



## Article

# Energy Consumption Reduction in Wireless Sensor Network-Based Water Pipeline Monitoring Systems via Energy Conservation Techniques

Valery Nkemeni <sup>1,\*</sup> , Fabien Mieyeville <sup>2</sup> and Pierre Tsafack <sup>1</sup><sup>1</sup> Laboratory of Electrical Engineering and Computing, University of Buea, Buea P.O. Box 63, Cameroon<sup>2</sup> University de Lyon, Université Claude Bernard Lyon 1, Ecole Centrale de Lyon, INSA Lyon, CNRS, Ampère, F-69621 Villeurbanne, France; fabien.mieyeville@univ-lyon1.fr

\* Correspondence: nkemeni.valery@ubuea.cm

**Abstract:** In wireless sensor network-based water pipeline monitoring (WWPM) systems, a vital requirement emerges: the achievement of low energy consumption. This primary goal arises from the fundamental necessity to ensure the sustained operability of sensor nodes over extended durations, all without the need for frequent battery replacement. Given that sensor nodes in such applications are typically battery-powered and often physically inaccessible, maximizing energy efficiency by minimizing unnecessary energy consumption is of vital importance. This paper presents an experimental study that investigates the impact of a hybrid technique, incorporating distributed computing, hierarchical sensing, and duty cycling, on the energy consumption of a sensor node in prolonging the lifespan of a WWPM system. A custom sensor node is designed using the ESP32 MCU and nRF24L01+ transceiver. Hierarchical sensing is implemented through the use of LSM9DS1 and ADXL344 accelerometers, distributed computing through the implementation of a distributed Kalman filter, and duty cycling through the implementation of interrupt-enabled sleep/wakeup functionality. The experimental results reveal that combining distributed computing, hierarchical sensing and duty cycling reduces energy consumption by a factor of eight compared to the lone implementation of distributed computing.



**Citation:** Nkemeni, V.; Mieyeville, F.; Tsafack, P. Energy Consumption Reduction in Wireless Sensor Network-Based Water Pipeline Monitoring Systems via Energy Conservation Techniques. *Future Internet* **2023**, *15*, 402. <https://doi.org/10.3390/fi15120402>

Academic Editor: Giovanni Pau

Received: 15 November 2023

Revised: 6 December 2023

Accepted: 9 December 2023

Published: 14 December 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** wireless sensor network-based water pipeline monitoring; green wireless sensor networks; energy conservation techniques; distributed computing; hierarchical sensing; duty cycling

## 1. Introduction

### 1.1. Green Wireless Sensor Networks

A Wireless Sensor Network (WSN) consists of several embedded nodes with sensing, processing, and wireless communications capabilities, distributed over an area of interest to monitor physical or environmental conditions [1]. The sensor nodes in WSNs serve as the backbone of the Internet of Things (IoT), where they act as data collection modules (perception layer) of IoT applications. They are the low-end devices of IoT [2]. Being spatially distributed systems, WSNs exploit wireless communication as the means of communication between nodes. This makes them effective for a myriad of applications such as geographical monitoring, habitat monitoring, transportation, military systems, business processes, structural health monitoring, microclimate research, medical care and many others [3]. Kandris et al. [4], in an up-to-date survey on the applications of WSN, classified the applications of WSN into six main categories (military, environmental, health, flora and fauna, industrial, and urban) based on the nature of their use. One of the urban applications of WSN is its use in monitoring water distribution networks, resulting in systems referred to as wireless sensor network-based water pipeline monitoring (WWPM) systems [5]. WWPM systems are greatly needed, especially in developing countries where the amount of non-

revenue water (NRW), mostly caused by leakages on the Water Distribution Network (WDN), sometimes exceed more than 70% of the total NRW [6,7].

In large-scale WSN applications exemplified by WWPM systems, a vital requirement prominently emerges: the attainment of low energy consumption. This requirement arises from the fundamental necessity to ensure the sustained operability of sensor nodes over extended durations, all without the requirement for frequent battery replacement. It is worth noting that the sensor nodes employed in such applications typically rely on battery power and are frequently situated in locations that are physically challenging to access [8]. This gives rise to the exigency for what is commonly referred to as “Green Wireless Sensor Networks” (GWSNs) [9], where the major concern is to maximize energy efficiency by reducing unnecessary energy consumption and minimizing environmental impact. A green WSN, by definition, is one that adeptly reduces its aggregate energy utilization, thereby prolonging the operational lifespan and enhancing the performance of the network, all while concomitantly mitigating its carbon footprint.

### *1.2. Energy Management Techniques for Extending WSN Lifetime*

As earlier stated, energy efficiency is a critical aspect in most WSN applications because of the energy constrain of sensor nodes and the need to prolong WSN lifetime. Yetgin et al. [10] provide several different definitions of network lifetime depending on the specific application, the objective function and the network topology considered. However, we will stick to the definition of network lifetime as being the total amount of time during which the WSN is capable of effectively performing its functions and meeting up with the application’s requirements. In that light, a WSN composed of battery-powered sensor nodes without energy harvesting capabilities is considered dead when it is unable to forward any data to the base station [11]. The lifetime for such WSNs is constrained by the battery of the individual sensor nodes in the WSN [12]. Thus, the critical issues to be considered when maximizing WSN lifetime are how to reduce the energy consumption of the nodes or how to replenish their energy sources in an efficient and realistic way [12,13]. The quest to prolong WSN lifetime has in recent years led to a plethora of energy management techniques in the literature. The techniques can be broadly classified as energy conservation techniques (data-driven and duty cycling) [9], energy balancing techniques (energy-efficient routing protocols and mobile sink) [14], energy harvesting techniques (solar, vibration, RF, wind and thermal) [15] and associated predictive energy management approaches [16]. In this paper, we focus on energy conservation techniques.

Conserving the energy of sensor nodes requires a compromise between various activities at both the node and network levels [11,17–19]. This can be accomplished by implementing energy-efficient protocols that are aimed at minimizing the energy consumption during network activities and/or implementing power management schemes that involve switching off node components that are not temporarily needed. The reason is that a large amount of energy is consumed by node components (CPU, radio, sensor, etc.) even in the idle mode [12]. The techniques for minimizing energy consumption at the node level include duty cycling techniques (radio optimization, sleep/wake-up schemes, dynamic voltage frequency scaling), and data-driven approaches (data compression, data aggregation, data prediction, hierarchical sensing, adaptive sampling, and model-based active sensing) [12,13,20]. A detailed review and a commonly adopted taxonomy of energy conservation strategies used for preserving WSN lifetime is presented in [12,13,17–20].

Duty cycling techniques reduce the sensor node’s energy consumption by turning off the sensor node’s hardware components when they are not needed and waking them up whenever necessary [21]. This establishes a small duty cycle for the nodes based on events occurring in the monitored environment [22]. Thus, techniques based on duty cycling rely on the fact that active nodes do not need to keep their radios, processor, and sensing devices continuously on. According to the survey by Anastasi et al. [12], duty cycling is achieved by two complementary approaches, with one approach taking advantage of the redundancy in WSNs by adaptively selecting only a minimum subset of nodes to remain

active for maintaining connectivity, while the other approach ensures that active nodes do not maintain their radio and sensors continuously on by constantly switching them off (i.e., put it in the low-power sleep mode) when there is no network activity. The authors termed the former topology control and the latter power management.

Data-driven techniques can be classified into either data reduction or energy-efficient data acquisition schemes depending the problem they address [12,23,24]. The data reduction schemes on the one hand are primarily involved in reducing the number of data transmissions and the amount of data transmitted, as data moves from the sensor nodes to the base station. Energy-efficient data acquisition schemes, on the other hand, are involved in reducing the energy spent by the sensing subsystem by typically reducing the number of samples generated by the sensors.

The data reduction data-driven schemes are performed via in-network processing within the WSN. In-network processing involves distributed computing, requiring the processing of data as it travels via the WSN to the sink. It involves actions such as fusion and aggregation on the data as it moves within the WSN from one sensor node to another. This reduces the number of redundant data that needs to be transmitted. Popular in-network processing techniques in the literature include techniques such as data aggregation, data compression, and data prediction [12,13,24,25]. Data aggregation techniques increase the network lifetime by merging data in an efficient manner as it traverses the network from one node to the another until it gets to the sink [26,27]. Likewise, data compression involves encoding information at the sensor nodes and decoding it at the sink, and it can be applied to reduce the amount of information sent by source nodes [12]. This reduction in the amount of data transmitted and received also reduces the radio module's active time, which also decreases the sensor node's energy consumption. Similarly, data prediction techniques aim at reducing the node's energy consumption by minimizing the communication cost. They achieve this by building an abstraction of a sensed phenomenon, i.e., a model describing data evolution. The model can predict the values sensed by sensor nodes within certain error bounds, and resides both at the sensor nodes and at the sink [12]. Transmissions between the nodes and the sink occur only when sensor nodes measure values outside the threshold of the prediction models [13]. This reduces the frequency of transmission and the energy needed for communication as well.

The energy-efficient data acquisition data-driven schemes include adaptive sensing techniques such as hierarchical sensing, adaptive sampling, and model-based active sensing [22]. By reducing the number of samples generated by the sensors, an efficient sensing strategy also reduces the amount of data to be processed and possibly transmitted by sensor nodes, and thus generate further energy savings [22]. In hierarchical sensing, a sensor node has multiple sensing devices monitoring the same physical parameter, but with each having a different sensing accuracy and power consumption. Accuracy can be traded off for energy efficiency by using the low-power sensors to get a rough estimate of the monitored parameter [12]. Once an event has been detected, the accurate power-hungry sensors can be activated to give more accurate readings of the physical property at the cost of greater energy consumption [22].

### *1.3. Review of Energy Management Techniques for Extending the Lifespan of WWPM Systems*

In the literature of WWPM, very few studies have tackled the issue of power consumption and sought ways to reduce the energy consumption so as to prolong the monitoring lifetime. However, the issue of energy consumption in WWPM systems is of vital importance since a WWPM system is required to go for long periods of time unattended as a result of the fact the nodes may be difficult to access, especially in the case of buried pipelines. In such cases, the pipelines are required to be monitored throughout their life span, which can extend into years. In this sub-section, we survey some energy conservation management techniques that were discussed in Section 1.2 and how they were applied in previous WWPM studies.

