جامعة سيدي محمد بن عبد الله بفاس
UNIVERSITÉ SIDI MOHAMED BEN ABDELLAH DE FES

كلية العلوم ظهر المهراز فاس
FACULTE DES SCIENCES DHAR EL MAHRAZ FES

N° d'ordre : 70 / 2024

Centre des Etudes Doctorales : ***Sciences et Techniques et Sciences Médicales***

Formation Doctorale : ***Sciences et Technologies de l'Information et de Communication***

Discipline : ***Sciences de l'Ingénieur***
Spécialité : ***Génie Électrique***
Laboratoire : ***Laboratoire d'Ingénierie, Modélisation et Analyse des Systèmes***

THÈSE DE DOCTORAT

Fault Detection in Wireless Sensor Networks for Smart Irrigation

Présentée par :

Nassima JIHANI

Soutenue le 14/12/ 2024 devant le jury :

Nom et Prénom	Grade	Etablissement	Qualité
EL BEKKALI Chakib	PES	Faculté des Sciences Dhar El Mahraz - Fès	Président
MOUHIB Omar	PES	Faculté des Sciences -Kénitra	Rapporteur& examinateur
MARHRAOUI HSAINI Abdallah	MCH	Ecole Supérieure de Technologie - Meknès	Rapporteur& examinateur
BENCHEQROUNE Asmae	PES	Faculté des Sciences Dhar El Mahraz - Fès	Rapporteur& examinateur
BOUAZI Aziz	PES	Ecole Supérieure de Technologie - Meknès	Examinateur
BENBRAHIM Mohammed	PES	Faculté des Sciences Dhar El Mahraz - Fès	Examinateur
KABBAJ Mohammed Nabil	PES	Faculté des Sciences Dhar El Mahraz - Fès	Directeur de thèse

Année Universitaire : 2023-2024

Résumé de la thèse

Cette thèse aborde la problématique cruciale de la détection et de l'isolation des défauts (FDI) dans les Réseaux de Capteurs Sans Fil (RCSF) déployés pour les systèmes d'irrigation intelligente. La fiabilité et la précision des capteurs sont essentielles pour le bon fonctionnement de l'irrigation intelligente, car ces systèmes dépendent fortement des données en temps réel pour optimiser l'utilisation de l'eau et assurer la santé des cultures. Les défauts de capteurs, s'ils ne sont pas détectés, peuvent entraîner des inefficacités majeures, notamment le gaspillage d'eau et les dommages aux cultures. Nous proposons et développons deux stratégies innovantes pour la FDI dans les RCSF adaptées aux applications de l'irrigation intelligente, en mettant particulièrement l'accent sur les contraintes de ressources des nœuds de capteurs. La première approche repose sur l'espace de parité et utilise la corrélation spatiale entre les nœuds de capteurs pour détecter et isoler les défauts. La seconde approche emploie un modèle autorégressif combiné avec un filtre de Kalman, exploitant la corrélation temporelle des données des capteurs pour la détection et l'isolation des défauts. Les deux méthodes sont conçues pour être faciles à implémenter et efficaces sur le plan computationnel, garantissant ainsi une consommation énergétique minimale. Pour valider l'efficacité de nos approches, nous avons mené des implémentations réelles dans un environnement contrôlé de l'irrigation intelligente. Les résultats de ces implémentations montrent une amélioration significative de la précision de détection des défauts et de l'efficacité de l'isolation, tout en maintenant une faible complexité computationnelle. Nos solutions renforcent non seulement la robustesse et la fiabilité des systèmes d'irrigation intelligente, mais elles contribuent également à des pratiques agricoles durables en optimisant l'utilisation de l'eau. Cette thèse contribue de manière significative au domaine de l'agriculture intelligente en proposant des stratégies de FDI pour les RCSF, qui sont à la fois pratiques, évolutives et adaptées aux contraintes de ressources des nœuds capteurs. Ces stratégies garantissent la fiabilité des systèmes d'irrigation pilotés par capteurs et ouvrent la voie à de futures recherches et développements dans le domaine de la fiabilité des réseaux de capteurs et des technologies d'irrigation intelligente.

