



HAL
open science

Monitoring of in-field risk of infection events by foliar pathogens using smart IoT nodes

Hassan Hammoud, Frédéric Weis, Christophe Langrume, Melen Leclerc,
Jean-Marie Bonnin

► **To cite this version:**

Hassan Hammoud, Frédéric Weis, Christophe Langrume, Melen Leclerc, Jean-Marie Bonnin. Monitoring of in-field risk of infection events by foliar pathogens using smart IoT nodes. 2024 IEEE Cybermatics Congress - International Conference on Internet of Things, Aug 2024, Copenhagen, Denmark. hal-04752653

HAL Id: hal-04752653

<https://inria.hal.science/hal-04752653v1>

Submitted on 25 Oct 2024

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

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



Distributed under a Creative Commons Attribution 4.0 International License

Monitoring of in-field risk of infection events by foliar pathogens using smart IoT nodes

Hassan Hammoud*[§], Frédéric Weis[‡], Christophe Langrume*, Melen Leclerc* and Jean-Marie Bonnin[†]

[†]IRISA, IMT Atlantique, Rennes, France

Email: jm.bonnin@imt-atlantique.fr

*IGEPP, INRAE, Le Rheu, France

Email: christophe.langrume@inrae.fr

Email: melen.leclerc@inrae.fr

[‡]IRISA, Rennes University, Rennes, France

Email: frederic.weis@univ-rennes.fr

[§]INRIA, Rennes University, Rennes, France

Email: hassan.hammoud@inria.fr

Abstract—In precision agriculture, risk of infection has emerged as a critical approach to optimize crop yields and resource usage. However, the current paradigm largely relies on massive data collection which expends energy resources and gathers a substantial amount of non-essential data. This paper challenges this paradigm by proposing a localized approach to precision agriculture through the deployment of Internet of Things (IoT) technology for real-time crop monitoring and resource management. The primary objective is experimenting local analysis depending on the environmental data collected by an implemented Wireless Sensor Network (WSN), and addressing a power consumption study for a long-lasting experimentation. We propose a test scenario based on a well-known infection model. Our results demonstrate the feasibility and efficiency of a localized approach over massive data collection by a robust and scalable monitoring system, showcasing the positive impact of orchestrated homogeneity among nodes in a local analysis, and the associated power consumption efficiency gains.

Index Terms—IoT, WSN, Edge Computing, Crop Monitoring, Energy Consumption, Local Analysis, RIOT OS

I. INTRODUCTION

Over the past two decades, sensor development has significantly enhanced environmental and agricultural monitoring capabilities [1, 2, 3]. Numerous studies demonstrate that Wireless Sensor Networks (WSNs) can notably improve the detection of crop infection events during the growing season [4, 5]. This enhancement is particularly valuable for developing decision support systems aimed at disease management in agriculture.

Since the automatic detection of plant pathogens (e.g., spores) in the environment remains challenging, the risk of crop infection is often predicted from climatic data. Most used systems rely on massive data collection, where data is acquired continuously by sensors and transmitted in real time to remote servers for subsequent analysis, as illustrated in Fig. 1-A. Nevertheless, with such an approach, a large amount of data can be transmitted and stored without providing any useful information to detect crop infection events. In non-tropical conditions, even when infection events are frequent, most of the data collected throughout the growing season are irrelevant

and do not correspond to climatic conditions favorable to disease occurrence. Developing more frugal data collection is therefore a pressing challenge to reduce the environmental cost of crop disease surveillance based on climatic conditions.

Key weather variables such as temperature and leaf wetness are crucial for predicting the risk of infection by foliar pathogens. These variables can vary significantly within a farm, across different plots, and even under the canopy. Although most commercially available solutions provide weather stations suitable for meso-climatic conditions, they do not support dense spatial deployment of sensors, complicating the capture of micro-climates. Advanced IoT technology could allow for more precise and spatially detailed assessments of disease risk. In the short term, this would enhance epidemiologists' understanding of the spatiotemporal dynamics of infection risks. In the medium term, it could aid in precisely localizing and optimizing treatments within precision agriculture.