In [28], the authors used duty cycling with scheduled wake-up to reduce the power consumption and thus prolong the WWPM lifetime. The power consumption of the sensor nodes was minimized to 2.2  $\mu$ W based on taking one measurement every six hours, thus given the sensor node a theoretical lifespan of 100 years when powered with two AA batteries. Although the sleep/wake-up method used in this study achieves great energy reduction, the WWPM solution does not provide real-time monitoring. In a more recent study, Liu et al. [29] proposed a leakage-triggered networking method (radio-controlled wake-up) to reduce the wireless sensor network's energy consumption and prolong the lifetime of their proposed WWPM system. The authors proposed the use of three types of control frames (i.e., join frame, active frame, and wave frame) to trigger the network according to the leakage detection results. The coordinator and routing nodes were consistently in the working state, and the terminal nodes within the routing node network were sequentially working and sleeping. The terminal nodes only wake up from sleep when they receive an active frame from the routing nodes. The power consumption of the sensor node was 30 mW, and the proposed energy reduction technique increased the WSN lifetime by a factor of 2. In [30], the authors fused duty cycling and data driven-based schemes for maximizing the information gain about the leak, as well as minimizing the power consumption. In their proposed solution, the duty cycling part provides the sleep/wake-up schedule for the nodes to minimize the sensing, communication and processing-related energies, while the data-driven part implements adaptive sampling where nodes closer to the leak location operate at a higher sampling rate, whereas nodes farther away from the leak operate at a lower sampling rate. The power consumption of the sensor node when there was no leak was 2.5 mW, and they achieved an energy reduction that extended the WSN lifetime by a factor of 2.6. The same author in another study [31] combined duty cycling and data-driven approaches like hierarchical sensing and compression. The scheme relies on implementing hierarchical sensing by using vibration sensors of different sensitivities to detect vibrations due to a leak [32], and on exploiting duty-cycling and wavelet-based signal compression in order to reduce sensing, computation and communication energies. The authors failed to mention the energy consumption of the sensor node and did not provide an overall quantitative measure of the energy reduction of their proposed hybrid technique. Both [30,31] employed centralized data processing to detect and localize leaks, and the results were validated only via simulations.

Rashid et al. [33] used clustering and in-network processing to reduce the energy consumed in the network. The authors used wavelength transform and moving average filter to implement in-network processing within the WSN. By integrating the signal processing algorithm in the sensor nodes for distributed event detection and by performing aggregation on data within the cluster, the energy consumed in the network is far less than when all readings are sent to the base station in a centralized network. The authors failed to mention the energy consumption of the sensor nodes, but the results reported an approximately 50% energy reduction when they compared their proposed distributed approach with the centralized approach. References [34,35] also implemented clustering whereby sensor nodes acquire vibration data from the pipe and transmit to a closely located cluster head node that performs local processing and finally transmits to a centralized base station that performs the leak detection decision. In another study, Kartakis et al. [8] presented an end-to-end water leak localization system, which exploits compression as the data reduction approach for conserving energy consumption. The proposed system combined a lightweight edge anomaly detection algorithm based on Kalman filtering on compression rate stream. The results revealed that the proposed solution reduced the communication by 99% compared to the traditional periodic communication, thus enabling the use of battery-powered sensor nodes for an extended period. The limitation of this work is that the authors failed to mention the energy consumption of the sensor node and to what extent their proposed technique increases the WSN lifetime. Lastly, Nkemeni et al. [36] proposed a distributed computing solution for detecting leaks and reducing energy consumption in WWPM systems by implementing a distributed Kalman

filter (DKF). The distributed computing technique performed leak detection locally on the sensor nodes and transmitted leak decisions to the base station only when a leak was detected on the pipeline. A 10-node linear WSN was simulated, where scenarios for both the proposed distributed approach and the centralized approach were tested. The reported results demonstrate that the fully distributed solution enhances both leak detection accuracy and reduces power consumption compared to the centralized approach.

#### 1.4. Objective

The issue of energy conservation is of vital importance in WWPM. However, most WWPM studies focus on leak localization while neglecting the energy consumption of their proposed WWPM solutions. As such, there are very few studies in the literature that concern themselves with the evaluation of the energy consumption of their proposed WWPM solution and means on how to conserve energy in order to prolong the lifespan of the WWPM system. Additionally, most of the studies that have treated energy consumption with their proposed WWPM solutions have been based on simulations, and very few studies go to the extent of performing physical experiments on a laboratory testbed. Thus, to the best of our knowledge, no existing study has experimentally evaluated the combined effected duty cycling, hierarchical sensing, and distributed computing on the energy consumption and lifespan of WWPM systems. In this paper, the aim is to experimentally investigate the impact of a hybrid technique, incorporating distributed computing, hierarchical sensing, and duty cycling on the energy consumption of a sensor node in prolonging the lifespan of a WWPM system. From the existing literature, it is evident that a limited number of studies are experimental, and only a scant few employ an approach that integrates both simulation and physical deployment. The novelty of this study is two folds. Firstly, it evaluates the energy consumption of three variants of Distributed Kalman Filter (DKF) applied in the context of a distributed computing solution for WWPM via simulations and laboratory experiments. Secondly, it implements a hybrid energy reduction technique involving duty cycling, hierarchical sensing, and distributed computing and then compares the energy reduction achieved from this hybrid technique with the lone implementation of distributed computing. From results obtained from experiments, it shows that this hybrid technique achieves an energy reduction that is 8 times lower compared to the lone implementation of distributed computing.

## 2. Materials and Methods

We adopted a method that involves simulations and laboratory experiments. The physical experiments were carried out on a WDN laboratory testbed, and the simulations were performed on the CupCarbon simulator (version 5.2). CupCarbon is a Smart City and Internet of Things Wireless Sensor Network (SCI-WSN) simulator that is used to design, visualize, debug and validate distributed algorithms [37,38]. CupCarbon is based on a multi-agent system approach and was then used because of its inherent property that permit the simulation of distributed algorithms, monitoring of the energy profiles of sensor nodes during simulations [39], easy to use GUI, and native support of the ESP32 microcontroller and communication protocols such as IEEE 802.15.4, ZigBee, WiFi, and Lora [40]. It supports recent application layer protocols in IoT such as MQTT making it one of the popular simulators used in IoT research [41]. Lastly, it supports programming in python, which makes it a suitable choice for our future works requiring machine learning.

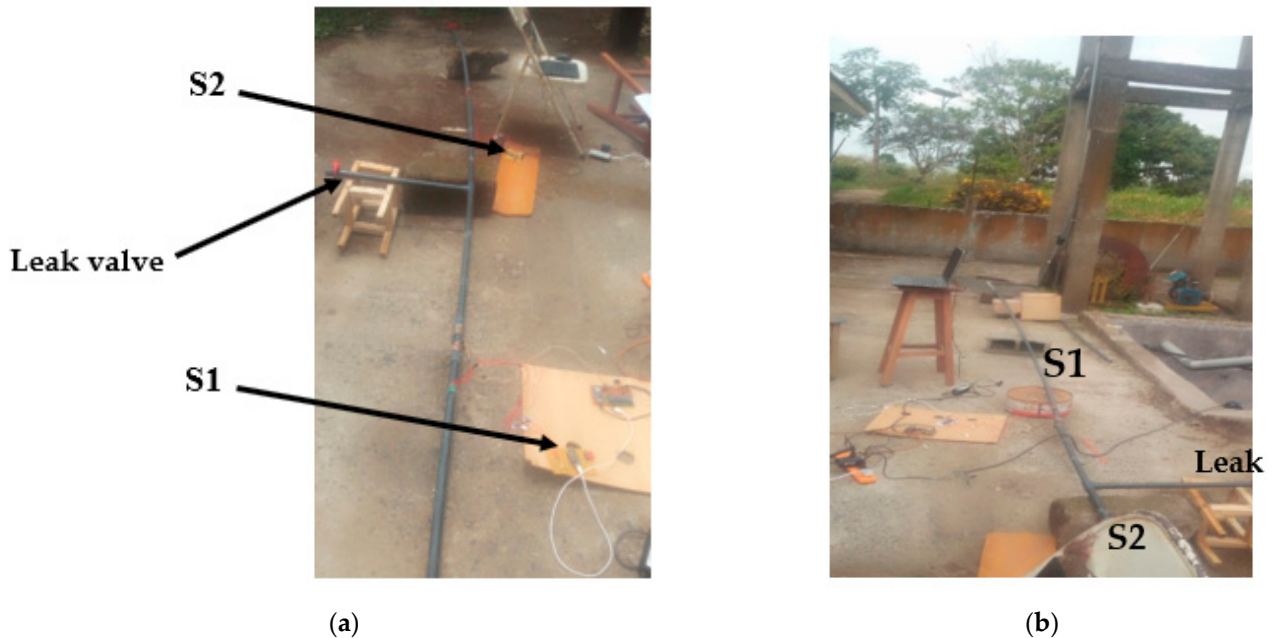
The following subsections describe more precisely the experimental setup and the energy characterization tools.

#### 2.1. Experimental Setup

The laboratory testbed used for physical experiments consisted of two high-pressure pipes, each with a diameter of 25 mm and a length of 6 m. These pipes are interconnected to form an L-shaped structure. Two valves are installed on the pipeline: one at the end, designed as the service valve to simulate water consumption at the client's premise, and



another positioned 8 m from the water inlet into the distribution pipe, serving as the leak valve to replicate leaks in the Water Distribution Network (WDN). Figure 1 below illustrates the setup of the laboratory testbed.



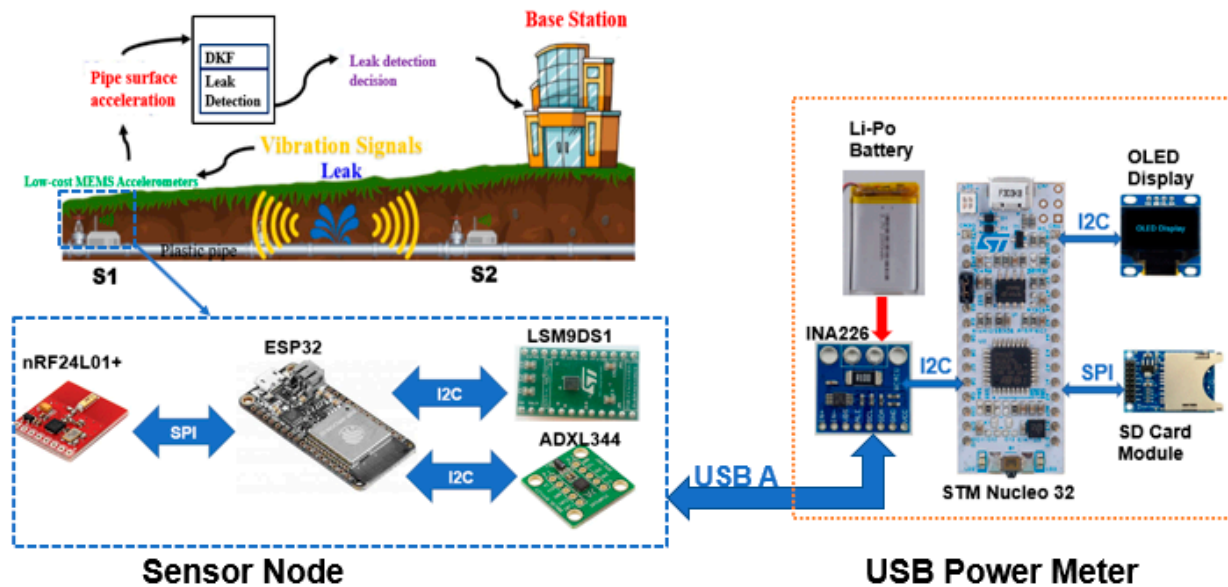
**Figure 1.** Laboratory testbed setup. (a) Position of sensor node's S1 and S2 (b) water distribution pipeline.