Mots clés : Irrigation Intelligente, Réseau de Capteurs Sans Fil, Détection et Isolation des Défauts de Capteurs, Espace de Parité, Filtre de Kalman, Modèle Autorégressif.

Fault Detection in Wireless Sensor Networks for Smart Irrigation

Nassima JIHANI

Sidi Mohammed Ben Abdellah University

This dissertation is submitted for the degree of
Doctor of Philosophy

February 2025

Declaration

I, Nassima JIHANI, declare that this thesis titled "Fault Detection in Wireless Sensor Networks for Smart Irrigation" and the work presented within it are my own and have been generated by me as the result of my own original research.

I confirm that:

- I have acknowledged all sources used, and have quoted from them as appropriate.
- This thesis is my own original work, and where I have used the work of others, it has been fully acknowledged and referenced.
- I have not previously submitted this work or any version of it for assessment at any other institution.
- This thesis has not been submitted for any other degree or professional qualification.
- The content of this thesis is the result of work carried out since the official start date of my research program.

Nassima JIHANI
February 2025

Acknowledgements

Alhamdulillah, I am greatly indebted to Allah SWT for His mercy and blessing for making this research successful.

First and foremost, I would like to express my deepest gratitude and sincere thanks to my thesis supervisor, Prof. Mohammed Nabil KABBAJ, for his invaluable guidance, insightful advice, and trust throughout my research journey. His continuous support and constructive criticisms have greatly helped me in accomplishing this thesis.

I would like to take this opportunity to express my deepest gratitude and thanks to Prof. Mohammed BENBRAHIM, for his constant guidance, assistance and support as well as all the knowledge they shared during the course of this research.

I would like to express my appreciation and gratitude to Prof. Chakib EL BEKKALI for accepting to chair the jury of my thesis.

I express my deep sense of gratitude and sincere thanks to Prof. Omar MOUHIB, Prof. Abdallah MARHRAOUI HSAINI and Prof. Asmae BENCHEQROUNE for their time and efforts for reviewing this manuscript.

I would like to thank Prof. Aziz BOUAZI for agreeing to examine this thesis and for his valuable comments and helpful suggestions.

To my closer friends, I express my gratitude for their unconditional friendship, support and patience throughout these years.

Finally, I cannot end without thanking all my family who supported me throughout my studies with their unconditional love and constant support.

Abstract

This thesis addresses the critical issue of fault detection and isolation (FDI) in wireless sensor networks (WSNs) deployed for smart irrigation systems. The reliability and accuracy of the sensor nodes are paramount for the efficient operation of smart irrigation, as these systems rely heavily on real-time data to optimize water usage and ensure crop health. Sensor faults, if undetected, can lead to significant inefficiencies, including water wastage and crop damage.

We propose and develop two innovative strategies for FDI in WSNs designed for smart irrigation applications, with a specific focus on the resource constraints of sensor nodes. The first approach is based on parity space and utilizes spatial correlation among sensor nodes to detect and isolate sensor faults. The second approach employs an autoregressive model combined with a Kalman filter, leveraging temporal correlation in the sensor data for fault detection and isolation. Both methods are designed to be easy to implement and computationally efficient, ensuring minimal energy overhead.

To validate the effectiveness of our approaches, we conducted real-world implementations in a controlled smart irrigation environment. The results from these implementations demonstrate a significant improvement in fault detection accuracy and isolation efficiency, while maintaining low computational complexity. Our solutions not only enhance the robustness and reliability of smart irrigation systems but also contribute to sustainable agricultural practices by optimizing water usage.

This thesis offers a substantial contribution to the field of smart agriculture by providing practical, scalable, and resource-efficient FDI strategies for WSNs. These strategies ensure the reliability of sensor-driven irrigation systems and lay the groundwork for future research and development in sensor network reliability and smart irrigation technologies.

Keywords: Smart Irrigation, Wireless Sensor Network, Sensor Fault Detection and Isolation, Parity Space, Kalman Filter, Autoregressive Model.