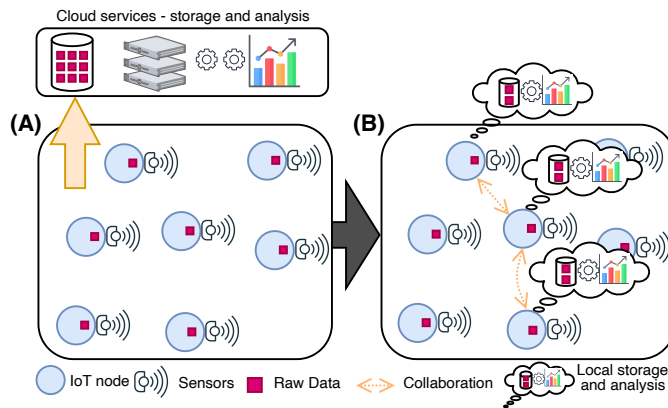


Fig. 1. In (A), classical function of IoT node, getting measurements from sensors and transmitting it to the server for storing and further analysis. In (B), IoT node performing local analysis and storage, as well as communicating with neighboring nodes to collaborate in the observation when necessary.

In this context, a comprehensive monitoring solution involving dense deployment in the environment is essential,

generating data efficiently and selectively. The advent of low-power wireless mesh networks offers a promising avenue for such deployments. This technology empowers a network of compact, battery-operated nodes, each equipped with various sensors. Upon activation, these nodes seamlessly integrate into a WSN, extending over vast areas and capable of functioning for extended periods. Unlike traditional WSNs, where nodes, or 'motives', face significant limitations in energy, processing power, and memory, the current approach emphasizes the development of software mechanisms tailored to the unique hardware and application requirements, focusing primarily on data acquisition and transmission to a centralized location.

In recent years, the architecture of traditional nodes has evolved towards more powerful microcontrollers (MCUs), which offer better trade-offs between processing capacity and energy consumption. This hardware evolution has been accompanied by software advancements. Beyond a network stack, these nodes can now run full-featured operating systems and provide software primitives and APIs. These advances greatly simplify the development of computer code intended to be executed directly on the node. The operating systems such as TinyOS, RIOT, and Contiki [6] are good examples. They rely on component libraries and IoT standards, and are supported by a large community. These systems have enabled extensive concept validation in WSNs, especially via open testbeds [7].

By leveraging local processing within sensor networks and utilizing advanced statistical methods, we can significantly enhance data collection efficiency, enabling the generation of relevant information directly on-site. This move towards local data processing and peer-to-peer node collaboration marks a shift towards environmental monitoring systems that operate independently of cloud infrastructures, as illustrated in Fig. 1-B. However, it is crucial to balance the optimization of local processing capabilities with the sustainability of the system's operational lifespan, particularly given the high energy demands of environmental sensors, which often require continuous power for activation. In this paper, we therefore set out to test the use of these programmable sensor nodes in a setting approaching a real case, through an experimental approach.

Our approach is based on the following steps:

- We select a well-recognized infection model within the phytopathologist community.
- We design a generic architecture that relies on an MCU capable of running a full-fledged operating system and driving various types of sensors while controlling energy consumption.
- We implement this model on the MCU and propose a scenario where the model is executed on nodes in a state-of-the-art mesh network.
- We conduct experiments with this scenario over a week, evaluating the nodes' ability to reliably obtain the expected observations and the impact on their energy consumption.

The remainder of this article is organized as follows. Section 2 presents the related works. In section 3, we present the selected infection model, the node architecture and the experimentation

and evaluation methodology. The results obtained are presented in section 4. Finally, section 5 concludes this paper.

II. RELATED WORKS

A lot of studies have been done regarding IoT in agriculture [1, 8]. Crop and soil monitoring is an important part of this. A significant amount of work relies on a single MCU, equipped with several low-cost sensors, where captured data is stored and analyzed on a remote server [9]. The systems presented in [4] and [5] monitor the health of a plant via a simple Arduino equipped with three sensors: temperature, humidity and color. The data are transmitted directly via a WiFi network to a remote server, which compares the collected data with a dataset to determine whether the plant is healthy or diseased. Similarly, authors in [10] employed ML techniques to predict the occurrence of plant diseases based on the environmental conditions captured by IoT sensors in the crop field.