## 2.2. Sensor Node Components

The WWPM system used for monitoring the laboratory WDN was composed of two sensor nodes S1 and S2, which were placed before and after the leak valve, respectively, in the direction of water flow (left to right) as shown in Figure 1. The main components of the sensor nodes comprised an ESP32 as the processing unit, a nRF24L01+ transceiver as the communication unit, a LSM9DS1 as the sensing unit, and a 3.7 V 2000 mAh Li-Po rechargeable battery as the power source, as depicted in Figure 2. Depending on the specific scenario under investigation, as discussed in Section 3, the connection or disconnection of an ADXL344 accelerometer may be optional, contingent upon the implementation of hierarchical sensing. For the processing unit, the ESP32 was used because it is a powerful, low-cost, and low-power MCU. It incorporates a double-core 32-bit Xtensa LX6 microprocessor and an ultra-low-power coprocessor (ULP).

The architecture of the ESP32 frequently renders it a popular choice for deployment in intermediate-level devices within the context of IoT/WSN applications [2]. The ULP coprocessor consumes minimal current (between 10  $\mu$ A and 150  $\mu$ A) when the core is sleeping and can be used for simple control. This ULP feature makes the ESP32 a suitable processing unit for a sensor node that will be battery-powered. Thus, the ESP32 can be used to implement duty cycling via an interrupt-driven sleep/wakeup, where the low-power ULP coprocessor is needed for control during deep sleep mode when no leak event is detected while the high-power dual-core processor is needed in the active mode to perform powerful computations when a leak event is detected. Moreover, the ESP32 has a feature that permits the remaining charge of its battery supply to be estimated by reading the battery voltage from one of its analogue pins. This feature makes the ESP32 suitable for energy-aware applications as it can compute its energy consumption online and thus regulate its operations. Another advantage of the ESP32 is its lower cost, which is an attractive feature for sensor nodes that are to be deployed in a developing country's WWPM system. In addition, the ESP32 also incorporates Wi-Fi and Bluetooth modules, which makes it IoT-compatible. In our experiments, the core of the ESP32 MCU on each

sensor node was configured to operate at a speed of 80 MHz and in the modem sleep mode (ESP32's Wi-Fi transceiver turned off).



**Figure 2.** Setup model depicting hardware interfacing of the sensor node's components and power measurement device.

The nRF24L01+ was used as the communication unit and not the ESP32's Wi-Fi or Bluetooth transceiver because it exhibits current peaks in RX/TX modes lower than 14 mA (one of the lowest consumptions on the market), a sub- $\mu$ A power-down mode, and advanced power management. The nRF24L01+ is hence a true ultra-low power solution enabling months to years of battery life from coin cell or AA/AAA batteries. In addition, compared to other IEEE 802.15.4-based radio chips, the nRF24L01+ transceiver is low-cost. Finally, the enhanced ShockBurst feature of the nRF24L01+ enables it to achieve high data rates (up to 2 Mbps) at lower power consumption. The burst mode is particularly interesting for distributed computing, which involves only short-distance communications between neighbouring nodes. For the sensing unit, we selected the LSM9DS1 because of its low cost, high sensitivity, and availability. To reduce the power consumption of the sensor node, we also selected a low-power accelerometer to compensate for the high-power consumption of the more sensitive LSM9DS1 accelerometer by implementing hierarchical sensing. The low-power accelerometer is less sensitive and is continuously used to monitor the pipeline. Once it detects an acceleration larger than a defined threshold, it triggers an interrupt to wake up the more sensitive accelerometer from sleep to collect more accurate measurements that will be used for leak detection. The ADXL344 was selected as the low-power accelerometer because of its low-cost, low-power consumption, and wide bandwidth. The accelerometers of both sensor nodes were glued on the pipe surface using hot glue, and wires were used to connect them to the ESP32 MCU. Table 1 presents a comparison between the ADXL344 and LSM9DS1 accelerometers.

**Table 1.** Comparison of ADXL344 and LSM9DS1 accelerometers.

Accelerometer	Resolution	Bandwidth (Hz)	Sensitivity (LSB/g)	Sensing Range	Noise Floor Level ( $\mu\text{g}/\sqrt{\text{Hz}}$ )	Current Consumption ( $\mu\text{A}$ )
ADXL344	13	0–1600	4096	$\pm 2\text{ g}, \pm 4\text{ g}, \pm 8\text{ g}, \pm 16\text{ g}$	530	23
LSM9DS1	16	0–400	32,768	$\pm 2\text{ g}, \pm 4\text{ g}, \pm 8\text{ g}, \pm 16\text{ g}$	N.A.	600

g is the gravitational constant, where  $1\text{ g} = 9.8\text{ ms}^{-2}$ .

### 2.3. Energy Measurement

We used a custom USB power meter developed in [36] to measure and record the energy consumption of the sensor nodes. The custom power measurement device was based on the intrusive direct measurement method and was composed of an INA226 module with a 100 mΩ shunt resistor, STM nucleo-32 F303k8 microcontroller, 128 × 64 OLED display, SD card and USB port. The block diagram of the USB power meter is presented in Figure 2. We developed this device for two main reasons: (1) to enable us to measure very low currents in the μA range, particularly the current consumption of the node when the ESP32 is operating in deep sleep mode, and (2) to be able to monitor and store the power consumption of the nodes without being physically present (i.e., recording of power measurements collected periodically over a long period of time).

The energy consumption metric represents the total amount of energy spent by the sensor node while performing sensing, processing and communication operations. It is expressed as:

$$E = \sum_k (E_{MCU,k} + E_{TRX,k} + E_{SEN,k}) \quad (1)$$

where  $E_{(MCU,k)}$ ,  $E_{(TRX,k)}$ , and  $E_{(SEN,k)}$  are the energy consumed by the sensor node's processing unit, communication unit, and sensing unit, respectively, at time step  $k$ .

The energy consumed by the constituent parts of the sensor node are calculated from the integration of the product of current and voltage over the duration of operation. Thus, the expressions of the energy consumption of the sensor node's constituent parts are given as follows.

$$E_{MCU,k} = I_{MCU} \times V_{MCU} \times t_{proc} \quad (2)$$

where  $I_{MCU}$  is the current consumption of the MCU,  $V_{MCU}$  is the operation voltage of the MCU, and  $t_{proc}$  is the time used by the MCU for processing.

$$E_{SEN,k} = I_{SEN} \times V_{SEN} \times t_{sen} \quad (3)$$

where  $I_{SEN}$  is the current consumption of the sensor,  $V_{SEN}$  is the operation voltage of the sensor, and  $t_{sen}$  is the amount of time the sensor is active during the sensing process.

$$E_{TRX,k} = \frac{L}{R} V_{TRX} (I_{TX} + I_{RX}) \quad (4)$$

where  $L$  is the length of the packet transmitted/received,  $R$  is the data rate, and  $V_{TRX}$  is the operational voltage of the transceiver, while  $I_{TX}$  and  $I_{RX}$  are the current consumption of the transceiver when operating in the transmit and receive modes, respectively.

### 2.4. Battery Lifetime Estimation

When the node is powered by a lithium-polymer (Li-Po) battery, the state of charge (SOC) of the battery is estimated and used as the energy consumption metric. There are several techniques for estimating the SOC of a battery. They include the coulomb counting method, voltage method, Kalman filter method, impedance spectroscopy, etc. [42]. However, we used the voltage method because the battery voltage can be read directly from one of the ESP32 pins without needing extra circuitry. From the SOC-versus-voltage-discharge curve obtained from the 3.7 V 2000 mAh Li-Po rechargeable battery's datasheet [43,44], we used polynomial interpolation to derive the analytic relationship between SOC and the voltage of the battery given by Equation (5).

$$SOC = pV^4 - qV^3 + rV^2 - sV + t \quad (5)$$

where  $V$  is the battery's voltage, with  $p = 2808.3808$ ,  $q = 43,560.9157$ ,  $r = 252,848.5888$ ,  $s = 650,767.4615$ , and  $t = 626,532.5703$



Now that the experimental testbed and power characterization tools have been introduced, the next section will discuss the experimental scenarios that were investigated in this study.

### 3. Energy Consumption of Distributed Solutions under Different Scenarios

#### 3.1. Distributed Computing on WWPM Systems

In our previous study [36], we compared the lifetime of a WWPM system for both a centralized solution and a fully distributed solution. In this study, we focus on the fully distributed solution, which implements distributed computing, and investigate further ways to improve the WSN lifetime by reducing the energy consumption at the node level. We first evaluate the energy performance of the distributed solution by implementing three variants of Distributed Kalman Filter (DKF) and then proceed further to improve the node's lifetime by using duty cycling and hierarchical sensing techniques. Thus, we study the energy consumption of the fully distributed WWPM solution under two main scenarios. In scenario 1, we assess the energy consumption of three DKFs implemented in the context of WWPM systems. In this scenario, we implement distributed computing by implementing a DKF on the sensor nodes as the sole energy reduction technique. In scenario 2, we employ a hybrid energy reduction technique that combines distributed computing, hierarchical sensing, and duty cycling. Distributed computing is achieved by implementing DKFs on the sensor nodes, hierarchical sensing is achieved by using both a high-power and high-accuracy accelerometer (LSM9DS1) and a low-power and low-accuracy accelerometer (ADXL344), and duty cycling is implemented via an interrupt-driven sleep/wakeup mechanism. We then analyze the contribution of each energy conservation technique on the energy reduction achieved by the hybrid technique. Table 2 presents the main rationale behind each energy conservation technique and the components of the sensor node that it affects to reduce energy consumption.

**Table 2.** Rationale behind each energy conservation technique and the sensor node components affected.

Technique	Rationale and Components Affected
Distributed computing	Reduces energy consumption by reducing the number of transmissions to the base station
Duty cycling	Reduces energy consumption by switching sensor node components (MCU, transceiver, and sensors) to low power modes.
Hierarchical Sensing	Reduces energy consumption by performing a tradeoff (compromise) between accuracy and energy consumption. This involves switching between high accuracy high power sensor and low accuracy low power sensor

Detailed explanations of scenarios 1 and 2 are provided in the following Sections 3.2 and 3.3, respectively.