Résumé

Cette thèse aborde la problématique cruciale de la détection et de l'isolation des défauts (FDI) dans les Réseaux de Capteurs Sans Fil (RCSF) déployés pour les systèmes d'irrigation intelligente. La fiabilité et la précision des capteurs sont essentielles pour le bon fonctionnement de l'irrigation intelligente, car ces systèmes dépendent fortement des données en temps réel pour optimiser l'utilisation de l'eau et assurer la santé des cultures. Les défauts de capteurs, s'ils ne sont pas détectés, peuvent entraîner des inefficacités majeures, notamment le gaspillage d'eau et les dommages aux cultures.

Nous proposons et développons deux stratégies innovantes pour la FDI dans les RCSF adaptées aux applications de l'irrigation intelligente, en mettant particulièrement l'accent sur les contraintes de ressources des nœuds de capteurs. La première approche repose sur l'espace de parité et utilise la corrélation spatiale entre les nœuds de capteurs pour détecter et isoler les défauts. La seconde approche emploie un modèle autorégressif combiné avec un filtre de Kalman, exploitant la corrélation temporelle des données des capteurs pour la détection et l'isolation des défauts. Les deux méthodes sont conçues pour être faciles à implémenter et efficaces sur le plan computationnel, garantissant ainsi une consommation énergétique minimale.

Pour valider l'efficacité de nos approches, nous avons mené des implémentations réelles dans un environnement contrôlé de l'irrigation intelligente. Les résultats de ces implémentations montrent une amélioration significative de la précision de détection des défauts et de l'efficacité de l'isolation, tout en maintenant une faible complexité computationnelle. Nos solutions renforcent non seulement la robustesse et la fiabilité des systèmes d'irrigation intelligente, mais elles contribuent également à des pratiques agricoles durables en optimisant l'utilisation de l'eau.

Cette thèse contribue de manière significative au domaine de l'agriculture intelligente en proposant des stratégies de FDI pour les RCSF, qui sont à la fois pratiques, évolutives et adaptées aux contraintes de ressources des nœuds capteurs. Ces stratégies garantissent la fiabilité des systèmes d'irrigation pilotés par capteurs et ouvrent la voie à de futures recherches et développements dans le domaine de la fiabilité des réseaux de capteurs et des technologies d'irrigation intelligente.

Mots clés: Irrigation Intelligente, Réseau de Capteurs Sans Fil, Détection et Isolation des Défauts de Capteurs, Espace de Parité, Filtre de Kalman, Modèle Autorégressif.

Table of contents

Acknowledgements **iii**

Abstract **iv**

Résumé **v**

List of figures **ix**

List of tables **xi**

General Introduction **1**

Publications **5**

1 Wireless Sensor Network for Smart Irrigation Application **6**

1.1 Water Challenges and Smart Solutions for Efficient Agricultural Irrigation . 7

1.2 Wireless Sensor Network 8

1.3 Wireless Sensor Network Applications 10

1.3.1 Military 11

1.3.2 Health 11

1.3.3 Home 12

1.3.4 Structural Health Monitoring 12

1.3.5 Environmental 13

1.4 Sensor Node 14

1.4.1 Sensing Sub-System 15

1.4.2 Computation Sub-System 16

1.4.3 Communication Sub-System 16

1.4.4 Power Supply Sub-System 17

1.5 Network Topology 17

1.5.1 Star Topology 17

1.5.2 Mesh Topology 18

1.5.3 Cluster-Tree Topology . . . 19
1.6 Wireless Communication Protocols Applied in The Context of Smart Agriculture . . . 20
1.6.1 Zigbee . . . 20
1.6.2 Wi-Fi . . . 22
1.6.3 Sigfox . . . 23
1.6.4 Narrow Band IoT (NB-IoT) . . . 24
1.6.5 Long-Range Wireless Area Network (LoRaWAN) . . . 25
1.7 Challenges in Wireless Sensor Network . . . 27
1.8 Wireless Sensor Network Reference Model . . . 29
1.9 Energy Comsuption Model of the Sensor Node . . . 31
1.10 Conclusion . . . 34