Other research aims to extend the range of measurements to cover the entire crop. This type of deployment is enabled by low-power wireless mesh networks. [3] presents an experiment with this type of network in a peach orchard, in order to detect the risk of frost. The network relies on very low-power motes; each node continuously transmits a set of data (temperature, node status) to a remote server. [11] and [12] propose software architectures enabling the deployment of monitoring and risk analysis services in the cloud. Finally, [13] provides an overview of the adaptation of deployable AI techniques in the cloud, specifically in the field of smart agriculture and environmental monitoring. In these studies, data storage and analysis are performed in the cloud [2], with sensor nodes deployed in the environment being used to perform simple acquisition and communication tasks.

However, sensor node architecture is heterogeneous, ranging from the smallest 8-bit MCUs to the use of much more powerful systems-on-a-chip and microprocessors. The latter offer an enhanced trade-off between processing capabilities and energy consumption. With these more advanced architectures, a strong trend is to bring more intelligence to the sensor nodes. [14] presents a comprehensive overview of this IoT evolution, introducing an *extreme edge* and *edge* levels between sensing layer and cloud services. At the extreme edge, sensor nodes can leverage fully-fledged operating systems such as RIOT and Contiki, with software primitives to facilitate software development [15]. To the best of our knowledge, exploring these enhanced capabilities for the development of autonomous decision-making mechanisms and local risk analysis remains an unexplored area.

III. IoT-BASED SURVEILLANCE OF DISEASE RISK IN AGRICULTURE: MATERIALS AND METHODS

Regarding the monitoring of disease risks associated with weather conditions in agriculture, a more efficient and frugal approach would be to calculate the risk locally and retrieve only risk events to the server. Moreover, when favorable conditions occur, it would be relevant to expand surveillance to improve the spatial resolution of the risk assessment. Here, we aim to

experiment with this type of approach with an IoT solution that consists of low-cost and low-power consumption nodes, distributed within a wireless mesh network, onto which a wide range of commercial climatic sensors can be connected. The operating system embedded in the nodes is open-source and used to manage wireless communications, sleep-wake cycles, sensor measurements, and local processing. In addition to the local risk assessment by a main node, that monitors the risk associated with its local conditions at each wake-up, we also propose a mechanism that allows those main nodes to enroll their neighboring nodes and thus expand the spatial resolution of the monitoring. In this section, we present the selected infection model, the hardware and software architecture of the sensor nodes, and the test scenario and evaluation methodology.

A. Weather-Based Infection Model: The Magarey Model

In this study, we consider an infection model, proposed by Magarey, Sutton and Thayer [16], that predicts the risk of crop infection by foliar pathogens from temperature and leaf wetness. This model has been validated for numerous aerial plant pathogens and is well recognized within the community of phytopathologists. As shown by [12], this infection model is relatively simple and can be easily implemented into IoT nodes, connected with true sensors, to predict disease risk.

The model is formulated through an indicator (or step function) $R(x) = \mathbb{1}_{\mathcal{A}}$, which indicates the subset of environmental conditions \mathcal{A} that are favorable for plant infection by a given pathogen. The subset of favorable conditions \mathcal{A} corresponds to a two-dimensional domain defining the required leaf wetness duration W (in hours) for an infection to occur at temperature T (in °C):

$$W(T) = \frac{W_{min}}{f(T)} \leq W_{max} \quad (1)$$

$$f(T) = \left(\frac{T_{max} - T}{T_{max} - T_{opt}} \right) \left(\frac{T - T_{min}}{T_{opt} - T_{min}} \right)^{(T_{opt} - T_{min}) / (T_{max} - T_{opt})} \quad (2)$$

where $W(T)$ is the wetness duration requirement (in hours), W_{min} and W_{max} representing the minimal and maximal wetness durations, respectively, regardless of temperature. $f(T)$ is the temperature-dependent functional response which is utilized for calculating $W(T)$, while T_{min} , T_{max} , and T_{opt} are the minimal, maximal, and optimal temperatures for the occurrence of infections.

In our study, we consider the risk of pea crop infection by the fungal pathogen *Mycosphaerella pinodes* that causes the ascochyta blight disease. For this pathosystem, the estimated parameters of the infection model are:

- 14, 35 and 20 °C for T_{min} , T_{max} and T_{opt} respectively
- 6 and 72 hours for W_{min} and W_{max} respectively

In practice, temperature T is obtained by standard sensors used in commercial weather stations, while leaf wetness duration W can be measured with a dielectric leaf wetness sensor. The characteristics of the two sensors used for the experiment are detailed in the following section.