#### 3.2. Scenario 1

The performance of different classes of DKFs used in a WSN have been evaluated and compared in [45]. From [45], we selected three DKFs in which we evaluated their leak detection performance in the context of WWPM systems in [46]. The three DKFs were selected from data fusion strategies—specifically, diffusion, gossip, and consensus. The consensus-based DKF selected was the Information-Weighted Consensus Filter (ICF) proposed by Kamal et al. [47], the gossip-based DKF selected was the Sample Greedy Gossip Information-Weighted Consensus Filter (SGG-ICF) proposed by Shin et al. [48], and the selected diffusion-based DKF was the Event-triggered Diffusion-based Kalman Filter (EDKF) proposed by Battistelli et al. [49]. In [46], we presented the algorithms, operation and detailed explanations for the choice of the selected DKFs. Table 3 provides a summary of the values of the parameters assigned to the various DKF algorithms. However, no work has been done to evaluate the energy consumption of DKFs used in a WSN and more specifically in the context of WWPM. If we have previously demonstrated that distributed

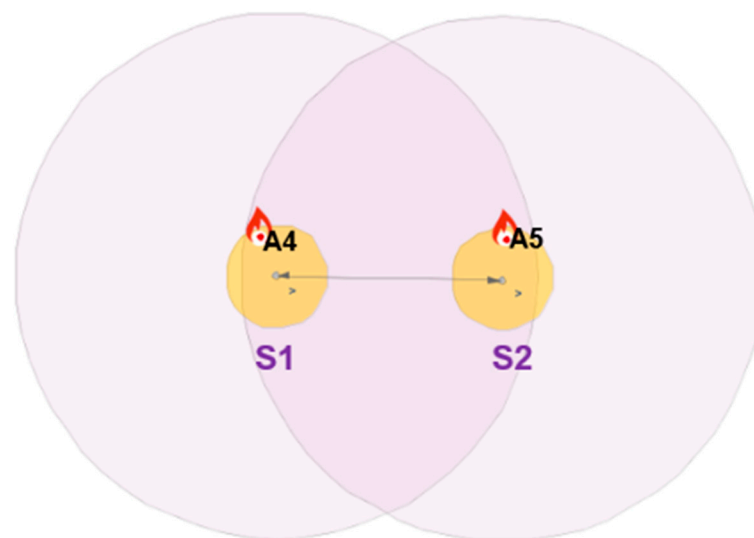
computing is feasible on low-end IoT devices and yields similar or superior results compared to centralized computing, the energy aspects in those prior works primarily focused on establishing feasibility and quantifying energy consumption orders. However, in this article, our objective is to conduct a precise evaluation of the IoT device energy consumption in a distributed computing solution. We also investigate hierarchical sensing as a means to prolong the system's lifespan without compromising the leak detection reliability and real-time monitoring. For this reason, in scenario 1, we are interested in assessing the energy consumption of the three selected DKFs whose leak detection performances were evaluated in [46].

**Table 3.** Values assigned to DKF parameters.

Parameter	Value	Concerned Algorithms
State transition matrix (A)	1	All
Measurement matrix (H)	1	All
Process noise covariance (Q)	0.001	All
Measurement noise covariance (R)	0.0081	All
Network size (N)	2	ICF and SGG-ICF
Number of consensus or gossip iterations (L)	5	ICF and SGG-ICF
Consensus speed factor ( $\epsilon$ )	0.65	ICF
Sensor activation probability (p)	0.5	SGG-ICF
Information transmission rate ( $\alpha$ , $\beta$ , and $\delta$ )	0.001, 40, 40, respectively	EDKF

In scenario 1, we started with simulations on CupCarbon before proceeding with physical experiments on the laboratory testbed. The simulation platform was fed with real data from the laboratory testbed. The simulation and experimental setups are the same as those described in [46], which was focused on evaluating the leak detection performance of three DKF variants. However, in this study, we focus on the detailed energy evaluation of these algorithms, the impact of using hierarchical sensing and duty cycle on energy consumption, and battery management.

The simulation setup is shown in Figure 3 and is composed of two sensor nodes (S1 and S2) and natural event generators (A4 and A5). The natural event generators emulated the physical accelerometer sensors and were loaded with acceleration data collected from the laboratory testbed. The sensor nodes (S1 and S2) were loaded with scripts that implement the selected DKF algorithms during the simulations.



**Figure 3.** Simulation setup in CupCarbon.

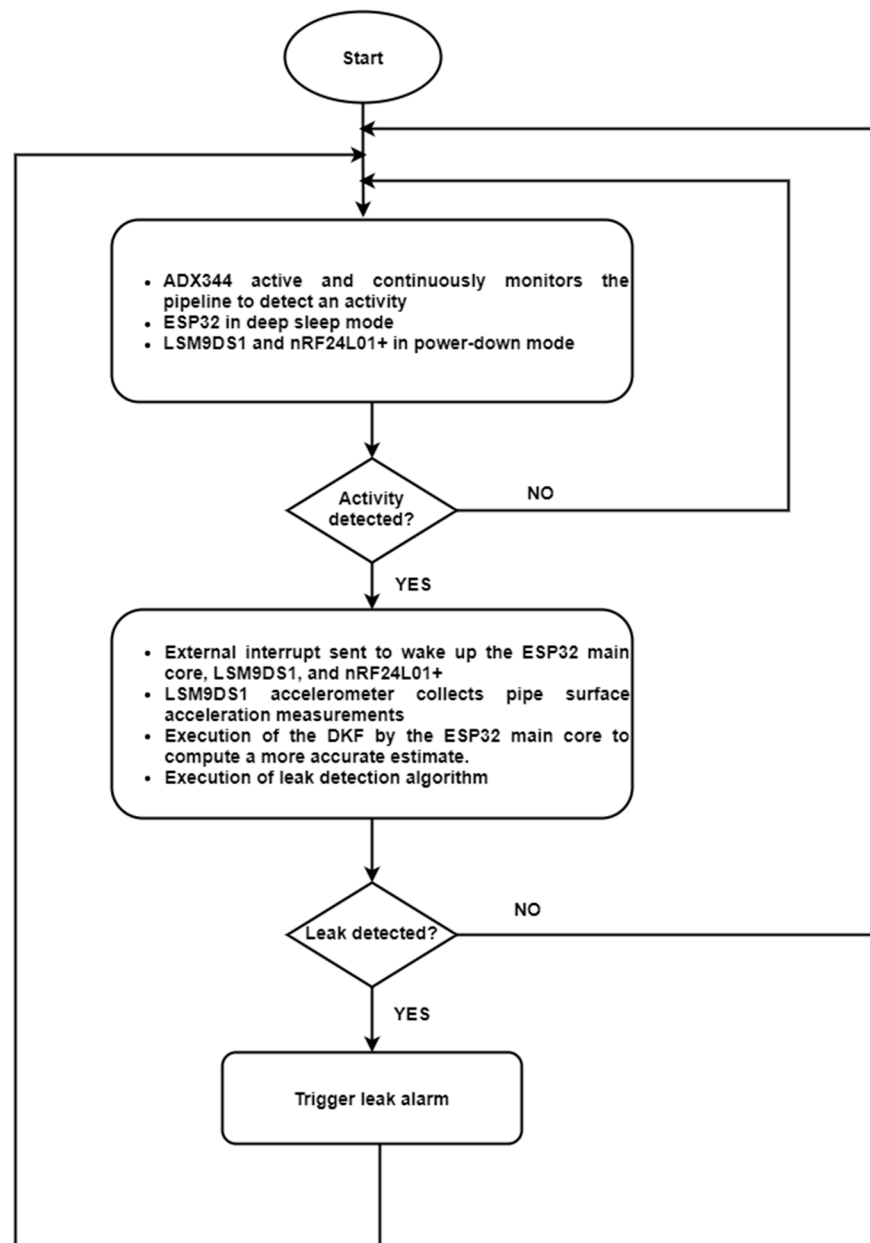
We conducted several simulations on the setup depicted in Figure 3 to assess the power consumption of each of the selected DKF algorithm. The simulations are conducted on a two-node WSN since a WWPM solution is inherently a linear WSN. In this configuration, the nodes at the beginning and end of the chain each have only one directly connected neighbour, while all intermediate nodes have two directly connected neighbours each. As illustrated in [36], the use of a two-node setup is sufficient for modelling a linear WSN solution. We ran the simulations for a period of 86,400 s (1 day) to monitor the power consumption of the nodes when executing each of the selected DKF algorithm. The simulations permitted us to determine the energy consumed by the nodes and to estimate the size of the battery that will have sufficient energy to power the sensor nodes for a complete day of experimentation. We later conducted physical experiments on the laboratory testbed to measure the power consumption of each of the selected DKF algorithm. We used the USB power meter to monitor the node's power profile for each of the DKF algorithm for a period of 1 h. We later supplied each node with a 3.7 V, 2000 mAh Li-Po battery and monitored the battery's voltage for a period of 1 day. The measured voltage was then used to estimate the state of the battery discharge using equation 5. The results and discussion of the power consumption of each of the DKF algorithms derived from simulations and physical experiments are found in Section 4.1.

### 3.3. Scenario 2

At the end of scenario 1, the DKF with the the highest power consumption from both simulations and experiments is selected. The selected DKF is then implemented on the sensor nodes alongside hierarchical sensing and duty cycling, and the energy consumption is monitored. The worst case was selected in order to evaluate the extent to which the hybrid technique will affect the energy consumption of the sensor nodes. The objective was to develop a WWPM system that can achieve both low-power consumption and real-time monitoring. Figure 4 depicts the flow of operations in our proposed fully distributed, real-time and low-power leak detection solution for WWPM using low-cost MEMS accelerometers that implements the hybrid technique.

The operation of the sensor nodes implementing this hybrid technique is described as follows:

- On starting, the main core of the ESP32 is in deep sleep mode and the nRF24L01+ and LSM9DS1 are in the power down mode, while the ESP32 ULP and ADXL344 are in the active mode.
- The ADXL344 continuously monitors the pipeline to detect an activity. An activity (a leak event) is detected once the measured acceleration is above the predefined threshold value (1.01 g) that is stored in the activity register of the ADXL344 accelerometer. Once an activity is detected, an external interrupt is sent to wake up the other components of the sensor node.
- When no activity is detected by the ADXL344 accelerometer, the ESP32 stays in the deep sleep mode with the ULP coprocessor active, while the nRF24L01+ transceiver and the LSM9DS1 accelerometer both remain in the power-down mode.
- Once an activity is detected, the ADXL344 triggers a wake-up interrupt to the ESP32, LSM9DS1, and nRF24L01+. The LSM9DS1 wakes up and collects more accurate measurements. The measurements are then processed by the ESP32 main core by running the DKF algorithm. The nRF24L01+ is used to communicate the local estimates of the sensor node to its direct neighbours to achieve distributed data fusion.
- Once the fusion of local estimates from neighbouring nodes has been performed, the computed estimate is then compared with the baseline value for leak detection. If the final estimate exceeds the baseline value by some threshold value, then a leak alarm is triggered and the node goes back to sleep. Otherwise, no leak alarm is triggered and the node goes back to sleep.



**Figure 4.** Proposed fully distributed, real-time and low-power leak detection solution for WWPM using low-cost MEMS accelerometers.