2 Fault Detection in Wireless Sensor Network: State of Art 36
2.1 Generality on Fault Diagnosis in Wireless Sensor Networks . . . 37
2.2 Motivations for Fault Diagnosis in Wireless Sensor Networks . . . 37
2.3 Definitions and Terminologies . . . 39
2.4 Fault Classification and Modeling . . . 41
2.5 Methods of Fault Detection and Diagnosis in WSN . . . 46
2.5.1 Data Correlations . . . 52
2.5.2 Model Structure . . . 53
2.5.3 Operation Mode . . . 55
2.6 Fault Detection and Isolation . . . 55
2.7 Fault Diagnosis Challenges in Wireless Sensor Networks . . . 56
2.8 Limitations of Existing Approaches for Fault Detection in Wireless Sensor Networks . . . 58
2.9 Conclusion . . . 59

3 Spatial Correlation-Based Fault Detection and Isolation for Smart Irrigation Systems in Wireless Sensor Networks 61
3.1 Soil Monitoring Background . . . 62
3.2 Methodology for Sensor Deployment . . . 64
3.3 Parity Space . . . 65
3.3.1 Fault Detection . . . 68
3.3.2 Fault Isolation . . . 68
3.4 Proposed FDI Algorithm . . . 68
3.5 Laboratory Experiments . . . 70

3.6 Results and Analysis 72
3.7 Conclusion 77

4 Sensor Fault Detection in Wireless Sensor Network for Smart Irrigation Application using Kalman Filter 79
4.1 Sensor Readings Prediction 80
4.2 Sensor Fault Detection using Kalman Filter 82
4.2.1 Autoregressive Model 83
4.2.2 Kalman Filter 83
4.2.3 Residual Generation and Evaluation 86
4.2.4 Setting a Threshold 87
4.3 Experiments 88
4.3.1 System Overview 88
4.3.2 Real-World Application of Fault Detection System 90
4.3.3 Hardware 92
4.3.4 Communication System 93
4.3.5 Network Infrastructure 94
4.3.6 Monitoring Dashboard 96
4.4 Results and Analysis 97
4.5 Conclusion 102

Conclusion and Future Work 103

References 106

Appendix A Calculation of Autoregressive (AR) Model Coefficients 117
A.1 Overview of the AR Model 117
A.2 Steps to Calculate AR Coefficients 118

Appendix B Technical Datasheet of Decagon's ECH2O EC-5 Soil Moisture Sensor 120

List of figures

1.1 Smart irrigation system 8
1.2 Benefits of smart irrigation 8
1.3 Structure of wireless sensor network 9
1.4 Forest-fire monitoring application 14
1.5 Sensor architecture 15
1.6 Star topology 18
1.7 Mesh topology 19
1.8 Cluster topology 20
1.9 Wireless sensor network architecture 30
1.10 Sensor working scenario 32

2.1 The principles behind hardware redundancy and analytical redundancy . . . 38
2.2 Taxonomy of faults in wireless sensor network 42
2.3 Fault categorization based on their duration 43
2.4 Offset fault in WSN [87] 45
2.5 Gain fault in WSN [87] 45
2.6 Stuckat fault in WSN [87] 46
2.7 Data loss fault in WSN [87] 47
2.8 Fault detection techniques for Wireless Sensor Networks 48
2.9 The clustering-based fault detection method 50
2.10 SVM classifier 51
2.11 Sensor fault diagnosis in centralized architecture 54
2.12 Sensor fault detection and isolation process 56

3.1 Variance in soil type at different depths [21] 63
3.2 Sensors nodes distribution 65
3.3 Direct redundancy 66
3.4 Fault signature directions in the parity space 67

3.5 Experimental prototype of the wireless sensor network deployed in soil for measuring soil temperature and soil moisture 70
3.6 Illustrative schematic of the experimental prototype for measuring soil temperature and soil moisture. 72
3.7 Soil temperature recorded by the wireless sensor network in its healthy mode 72
3.8 Soil moisture measurement recorded by the wireless sensor network in its healthy mode 73
3.9 Residual evolution of soil temperature 73
3.10 Residual evolution of soil moisture 74
3.11 Fault isolation process of soil temperature 74
3.12 Fault isolation process of soil moisture 75
3.13 Residual evolution of soil temperature 75
3.14 Residual evolution of soil moisture 76
3.15 Fault isolation process of soil temperature 76
3.16 Fault isolation process of soil moisture 77

