agriculture

MDPI

Article

Wireless Sensor Network Synchronization for Precision Agriculture Applications

Alexandros Zervopoulos [1,*], Athanasios Tsipis [1], Aikaterini Georgia Alvanou [1], Konstantinos Bezas [1], Asterios Papamichail [1], Spiridon Vergis [1], Andreana Stylidou [1], Georgios Tsoumanis [2], Vasileios Komianos [3], George Koufoudakis [1] and Konstantinos Oikonomou [1]

[1] Department of Informatics, Ionian University, GR-49100 Corfu, Greece; atsipis@ionio.gr (A.T.); akorina@ionio.gr (A.G.A.); kbezas@ionio.gr (K.B.); aspapa@ionio.gr (A.P.); svergis@ionio.gr (S.V.); astylidou@ionio.gr (A.S.); gkoufoud@ionio.gr (G.K.); okon@ionio.gr (K.O.)

[2] Department of Informatics and Telecommunications, University of Ioannina, GR-47100 Arta, Greece; gtsoum@uoi.gr

[3] Department of Audio and Visual Arts, Ionian University, GR-49100 Corfu, Greece; vkomianos@ionio.gr

* Correspondence: azervop@ionio.gr

Received: 3 February 2020; Accepted: 17 March 2020; Published: 24 March 2020

Abstract: The advent of Internet of Things has propelled the agricultural domain through the integration of sensory devices, capable of monitoring and wirelessly propagating information to producers; thus, they employ Wireless Sensor Networks (WSNs). These WSNs allow real time monitoring, enabling intelligent decision-making to maximize yields and minimize cost. Designing and deploying a WSN is a challenging and multivariate task, dependent on the considered environment. For example, a need for network synchronization arises in such networks to correlate acquired measurements. This work focuses on the design and installation of a WSN that is capable of facilitating the sensing aspects of smart and precision agriculture applications. A system is designed and implemented to address specific design requirements that are brought about by the considered environment. A simple synchronization scheme is described to provide time-correlated measurements using the *sink* node's clock as reference. The proposed system was installed on an olive grove to assess its effectiveness in providing a low-cost system, capable of acquiring synchronized measurements. The obtained results indicate the system's overall effectiveness, revealing a small but expected difference in the acquired measurements' time correlation, caused mostly by serial transmission delays, while yielding a plethora of relevant environmental conditions.

Keywords: smart agriculture; precision agriculture; internet of things; wireless sensor network; network synchronization; crop monitoring; olive grove monitoring

1. Introduction

Rapid advances in technology observed in recent times have positively affected most facets of everyday life, allowing for new possibilities in expressing human creativity and becoming an integral part of most professions. Innovation has become an integral aspect of agriculture, with the ulterior goal of maximizing productivity and yields. A testament to the integration of technological means in the agricultural domain constitutes precision agriculture [1]. According to precision agriculture, real time monitoring is utilized to inform decisions centered around crops and provide immediate and appropriate response.

Such an undertaking would be hardly feasible without the capability of remotely monitoring environmental, crop and soil conditions. This process is enabled by the deployment of a wide assortment of sensors, which are able to establish wireless communication, forming potentially vast networks, more widely termed as Wireless Sensor Networks (WSNs) [2]. WSNs have rapidly grown, largely due to progress in the field of microelectronics, becoming a scientific domain of its own, which is expected to play a significant role in Internet of Things (IoT) [3] environments, where sensors are integrated into everyday devices to automate and optimize a large number of processes.

A significant portion of the scientific world considers WSN applications in the agricultural domain as state-of-the-art and not unjustifiably. Given the certain impact brought about from shifting from rough estimations and hypotheses to accurate sensing and mathematical models, new horizons have opened up in precise agriculture, facilitating intelligent decision-making and thereby increasing productivity and minimizing cost, while simultaneously diminishing harm to the environment, all key enablers of future smart farming [4].

Aside from the sensing process, smart farming [5] and similar agricultural approaches deal with: (i) the data analysis process, including data mining and visualization; (ii) the decision-making process, involving decision-support systems, modeling and planning tools; and (iii) the action process, which includes online applications, monitoring robots and automation technologies. The aggregation and analysis of large volumes of data within short time periods grants producers the ability to act forthwith, contributing to the prediction and prevention of crop damage, the frequency of which never ceases to decrease, potentially as a consequence of climate change [6,7]. Thus, intelligent agriculture and other cross-disciplinary approaches in agricultural production, such as the one found in [8], is not merely a passing trend in innovation, but an auspicious endeavor for supplying satisfactory and adequate levels of food production of high quality [9,10], for an increasingly growing global population.

Although the opportunities arisen are multiple [11], multiple are also the challenges that need to be tackled, since these emerging technologies are still under development. One such challenge is the need for synchronization, which often comes up in networking environments, and especially so in distributed systems [12,13]. Network synchronization is required to provide time-correlated measurements from remote nodes, so that, when the time comes for the individual data to be processed or fused to compose a new set of information, the latter can be successfully interpreted by the end-user or application and utilized for the decision-making process [14]. Accurate synchronization can be difficult to achieve, due to multiple delays that can accumulate, depending on the utilized hardware and software, alike.

Additionally, a variety of factors need to be considered when deploying a WSN outdoors, as nodes may be exposed to harsh environmental conditions, which can cause malfunctions [15]. Furthermore, preemptive measures need to be taken to ensure the system's seamless operation and prevent intervention from external factors (e.g., human intrusion [16] and animal attacks [17]). Hence, appropriate hardware, materials and installation procedures need to be carefully selected for the deployment environment, as they are primary factors in the WSN's accountability and durability.

In this work, issues regarding the design and installation of a WSN system for synchronized sensing in the context of smart and precision agriculture applications are discussed. More specifically, various components, including Arduino Boards [18,19], a plethora of sensors and XBee modules [20] are integrated into a unified system to address specific design requirements, defined by the agricultural applications in mind. The proposed system is capable of monitoring a wide range of environmental and soil conditions through the integrated sensors. Additionally, a synchronization scheme is described in order to provide measurements correlated with the network sink's clock, as a reference point, and facilitate precision agriculture approaches. The particular synchronization scheme utilizes as few transmissions as possible, so as to minimize the system's energy footprint and thereby elongate the network's lifetime. The main contribution is the implementation and deployment of the proposed system, consisting of 30 nodes, on an olive grove aiming to identify major factors that need to be taken into account regarding synchronization. The system was experimentally evaluated with respect to the hardware's appropriateness and the software's precision in providing synchronized measurements that can be of utility for decision-making to producers and end-users. Significant attention is paid to implementation-specific feedback, regarding the synchronization results, which mostly focus on serial communication delays. Results acquired from the experiments indicate that the system is effective in accomplishing the specified goals, while parameters are identified for the serial communication interface to minimize synchronization delay.

The rest of this paper is structured in the following manner. Previous work related to the subject at hand is presented in Section 2. Design requirements and the system put together to address them is discussed in Section 3. The utilized scheme for acquiring synchronized measurements is described in Section 4. The conditions present in the olive grove, the experiments that were conducted on it and the corresponding results are detailed in Section 5. Finally, conclusions are drawn in Section 6.

2. Related Work

The vast majority of IoT, and by extension WSN, research revolves around their appliances in smart city environments [21–23]. This is especially true for IoT environments that are realized through WSNs [24]. More specifically, the abundance of small but powerful smart devices, which act as sensors and are equipped with high processing and network functionalities, has led to the formation of flexible and scalable WSNs with high accuracy and quality performance in many fields of modern life [25]. Characteristic examples refer to their adoption in domains such as healthcare [26], industrial infrastructures [27,28], marine environments [29], underwater [30,31] and air [32,33] pollution detection systems, structural health monitoring systems [34], unmanned aerial vehicles [35] and even the military [36]. Nevertheless, a popular trend nowadays refers to WSNs' application in the agricultural sector. This section lists past related research in the field focusing on modern challenges, one of the most crucial being time synchronization. It also provides insight regarding various past and ongoing projects in precision agriculture.

2.1. WSNs in the Agricultural Domain

There exists an increasing research volume [37–42] investigating WSNs' potential in smart farming and precision agriculture [11]. Until recently, these domains faced many challenges, mainly due to the lack of wide-area connectivity, energy resources and sometimes harsh environmental conditions that made difficult or even impossible the deployment of WSNs. Innovative and cutting-edge technological advances, however, have paved the way for new low-cost wireless technologies, e.g., ZigBee [43], that take us to places previously unexplored, allowing us to test, administrate and record the dynamics of such systems in secure and credible ways [44].

As such, many systems have been developed that focus not only on classical agricultural applications [45], such as crop monitoring [40], disease countermeasures [46] and pest detection [47], but also on various other aspects of farming, as stated in [4], where the needs of future agriculture are determined. These include (but are not limited to) management of underground planting [48], modern drone spraying and control techniques [49,50], robotic enhancements [28,51], security and privacy matters regarding the sensed data [52,53] and of course food safety and quality issues [9,10,37].

One of the current trends, however, focuses on the integration of cloud technologies to manage the huge volume of procured data. Cloud technologies free the end users, in this case the farmers, from the necessity of owning specialized equipment (hardware and software) in order to embed new services in their production lines, as indicated by Mekala and Viswanathan [54], who surveyed solutions that migrate cloud information and communication technologies, with the aim of building sustainable and remotely controlled agricultural applications. In a similar approach, Mekala and Viswanathan [55] in their project adopted the cloud computing paradigm to offer real-time services comprised of spraying, weeding, bird and animal scaring, keeping vigilance, moisture sensing, etc., while providing a data storage system for facilitating all the generated data in a smart and organized manner.