In scenario 2, only physical experiments were conducted using the laboratory testbed described in Section 2.1. The power consumption of the sensor nodes implementing the hybrid energy reduction technique were monitored using the power meter over a period of 1 day. We introduced leak events every hour over a twelve-hour period during the 1-day experimentation phase. The results and discussion of the power consumption reduction are found in Section 4.2.

#### 4. Results and Discussion

In this section, we present and discuss the results of scenario 1 in Section 4.1 and the results of scenario 2 in Section 4.2.

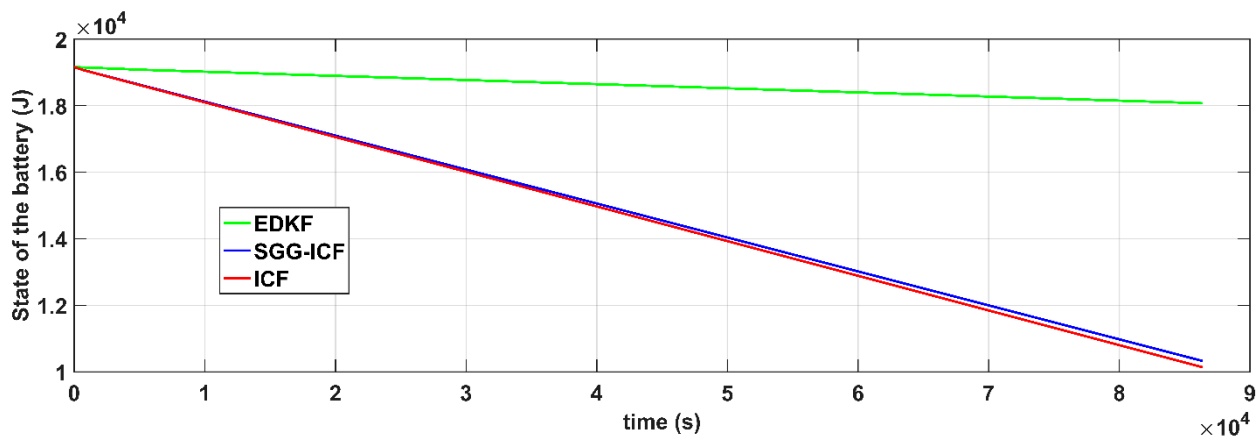
##### 4.1. Power Consumption Evaluation of the Three DKF Algorithms: Scenario 1

In this sub-section we examine the power consumption of the three selected DKF algorithms by presenting and discussing the results obtained from both simulations and

physical measurements. We start by presenting the results from simulations in Section 4.1.1 then validate them with results from laboratory experiments in Section 4.1.2.

#### 4.1.1. Results from Simulations

We conducted the simulations on the two-node linear WSN presented in Section 3.2. For each of the selected DKF algorithms that were implemented on the two-node linear WSN, we ran the simulation for a period of 1 day (86,400 s). Figure 5 depicts the energy profile of sensor node S2 for all the three selected DKF implementations.



**Figure 5.** Energy profile of the selected DKF algorithms over a simulation period of 1 day.

From Figure 5, it is clear that EDKF has the lowest energy consumption, while ICF consumes more battery energy compared to the other DKF algorithms. We also observed that the energy consumption of ICF and SGG-ICF were close. This is because both of them were involved in the same number of consensus/gossip iterations. From the results displayed in Figure 5, we see at time  $t = 80,000$  s, the battery energy of sensor node S2 has not crossed the  $1.8 \times 10^4$  J level for the EDKF implementation, whereas the battery energy of S2 had crossed the  $1.8 \times 10^4$  J level at time  $t = 10,000$  s for both ICF and SGG-ICF. Summarily, the results revealed that for a period of 1 day ( $t = 86,400$  s), sensor node S2 had exhausted 5.7%, 46.1%, and 47.1% of its total battery energy in the case of EDKF, SGG-ICF, and ICF implementations, respectively. Thus, we see that the energy consumption of SGG-ICF and ICF are over 8 times greater than the energy consumption of EDKF. The results revealed that EDKF is a more energy-efficient solution and will provide a longer WWPM lifetime when compared to ICF and SGG-ICF. Table 4 summarizes the battery energy usage of the sensor node for all three DKF implementations over a simulation period of 1 day.

**Table 4.** Battery energy consumption from simulations.

	Time (min)	Battery Energy (J)		Battery Energy Usage (%)
		0	1440	
DKF Algorithm	EDKF	19,160	18,069	5.7
	ICF	19,160	10,133	47.1
	SGG-ICF	19,160	10,321	46.1

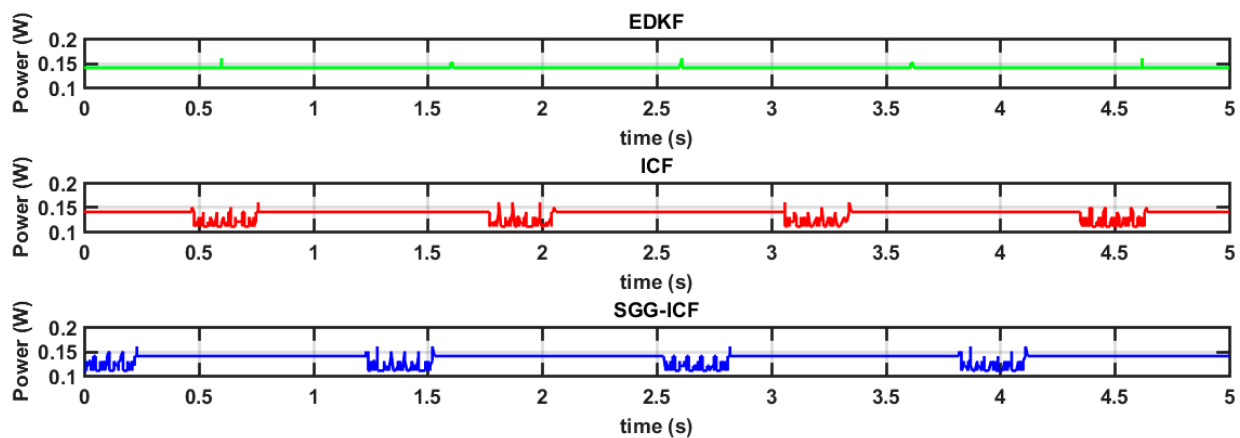
From the simulations, it shows that the maximum energy consumption for all three DKF implementations is 9000 J. Thus, the 3.7 V 2000 mAh Li-Po battery, which stores approximately 26,640 J of energy, is capable of supplying the nodes for more than a day of physical experimentation.



The next sub-sub-section presents the results of the physical experiments that measured the power consumption of the selected DKF algorithms and a comparison of the results with those obtained from simulations.

#### 4.1.2. Results from Laboratory Experiments

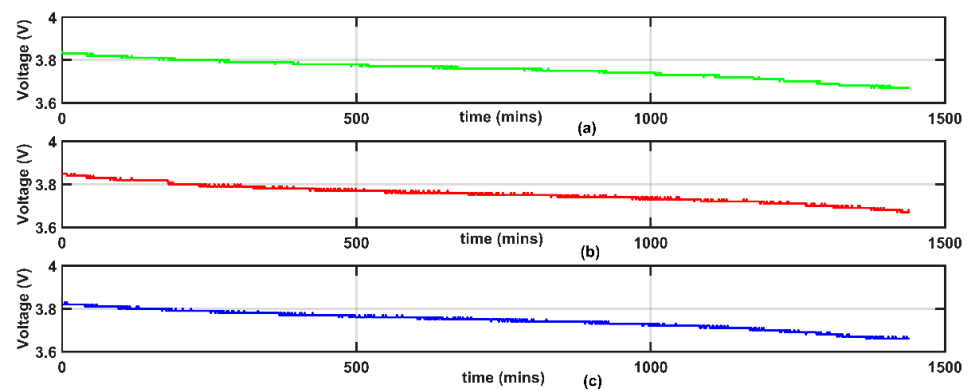
For each of the DKF implementation, we measured the current and voltage of the nodes. Figure 6 represents the power profile of sensor S2 in the case of the EDKF, ICF, and SGG-ICF implementations.



**Figure 6.** Power profile of the three DKF algorithms.

The average power consumed by the sensor node S2 for a period 1 h was found to be 135.616 mW for ICF and SGG-ICF, while it was 140.064 mW for EDKF. From these results, we can deduce that the lifespan of sensor node S2 when powered by the 2000 mAh capacity Li-Po battery will be approximately 54 h for ICF and SGG-ICF, and approximately 52 h for EDKF. Thus, the lifespan for the sensor node is approximately 2 days for all the DKF implementations. We also observed that the difference in power consumption for EDKF when compared to ICF and SGG-ICF was not as significant as what we had obtained from simulations. However, the results of the laboratory experiments validated the high communication burden of the ICF and SGG-ICF algorithms compared to the EDKF algorithm. As shown in Figure 6, the wavelets that occur every 1 s period on the power profile of ICF and SGG-ICF represent the five consensus/gossip communication iterations that these algorithms have to undertake to achieve an accurate estimate. This shows that the communication burden of these algorithms is high and so is their power consumption. The EDKF has only a single spike in its power profile after every 1 s because it requires just a single communication with the neighbouring node to fuse the estimates. We also observed something unexpected from the power measurement results obtained from the physical experiments. As can be seen from Figure 6, the power consumption of EDKF is higher than that of ICF and SGG-ICF. This is inconsistent with the results obtained from simulations and also from a theoretical perspective. For verification, we powered the sensor node for a period of 1 day using the Li-Po battery and monitored the rate of discharge of the battery's supply voltage. Figure 7 presents an experimental validation of the state of the sensor node battery's voltage over a period of 1 day for each of the DKF implementation.

Figure 7 demonstrates the accurate monitoring of the State of Charge (SOC) of the battery experimentally by observing the evolution of the battery's voltage over time. From observation, the battery discharge profiles for all three DKF implementations are found to be approximately the same at the end of one day, as indicated in Figure 7. The SOC of the battery is computed from the battery voltage measurements using Equation (5). Calculating the difference between the SOC at the beginning and end of the one-day experimental period yields the percentage of battery usage over that period. The results of these calculations are presented in Table 5.



**Figure 7.** State of charge of sensor node's battery for the selected DKF implementations: (a) EDKF, (b) ICF, (c) SGG-ICF.