4.1 Flowchart of the proposed scheme for Sensor readings prediction 81
4.2 Structural concept of the sensor fault detection 82
4.3 Framework of the Kalman filter algorithm. 86
4.4 Schematic of IrrigSys system 89
4.5 The plot for typical installation at the faculty [43] 91
4.6 Illustrative scheme of our laboratory experiment 91
4.7 Evaluation of data acquisition, transmission, and processing [43] 92
4.8 Decagon's ECH2O EC-5 soil moisture [43] 93
4.9 Installation of the LoRa node at the FSDM [43] 94
4.10 LoRa gateway [43] 95
4.11 The GUI of the real-time monitoring using Node-RED 97
4.12 Soil temperature measurements and predicted data in non-faulty mode . . . 99
4.13 Soil moisture measurements and predicted data in non-faulty mode 99
4.14 Soil temperature measurements with its predicted data in faulty mode . . . 100
4.15 Residuals of soil temperature sensor before and after fault introduction . . . 100
4.16 Soil moisture measurements with its predicted data in faulty mode 101
4.17 Residuals of soil moisture sensor before and after fault introduction 101

List of tables

1.1 Comparison of Star, Mesh, and Cluster-Tree Topologies 21
1.2 Wireless communication protocols features 27

4.1 Parameters of the Kalman Filter-Based AR(1) Model for Sensor Data . . . 98

Abbreviations

FDI Fault Detection and Isolation

WSN Wireless Sensor Network

IoT Internet of Things

SHM Structural Health Monitoring

RF Radio Frequency

PAN Personal Area Network

LoRaWAN Low Range Wide Area Network

USB Universal Serial Bus

WPAN Wireless Personal Area Network

GPS Global Positioning System

WLANs Wireless Local Area Networks

OSI Open System Interconnection

SN Sink Node

BS Base Station

PDF Probability Distribution Function

KNN K-Nearest Neighbors

SVM Support Vector Machine

NN Neural Networks

PCA Principal Component Analysis

AR Autoregressive

General Introduction

Global water scarcity is increasing, creating significant challenges for sustainable agricultural production [22, 32, 101]. By 2050, more than 80% of the world's countries are expected to face heightened agricultural water scarcity [76]. This issue, combined with a growing population, the impacts of climate change, and competition from other sectors, puts food security at stake. Irrigated agriculture is a major consumer of water, accounting for about 70% of freshwater withdrawals [24]. It contributes 40% of the world's food and fiber production, making it crucial for feeding the projected 9.7 billion people by 2050 [47]. Water withdrawals for irrigation are projected to reach 29,000 km^3 by 2050, with the highest demand in developing countries, leading to 20 million hectares under irrigation [120].

Given these circumstances, improving water use efficiency in irrigated agriculture is essential. Effective water management involves using available resources to meet real-time crop irrigation needs, which requires careful monitoring and control of irrigation practices. Smart irrigation systems play a vital role in this context. By continuously monitoring the crop environment and adjusting water application, these systems ensure that soil moisture remains within specific thresholds, such as field capacity and the permanent wilting point. The importance of smart irrigation lies in its ability to optimize water use, thus contributing to sustainable agricultural practices and enhancing food security in a world facing increasing water scarcity.

An essential technology supporting smart irrigation systems is wireless sensor networks (WSNs). WSN technology involves deploying a network of spatially distributed sensors nodes to monitor and collect data on various environmental parameters, such as soil moisture, and soil temperature. This data is then transmitted wirelessly to a central system for analysis and decision-making.

The use of a WSN in smart irrigation offers several advantages. Firstly, it enables real-time monitoring of soil and crop conditions, ensuring that irrigation is applied with optimal

timing and precision. This technology (WSN) contributes significantly to reducing water waste while enhancing the overall efficiency of water use. Secondly, WSN can cover large agricultural areas with minimal manual intervention, making the system scalable and suitable for both small and large farms. Additionally, the data collected by WSN can be integrated with weather forecasts and crop models to optimize irrigation schedules further.