Due to their high customization capabilities, a popular WSN configuration, uses Arduino sensors [56] to generate environmental measurements and utilizes the ZigBee module for communication purposes, as described by Satyanarayana and Mazaruddin [57] and Papamichail et al. [58], whilst extensively documented by Kooijman [59]. However, with the coming of IoT, more microcontrollers have come to light in order to support or even remotely control the Arduino processes, such as Raspberry Pis [60,61], which can be setup to act as sensors [62], gateways [63], databases [64], servers [65] or fog devices [66].

Dwelling deeper into developed prototypes, that also find relevance to current work, the authors of [67] presented a WSN testbed to monitor the micro-climate of grapevine in Sicily, with the aim of managing spring period hazards. The WSN was then compared to a traditional fixed meteorological station and was proven to be more efficient in monitoring the local environmental conditions. On the other hand, Jeličić et al. [68], similarly to the current work, focused on olive groves. Olive trees, similar to grapevines, are very susceptible to diseases and weather changes and require continuous care in order to produce high quality products (mainly olive oil). Therefore, the authors described the design and deployment of a prototype, named MasliNET, that utilizes low-power sensors to keep watch over the olive groves in an energy optimized way. However, the proposed prototype is connected to the Internet, thus synchronization issues are not considered.

2.2. WSNs Challenges and Opportunities

One of the great benefits of employing WSNs for smart agriculture, as already mentioned, is the opportunities they offer through the use of low-cost mobile devices, microcontrollers and sensors, as discussed extensively in [69]. The abundance of such devices has triggered new methods that provide increased land coverage, reliable transmissions, flexibility and scalability, even under harsh environmental conditions. For example, the authors of [63] reported the design and development of an end-to-end system in Northern Wales comprising Arduino sensors and incorporating Raspberry Pis, used as gateways for interconnection with a cloud server.

To support the main cloud infrastructure and achieve reduced response time and even distribution of the network load, a subject of critical importance for system success, especially with the future coming of 5G technologies, the research community has turned to alternative solutions that take advantage of the fog computing paradigm to propose new hybrid cloud/fog models. For instance, Castillo-Cara et al. [70] proposed a scalable network architecture for monitoring and managing farms in rural areas that embeds fog computing capabilities to increase coverage and throughput. To even further enhance agricultural automation and remote sensing, these systems require processing of enormous amounts of remotely sensed data from different platforms and, therefore, greater attention is currently being devoted to machine learning methods and artificial intelligence algorithms [71,72].

Another crucial part of WSNs relates to energy consumption, which must be kept at minimum and manageable levels in order to prolong their lifetime. Many works have been reported in literature that attempt to tackle this challenge, like the ones found in [39,73]. At the same time, for efficient power management, many works analyze the role of the sink node, i.e., the node ultimately responsible for collecting all the sensed data. In [74], the authors proposed an analytical model for tackling the energy hole problem, by solving a facility location problem for selecting the optimum position for placing the sink node.

2.3. Synchronization in WSNs

In the agricultural context, the temporal variability of acquired measurements is an essential aspect that needs to be taken into account for effective monitoring and decision-making [75]. Thus, it is necessary to know the point in time that a measurement is made or an event occurs. The timestamp associated with each such measurement is usually acquired by the device carrying it out. In traditional computing environments, time issues are tackled by being connected to the Internet, which is generally not an option in WSNs due to strict cost, size and energy restrictions.

As such, the problem of synchronization arises in WSNs for multiple reasons [12]. Generally, each node is equipped with a local clock capable of tracking time locally with a certain degree of precision, determined by manufacturing parameters of the device responsible for keeping track of time. As a rule of thumb, the more expensive is the time-keeping device, the higher is the accuracy it provides and the more energy it requires [76]. In the case of Arduino boards, this clock keeps track of how long the device has been powered-on for.

Since each device maintains its own local clock and no other means of physically keeping track of time or correlation to a certain time zone are provided, it is necessary to somehow associate these timestamps to an actual point in physical time interpretable by humans. After all, if timestamp indications that are taken at the same point in physical time are completely disparate, the timestamps might as well be meaningless. Consider the case where a WSN consisting of a large number of nodes is deployed to cover a particular area. It is nigh impossible for all nodes to start operating simultaneously; consequently, their initial clocks are different and supplementing measurements with these clocks is pointless.

Correlation to physical time is often accomplished using GPS modules [13], which provide time-correlated measurements through a satellite connection. However, equipping each node with these modules can tremendously increase deployment cost, while other issues can arise, such as signal loss in suburban areas, etc. The correlation of a local timestamp to a physical point in time can be accomplished on the external device receiving the measurements, such as a laptop or remote server.

Typically, in WSNs, a single node is responsible for delivering the entire network's measurements to the outside world, known as the sink node. Because the sink node is responsible for data collection across the entire network, a reasonable choice is to instead correlate the remote nodes' clocks to that of the sink's [77], which is then associated with a physical point in time. This is a form of network synchronization that avoids the need for each individual node's clock to be associated with physical time.

Furthermore, time-keeping devices are known to drift apart, especially when subjected to harsh environmental conditions [78], which showcases the need for effective communication-based synchronization [79]. For reference, under normal conditions, the resonator of an Arduino Uno or Mega, operating at 16 MHz, lose 10s of seconds per day, a performance that is fairly poor [80] for time-critical applications. The significance of this deviation might not impact that many agricultural applications, but it is clear that even if the clocks of two nodes are synchronized at a certain point in time, their indications will eventually drift apart, creating the need for regular repeats of the synchronization process.

Another element that makes synchronization a challenging task is the fact that most applications usually only have access to the operating protocol's upper networking layers. Thus, while approaches, such as MAC layer timestamping [81] are effective, a large number of applications are not capable of utilizing them. Similarly, other approaches may require multiple rounds of communication to increase precision [82], which further drains the nodes' energy, reducing the network's lifetime and, as such, its sustainability and monitoring capacity.

3. System Setup

In this section, a comprehensive overview of the design requirements is procured, specifying an accessible, energy efficient and versatile system, designed for applications revolving around agriculture. Additionally, a system, along with its various components, is proposed to satisfy these requirements.

3.1. Design Requirements

According to Akyildiz et al. [25], a node in WSNs must comprise at minimum a processing unit, a transceiver, a set of sensors and a power unit. For any agriculture-based WSN, the selection of relevant environmental conditions to monitor is a fundamental preparatory step. Consequently, the set of sensors that can accurately measure these conditions, while keeping the cost and complexity of the resulting system as low as possible is equally important. Regardless of the choice of specific technologies, it is crucial for the deployed WSN to be capable of operating for extended periods of time before the need for recharging arises.

To enable wireless communication among the network's nodes, a transceiver is required. The suitability of the communication protocol is a key factor, especially when considering the system's energy efficiency and transmission delays. For the development of versatile and capable WSNs, a processing unit (typically a microcontroller) is required. This unit brings the whole system together, integrating the sensory components with the communication device and providing more fine-grained control of the system. When considering a power supply method, a primary consideration is its capacity, so that the network's lifetime is prolonged. Moreover, it would be optimal if the batteries could draw part of their energy from the environment [83], e.g., through solar panels, ultimately prolonging the WSN operation.

The equipment's size is also important in some cases. For example, when monitoring crops, some information might need to be gathered from the plants' leaves, in which case longer wires might be considered the optimal solution, in order to avoid housing the node close to the leaves where its weight can cause damage to the plant; therefore, the nodes' size and dimensions need to be carefully considered prior to field installation. Finally, it is worth pointing out that means of preventing and detecting malfunctions or external interference are worth considering, including sensors and protective cases.

3.2. Proposed System

The system proposed in this paper capitalizes on the Arduino prototyping board and XBee transceivers, which can be integrated along with sensory devices to form a versatile system, capable of serving as a WSN node. These components, along with relevant specifications, are discussed in the sequel.

3.2.1. Arduino Board

There is a wide variety of available Arduino boards, each featuring a different set of specifications in terms of both hardware and software, rendering them suitable for different applications. The most widespread board is the Arduino Uno [18], known to be a well-rounded board, which is heavily tailored towards new-comers in the world of electronics. Other boards worth mentioning include the Arduino Mega 2560 [19] (or just Arduino Mega), which boasts higher SRAM, more pins and a faster microcontroller, as well as the Arduino Nano [84], a much smaller board that can be customized according to an application's requirements, also boasting lower energy consumption, aimed at more experienced users.

Overall, selecting an appropriate processing unit is highly application-specific and, thus, the more generic and versatile boards are preferred in the context of this study. As such, the boards that have been used include Arduino Unos and a single Arduino Mega, serving as the sink node. While the Nano model could be considered the optimal choice for simple, real-world applications, the aforementioned boards form a fast and accessible prototyping system, which can be more easily augmented (compared to the Nano model) if the need arises, facilitating a wider variety of projects.

3.2.2. XBee Modules and Arduino Wireless SD Shield

XBee-PRO S2C [20] modules are transceivers operating at 2.4 GHz, which support three different protocols based on IEEE 802.15.4: (i) 802.15.4 itself; (ii) DigiMesh; and (iii) ZigBee [85]. However, for the current study, ZigBee is the protocol of choice, providing high level, mesh networking and self-healing capabilities, all while boasting high levels of energy efficiency [64].