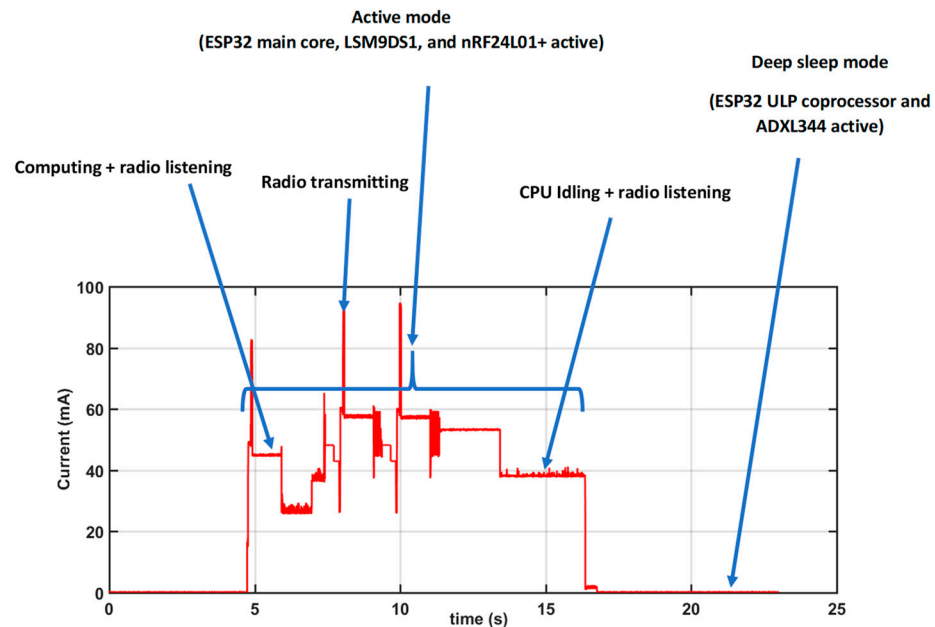
**Table 5.** Battery energy consumption from laboratory experiments.

	Time (min)	Battery Voltage (V)		SOC (%)		Battery Energy Usage (%)
		0	1440	0	1440	
DKF Algorithm	EDKF	3.83	3.67	67.32	26.61	40.71
	ICF	3.85	3.67	71.14	26.61	44.53
	SGG-ICF	3.82	3.66	65.26	23.92	41.34

Table 5 shows the battery energy consumption for the three selected DKF algorithms. As can be seen, EDKF has the lowest energy consumption (40.71%), followed by SGG-ICF (41.34%), and lastly ICF (44.53%). These results are consistent with those obtained from simulations. The results also show that there is no significant difference in the energy consumption of ICF and SGG-ICF, which is also consistent with the result obtained from the simulation. However, the difference between the energy consumption of EDKF and ICF is not as significant as was revealed by the simulation results. From the simulation results presented in Figure 5, the energy consumption of the ICF was 8 times higher than that of the EDKF, whereas the results from the laboratory experiments revealed that the energy consumption of the ICF is less than 2 times the energy consumption of the EDKF. One reason for this great difference can be explained by the fact that the CupCarbon simulator only models the energy consumption of the communication unit and does not model the energy consumption of the microcontroller and the sensor.

The presented results lead us to the conclusion that the simulation results align with the experimental results, validating our combined approach. The battery discharge performance of the three DKF algorithms is comparable, resulting in approximately equivalent node lifetimes for all implementations. The experimental results, depicted in Table 5, align with the simulation results, depicted in Table 4, with the exception of EDKF. Notably, from simulations (Figure 5), EDKF exhibits energy consumption over a one-day period that is eight times less than that of the other algorithms (ICF and SGG-ICF). This discrepancy arises from the fact that CupCarbon models solely the communication energy of the node, excluding the energy consumed by the MCU and sensor. Given that the ICF and SGG-ICF involve a communication frequency at least five times greater than that of the EDKF, their energy profiles appear similar but differ from the EDKF's. Figure 8 illustrates that, despite these differences, the battery discharge for all three algorithms is similar. Upon estimating the node's energy consumption from the State of Charge (SOC) of the battery, we find that there are no significant differences in the battery energy usage for all three DKF implementations over a period of one day. This similarity arises because the primary consumer of the node's battery energy in this case is the MCU rather than the communication unit. As all three DKFs maintain the ESP32 MCU in an active state continuously, their energy

consumption is expected to be similar. Specifically, the energy consumed by the MCU (ESP32 core) in the active state (36.8 mA) far surpasses that consumed by the transceiver (nRF24L01+) during transmission (11.3 mA).



**Figure 8.** Current profile of sensor node implementing duty cycling and hierarchical sensing.

We now turn our attention to the computation of the node's lifespan for all three DKF implementations when powered by a 3.7 V 2000 mAh Li-Po battery. Traditionally, lifespan estimation involves calculating the total energy of the battery divided by the average power of the node computed over an hourly period. Additionally, we have the opportunity to assess lifespan through the direct extrapolation of data from real deployed batteries used to power the nodes. Thus, we will employ two methods for estimating the node's lifespan, as summarized in Table 6.

**Table 6.** Sensor node's lifetime computed from power profile versus battery energy consumption measurements.

DKF Algorithm	Node's Lifetime from Power Measurements (Method 1)	Node's Lifetime from Battery Energy Consumption Measurements (Method 2)
EDKF	52 h	59 h
ICF	54 h	53.9 h
SGG-ICF	54 h	58.1 h

Method 1 utilizes the average power derived from power measurements presented in Figure 6 to calculate the node's lifespan. Method 2 leverages battery voltage measurements from Figure 7 and an approximate equation (cf. Equation (5)) to compute the State of Charge (SOC) of the battery at the beginning and end of the one-day experimental period. The energy usage for one day is then derived from the difference in SOC. With this energy consumption data, we calculate the time required to deplete the battery's energy.

Results presented in Table 6 reveal that the lifespan of the sensor nodes for all DKF implementations is approximately 2 days using both methods of estimation. The slight disparity in node lifespans using the two estimation methods may be attributed to the use of the approximate polynomial in Equation (5) for estimating the battery's SOC. Another contributing factor could be the variability in the average power of the sensor node over one-hour periods due to asynchronous events, causing fluctuations in power consumption.

For instance, higher communication interference leads to increased power consumption during such periods compared to periods with lower interference.

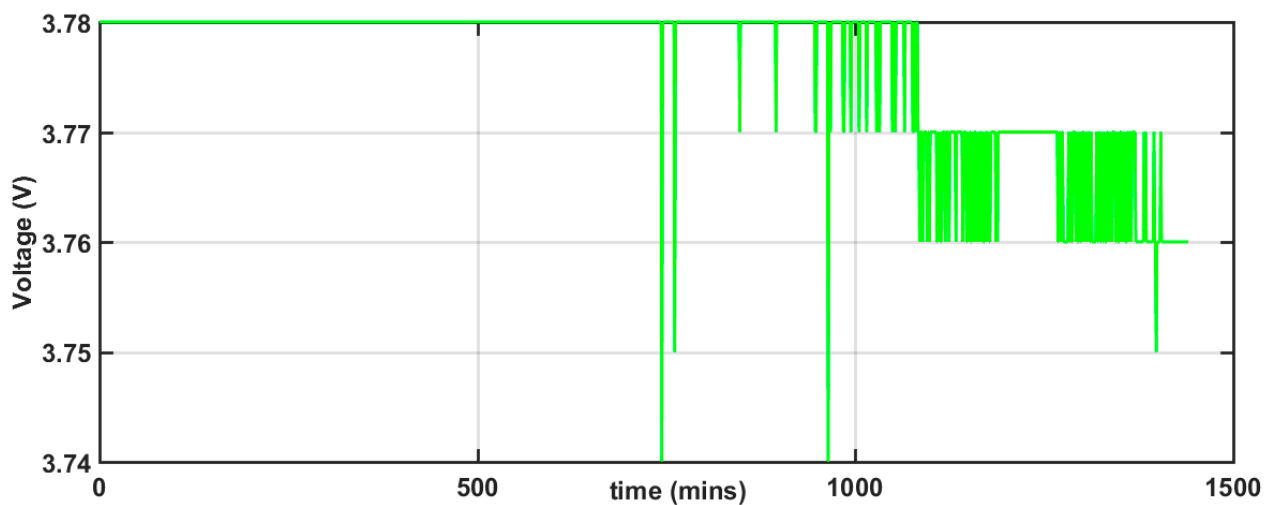
Moreover, the results presented in Table 6 indicate that the real implementation, as determined by Method 2, yields a lifespan that is longer compared to the estimated lifespan derived from power measurements (Method 1). This holds true for all DKF implementations except ICF, signifying that the real implementation offers extended node lifespan.

Lastly, from the results presented in Table 6, it can be seen that the lifespan of the sensor node is short (slightly greater than 2 days). This is because in these implementations, the ESP32 main core is operating in the modem sleep (active) mode throughout. As shown in [36], the ESP32 is a very power-hungry chip when operating in the modem sleep mode, given that it is a middle-end device. This huge power consumption is not appropriate for a WWPM system as it is required to operate for long periods without the need to recharge or replace the batteries. Thus, reducing the power consumption necessitates that the ESP32 be used as a low-end device by putting it most of the time in the deep sleep mode (relying only on the ULP coprocessor). However, to ensure real-time leak detection, it is necessary that the ESP32 awakes from sleep when there is a leak event to perform the processing tasks required to reliably detect the leak. Thus, the challenge is how to achieve low-power consumption without compromising the real-time performance of the WWPM system. We tackle this by proposing hierarchical sensing and duty cycling to achieve both low-power consumption and real-time detection. In the next sub-section, we present how we reduced the sensor node power consumption via duty cycling and hierarchical sensing, while maintaining real-time leak detection.

#### 4.2. Energy Consumption Reduction from Proposed Hybrid Technique: Scenario 2

The results presented in Section 4.1 show that the WSN lifespan obtained from the configurations of scenario 1 is too low. Though we are able to achieve real-time leak detection in scenario 1, such a deployment will require frequent battery replacement, thus making it not energy-efficient and suitable for battery-powered WWPM systems. Therefore, there is a need to further decrease the sensor node energy consumption so as to prolong the lifespan of the WWPM system. This sub-section presents and discusses the results of the implementation of the hybrid energy reduction technique that adds hierarchical sensing and duty cycling to the already-existing distributed computing at the node level to achieve both low-power consumption and real-time monitoring. For illustration purposes, we implemented the proposed solution on the laboratory testbed and evaluated the power consumption of the ICF algorithm (since it has the highest power consumption as shown in Section 4.1). Figure 8 displays the current profile of sensor node S2, and Figure 9 depicts the battery voltage discharge over a period of 1 day. The measurements presented in these figures were collected using the custom power measurement device we described in Section 2.3.

The results in Figure 8 revealed a current consumption of 300  $\mu\text{A}$  (0.99 mW) when the node is in the deep sleep mode. This is significantly close to the theoretical datasheet value of 290  $\mu\text{A}$  (150  $\mu\text{A}$  for ESP32 ULP coprocessor and 140  $\mu\text{A}$  for ADXL344 measurement mode supply current). From the results, it can be seen that there is a significant decrease in the power consumption of the sensor node. When there is no leak on the pipeline, the sensor node consumed a current as low as 0.3 mA when duty cycling and hierarchical sensing were implemented compared to 31.8 mA when duty cycling and hierarchical sensing were not implemented. The results in Figure 9 showed that the battery voltage dropped from 3.78 V to 3.76 V for a period of 1 day when both duty cycling and hierarchical sensing were implemented at the sensor node level. This corresponded to a battery energy consumption of 5.03% compared to 44.53% when duty cycling and hierarchical sensing were not implemented. Thus, it can be seen that the implementation of duty cycling at the sensor node level can lead to an increase in the sensor node's lifetime by a factor of 8. This is because the node stays in the ultra-low power state (where its current consumption is 0.3 mA) whenever there is no leak on the pipeline.