However, deploying WSN in agricultural environments presents significant challenges. The harsh conditions where sensor nodes are deployed, such as extreme temperatures, high humidity, soil salinity, and physical disturbances, can lead to sensor failures. These sensors are prone to malfunctions and can provide erroneous measurements, which can negatively impact the operation of smart irrigation systems. Inaccurate data can lead to improper irrigation scheduling, resulting in either water wastage or insufficient watering, both of which can harm crop yields.

Addressing the issue of sensor reliability requires the implementation of sensor fault detection and isolation (FDI) techniques. FDI involves continuously monitoring the performance of sensors to detect and isolate faults. By identifying faulty sensors promptly, these systems can either exclude the erroneous data from the analysis or trigger maintenance actions to replace or repair the defective sensors. FDI methods can use various approaches, including statistical analysis, machine learning algorithms, and redundancy strategies, to ensure data accuracy and system reliability.

Incorporating FDI techniques into WSNs for smart irrigation enhances the robustness of the system, ensuring that it can operate effectively even in challenging environments. This integration not only improves water use efficiency but also increases the resilience and dependability of smart irrigation systems, making them more suitable for widespread adoption in agriculture. Consequently, the combination of WSN technology and advanced fault detection methods is essential for advancing sustainable agricultural practices and securing food production in the face of global water scarcity.

Motivation and Objectives of Research:

Implementing FDI in WSNs is crucial for ensuring accurate and reliable sensor data in smart irrigation systems. This integration is driven by the necessity to maintain precise irrigation practices and optimize resource usage. Accurate sensor data is paramount for informed irrigation decisions, preventing over or under-watering which can harm crops.

FDI ensures timely detection and isolation of faulty sensors, preventing erroneous decisions and conserving water. Optimizing water resources is essential for sustainable agriculture. Reliable sensor data, coupled with robust FDI, minimizes wastage and supports efficient water application. Economically, smart irrigation systems with FDI offer significant benefits to farmers, lowering water costs and increasing crop yields. Moreover, FDI contributes to environmental sustainability by conserving water and reducing wastage, supporting ecosystem preservation. The consistent and reliable performance of smart irrigation systems builds user trust, fostering widespread adoption and advancement in smart farming practices. In summary, integrating sensor fault detection in WSNs for smart irrigation enhances accuracy, optimizes resource usage, and fosters economic and environmental sustainability, thereby promoting trust and wider adoption of smart irrigation technologies.

Contributions:

This thesis makes several significant contributions to the field of sensor fault detection in WSNs for smart irrigation:

- Innovative algorithms are presented for detecting and isolating sensor faults in WSN for smart irrigation. These algorithms utilize analytical redundancy, offering a more cost-effective solution compared to hardware redundancy.

- The proposed approaches enable real-time detection of sensor faults, ensuring the reliability and accuracy of data for smart irrigation systems. By enhancing the reliability of WSN, this research contributes to more effective water management practices in agriculture.

- The algorithms are designed to be resource-efficient, accommodating the limited computational and energy resources of sensor nodes in WSN, making them suitable for practical deployment. Our technique addresses the challenge of lacking ground truth information, which is often a significant hurdle in practical WSN applications.

- The algorithms developed are simple to implement in real-world applications without requiring extensive technical expertise. By maintaining minimal computational complexity, our approaches preserve the overall efficiency of the smart irrigation application, making them suitable for real-time operations.

- The research addresses critical needs in modern agriculture for reliable, real-time monitoring of soil parameters, thereby supporting sustainable agricultural practices.

In summary, this thesis offers significant advancements in the field of WSN for smart irrigation by developing cost-effective, reliable, and efficient sensor fault detection algorithms that are practical for real-world applications. These contributions collectively improve the reliability and effectiveness of smart irrigation systems, promoting better water management and conservation in agriculture.

Outline of Thesis:

This thesis is structured into four chapters, outlined as follows:

Chapter 1: This chapter presents the role of WSNs in smart irrigation and various other applications. It covers three network topologies and discusses wireless communication protocols used in smart agriculture. Additionally, the chapter addresses challenges faced by WSN technology and provides an energy consumption model for each sensor node.