ZigBee additionally defines, according to Farahani [86], three distinct device types: (i) the *Coordinator*; (ii) the *Router*; and (iii) the *End Device*. Each of these types provides different functionality to address specific application needs as follows.

Routers and Coordinators are more generic device types, constantly having to be powered-on, as they are essential for establishing multi-hop wireless communication. Routers are responsible for propagating messages among nodes that are too far away to communicate directly, while the

Coordinator sets up the necessary conditions and parameters for the network's operation. Only a single Coordinator may exist in a network.

In comparison, the only nodes that are able to utilize sleep mode are End Devices, which may occasionally power off to preserve their energy levels. However, End Devices are restricted in that they may only communicate with a single node, referred to as their parent. End Device parents are responsible for buffering messages until their associated End Devices "wake up" to receive them. Even though End Devices can significantly aid in decreasing energy consumption, they are not used for the purposes of the present study, as the setup needs to remain as simple as possible.

To accommodate the interfacing between the Arduino boards and the XBee modules, the Arduino Wireless Secure Digital (SD) Shields [87] are incorporated. Shields are used to augment an Arduino board's capabilities. The XBee module can be attached to the shield using a designated slot, which is attached to the board's pins. Thus, the module is able to exchange data and commands with the board via serial connection. This particular shield also features an SD memory card slot, which can prove useful as a secondary form of storage for measurements. In this case, the Coordinator is the only node of the system utilizing an SD memory card [88], in order to store measurements to be further analyzed.

Serial communication between the Arduino and the XBee module takes place in one of two modes: *Transparent* and *Application Programming Interface (API)*. When utilizing the Transparent mode of operation, the XBee module functions based on its preconfigured settings; these cannot be changed during runtime, making Transparent mode too basic for most applications. On the other hand, in the API mode of operation, communication between the two devices is established by arranging data into frames. Overall, XBee modules operating in API mode are much more capable and powerful networking units and, as such, they are well-suited for the proposed system's requirements. Forming the API frames manually is a painstaking process, however, which can be simplified using external libraries. In this instance, the Arduino library by GitHub user "Andrew Rapp" [89] has been utilized.

3.2.3. Sensors

Related work [38] indicates that environmental parameters necessary for the monitoring of agriculture include, but are not limited to: soil moisture, temperature, humidity and ultraviolet (UV) radiation. Additionally, means of detecting unanticipated node movement, which may occur due to intervention from humans and animals, or even natural disasters, should be set up preemptively.

Accordingly, a set of sensors has been selected for the purposes of this study, which satisfy the aforementioned requirements. The RHT-03 sensor [90] is responsible for monitoring temperature and humidity. To measure UV radiation, the VEML-6070 [91] UV light sensor is incorporated into the system, providing unitless measurements, based on how much UV light is sensed. Complementary is the YL-69 sensor [92], being responsible for measuring soil moisture, while the MPU [93] accelerometer is incorporated in order to estimate node movement by indicating sudden displacements.

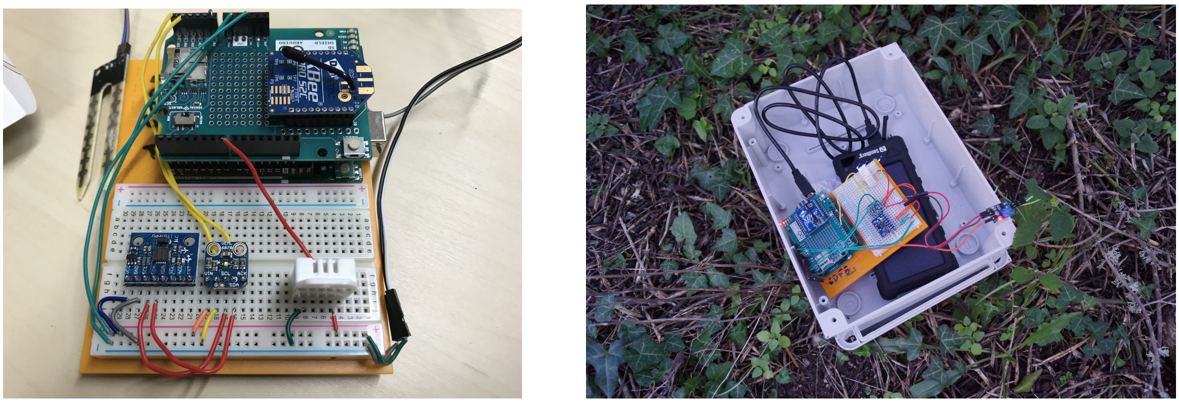
3.2.4. Power Supply

To maximize network lifetime and provide a reliable and durable source of power, a heavy duty power bank with the capability of recharging through solar energy would be optimal. In this case, Sandberg Outdoor Solar Powerbanks [94] have been utilized, which are designed to be maximally resilient and even rain proof. Moreover, they feature two USB ports each, a capacity of 16,000 mAh and built-in solar cells in order to permit energy harvesting.

3.3. Design Summary and Realistic Deployment

Putting everything together, a node is made up of either an Arduino Uno or Mega, playing the role of the microcontroller, equipped with an arsenal of sensors for monitoring UV radiation, temperature, humidity and soil moisture. XBee-PRO S2C modules, operating with ZigBee as the protocol of choice, interact with the Arduinos through an Arduino Wireless SD Shield, so as to provide energy efficient wireless communication.

A single node, along with a power bank, is stored in a protective case, which aims to prevent any malfunctions that may occur due to weather conditions, human intervention or any other external factors. A fully assembled node is depicted in Figure 1. Unfortunately, the particular protective cases being used block incoming light, affecting the UV radiation sensor's measurements.



(**a**) (**b**)

Figure 1. Fully assembled Arduino Uno nodes: (**a**) a node with all accompanying sensors; and (**b**) a node operating within the protective case.

Some of the decisions made for the system's design aim to simplify its scope to more accurately evaluate its effectiveness in an experimental setup, without compromising the system's capabilities in actual deployment. The study's primary concern is to recommend a realistic system capable of providing accurate and synchronized measurements of environmental conditions. Optimizing a system for real-world deployment is highly specific to the environment and application under consideration, which is outside the scope of this study. However, a few suggestions are made when considering deployment of WSNs for real applications.

Regarding development boards, the sensing and networking aspects considered in this paper have very trivial computational requirements. Sink nodes do need more resources, especially so for memory, as they need to handle incoming messages and store the collected measurements in an SD memory card; thus, the use of a more powerful board, such as the Mega, is recommended. For reference, an Arduino Uno program merely including the library for managing the SD memory card requires about 82% of its available memory. Furthermore, the higher energy requirements for the Mega model are alleviated, as it is always connected to a computer to offload the measurements. However, fully-fledged applications can be much more computationally demanding, as they may employ Machine Learning models, advanced constructs to facilitate information dissemination and routing, such as Connected Dominating Sets [95], etc. Overall, a balance must be stricken among computational resources, price and energy consumption depending on the application's requirements.

With respect to ZigBee device types, the use of End Devices is optimal. In this case, they have been discarded to minimize factors that might interfere with the synchronization's experimental assessment, as delays brought about by devices sleeping would be hard to account for. The impact of End Devices in terms of energy consumption cannot be understated, making them essential for real applications, without significantly affecting the synchronization's effectiveness.

4. Proposed Measurement Synchronization Scheme

Under the strict restrictions imposed by WSNs, a simple synchronization scheme, capable of operating at the application layer, is imperative. The approach studied in this paper aims at providing the sink node's clock to the entire network, so that it may be used as a point of reference. Having received and stored a reference clock, the rest of the nodes have sufficient information to estimate the sink's clock, whenever it is needed. It should be clear that this approach does not provide any correlation to real time, but merely a means of synchronizing acquired measurements to a common point of reference.

To define things more formally, let t_s and t_n be the clock indications at a given point in time of the network's sink s and node n, respectively. Assume for simplicity's sake that $t_n > t_s$. At any future instance $t'_n = t_n + r$, where $r > 0$ is the time elapsed since t_n, it may be assumed that the sink's clock will correspondingly be $t'_s = t_s + r$. Thus, it follows that $t'_s - t_s = t'_n - t_n$ or equivalently $t'_s = t_s + (t'_n - t_n)$. Therefore, if node n is aware of t_s at a local point in time t_n, it is possible to estimate the sink's clock t'_s at any future point in time t'_n.

However, there are a few things that complicate this process. First, due to the jitter inherent in the nodes' clocks, in reality, there will be a small deviation for r between the sink's and the node's clock. This is not necessarily a problem, if the timeframe between t_n and t'_n is not too large, as clock inaccuracies will not have accumulated as much in shorter time spans. Additionally, in order for the sink to inform node n of its clock, a transmission is required, which introduces additional time cost that needs to be taken into consideration.

Moreover, miscellaneous delays t_m occur due to other parameters, including the serial interface between Arduino and XBee devices, ZigBee MAC layer retransmissions for collision avoidance, etc. Such delays accumulate on a hop-by-hop basis and are hard to estimate, without extensive experimentation. These miscellaneous delays are also largely hardware and implementation-specific, which are not always easily portable across different architectures and devices.