**Figure 9.** State of charge of sensor node's battery when duty cycling and hierarchical were implemented.

Finally, the results presented in this section demonstrates that our proposed solution can achieve real-time leak detection while at the same time preserving the lifespan of the WWPM system. Furthermore, the hierarchical sensing utilized in this hybrid approach does not reduce the leak detection reliability. From [46], it was shown that the magnitude of vibrations resulting from leaks are in the order of 1.01 g, leading to 1.01 g being set as the baseline value for leak detection. Thus, a leak alarm is triggered each time the difference between the estimated pipe surface acceleration and the baseline value exceeds the threshold value ( $\delta$ ) by 0.01 g. Such vibrations were effectively detected by the ADXL344 accelerometer when configured in the  $\pm 2$  g sensing range given its sensitivity value of 4096 LSB/g and noise floor level of  $530 \mu\text{g}/\sqrt{\text{Hz}}$ . To be sure of the leak event detected by the ADXL344 accelerometer, the ADXL344 always triggered a wakeup to the LSM9DS1 accelerometer to make more accurate measurements to confirm the leak event. In this way, the leak detection reliability is not reduced in any way.

In conclusion, the eightfold improvement (approximately 480 h or 20 days) observed in scenario 2 compared to scenario 1 still presents a relatively short lifespan in a practical sense. This limited duration arises from the deliberate introduction of leak events every hour for twelve hours within the 1-day experimentation period of scenario 2. The intention was to induce a substantial and measurable change in the battery's voltage, given the node's low power consumption during deep sleep mode, which otherwise showed no significant alteration in the battery's voltage over a 1-day span. Consequently, the 5% reduction in the node's battery within this timeframe can be attributed to the energy expended in detecting the leak events introduced during experimentation. This detection process necessitates the node's transition from deep sleep to active mode, consuming a considerable amount of energy.

However, it is crucial to note that in real-world scenarios, the frequency of leak events is typically lower. Assuming no leaks occur over a one-year period, the node would continuously remain in deep sleep mode, consuming a current of 0.3 mA. In this case, the estimated lifespan of the node, supplied with a 3.7 V 2000 mAh Li-Po battery, extends to approximately 278 days. Additionally, the node's lifespan can be prolonged by optimizing the balance between leak detection reliability and energy consumption. For instance, configuring the ADXL344 to operate at different Output Data Rates (ODR) during specific times, such as employing an  $\text{ODR} < 10$  Hz (with a measurement mode supply current of 30  $\mu\text{A}$ ) for 90% of every 10-s period and an  $\text{ODR} \geq 100$  Hz (with a measurement mode supply current of 140  $\mu\text{A}$ ) for the remaining 10%, will significantly enhance the sensor node's lifespan.



#### 4.3. Analysis of Energy Consumption Reduction Contribution of the Different Techniques

In this sub-section, we present an analysis of the contribution of each energy conservation technique on the energy reduction achieved by our proposed hybrid technique.

From our previous study, we compared the energy consumption of the centralized and distributed solution. The results revealed an increase in the lifespan of the WWPM by a factor of 13 in favor of the distributed solution. In this study, we focus on the distributed solution that implements distributed computing by using a Distributed Kalman Filter. Duty cycling and hierarchical sensing are then implemented on the distributed solution, and the contribution of each energy conservation technique (duty cycling and hierarchical sensing) to the energy reduction of the distributed solution is analyzed. Analysis is performed for the case where hierarchical sensing and duty cycling are implemented separately with the distributed solution (distributed computing) and the case where they are combinedly implemented with the distributed solution. For the analysis, we consider a one-hour period with no leak event. The result of the energy consumption analysis is presented in Table 7.

**Table 7.** Energy reduction contribution of different energy conservation techniques on the distributed solution.

Technique	Energy Consumption (mAh)
Distributed Computing only	37.89
Distributed Computing + Hierarchical Sensing	37.39
Distributed Computing + Duty Cycling	0.75
Distributed Computing + Hierarchical Sensing + Duty Cycling	0.3

From Table 7, it can be clearly seen that duty cycling has the highest impact on the energy reduction of the hybrid technique. Implementing duty cycling saves the node approximately 37 mAh of energy every hour in the case of no leak. This is because duty cycling affects the most energy consuming parts of the sensor node, i.e., the MCU and the transceiver, by switching them from the power-hungry active modes to the power-efficient deep sleep mode. The MCU and transceiver are switched to their active modes only when an interrupt is initiated by the accelerometer due to the detection of a leak event. Hierarchical sensing has a minimal effect on the energy reduction because it does not affect the MCU and transceiver. It only affects the sensors and thus influences only the energy consumed by the sensing unit. The result shows that implementing hierarchical sensing saves the node approximately 0.4 mAh of energy every hour in the case of no leak. This is because hierarchical sensing reduces the energy consumed by the sensing unit from 600  $\mu$ A to 140  $\mu$ A when no leak is detected on the pipeline.

## 5. Conclusions and Perspectives

This study delved into a critical aspect of wireless sensor network-based water pipeline monitoring (WWPM) systems—the imperative need for low energy consumption. The sustained operability of sensor nodes over extended periods without the requirement for frequent battery replacement is vital in such applications, especially when considering the typically battery-powered and physically inaccessible nature of these nodes. This study was centered on a comprehensive experimental investigation into a hybrid technique that incorporates distributed computing, hierarchical sensing, and duty cycling to optimize the energy consumption of sensor nodes, thereby extending the lifespan of WWPM systems. The energy consumption of three DKFs selected from data fusion strategies—specifically, diffusion, gossip, and consensus—were evaluated in the context of a distributed computing solution for WWPM. In all three DKF implementations, the sensor node's lifetime when powered with a 3.7 V 2000 mAh Li-Po rechargeable battery was between 52 to 54 h. The implementation of hierarchical sensing and duty cycling on the worst-case DKF shows an improvement in energy reduction by 800%. The proposed solution also achieved real-time

leak detection and preserved the lifespan of the WWPM system, all without degrading the leak detection reliability.

In conclusion, combining the results of leak detection performance in [46] and power consumption presented in Section 4.1, a compromise between leak detection performance and energy efficiency shows that EDKF is more optimal for a fully distributed leak detection solution in WWPM when compared to ICF and SGG-ICF, since its estimation performance is less affected by packet loss. Rather than implementing an ICF or SGG-ICF, which have higher sensitivities than an EDKF but have a high communication requirement (which has an adverse impact on the power consumption), we propose that the sensitivity of a WWPM solution implementing an EDKF can be improved by using machine learning (ML) techniques at the decision level [50]. Presently, it is feasible to implement ML on embedded systems with limited computational power, such as microcontrollers [51], as demonstrated in previous works that have employed deep neural networks [52]. Furthermore, ML is currently regarded as a strong candidate for enhancing security in IoT [53] and has already seen partial implementation in water monitoring applications [54]. Therefore, we advocate for the use of EDKF for filtering at the feature extraction level to provide more accurate features. These features can then be classified by an ML algorithm at the decision level to improve leak detection accuracy. Since this approach involves increased computation rather than communication, it is likely to reduce energy consumption at the sensor nodes. Consequently, it emerges as a promising choice for implementation in battery-powered sensor nodes utilized in WWPM, thanks to its energy-aware design. This makes it a strong candidate for a green WSN solution for water pipeline monitoring.

**Author Contributions:** Conceptualization, V.N.; methodology, V.N.; software, V.N.; validation, V.N. and F.M.; writing—original draft preparation, V.N.; writing—review and editing, V.N., F.M. and P.T.; supervision, F.M. and P.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless Sensor Networks: A Survey. *Comput. Netw.* **2002**, *38*, 393–422. [CrossRef]
2. Ojo, M.O.; Giordano, S.; Prociassi, G.; Seitanidis, I.N. A Review of Low-End, Middle-End, and High-End Iot Devices. *IEEE Access* **2018**, *6*, 70528–70554. [CrossRef]
3. Ramson, S.R.J.; Moni, D.J. Applications of Wireless Sensor Networks—A Survey. In Proceedings of the International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT), Coimbatore, India, 3–4 February 2017; pp. 325–329. [CrossRef]
4. Kandris, D.; Nakas, C.; Vomvas, D.; Koulouras, G. Applications of Wireless Sensor Networks: An Up-to-Date Survey. *Appl. Syst. Innov.* **2020**, *3*, 14. [CrossRef]
5. Ayadi, A.; Ghorbel, O.; BenSalah, M.S.; Abid, M. A Framework of Monitoring Water Pipeline Techniques Based on Sensors Technologies. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 47–57. [CrossRef]
6. Daniel, P.D. Non-Revenue Water Loss: Its Causes and Cures. 2016. Available online: <https://www.waterworld.com/home/article/14070145/nonrevenue-water-loss-its-causes-and-cures> (accessed on 27 June 2021).
7. van Zyl, J.E.; Clayton, C.R.I. The Effect of Pressure on Leakage in Water Distribution Systems. *Proc. Inst. Civ. Eng.-Water Manag.* **2007**, *160*, 109–114. [CrossRef]
8. Kartakis, S.; Yu, W.; Akhavan, R.; McCann, J.A. Adaptive Edge Analytics for Distributed Networked Control of Water Systems. In Proceedings of the First International Conference on Internet-of-Things Design and Implementation (IoTDI), Berlin, Germany, 4–8 April 2016. [CrossRef]
9. López-Ardao, J.C.; Rodríguez-Rubio, R.F.; Suárez-González, A.; Rodríguez-Pérez, M.; Sousa-Vieira, M.E. Current Trends on Green Wireless Sensor Networks. *Sensors* **2021**, *21*, 4281. [CrossRef] [PubMed]
10. Yetgin, H.; Cheung, K.T.; El-Hajjar, M.; Hanzo, L. A Survey of Network Lifetime Maximization Techniques in Wireless Sensor Networks. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 828–854. [CrossRef]