Chapter 2: This chapter presents various fault detection and isolation approaches for WSNs. It covers different types of faults, fault detection and isolation methods, their objectives, and classifications. Additionally, it highlights the challenges of fault diagnosis in WSNs and the limitations of current fault detection methods.

Chapter 3: This chapter describes the application of the parity space approach for detecting and isolating sensor faults in WSN. The results demonstrate the effectiveness of the algorithm in detecting and isolating sensor faults in WSN.

Chapter 4: This chapter addresses the problem of sensor fault detection in WSNs for smart irrigation. An algorithm based on the Kalman filter is designed to detect sensor faults. The practical deployment and effectiveness of the algorithm in detecting and isolating sensor faults in WSN are detailed.

Publications

The related works of this thesis have given rise to the following publications:

Journals

- Jihani, N., Kabbaj, M. N., Benbrahim, M. (2023). "Kalman filter based sensor fault detection in wireless sensor network for smart irrigation". Results in Engineering (Elsevier), 20, 101395.
  https://doi.org/10.1016/j.rineng.2023.101395

- Jihani, N., Kabbaj, M. N., Benbrahim, M. (2023). "Sensor fault detection and isolation for smart irrigation wireless sensor network based on parity space". International Journal of Electrical and Computer Engineering (IJECE), 13(2), 1463.
  http://doi.org/10.11591/ijece.v13i2.pp1463-1471

Conferences

- Jihani, N., Kabbaj, M. N., Benbrahim, M. (2021, January). Wireless sensor network fault detection and isolation for smart irrigation. In International Conference on Digital Technologies and Applications (ICDTA) (pp. 759-768). Cham: Springer International Publishing.
  https://doi.org/10.1007/978-3-030-73882-2_69

Chapter 1

Wireless Sensor Network for Smart Irrigation Application

In this chapter, the integration of wireless sensor networks (WSNs) in modern agricultural practices is explored, specifically focusing on their application for smart irrigation techniques. Our investigation begins by addressing the critical importance of water in agriculture and the imperative for its judicious use to meet growing global demand in the face of growing water scarcity challenges.

Subsequently, a comprehensive overview of WSNs is provided, clarifying their fundamental principles and operational mechanisms. Building upon this foundational understanding, we proceed to explore the diverse applications of WSNs.
Moreover, we explore the components of sensor node, network topologies, and the communication protocols employed in smart agriculture, delineating their characteristics and suitability for diverse agricultural environments.

While investigating the benefits offered by WSNs, the inherent challenges encountered in WSN deployments are also addressed, including resource limitations, environmental dynamics, scalability concerns, and security considerations.
Moreover, a comprehensive overview of the WSN reference model is presented, encompassing the application layer, transport layer, network layer, data link layer, and physical layer. Furthermore, we present a simplified model for understanding the energy consumption dynamics of sensor nodes, providing the parties involved with the knowledge needed to optimize power efficiency and maximize the longevity of WSN deployments.

1.1 Water Challenges and Smart Solutions for Efficient Agricultural Irrigation

Smart irrigation plays a pivotal role in modern agriculture, providing a controlled and reliable water supply that is essential for crop growth and yield. Agriculture's heavy dependence on water availability highlights the importance of smart irrigation to ensure consistent agricultural productivity. Lack of adequate water can lead to disastrous consequences such as drought stress, reduced yields, or even complete crop failure. With the world's population increasing and arable land becoming scarcer, the importance of efficient irrigation practices becomes increasingly crucial to meeting food production demands and ensuring global food security.

The pressing issue of water scarcity resonates globally, particularly in agricultural regions. Traditional irrigation methods often contribute to water wastage due to inefficient distribution systems and the absence of reliable soil moisture measurement. These inefficient practices not only harm water resources but also lead to environmental degradation, soil salinization, and energy inefficiency. The urgency of the situation is exacerbated by the effects of climate change, which are further intensifying water scarcity. In response, innovative and sustainable irrigation solutions are imperative. Smart solutions that use technology and data-driven approaches are the key to optimizing water usage, increasing crop yields, and mitigating the adverse effects of water scarcity. These solutions will transform irrigation practices by enabling precise control and informed decision-making.