In the current system, transmitting a frame can roughly be broken down into the following steps: (i) Arduino writes the frame to serial buffer; (ii) XBee transmitter reads the message from serial buffer; (iii) XBee transmitter attempts to transmit read frame; (iv) remote XBee receives message; (v) remote XBee writes frame to serial buffer; and (vi) Arduino reads frame from serial buffer. Since the Arduino microcontroller is involved only in the first and final steps, the intermediate steps cannot be accurately timed.

However, since transmission time over the air is incredibly fast, compared to serial data exchange, only the latter is worth accounting for. The rate by which data are transmitted over a serial connection on the Arduino platform is determined by the baud rate, which by default is equal to 9600. Similarly, using the default serial settings, each transmitted byte requires 10 bits to be transmitted over serial, due to start and stop bits being added. As such, the overall transmission rate is equal to 960 bytes per second, or, equivalently, it takes about 1.041 ms to transmit a byte of data.

When measurements are transmitted in the present system, they are converted to a string datatype. Using the current string format, each measurement message is on average 91 bytes long. Therefore, its transmission from the Arduino microcontroller to the XBee module is supposed to take approximately 94.73 ms, without considering frame headers. Similarly, when this measurement message is received, it will take equally long for the module to transmit it to the Arduino over serial. Therefore, a minimum deviation of approximately 190 ms is expected between the clocks of a remote node and the sink. In the experiments conducted in this study, the acquired data were expected to provide feedback about the correctness of these calculations, enabling future optimizations and parameter-tweaking. Furthermore, the experiments could provide an indication of the delays that the ZigBee protocol introduces and how these may affect synchronization applications.

Nevertheless, such delays are not preemptively accounted for at this stage in the present system, due to the fact that additional transmissions and calculations are required to approximate them during operation. Consequently, the resulting software (and potentially hardware) becomes harder to reason about, while demanding further resources from the already constrained system and thereby limiting the application's scope. Therefore, it is expected that the synchronization's precision will diminish across larger networks, which is not deemed impactful for the purposes of the present study's system or experiments, but should be considered in actual deployment scenarios.

In summary, the deployed synchronization scheme is depicted in Figure 2. The sink node initially transmits its clock indication at time t_s to all of the network's nodes. Remote nodes, upon being switched on, periodically transmit to the sink, including their time estimation, which is calculated as the initially received t_s, offset by the duration r since it is first received. Since miscellaneous delays t_m are not accounted for, the synchronization's precision will be off by at least $2t_m$, which is further increased by distance in hops between sensing and sink node. The synchronization process is repeated at regular intervals, so that jitter among the nodes' clocks is accounted for.

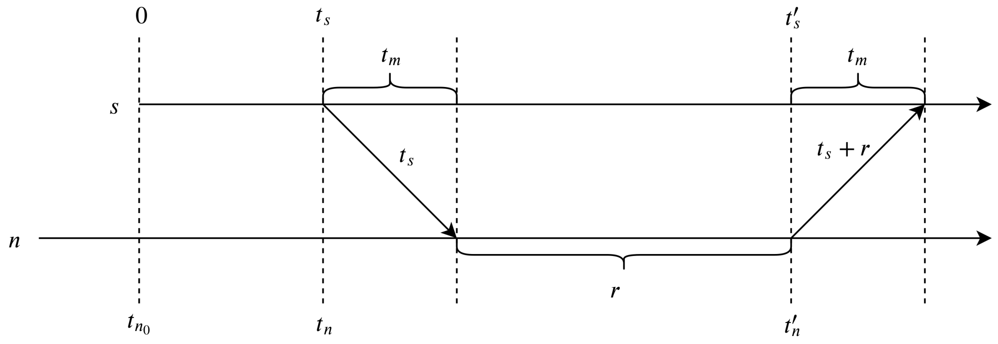


Figure 2. The synchronization scheme utilized in this study.

In a real world situation, the frequency of data collection would have to be selected depending on the appropriate frequency for the considered application. Unnecessarily high frequency of data collection leads to higher energy consumption and nodes needing to be recharged sooner, while too low frequency could lead to less effective decision-making, resulting in lost yields, crop damage, etc. In the case of olive groves, for instance, environmental condition fluctuations within a ten second (10 s) time span are unlikely to influence decision-making much. There are a few cases where measurement collection frequency might need to be significantly increased, as synchronization accuracy is extremely important. These include wildfire monitoring, as wildfires can spread at incredibly high velocity and unpredictable directions [96]; controlled-environment farming, such as hydroponic cultivations [97,98]; and finally olive grove irrigation scheduling [99], which can also be used in wildfire prevention [100]. Such strict requirements define the presented work's objective for precise synchronization even in time-critical applications.

Regarding the frequency of synchronization messages, it is entirely dependent on the synchronization accuracy requirements imposed by application-specific needs and the time-keeping device's accuracy. Each time-keeping device of the same model exhibits slightly different behavior as a result of manufacturing, while different models and types of devices (e.g., crystal vs. ceramic resonators) widely differ in their performance. Additionally, under varying environmental conditions, it may be beneficial to dynamically adjust this parameter to accurately match the required precision. Another parameter to keep in mind is the potential loss of synchronization messages, as the clocks of nodes that do not successfully receive this message will not be as accurate, which may have an impact on certain applications.

5. Experiments and Results

This section presents the results obtained from experiments conducted in an olive grove, where the proposed system was deployed to monitor environmental conditions. These experiments aimed to evaluate the system's effectiveness, not only in terms of networking efficiency, but also on the accuracy and aptness of the sensory devices.

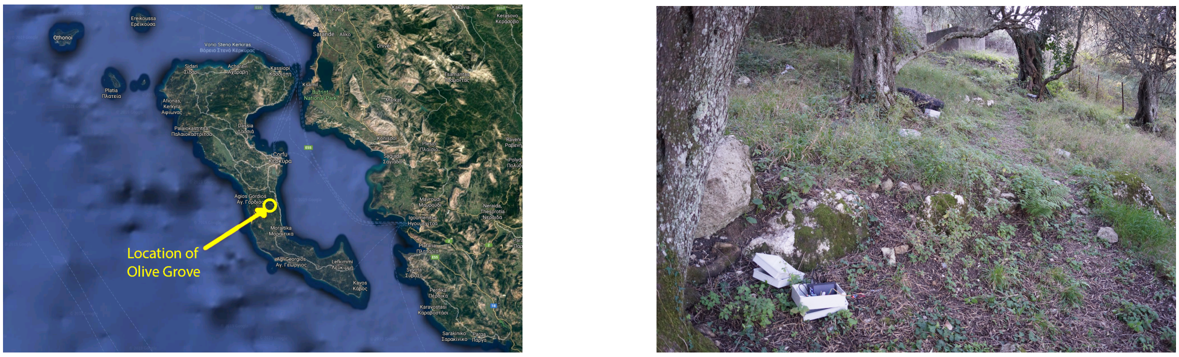
5.1. Test Environment

The contribution of this study was mostly in the sensing aspect of smart agriculture, i.e., the technical decisions that allowed the proposed system to deliver accurate and, in particular, time correlated measurements, so that they may be of use to the later stages of decision-making and action-taking. Therefore, determining an appropriate environment to conduct the experiments was a challenge in and of itself, as the conditions under which the WSN would operate needed to be representative of the final system, in order to identify potential malfunctions and evaluate the system's overall effectiveness. Conducting the experiments in an outdoor environment was important, due to node clock accuracy varying significantly in different environmental conditions, with temperature being especially important [78].

While olive trees are very resilient and require little care, they can be affected by shifts in environmental conditions and diseases, as well as pest infestations and fungal diseases [101]. The environmental conditions that affect them are fairly generic, with temperature and relative humidity being particularly relevant to the health of a tree's trunk, leaves and fruit, while also influencing the development of fungal diseases and pests [102,103]. The fact that olive trees require little human intervention in a short time span, while also potentially gaining from the benefits provided by smart agriculture applications utilizing WSNs, makes olive groves an ideal testing ground in the context of this study, focusing on the sensing, rather than the decision-making, aspects.

Additionally, the employed set of sensors is capable of monitoring these conditions, allowing for potential applications to identify factors affecting their health, while also taking advantage of a WSN's full breadth of capabilities in agricultural applications, such as maximizing yields. For instance, trees that may not be as exposed to the sun or spots in the ground that are particularly moist—both factors that can influence an olive tree's growth—can be identified through the relevant acquired measurements.

Thus, the proposed system was deployed in an olive grove located in Chalidiata, Corfu, Greece. The olive grove's location is pinpointed in Figure 3a. Experiments took place in February 2019. This particular olive grove is adjacent to a road, which facilitates transportation of the equipment. Its premises, spanning approximately 1000 m^2, partly lay on a hill facing away from the road, which hinders visibility. In combination with the fact that the area is surrounded by a wire fence, it can be presumed that unauthorized human interference would be minimized.



(**a**) (**b**)

Figure 3. Deployment site for the experiment: (**a**) satellite view of the olive grove; and (**b**) installation sites on the olive grove.

For the purpose of the experiments, a network consisting of 30 nodes made up of Arduino Unos and a single Mega board was deployed on the selected olive grove. Up to three nodes were placed at a single installation site, generally placed close to an olive tree, and all nodes were enclosed in their protective cases to protect them from potential harm or exposure to weather elements. The soil moisture sensor was connected via wires and remained outside the case, next to roots of olive trees. Installation sites were typically up to 10 m away from one another. Do note that the employed XBee modules have a communication radius of close to 100 m, thus installation sites could be further spread out. An example of an installation site is shown in Figure 3b.