B. Design of the nodes

RIOT [17] is the operating system used for this experimentation. RIOT is an open-source OS licensed under LGPL [18]. It provides a programming environment similar to what is experienced in Linux, notably through standardized APIs. Applications and libraries can be implemented in C and C++. Furthermore, RIOT supports essential IoT protocols (mainly full 6LowPAN/RPL stack), enabling the straightforward setup of a mesh network. Lastly, RIOT supports various types of IoT hardware (MCU 8/16/32 bits), and is developed by a large community. These features position RIOT as a full-fledged OS for IoT, offering runtime support for local processing on each node.

The Particle Xenon is the board used for the experimentation. It is an off-the-shelf MCU. It features the Nordic nRF52840 chip, a system-on-chip (SOC) with a 32 bits ARM Cortex-M4F and an IEEE 802.15.4 short distance radio [19]. The architecture of this type of MCU is very generic. It offers a fairly limited number of I/O interfaces: GPIO port, UART, etc. However, the environmental sensors used for crop monitoring can have highly variable characteristics: digital or analog, various levels of power supply, different communication protocols, etc. In our experiment, we use two sensors representative of the diversity of technical solutions used to sense the environment. The temperature sensor uses the SDI-12 interface [20] to communicate with the MCU. SDI-12 (Serial Digital Interface at 1200 baud) is an asynchronous serial communication protocol specifically designed for interacting with intelligent sensors monitoring environmental data. The wetness sensor [21] estimates leaf surface wetness by measuring the dielectric constant of the sensor's upper surface. Using this principle, it is able to detect the presence of tiny amount of water. However, it is particularly energy-intensive when powered by the MCU.

To cope with this heterogeneity, we designed an additional board to provide different voltage levels and control mechanisms for different types of sensor. Particular attention has been paid to energy efficiency. The use of a complete OS and advanced sensors is not free. To ensure a long life for the battery powering the node, the set 'MCU + sensors' must be put into the deepest sleep as often as possible. In the case of a WSN used to monitor a crop, node operation can be synchronized, with MCUs waking up together only during short observation and analysis phases. To implement this effective duty cycle mechanism, an RTC (Real-Time Clock) component has been integrated into the additional board which keeps track of the current time, independently of the MCU. We also developed a software mechanism integrated into the official RIOT code, enabling the MCU to be woken up from a deep sleep by programming a simple RTC alarm.

The additional board is based on a low-cost PCB (Printed Circuit Board) using a set of generic electronic components, enabling several types of sensor to operate efficiently. This architecture*, represented schematically in Fig. 2, is very low-cost and can be easily adapted to different MCUs.

*The hardware architecture design will soon be released as open-source.

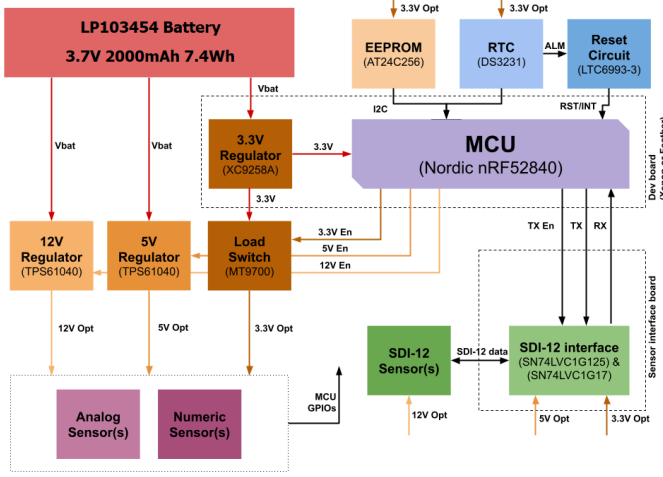


Fig. 2. Hardware connections of MCU with additional components on a single board

Fig. 3 depicts the set of components of the node used for our experimentation. This node is powered by a battery with a capacity of 2000mAh.