11. Kacimi, R.; Dhaou, R.; Beylot, A. Load Balancing Techniques for Lifetime Maximizing in Wireless Sensor Networks. *Ad Hoc Netw.* **2013**, *11*, 2172–2186. [\[CrossRef\]](#)
12. Anastasi, G.; Conti, M.; Francesco, M.D.; Passarella, A. Energy Conservation in Wireless Sensor Networks: A Survey. *Ad Hoc Netw.* **2009**, *7*, 537–568. [\[CrossRef\]](#)
13. Engmann, F.; Katsriku, F.A.; Abdulai, J.; Adu-Manu, K.S.; Banaseka, F.K. Prolonging the Lifetime of Wireless Sensor Networks: A Review of Current Techniques. *Wirel. Comm. Mob. Comput.* **2018**, *2018*, 8035065. [\[CrossRef\]](#)
14. Arjunan, S.; Pothula, S. A Survey on Unequal Clustering Protocols in Wireless Sensor Networks. *J. King Saud Univ.-Comput. Inf. Sci.* **2019**, *31*, 304–317. [\[CrossRef\]](#)
15. Elahi, H.; Munir, K.; Eugeni, M.; Atek, S.; Gaudenzi, P. Energy Harvesting towards Self-Powered IoT Devices. *Energies* **2020**, *13*, 5528. [\[CrossRef\]](#)
16. Wu, Y.; Li, B.; Zhang, F. Predictive Power Management for Wind Powered Wireless Sensor Node. *Future Internet* **2018**, *10*, 85. [\[CrossRef\]](#)
17. Abdulzahra, S.A.; Al-Qurabat, A.K.; Idrees, A.K. Energy Conservation Approach Oof Wireless Sensor Networks for IoT Applications. *Karbala Int. J. Mod. Sci.* **2021**, *7*, 340–351. [\[CrossRef\]](#)
18. Odeyinka, O.J.; Ajibola, O.A.; Ndinechi, M.C.; Nosiri, O.C.; Onuekwusi, N.C. A Review on Conservation of Energy in Wireless Sensor Networks. *Int. J. Smart Sens. Technol. Appl. IJSSA* **2020**, *1*, 1–16. [\[CrossRef\]](#)
19. Abdul-Qawy, A.S.H.; Almurisi, N.M.S.; Tadisetty, S. Classification of Energy Saving Techniques for IoT-Based Heterogeneous Wireless Nodes. *Procedia Comput. Sci.* **2020**, *171*, 2590–2599. [\[CrossRef\]](#)
20. Obeid, A.M.; Atitallah, N.; Loukil, K.; Abid, M.; Bensalah, M. A Survey on Efficient Power Consumption in Adaptive Wireless Sensor Networks. *Wirel. Pers. Commun.* **2018**, *101*, 101–117. [\[CrossRef\]](#)
21. Singh, S.; Anand, V.; Bera, P.K. A Delay-Tolerant Low-Duty Cycle Scheme in Wireless Sensor Networks for IoT Applications. *Int. J. Cogn. Comput. Eng.* **2023**, *4*, 194–204. [\[CrossRef\]](#)
22. Delicato, F.C.; Pires, P.F. Energy Awareness and Efficiency in Wireless Sensor Networks: From Physical Devices to the Communication Link. In *Energy-Efficient Distributed Computing Systems*; John Wiley & Sons: Hoboken, NJ, USA, 2012.
23. Sahar, G.; Bakar, K.A.; Rahim, S.; Khani, N.A.K.K.; Bibi, T. Recent Advancement of Data-Driven Models in Wireless Sensor Networks: A Survey. *Technologies* **2021**, *9*, 76. [\[CrossRef\]](#)
24. Suryavansh, S.; Benna, A.; Guest, C.; Chaterji, S. A Data-Driven Approach to Increasing the Lifetime of IoT Sensor Nodes. *Sci. Rep.* **2021**, *11*, 22459. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Nedham, W.B.; Al-Qurabat, A.K.M. A Review of Current Prediction Techniques for Extending the Lifetime of Wireless Sensor Networks. *Int. J. Comput. Appl. Technol.* **2023**, *71*, 352–362. [\[CrossRef\]](#)
26. Tripathi, A.; Gupta, S.; Chourasiya, B. Survey on Data Aggregation Techniques for Wireless Sensor Networks. *IJARCCCE* **2014**, *3*, 7366–7371.
27. Bendjima, M.; Feham, M. Intelligent Communication in Wireless Sensor Networks. *Future Internet* **2018**, *10*, 91. [\[CrossRef\]](#)
28. Sadeghioon, A.M.; Metje, N.; Chapman, D.; Anthony, C. SmartPipes: Smart Wireless Sensor Networks for Leak Detection in Water Pipelines. *J. Sens. Actuator Netw.* **2014**, *3*, 64–78. [\[CrossRef\]](#)
29. Liu, Y.; Ma, X.; Li, Y.; Tie, Y.; Zhang, Y.; Gao, J. Water Pipeline Leakage Detection Based on Machine Learning and Wireless Sensor Networks. *Sensors* **2019**, *19*, 5086. [\[CrossRef\]](#)
30. Mysorewala, M.; Sabih, M.; Cheded, L.; Nasir, M.T. A Novel Energy-Aware Approach for Locating Leaks in Water Pipeline Using a Wireless Sensor Network and Noisy Pressure Sensor Data. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 675454. [\[CrossRef\]](#)
31. Mysorewala, M. Time and Energy Savings in Leak Detection in WSN-Based Water Pipelines: A Novel Parametric Optimization-Based Approach. *Water Resour. Manag.* **2019**, *33*, 2057–2071. [\[CrossRef\]](#)
32. Saqib, N.; Mysorewala, M.F.; Cheded, L. A Novel Multi-Scale Adaptive Sampling-Based Approach for Energy Saving in Leak Detection for WSN-Based Water Pipelines. *Meas. Sci. Technol.* **2017**, *28*, 125102. [\[CrossRef\]](#)
33. Rashid, S.; Qaisar, H.; Saeed, H.; Felemban, E. A Method for Distributed Pipeline Burst and Leakage Detection in Wireless Sensor Networks Using Transform Analysis. *Int. J. Distr. Sens. Netw.* **2014**, *10*, 939657. [\[CrossRef\]](#)
34. Yazdekhasti, S.; Piratla, K.R.; Sorber, J.; Atamturktur, S.; Khan, A.; Shukla, H. Sustainability Analysis of a Leakage-Monitoring Technique for Water Pipeline Networks. *J. Pipeline Syst. Eng. Pract.* **2020**, *11*, 04019052. [\[CrossRef\]](#)
35. Karray, F.; Garcia-Ortiz, A.; Jamal, M.W.; Obeid, A.M.; Abid, M. EARNPIPE: A Testbed for Smart Water Pipeline Monitoring Using Wireless Sensor Network. *Procedia Comput. Sci.* **2016**, *96*, 285–294. [\[CrossRef\]](#)
36. Nkemeni, V.; Mieyeville, F.; Tsafack, P. A Distributed Computing Solution Based on Distributed Kalman Filter for Leak Detection in WSN-Based Water Pipeline Monitoring. *Sensors* **2020**, *20*, 5204. [\[CrossRef\]](#) [\[PubMed\]](#)
37. Bounceur, A. CupCarbon: A New Platform for Designing and Simulating Smart-City and IoT Wireless Sensor Networks (SCI-WSN). In Proceedings of the International Conference on Internet of Things and Cloud Computing, New York, NY, USA, 22–23 March 2016. [\[CrossRef\]](#)
38. Lopez-Pavon, C.; Sendra, S.; Valenzuela-Valdes, J.F. Evaluation of CupCarbon Network Simulator for Wireless Sensor Networks. *Netw. Protoc. Algorithms* **2018**, *10*, 1–27. [\[CrossRef\]](#)
39. Abusair, M.; Sharaf, M.; Muccini, H.; Inverardi, P. Adaptation for Situational-Aware Cyber-Physical Systems Driven by Energy Consumption and Human Safety. In Proceedings of the 11th European Conference on Software Architecture: Companion Proceedings, New York, NY, USA, 11–15 September 2017; ACM: Canterbury, UK, 2017; pp. 78–84. [\[CrossRef\]](#)

40. Narayandas, V.; Maruthavanan, A.; Dugyala, R. Remote IoT Correspondence for Coordinating End-Devices over MANET via Energy-Efficient LPWAN. *Int. J. Nanotechnol.* **2023**, *20*, 346–360. [[CrossRef](#)]
41. Saez-de-Camara, X.; Flores, J.L.; Arellano, C.; Urbieto, A.; Zurutuza, U. Gotham Testbed: A Reproducible IoT Testbed for Security Experiments and Dataset Generation. *IEEE Trans. Dependable Secure Comput.* **2023**, 1–18. [[CrossRef](#)]
42. Murnane, M.; Ghazel, A. A Closer Look at State Of Charge (SOC) and State Of Health (SOH) Estimation Techniques for Batteries. *Analog. Devices* **2017**, *2*, 426–436.
43. EEMB Co., Ltd. Lithium-Ion Battery Datasheet. Available online: <http://eemb.com> (accessed on 20 May 2021).
44. Lady Ada. Li-Ion & LiPoly Batteries. Adafruit Learning System. Available online: <https://learn.adafruit.com/li-ion-and-lipoly-batteries/voltages> (accessed on 20 May 2021).
45. He, S.; Shin, H.; Xu, S.; Tsourdos, A. Distributed Estimation over a Low-Cost Sensor Network: A Review of State-of-the-Art. *Inf. Fusion* **2020**, *54*, 21–43. [[CrossRef](#)]
46. Nkemeni, V.; Mieyeville, F.; Tsafack, P. Evaluation of the Leak Detection Performance of Distributed Kalman Filter Algorithms in WSN-Based Water Pipeline Monitoring of Plastic Pipes. *Computation* **2022**, *10*, 55. [[CrossRef](#)]
47. Kamal, A.T.; Farrell, J.A.; Roy-Chowdhury, A.K. Information Weighted Consensus Filters and Their Application in Distributed Camera Networks. *IEEE Trans. Autom. Control* **2013**, *58*, 3112–3125. [[CrossRef](#)]
48. Shin, H.; He, S.; Tsourdos, A. Sample Greedy Gossip Distributed Kalman Filter. *Inf. Fusion* **2020**, *64*, 259–269. [[CrossRef](#)]
49. Battistelli, G.; Chisci, L.; Selvi, D. A Distributed Kalman Filter with Event-Triggered Communication and Guaranteed Stability. *Automatica* **2018**, *93*, 75–82. [[CrossRef](#)]
50. Virk, M.A.; Mysorewala, M.F.; Cheded, L.; Ali, I.M. Leak Detection Using Flow-Induced Vibrations in Pressurized Wall-Mounted Water Pipelines. *IEEE Access* **2020**, *8*, 188673–188687. [[CrossRef](#)]
51. Warden, P.; Situnayake, D. *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*; O'Reilly Media: Sebastopol, CA, USA, 2020.
52. Novac, P.-E.; Boukli Hacene, G.; Pegatoquet, A.; Miramond, B.; Gripon, V. Quantization and Deployment of Deep Neural Networks on Microcontrollers. *Sensors* **2021**, *21*, 2984. [[CrossRef](#)]
53. Tomer, V.; Sharma, S. Detecting IoT Attacks Using an Ensemble Machine Learning Model. *Future Internet* **2022**, *14*, 102. [[CrossRef](#)]
54. Fan, X.; Zhang, X.; Yu, X.B. Machine Learning Model and Strategy for Fast and Accurate Detection of Leaks in Water Supply Network. *J. Infrastruct. Preserv. Resil.* **2021**, *2*, 10. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.