5.2. Results

The experiments conducted aimed to provide an indication of the synchronization scheme's and the proposed system's effectiveness. The network consisted of a single Coordinator, which was connected to a laptop and carried no sensors, as well as 29 Routers, which were supplied with all sensors described in Section 3.

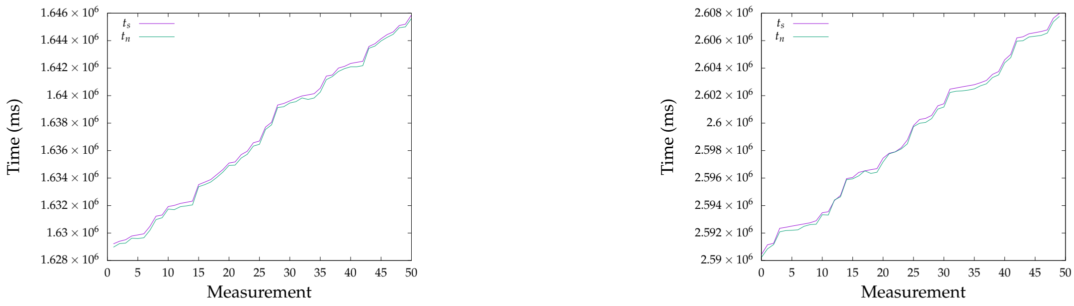
The experimentation procedure was organized in the following way. The Coordinator broadcasted its clock to all nodes at an 150 s fixed interval, upon being switched on. The Routers, on the other hand, upon being powered on, delivered messages to the Coordinator at a fixed interval of 10 s; these messages were not sent (or received) simultaneously, but were dependent on each node's switch-on time. These messages contained a set of current measurements from all attached sensors, accompanied by the sender's synchronized clock. When the Coordinator received a node's measurement, it was printed through the serial connection to the laptop's monitor and saved to an SD memory card for future examination.

The system was left to run for approximately 17 min, during the afternoon, to evaluate how the synchronization scheme fairs in the passage of time. The overall lifetime of the system turned out to be approximately three days, under normal conditions. Although the intervals of 10 and 150 s for measurement and synchronization messages, respectively, might seem excessively short, they were selected as a proof-of-concept to more accurately evaluate the synchronization scheme's effectiveness, even though a higher energy cost was incurred. These parameters should be optimized for specific applications, if real-world deployment is a consideration.

There are a few things worth mentioning regarding the experiments. Firstly, all of the required networking functionality was handled by ZigBee, utilizing most of the upper layers' capabilities, including network discovery, addressing, broadcasting and routing of messages. Secondly, although no information regarding the topology was available, the placement of the nodes effectively guaranteed that all nodes were a single hop away from the Coordinator. Thirdly, some of the nodes' protective cases were not sealed during the experiments, which was reflected in the relevant acquired measurements, such as UV radiation. This did not impact the system's capability of monitoring the relevant environmental parameters, which was the focal point of the experiments.

First, the synchronization's effectiveness is showcased from the acquired results. It is evident that the clocks received from remote Router nodes t_n are fairly close to that of the Coordinator t_s. More specifically, the difference $\Delta t = t_s - t_n$, where t_s and t_n are the clocks of the Coordinator and remote Router, respectively, has a mean value of 208.29 ms, standard deviation equal to 76.91 ms and 95% confidence interval of [205.39 ms, 211.20 ms]. Given that wildfire can spread from 0.29 to 1.95 ms^{-1} [104], the provided synchronization precision is considered to be effective, even for extremely time-critical applications.

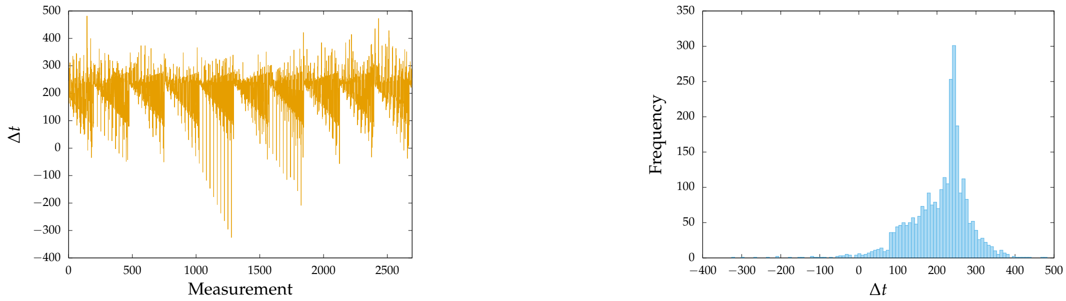
The clocks are depicted in Figure 4 for the first and last 50 measurements collected (Figure 4a,b, respectively), where in each data point the received clock indication from each node is compared to that of the sink's. For the first 50 measurements, the clocks of remote nodes are very close to that of the sink's. It is evident that the clocks are still heavily correlated in the last 50 measurements, even after the system has been active for a non-trivial amount of time. In fact, the window of the first 50 measurements has a difference in time between remote nodes and the sink Δt equal to 219.84 ms, whereas, for the last 50 measurements, the mean Δt is 211.54 ms. According to this metric, the last 50 estimations of the sink's clock are, on average, more accurate than the first 50 estimations.



(**a**) (**b**)

Figure 4. The Coordinator's (t_s) and Routers' (t_n) clocks for: (**a**) the first 50 measurements; and (**b**) the last 50 measurements.

Additionally, all Δt values along with the histogram of those values are presented in Figure 5 for all of the network's nodes, where each measurement corresponds to a Router clock indication received by the sink and Δt corresponds to the latter less the former's clock. The histogram shown in Figure 5b underlines the consistency of the proposed synchronization scheme, as most values are concentrated in the interval [200 ms, 300 ms]. There is an interesting trend worth pointing out that can be observed in Figure 5a: although occasional spikes in the values occur seemingly at random, there is a consistent downward slope. These slopes periodically reset, which would suggest that the Coordinator's synchronization message is the culprit, which is confirmed by the data. However, this downward trend occasionally continues, which might be caused by a remote node not receiving the synchronization message.

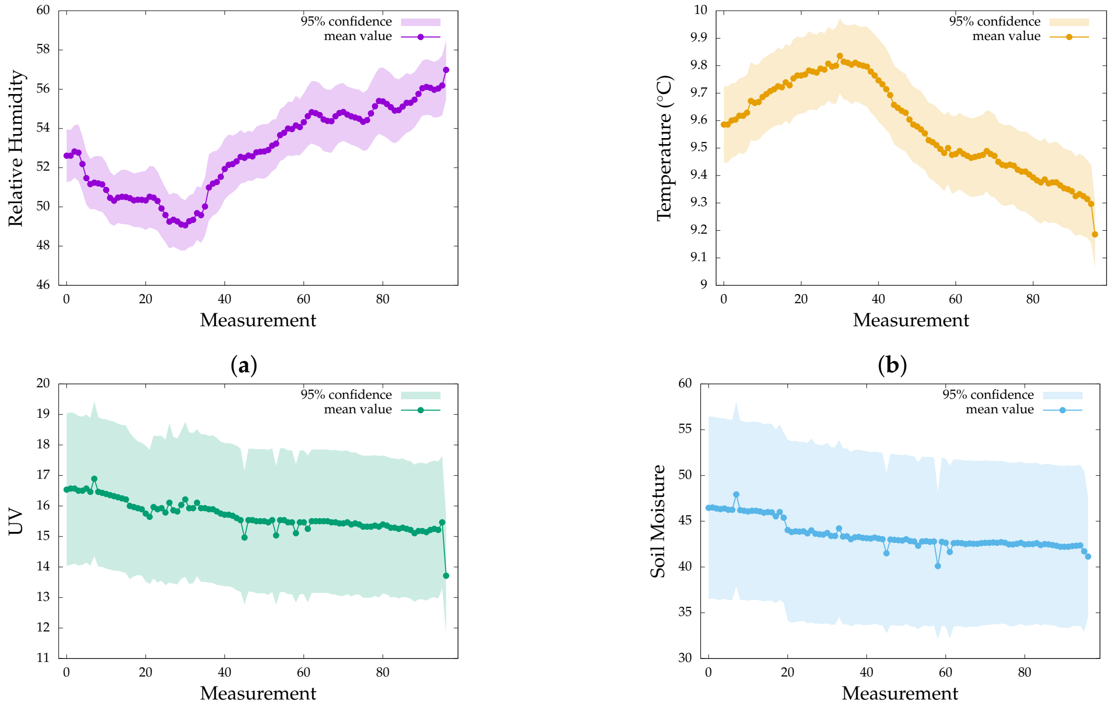


(**a**) (**b**)

Figure 5. Results related to synchronization: (**a**) time difference Δt between Coordinator and Router clocks; and (**b**) histogram of Δt.

As for the factors contributing to this delay, do recall from Section 4 that serial transmissions should account for approximately 190 ms. Given that the observed delay has a mean value of 208.29 ms, serial transmission takes up more than 90% of the miscellaneous delays t_m that were not accounted for. Considering frame headers are not taken into account in these calculations, it is clear that serial transmission forms a bottleneck for the entire transmission process. This also showcases that the ZigBee protocol introduces fairly low latency, when compared to the overhead introduced by the serial communication interface. Thus, increasing the serial baud rate is deemed essential for future applications, as a baud rate of 115,200 would reduce serial transmission time to 16 ms. Minimizing message length is equally important in reducing transmission time.