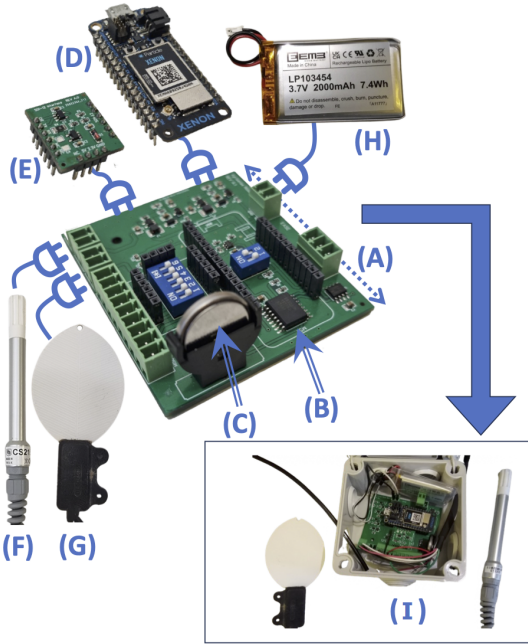


Fig. 3. Exploded view of all node components, where (A) represents the board with the RTC (B) battery-powered by (C), the Xenon MCU (D) and the SDI-12 interface for sensors (E) are directly plugged on the board, (H) is the battery for the node, (G) is Leaf Wetness Sensor, (F) is HygroVUE™10 Sensor for Temperature and Humidity. All components (except sensors) are placed in a watertight box (I) during the experiment.

C. Methodology

The experiment aims to:

- validate the stability of the system, including sleep-wake cycles, node synchronization, wireless communications, and the correct local execution of the infection model,

- assess the impact of commercial sensors and node behaviors on the battery life,
- test the robustness of our mechanism that expand the surveillance to neighbors.

To simultaneously test the correct implementation of the infection model and the impact of sensor consumption on the battery, we opted for a hybrid approach that combines true sensor activation with simulated temperature and leaf wetness duration values. Then, at each cycle, one node typically measures the required climatic variables. However, instead of using sensors data as inputs of the infection model, it uses predefined values.

Over 24 hours, the chosen pattern of climatic data consists of 1 hour at 1°C followed by 23 hours at 20 °C, and 6 dried hours followed by 18 wet hours (Fig. 4). Considering the Magarey infection model and *Mycosphaerella pinodes* on pea, the combination of these temperature and leaf wetness duration values gives 12 hours without risk (unfavorable conditions) followed by 12 hours with a risk of infection (favorable conditions) (Fig. 5). In real conditions daily fluctuations of temperature do occur and leaf wetness can appear daily with dew. However, our simulated climatic pattern has too stiff variations and is too regular to be realistic. It is designed to evaluate the correct execution of the model under all possible conditions regarding minimal and maximal required values of T and W.

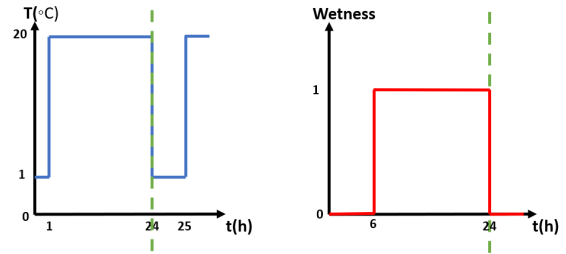


Fig. 4. 24 hours pattern used to test the implementation of the infection model.

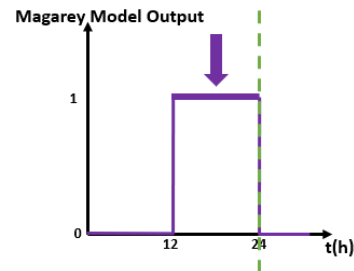


Fig. 5. Expected output of the infection model with the chosen temperature and leaf wetness duration values (Fig. 4). The output of the model is 1 when the climatic conditions are favorable and 0 otherwise.

Our experimentation is based on 8 nodes using the architecture presented in section III-B. These nodes, upon each wake-up, are programmed to automatically organize themselves into a mesh network, using the 6LowPAN/RPL protocols.

