . . . , 2014.

[3] V. Cantoni, L. Lombardi, and P. Lombardi, "Challenges for data mining in distributed sensor networks," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 1. IEEE, 2006, pp. 1000–1007.

[4] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler, "Design of a wireless sensor network platform for detecting rare, random, and ephemeral events," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.* IEEE, 2005, pp. 497–502.

[5] M. A. Razzaque and S. Dobson, "Energy-efficient sensing in wireless sensor networks using compressed sensing," *Sensors*, vol. 14, no. 2, pp. 2822–2859, 2014.

[6] D. Lin, Q. Wang, W. Min, J. Xu, and Z. Zhang, "A survey on energy-efficient strategies in static wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 17, no. 1, pp. 1–48, 2020.

[7] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," *Computer networks*, vol. 67, pp. 104–122, 2014.

[8] C. Alippi, G. Anastasi, M. Di Francesco, and M. Roveri, "An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 2, pp. 335–344, 2009.

[9] H. Harb and A. Makhoul, "Energy-efficient sensor data collection approach for industrial process monitoring," *IEEE Trans. Industrial Informatics*, vol. 14, no. 2, pp. 661–672, 2018.

[10] A. Makhoul, H. Harb, and D. Laiymani, "Residual energy-based adaptive data collection approach for periodic sensor networks," *Ad Hoc Networks*, vol. 35, pp. 149–160, 2015.

[11] J. Azar, C. Habib, R. Darazi, A. Makhoul, and J. Demerjian, "Using adaptive sampling and dwt lifting scheme for efficient data reduction in wireless body sensor networks," in *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2018, pp. 1–8.

[12] G. B. Tayeh, A. Makhoul, J. Demerjian, C. Guyeux, and J. Bahi, "Fault tolerant data transmission reduction method for wireless sensor networks," *World Wide Web*, vol. 23, no. 2, pp. 1197–1216, 2020.

[13] G. B. Tayeh, A. Makhoul, D. Laiymani, and J. Demerjian, "A distributed real-time data prediction and adaptive sensing approach for wireless sensor networks," *Pervasive and Mobile Computing*, vol. 49, pp. 62–75, 2018.