Moreover, the environmental conditions gathered during the experiment's measurements are jointly presented in Figure 6 with respect to time. Each data point in this figure represents a time window of 10 s, including a single measurement from all nodes. For each measurement in this time window, the mean value is displayed, along with a 95% confidence interval, as a means of showcasing variance of the measured environmental metrics, with respect to both space and time. Since most metrics are within reasonable ranges for the date and time the experiments were conducted, while also considering that they exhibit a relatively tight confidence interval, the system's effectiveness in providing accurate and synchronized measurements is showcased.



(**a**) (**b**)

(**c**) (**d**)

Figure 6. Environmental metrics monitored by the deployed system grouped in time windows with respect to time: (**a**) relative humidity levels; (**b**) temperature in degrees Celsius; (**c**) UV radiation; and (**d**) soil moisture mapped to [0, 100] range, with higher values indicating higher levels of moisture.

6. Conclusions

Smart agriculture is a field that has drawn heavy interest from the scientific community, especially through the contribution of WSNs. The measurements acquired from sensory devices are pinpoint accurate in addressing the challenges and requirements of crop cultivation on an individual basis, which is significantly affected by temporal variation of environmental, crop and soil conditions.

A primary factor in the utility of the acquired measurements is the ability to correlate them with respect to physical time, aided by the use of network synchronization approaches. Additionally, imperative for the accuracy of the yielded sensory indications is the utilization of appropriate equipment. The integration of both efficient software and proper hardware is conducive to the system's accountability, durability and overall effectiveness, so that later stages of precision agriculture, including data analysis, decision-making and taking action, can take place effectively.

Accordingly, in the current work, the design requirements of a WSN system facilitating agricultural applications are presented. A system setup is proposed to meet these design requirements, providing an accessible and energy efficient configuration that can be extended to accommodate beyond basic needs. A synchronization scheme is also described, which aims at providing time correlated measurements that are gathered by the network's sink node.

The proposed system configuration was experimentally evaluated in a selected olive grove, which was used as a testing ground representative of other agricultural applications. More specifically, the system operated in real conditions for a certain time period to assess the synchronization scheme's effectiveness in providing measurements that are time-correlated to the sink node's clock. Feedback from the conducted experiments indicates that, for this particular implementation, serial data transmission is to be blamed for most of the synchronization's inaccuracy. This inaccuracy can be addressed by maximizing serial baud rate, while minimizing the transmitted message's length.

Overall, the assembled system's hardware proved to be effective in monitoring relevant environmental conditions, while the developed software is effective in delivering the acquired measurements to the sink node with adequate accuracy. Therefore, the system in its entirety is deemed appropriate in accomplishing its goals of efficiently monitoring agricultural environments.

Author Contributions: Conceptualization, A.Z.; Data curation, A.T. and K.B.; Formal analysis, S.V. and G.K.; Funding acquisition, A.S.; Investigation, A.T., A.G.A., A.S., G.T., V.K. and K.O.; Methodology, K.B. and A.P.; Project administration, A.P.; Resources, A.G.A. and G.T.; Software, V.K. and K.O.; Validation, A.T., S.V. and G.K.; Writing—original draft, A.Z. and G.K.; and Writing—review and editing, A.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work was supported in part by project "A Pilot Wireless Sensor Networks System for Synchronized Monitoring of Climate and Soil Parameters in Olive Groves" (MIS 5007309), which is partially funded by European and National Greek Funds (ESPA) under the Regional Operational Programme "Ionian Islands 2014–2020."

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

Abbreviations

The following abbreviations are used in this manuscript:

5G	Fifth Generation of Cellular Network Technology
API	Application Programming Interface
AT	Attention
IoT	Internet of Things
WSN	Wireless Sensor Network
SD	Secure Digital
UV	Ultraviolet

References

1. Gebbers, R.; Adamchuk, V.I. Precision agriculture and food security. *Science* **2010**, *327*, 828–831. [CrossRef] [PubMed]
2. Baggio, A. Wireless sensor networks in precision agriculture. In Proceedings of the ACM Workshop on Real-World Wireless Sensor Networks (REALWSN 2005), Stockholm, Sweden, 20–21 June 2005; Volume 20, pp. 1567–1576.
3. Tan, L.; Wang, N. Future internet: The Internet of Things. In Proceedings of the 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), Chengdu, China, 20–22 August 2010; Volume 5, pp. V5:376–V5:380. [CrossRef]
4. Grift, T. The first word: The farm of the future. *Resour. Mag.* **2011**, *18*, 1.
5. Kamilaris, A.; Gao, F.; Prenafeta-Boldu, F.X.; Ali, M.I. Agri-IoT: A semantic framework for Internet of Things-enabled smart farming applications. In Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, USA, 12–14 December 2016; pp. 442–447.
6. Adams, R.M.; Hurd, B.H.; Lenhart, S.; Leary, N. Effects of global climate change on agriculture: An interpretative review. *Clim. Res.* **1998**, *11*, 19–30. [CrossRef]
7. Westermann, O.; Förch, W.; Thornton, P.; Körner, J.; Cramer, L.; Campbell, B. Scaling up agricultural interventions: Case studies of climate-smart agriculture. *Agric. Syst.* **2018**, *165*, 283–293. [CrossRef]
8. Channe, H.; Kothari, S.; Kadam, D. Multidisciplinary model for smart agriculture using internet-of-things (IoT), sensors, cloud-computing, mobile-computing & big-data analysis. *Int. J. Comput. Technol. Appl.* **2015**, *6*, 374–382.
9. Lin, J.; Shen, Z.; Zhang, A.; Chai, Y. Blockchain and IoT Based Food Traceability for Smart Agriculture. In Proceedings of the 3rd International Conference on Crowd Science and Engineering, ACM: Singapore, 28–31 July 2018; pp. 3:1–3:6. [CrossRef]

10. Klytchnikova, I.I.; Sadler, M.P.; Townsend, R.; Edmeades, S.; Choudhary, V.; Hussain, S.; Kray, H.A.; Fernandes, E.C.; Moses, E.; Cantrell, J.T.; et al. *Future of Food: Shaping a Climate-Smart Global Food System*; World Bank Group: Washington, DC, USA, 2015.
11. Bacco, M.; Berton, A.; Ferro, E.; Gennaro, C.; Gotta, A.; Matteoli, S.; Paonessa, F.; Ruggeri, M.; Virone, G.; Zanella, A. Smart farming: Opportunities, challenges and technology enablers. In Proceedings of the 2018 IoT Vertical and Topical Summit on Agriculture—Tuscany (IOT Tuscany), Tuscany, Italy, 8–9 May 2018; pp. 1–6. [CrossRef]
12. Lamport, L. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM* **1978**, *21*, 558–565. [CrossRef]
13. Sivrikaya, F.; Yener, B. Time synchronization in sensor networks: A survey. *IEEE Netw.* **2004**, *18*, 45–50. [CrossRef]
14. Tavares Bruscato, L.; Heimfarth, T.; Pignaton de Freitas, E. Enhancing time synchronization support in wireless sensor networks. *Sensors* **2017**, *17*, 2956. [CrossRef]
15. Andrade-Sanchez, P.; Pierce, F.J.; Elliott, T.V. Performance assessment of wireless sensor networks in agricultural settings. In Proceedings of the 2007 ASAE Annual Meeting. American Society of Agricultural and Biological Engineers, Minneapolis, MN, USA, 17–20 June 2007; p. 1.
16. Felemban, E. Advanced border intrusion detection and surveillance using wireless sensor network technology. *Int. J. Commun. Netw. Syst. Sci.* **2013**, *6*, 251. [CrossRef]
17. Giordano, S.; Seitanidis, I.; Ojo, M.; Adami, D.; Vignoli, F. IoT solutions for crop protection against wild animal attacks. In Proceedings of the 2018 IEEE International Conference on Environmental Engineering (EE), Milan, Italy, 12–14 March 2018; pp. 1–5. [CrossRef]
18. Arduino. Arduino Uno Rev3. Available online: https://store.arduino.cc/arduino-uno-rev3 (accessed on 14 December 2019).
19. Arduino. Arduino Mega 2560 Rev3. Available online: https://store.arduino.cc/mega-2560-r3 (accessed on 14 December 2019).
20. Digi. XBee-PRO Zigbee Through-Hole (Wire Antenna). Available online: https://www.digi.com/products/models/xbp24cz7wit-004 (accessed on 14 December 2019).
21. Atzori, L.; Iera, A.; Morabito, G. The internet of things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
22. Yin, C.; Xiong, Z.; Chen, H.; Wang, J.; Cooper, D.; David, B. A literature survey on smart cities. *Sci. China Inf. Sci.* **2015**, *58*, 1–18. [CrossRef]
23. Arasteh, H.; Hosseinnezhad, V.; Loia, V.; Tommasetti, A.; Troisi, O.; Shafie-Khah, M.; Siano, P. Iot-based smart cities: A survey. In Proceedings of the 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC), Florence, Italy, 7–10 June 2016; pp. 1–6.
24. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. A survey on sensor networks. *IEEE Commun. Mag.* **2002**, *40*, 102–114. [CrossRef]
25. Akyildiz, I.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless sensor networks: A survey. *Comput. Netw.* **2002**, *38*, 393–422. [CrossRef]
26. Alemdar, H.; Ersoy, C. Wireless sensor networks for healthcare: A survey. *Comput. Netw.* **2010**, *54*, 2688–2710. [CrossRef]
27. Güngör, V.Ç.; Hancke, G.P. *Industrial Wireless Sensor Networks: Applications, Protocols, and Standards*; CRC Press: Boca Raton, FL, USA, 2013.
28. Da Xu, L.; He, W.; Li, S. Internet of things in industries: A survey. *IEEE Trans. Ind. Informatics* **2014**, *10*, 2233–2243.
29. Xu, G.; Shen, W.; Wang, X. Applications of wireless sensor networks in marine environment monitoring: A survey. *Sensors* **2014**, *14*, 16932–16954. [CrossRef] [PubMed]
30. Cui, J.H.; Kong, J.; Gerla, M.; Zhou, S. The challenges of building scalable mobile underwater wireless sensor networks for aquatic applications. *IEEE Netw.* **2006**, *20*, 12.
31. Felemban, E.; Shaikh, F.K.; Qureshi, U.M.; Sheikh, A.A.; Qaisar, S.B. Underwater sensor network applications: A comprehensive survey. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 896832. [CrossRef]
32. Valente, J.; Sanz, D.; Barrientos, A.; Cerro, J.d.; Ribeiro, Á.; Rossi, C. An air-ground wireless sensor network for crop monitoring. *Sensors* **2011**, *11*, 6088–6108. [CrossRef]