In our setup, we consider a 'Main Node', which evaluates environmental conditions of the risk of infection at each cycle (Algo. 1). If favorable conditions arise, it requests the

'Collaborative Nodes' (its neighbors) to also assess their local risks of infection (Algo. 2). Furthermore, to have reference point and to better study the effects of sensors energy consumption on the early stage of battery discharge, we set one node without any sensor and another one with all sensors activated at each cycle (Algo. 3).

The nodes are outfitted with identical commercial sensors and identical batteries that are fully charged. The experiment runs for 7 days, featuring a short sleep-wake cycle of 3 minutes. At the end of each cycle we used the EEPROM implemented on the additional board (Fig. 2) to store the current battery level and the outcome of the infection model (0 or 1). These collected data are then used to quantitatively assess the behavior of nodes, the correct execution of the infection model and the battery discharge.

Algorithm 1 *Main – Node* Behavior

Inputs: T (Temperature Value), W (Wetness Value)

- 1: Initialize the power module and RTC
- 2: **if** First Cycle **then**
- 3: Initialize node's network
- 4: Request and setting *RTC Synchronization*
- 5: **end if**

Routine Process

- 6: Activate Sensors (retrieve T and W)
 - 7: Apply infection model on simulated data
 - 8: **if** Positive Response **then**
 - 9: Send Extension Request to other nodes
 - 10: **end if**
 - 11: Store Data
 - 12: Sleep 3 mins
-

Algorithm 2 Collaborative Node Behavior Using Sensors When-Necessary Cycle

Inputs: T (Temperature Value), W (Wetness Value)

- 1: Initialize the power module and RTC
- 2: **if** First Cycle **then**
- 3: Initialize node's network
- 4: Request and setting *RTC Synchronization*
- 5: **end if**

Routine Process

- 6: **if** *Main – Node* Request Received **then**
 - 7: Activate Sensors (retrieve T and W)
 - 8: Apply infection model on simulated data
 - 9: **end if**
 - 10: Store Data
 - 11: Sleep 3 mins
-

Algorithm 3 Collaborative Node Behavior Using Sensors Each Cycle

Inputs: T (Temperature Value), W (Wetness Value)

- 1: Initialize the power module and RTC
- 2: **if** First Cycle **then**
- 3: Initialize node's network
- 4: Request and setting *RTC Synchronization*
- 5: **end if**

Routine Process

- 6: Activate Sensors (retrieve T and W)
 - 7: **if** *Main – Node* Request Received **then**
 - 8: Apply infection model on simulated data
 - 9: **end if**
 - 10: Store Data
 - 11: Sleep 3 mins
-

IV. ANALYSIS OF THE RESULTS

The analysis of experimental data shows that all nodes behave as expected, with risk periods being perfectly predicted (Fig. 6). Moreover, the collaboration among nodes proves to be robust, as all nodes designated for surveillance extension evaluated the local risk of infection upon request from the 'Main Node', without detecting any loss of wireless communication.

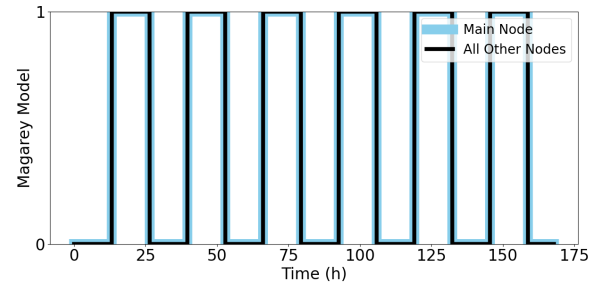


Fig. 6. Outputs of the Magarey model obtained in nodes during the experiment. All lines overlap.

Given the short duration of the experiment, our data, as shown in Fig. 7, only allows us to assess the initial phase of battery discharge, especially the small fluctuations and variations that are common in electronic components and measurement systems. To address that, we assumed a linear discharge over this period. Accordingly, we conducted an ANCOVA analysis [22], which is a statistical method for analyzing relationships between variables while adjusting for the effect of a continuous covariate, reveals a significant effect of the node type on the slope of the relationship ($p < 0.05$).

Post-hoc comparisons of nodes with Tukey's tests [23], which is a statistical method used to compare the means of multiple groups to determine if there are significant differences between them, show that:

- 'Node Without Sensors' and 'Node Activating Sensors Each Cycle' are statistically different.