33. Yi, W.Y.; Lo, K.M.; Mak, T.; Leung, K.S.; Leung, Y.; Meng, M.L. A survey of wireless sensor network based air pollution monitoring systems. *Sensors* **2015**, *15*, 31392–31427. [CrossRef]
34. Noel, A.B.; Abdaoui, A.; Elfouly, T.; Ahmed, M.H.; Badawy, A.; Shehata, M.S. Structural health monitoring using wireless sensor networks: A comprehensive survey. *IEEE Commun. Surv. Tutorials* **2017**, *19*, 1403–1423. [CrossRef]
35. Chen, Y.; Zhang, H.; Xu, M. The coverage problem in UAV network: A survey. In Proceedings of the Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Hefei, China, 11–13 July 2014; pp. 1–5.
36. Ðurišić, M.P.; Tafa, Z.; Dimić, G.; Milutinović, V. A survey of military applications of wireless sensor networks. In Proceedings of the 2012 Mediterranean conference on embedded computing (MECO), Bar, Montenegro, 19–21 June 2012; pp. 196–199.
37. Ruiz-Garcia, L.; Lunadei, L.; Barreiro, P.; Robla, I. A review of wireless sensor technologies and applications in agriculture and food industry: State of the art and current trends. *Sensors* **2009**, *9*, 4728–4750. [CrossRef] [PubMed]
38. Abbasi, A.Z.; Islam, N.; Shaikh, Z.A.; others. A review of wireless sensors and networks' applications in agriculture. *Comput. Stand. Interfaces* **2014**, *36*, 263–270. [CrossRef]
39. Anisi, M.H.; Abdul-Salaam, G.; Abdullah, A.H. A survey of wireless sensor network approaches and their energy consumption for monitoring farm fields in precision agriculture. *Precis. Agric.* **2015**, *16*, 216–238. [CrossRef]
40. Srbinovska, M.; Gavrovski, C.; Dimcev, V.; Krkoleva, A.; Borozan, V. Environmental parameters monitoring in precision agriculture using wireless sensor networks. *J. Clean. Prod.* **2015**, *88*, 297–307. [CrossRef]
41. Ojha, T.; Misra, S.; Raghuwanshi, N.S. Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges. *Comput. Electron. Agric.* **2015**, *118*, 66–84. [CrossRef]
42. Jawad, H.M.; Nordin, R.; Gharghan, S.K.; Jawad, A.M.; Ismail, M. Energy-efficient wireless sensor networks for precision agriculture: A review. *Sensors* **2017**, *17*, 1781. [CrossRef]
43. Baronti, P.; Pillai, P.; Chook, V.W.; Chessa, S.; Gotta, A.; Hu, Y.F. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards. *Comput. Commun.* **2007**, *30*, 1655–1695. [CrossRef]
44. Kumar, S.A.; Ilango, P. The impact of wireless sensor network in the field of precision agriculture: A review. *Wirel. Pers. Commun.* **2018**, *98*, 685–698. [CrossRef]
45. Popović, T.; Latinović, N.; Pešić, A.; Zečević, Ž.; Krstajić, B.; Djukanović, S. Architecting an IoT-enabled platform for precision agriculture and ecological monitoring: A case study. *Comput. Electron. Agric.* **2017**, *140*, 255–265. [CrossRef]
46. Azfar, S.; Nadeem, A.; Alkhodre, A.; Ahsan, K.; Mehmood, N.; Alghmdi, T.; Alsaawy, Y. Monitoring, Detection and Control Techniques of Agriculture Pests and Diseases using Wireless Sensor Network: A Review. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 424–433. [CrossRef]
47. Azfar, S.; Nadeem, A.; Basit, A. Pest detection and control techniques using wireless sensor network: A review. *J. Entomol. Zool. Stud.* **2015**, *3*, 92–99.
48. Yu, X.; Wu, P.; Han, W.; Zhang, Z. Overview of wireless underground sensor networks for agriculture. *Afr. J. Biotechnol.* **2012**, *11*, 3942–3948.
49. Moribe, T.; Okada, H.; Kobayashl, K.; Katayama, M. Combination of a wireless sensor network and drone using infrared thermometers for smart agriculture. In Proceedings of the 2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC), Las Vegas, NV, USA, 12–15 January 2018; pp. 1–2. [CrossRef]
50. Mogili, U.R.; Deepak, B. Review on application of drone systems in precision agriculture. *Procedia Comput. Sci.* **2018**, *133*, 502–509. [CrossRef]
51. Blackmore, S.; Stout, B.; Wang, M.; Runov, B. Robotic agriculture–the future of agricultural mechanisation. In Proceedings of the 5th European Conference on Precision Agriculture, Uppsala, Sweden, 9–12 June 2005; pp. 621–628.
52. Sykuta, M.E. Big data in agriculture: Property rights, privacy and competition in ag data services. *Int. Food Agribus. Manag. Rev.* **2016**, *19*, 57–74.
53. Ali, R.; Pal, A.K.; Kumari, S.; Karuppiah, M.; Conti, M. A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring. *Future Gener. Comput. Syst.* **2018**, *84*, 200–215. [CrossRef]

54. Mekala, M.S.; Viswanathan, P. A Survey: Smart agriculture IoT with cloud computing. In Proceedings of the 2017 International Conference on Microelectronic Devices, Circuits and Systems (ICMDCS), Vellore, India , 10–12 August 2017; pp. 1–7.
55. Mekala, M.S.; Viswanathan, P. A novel technology for smart agriculture based on IoT with cloud computing. In Proceedings of the 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), Coimbatore, India, 10–11 February 2017; pp. 75–82.
56. Banzi, M.; Shiloh, M. *Getting Started with Arduino: The Open Source Electronics Prototyping Platform*; Maker Media, Inc.: North Sebastopol, CA, USA, 2014.
57. Satyanarayana, G.; Mazaruddin, S. Wireless sensor based remote monitoring system for agriculture using ZigBee and GPS. In Proceedings of the Conference on Advances in Communication and Control Systems-2013, Mumbai, India, 18–19 January 2013.
58. Papamichail, A.; Alvanou, A.G.; Zervopoulos, A.; Bezas, K.; Vergis, S.; Koufoudakis, G.; Oikonomou, K.; Tsoumanis, G. Description of the Ionian University's Campus Wireless Network Testbed Infrastructure. In Proceedings of the 2019 South Eastern European Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Piraeus, Greece, 20–22 September 2019.
59. Kooijman, M. *Building Wireless Sensor Networks Using Arduino*; Packt Publishing Ltd.: Birmingham, UK, 2015.
60. Upton, E.; Halfacree, G. *Raspberry Pi User Guide*; John Wiley & Sons: Chichester, West Sussex, UK, 2014.
61. Saari, M.; bin Baharudin, A.M.; Hyrynsalmi, S. Survey of prototyping solutions utilizing Raspberry Pi. In Proceedings of the 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 22–26 May 2017; pp. 991–994.
62. Cabaccan, C.N.; Cruz, F.R.G.; Agulto, I.C. Wireless sensor network for agricultural environment using raspberry pi based sensor nodes. In Proceedings of the 2017 IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Pasay, Manila, Philippines, 1–3 December 2017; pp. 1–5.
63. Nundloll, V.; Porter, B.; Blair, G.S.; Emmett, B.; Cosby, J.; Jones, D.L.; Chadwick, D.; Winterbourn, B.; Beattie, P.; Dean, G.; et al. The Design and Deployment of an End-To-End IoT Infrastructure for the Natural Environment. *Future Internet* **2019**, *11*, 129. [CrossRef]
64. Nikhade, S.G. Wireless sensor network system using Raspberry Pi and zigbee for environmental monitoring applications. In Proceedings of the 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Chennai, India, 6–8 May 2015; pp. 376–381.
65. Flores, K.O.; Butaslac, I.M.; Gonzales, J.E.M.; Dumlao, S.M.G.; Reyes, R.S. Precision agriculture monitoring system using wireless sensor network and Raspberry Pi local server. In Proceedings of the 2016 IEEE Region 10 Conference (TENCON), Singapore, 22–25 November 2016; pp. 3018–3021.
66. Guardo, E.; Di Stefano, A.; La Corte, A.; Sapienza, M.; Scatà, M. A Fog Computing-based IoT Framework for Precision Agriculture. *J. Internet Technol.* **2018**, *19*, 1401–1411.
67. Catania, P.; Vallone, M.; Re, G.L.; Ortolani, M. A wireless sensor network for vineyard management in Sicily (Italy). *Agric. Eng. Int. CIGR J.* **2013**, *15*, 139–146.
68. Jeličić, V.; Ražov, T.; Oletić, D.; Kuri, M.; Bilas, V. MasliNET: A Wireless Sensor Network based environmental monitoring system. In Proceedings of the 2011 34th International Convention MIPRO, Opatija, Croatia, 23–27 May 2011; pp. 150–155.
69. Lipper, L.; Thornton, P.; Campbell, B.M.; Baedeker, T.; Braimoh, A.; Bwalya, M.; Caron, P.; Cattaneo, A.; Garrity, D.; Henry, K.; et al. Climate-smart agriculture for food security. *Nat. Clim. Chang.* **2014**, *4*, 1068–1072. [CrossRef]
70. Castillo-Cara, M.; Huaranga-Junco, E.; Quispe-Montesinos, M.; Orozco-Barbosa, L.; Antúnez, E.A. FROG: A robust and green wireless sensor node for fog computing platforms. *J. Sensors* **2018**, *2018*. [CrossRef]
71. Chlingaryan, A.; Sukkarieh, S.; Whelan, B. Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. *Comput. Electron. Agric.* **2018**, *151*, 61–69. [CrossRef]
72. Jha, K.; Doshi, A.; Patel, P.; Shah, M. A comprehensive review on automation in agriculture using artificial intelligence. *Artif. Intell. Agric.* **2019**. [CrossRef]
73. Hamouda, Y.; Msallam, M. Variable sampling interval for energy-efficient heterogeneous precision agriculture using Wireless Sensor Networks. *J. King Saud Univ. Comput. Inf. Sci.* **2018**. [CrossRef]