- 'Main Node' and 'Other Nodes' are grouped together due to the small fluctuations and variations that are common in electronic components and measurement systems.

Furthermore, as expected, it can be concluded that:

- Least power-consuming node was 'Node Without Sensors'.
- 'Collaborative-Node Activating Sensors Each Cycle' had a significantly faster discharge than 'Main Node'. This difference is attributed to its waiting period for collaboration requests at each cycle, highlighting the non-negligible impact of this period on battery level. This effect is due to the energy-intensive nature of passive listening in low-power radio technologies, such as 802.15.4, particularly in the absence of reception or transmission on the channel.
- Computation of the Magarey model has a negligible impact on battery level, unlike sensors usage which consumes power.

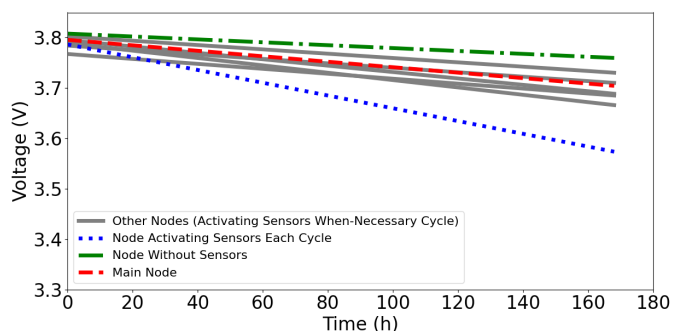


Fig. 7. Voltage levels dropping difference between nodes.

V. CONCLUSION

The outcomes from our experimental implementation validate the effectiveness of our approach across key parameters. The local execution of the Magarey model on nodes unfolded as anticipated, showcasing the system's robustness and reliability. This indicates successful network communication and resilience, as evidenced by the reception of all requests and active participation of nodes under each set of climatic conditions favorable to disease occurrence. A preliminary study on a limited number of nodes, not presented here, had already shown us that our IoT solution can capture accurately weather conditions in field conditions. Regarding the correct prediction of risk events in this study we believe that our nodes may predict accurately the risk of infection by foliar pathogens in field conditions.

Our results also illustrate the importance of the duty cycle mechanism, allowing a node to be turned off as frequently as possible. We explored different ways of setting up IoT nodes and how each setup affects their power usage. The specific way IoT nodes were used for local processing impacted the global power consumption. A short-term goal is to analyze the various sources of consumption more precisely, particularly based on the type of sensors used and their distribution among neighboring nodes.

These findings demonstrate that by limiting the amount of data collected and extending it only when necessary, a more targeted and frugal approach surpassing extensive data collection is feasible for effective monitoring. This underscores the suitability of our approach and points to promising directions for future research, highlighting its potential for decentralized, cost-effective, and efficient local data analysis and processing in WSNs. Research into the scalability and adaptability of the proposed methodology to various network architectures could offer significant enhancements.

Looking ahead, we aim to conduct larger-scale experiments using this model or explore other propagation models to further validate our methodology and broaden its applicability. Additionally, we plan to develop tools that will assist epidemiologists—those who are not necessarily specialists in computer science—in the deployment, maintenance, and configuration of models. Researching these user-friendly tools can greatly enhance the accessibility and practicality of our approach, promoting broader adoption and more versatile application across different environmental and operational contexts.

REFERENCES

- [1] V. Pawar and S. B. Deosarkar, "IoT-based smart agriculture: an exhaustive study," *Wireless Networks*, 2023, Volume 29, pp 2457–2470.
- [2] T. Ojha, S. Misra, N. S. Raghuvanshi, "Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges," *Computers and Electronics in Agriculture*, 2015, vol.118, pp. 66-84.
- [3] T. Watteyne, A. L. Diedrichs, K. Brun-Laguna, J. E. Chaar, E. Javier, D. Dujovne, J. C. Taffernaberry, and G. A. Mercado, "PEACH: Predicting Frost Events in Peach Orchards Using IoT Technology," *EAI Endorsed Transactions on Internet of Things*, Sept. 2016, vol. 2, num. 4.
- [4] M. A. Nawaz, T. Khan, R. M. Rasool, M. Kausar, A. Usman, T. F. N. Bukht, R. Ahmad, and J. Ahmad, "Plant Disease Detection using Internet of Thing (IoT)," *International Journal of Advanced Computer Science and Applications*, 2020, vol. 11, num. 1.
- [5] R. Yakkundimath, G. Saunshi, and V. Kamatar, "Plant Disease Detection using IoT," *Global Journal of Computer Science and Technology*, 2018, vol. 8, num. 9.
- [6] O. Hahm, E. Baccelli, H. Petersen, and N. Tsiftes, "Operating Systems for Low-End Devices in the Internet of Things: A Survey," *IEEE Internet of Things Journal*, October 2016, vol. 3, num. 5.
- [7] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, T. Watteyne, "FIT IoT-LAB: A Large Scale Open Experimental IoT Testbed," *IEEE World Forum on Internet of Things (IEEE WF-IoT)*, December 2015, Milan, Italy.
- [8] M. S. Farooq, S. Riaz, A. Abid, K. Abid and M. A. Naeem, "A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming," *IEEE Access*, 2019, vol. 7, doi:10.1109/ACCESS.2019.2949703.

- [9] P. Peddi, A. Dasgupta, and V. H. Gaidhane, "Smart Irrigation Systems: Soil Monitoring and Disease Detection for Precision Agriculture," IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), June 2022, pp. 1-7.
- [10] Liu, Zhiyan, et al. "Internet of Things (IoT) and machine learning model of plant disease prediction—blister blight for tea plant." *Ieee Access* 10 (2022): 44934-44944.
- [11] R. Filev-Maia, C. Ballester-Lurbe, A. Agrahari-Baniya, and J. Hornbuckle, "IRRISENS: An IoT Platform Based on Microservices Applied in Commercial-Scale Crops Working in a Multi-Cloud Environment," *Sensors*, publisher MDPI, 2020, vol. 20, num. 24.
- [12] S. Kim, M. Lee, and C. Shin, "IoT-Based Strawberry Disease Prediction System for Smart Farming," *Sensors*, publisher MDPI, 2018, vol. 18, num. 11.
- [13] H. K. Adli, M. A. Remli, K. N. S. Wan Salihin Wong, N. A. Ismail, A. González-Briones, J. M. Corchado, and M. S. Mohamad, "Recent Advancements and Challenges of AIoT Application in Smart Agriculture: A Review," *Sensors*, publisher MDPI, 2023, vol. 23, num. 7.
- [14] J. Portilla, G. Mujica, J. S. Lee, and T. Riesgo, "The Extreme Edge at the Bottom of the Internet of Things: A Review," *IEEE Sensors Journal*, May, 2019, vol. 19, num. 9.
- [15] R. Partha Pratim, D. Dash, and D. De, "Edge computing for Internet of Things: A survey, e-healthcare case study and future direction," *Journal of Network and Computer Applications* 140 (2019): 1-22.
- [16] R. D. Magarey, T. B. Sutton, and C. L., Thayer, C. L. "A simple generic infection model for foliar fungal plant pathogens," *Phytopathology*, 2005, vol. 95, num. 1, pp 92-100.
- [17] E. Baccelli, O. Hahm, H. Petersen, and K. Schleiser, "RIOT and the evolution of IoT operating systems and applications," *ERCIM News*, 2015, vol. 2015, no. 101.
- [18] "RIOT operating system," [Online], Available: <http://www.riot-os.org>.
- [19] "XENON DATASHEET," [Online], Available: <https://docs.particle.io/assets/pdfs/datasheets/xenon-datasheet.pdf>.
- [20] "SDI12 support group specification," [Online], Available: <https://www.sdi-12.org>.
- [21] "LWS Dielectric Leaf Wetness Sensor, Campbell Scientific" [Online], Available: <https://www.campbellsci.com/lws>.
- [22] J. Jamieson, (2004). "Analysis of covariance (ANCOVA) with difference scores," *International Journal of Psychophysiology*, 52(3), 277-283.
- [23] A. Nanda, B. B. Mohapatra, A. P. K. Mahapatra, A. P. K. Mahapatra A. P. K. Mahapatra (2021), "Multiple comparison test by Tukey's honestly significant difference (HSD): Do the confident level control type I error," *International Journal of Statistics and Applied Mathematics*, 6(1), 59-65.