74. Tsoumanis, G.; Oikonomou, K.; Koufoudakis, G.; Aïssa, S. Energy-Efficient Sink Placement in Wireless Sensor Networks. *Comput. Netw.* **2018**, *141*, 166–178. [CrossRef]
75. Zhang, N.; Wang, M.; Wang, N. Precision agriculture—A worldwide overview. *Comput. Electron. Agric.* **2002**, *36*, 113–132. [CrossRef]
76. Sadler, B.M.; Swami, A. Synchronization in Sensor Networks: An Overview. In Proceedings of the IEEE Military Communications Conference (MILCOM), Washington, DC, USA, 23–25 October 2006; pp. 1–6. [CrossRef]
77. Li, Q.; Rus, D. Global clock synchronization in sensor networks. *IEEE Trans. Comput.* **2006**, *55*, 214–226.
78. Elsts, A.; Fafoutis, X.; Duquennoy, S.; Oikonomou, G.; Piechocki, R.; Craddock, I. Temperature-resilient time synchronization for the internet of things. *IEEE Trans. Ind. Inform.* **2017**, *14*, 2241–2250. [CrossRef]
79. Elson, J.; Römer, K. Wireless sensor networks: A new regime for time synchronization. *ACM SIGCOMM Comput. Commun. Rev.* **2003**, *33*, 149–154. [CrossRef]
80. Mani, S.K.; Durairajan, R.; Barford, P.; Sommers, J. An Architecture for IoT Clock Synchronization. In Proceedings of the 8th International Conference on the Internet of Things, Association for Computing Machinery, Santa Barbara, CA, USA, 15–18 October 2018; doi:10.1145/3277593.3277606. [CrossRef]
81. Skiadopoulos, K.; Tsipis, A.; Giannakis, K.; Koufoudakis, G.; Christopoulou, E.; Oikonomou, K.; Kormentzas, G.; Stavrakakis, I. Synchronization of data measurements in wireless sensor networks for IoT applications. *Ad Hoc Netw.* **2019**, *89*, 47–57. [CrossRef]
82. Bezas, K.; Komianos, V.; Oikonomou, K.; Koufoudakis, G.; Tsoumanis, G. Structural Health Monitoring In Historical Buildings Using A Low Cost Wireless Sensor Network. In Proceedings of the 2019 4th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Piraeus, Greece, 20–25 September 2019; pp. 1–4. [CrossRef]
83. Seah, W.K.; Eu, Z.A.; Tan, H.P. Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP)-Survey and challenges. In Proceedings of the Wireless VITAE 2009, 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, Aalborg, Denmark, 17 May 2009; pp. 1–5.
84. Arduino. Arduino Nano. Available online: https://store.arduino.cc/arduino-nano (accessed on 14 December 2019).
85. Digi. XBee®/XBee-PRO S2C Zigbee® RF Module User Guide. Available online: https://tinyurl.com/y5posdyh (accessed on 14 December 2019).
86. Farahani, S. *ZigBee Wireless Networks and Transceivers*; Newnes, Elsevier: Burlington, MA, USA, 2011.
87. Arduino. Arduino Wireless SD Shield. Available online: https://store.arduino.cc/arduino-wirelss-sd-shield (accessed on 14 December 2019).
88. Kingston. MicroSDHC Class 4 Memory Card. Available online: https://www.kingston.com/us/memory-cards/microsdhc-card-class-4 (accessed on 14 December 2019).
89. Andrewrapp. xbee-arduino. Available online: https://github.com/andrewrapp/xbee-arduino (accessed on 14 December 2019).
90. SparkFun. Humidity and Temperature Sensor - RHT03. Available online: https://www.sparkfun.com/products/10167 (accessed on 14 December 2019).
91. Vishay. VEML6070 UVA Light Sensor With I^{2}C Interface. Available online: https://www.vishay.com/ppg?84277 (accessed on 14 December 2019).
92. oddWires. YL-69 Soil Hygrometer Humidity & Soil Moisture Detection Sensor. Available online: https://www.oddwires.com/yl-69-soil-hygrometer-humidity-soil-moisture-detection-sensor/ (accessed on 14 December 2019).
93. InvenSense. MPU-6050. Available online: https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/ (accessed on 14 December 2019).
94. Sandberg. Sandberg Outdoor Solar Powerbank 16000. Available online: https://sandberg.it/en-mt/product/Outdoor-Solar-Powerbank-16000 (accessed on 14 December 2019).
95. Yu, J.; Wang, N.; Wang, G.; Yu, D. Connected dominating sets in wireless ad hoc and sensor networks—A comprehensive survey. *Comput. Commun.* **2013**, *36*, 121–134. [CrossRef]
96. Morvan, D.; Dupuy, J. Modeling the propagation of a wildfire through a Mediterranean shrub using a multiphase formulation. *Combust. Flame* **2004**, *138*, 199–210. [CrossRef]

97. Theopoulos, A.; Boursianis, A.; Koukounaras, A.; Samaras, T. Prototype wireless sensor network for real-time measurements in hydroponics cultivation. In Proceedings of the 2018 7th International Conference on Modern Circuits and Systems Technologies (MOCAST), Thessaloniki, Greece, 13–15 May 2018; pp. 1–4.
98. Kassim, M.R.M.; Harun, A.N.; Yusoff, I.M.; Mat, I.; Kuen, C.P.; Rahmad, N. Applications of wireless sensor networks in Shiitake Mushroom cultivation. In Proceedings of the 2017 Eleventh International Conference on Sensing Technology (ICST), Sydney, Australia, 4–6 December 2017; pp. 1–6.
99. Capra, A.; Scicolone, B. Irrigation Scheduling Optimisation in Olive Groves. *J. Exp. Agric. Int.* **2018**, doi:10.9734/JEAI/2018/44582 . [CrossRef]
100. Vicente-Charlesworth, L.; Galmés, S. On the development of a sensor network-based system for wildfire prevention. In Proceedings of the International Conference on Cooperative Design, Visualization and Engineering, Hong Kong, China, 11–14 September 2011; pp. 53–60.
101. Chliyeh, M.; Touati, J.; Selmaoui, K.; Touhami, A.O.; Filali-Maltouf, A.; El Modafar, C.; Douira, A. Bibliographic inventory of the olive tree (Olea europaea L.) fungal diseases in the world. *Int. J. Pure Appl. Biosci.* **2014**, *2*, 46–79.
102. Broufas, G.D.; Pappas, M.L.; Koveos, D.S. Effect of Relative Humidity on Longevity, Ovarian Maturation, and Egg Production in the Olive Fruit Fly (Diptera: Tephritidae). *Ann. Entomol. Soc. Am.* **2009**, *102*, 70–75. [CrossRef]
103. Koubouris, G.C.; Metzidakis, I.T.; Vasilakakis, M.D. Impact of temperature on olive (Olea europaea L.) pollen performance in relation to relative humidity and genotype. *Environ. Exp. Bot.* **2009**, *67*, 209–214. [CrossRef]
104. Cruz, M.G. Monte Carlo-based ensemble method for prediction of grassland fire spread. *Int. J. Wildland Fire* **2010**, *19*, 521–530. [CrossRef]