Smart irrigation represents a paradigm shift in irrigation management through technology and data-driven methodologies. This innovative technique automates irrigation systems while conserving water resources. By integrating sensors, devices and software, smart irrigation optimizes water consumption, reduces waste, and increases overall irrigation efficiency. This approach dynamically adjusts irrigation schedules based on real-time soil and weather conditions, allowing farmers to meet their water needs in an innovative way that conserves water resources. Figure 1.1 illustrates the overall structure of the smart irrigation system. The core objective of smart irrigation is to deliver precise amounts of water at the optimal times, accounting for variables such as weather dynamics, soil moisture levels, plant types, and other pertinent parameters. Through such an approach, smart irrigation ensures sustainable agricultural practices while maximizing yield potential [1]. Figure 1.2 summarize the benefits of using smart irrigation.

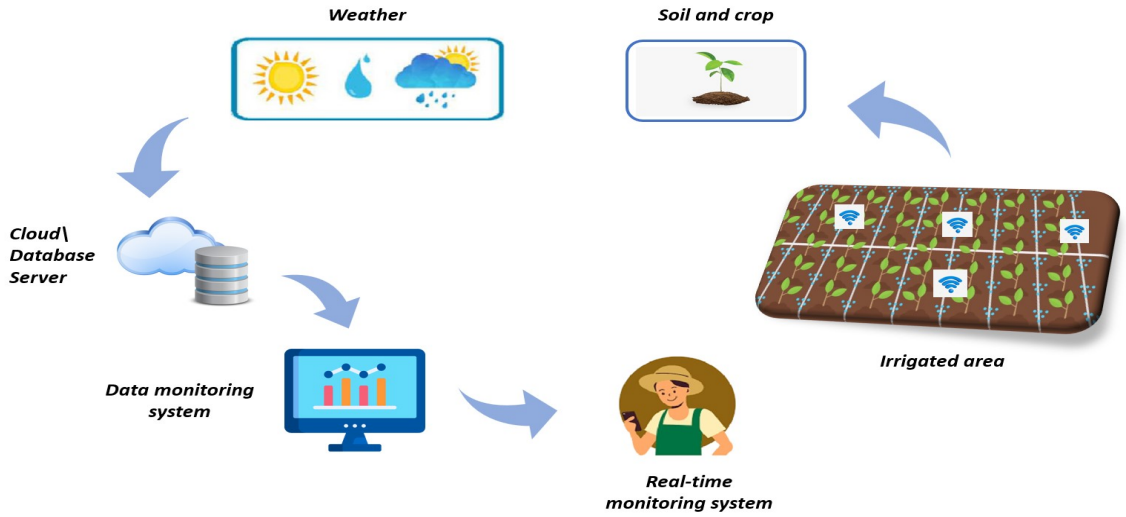


Fig. 1.1 Smart irrigation system



Fig. 1.2 Benefits of smart irrigation

1.2 Wireless Sensor Network

The remarkable advancement in the development of low-cost, low-power, multifunctional sensor nodes (also known as "motes"), combined with improvements in wireless communication, have paved the way for the creation of a new type of smart device. Leveraging these

capailities in a self-organizing network has led to emergence of Wireless Sensor Networks (WSNs).

A WSN is an ad hoc network that can be defined as a group of tiny devices capable of sensing a variety of environmental information in the area where they are placed, such as temperature, soil moisture, wind speed, and light intensity. In a WSN, the sensor nodes are spread over the field (a few tens to thousands) with restricted power, which are equipped with one or more sensors, a processor, memory, a power source, and a radio. Every node utilizes its capabilities to collect and transmit data, with the ultimate goal of constructing a comprehensive overview of the controlled field. The gathered data is routed directly or via other sensors employing a multi-step architecture to a central collection point, commonly referred to as a base station, for subsequent processing and analysis. The base station also has the capability to function as a gateway node when there is a requirement to establish communication and connections with an external network, primarily for the purpose of data analysis and decision-making [3, 119, 46] as shown Figure 1.3.

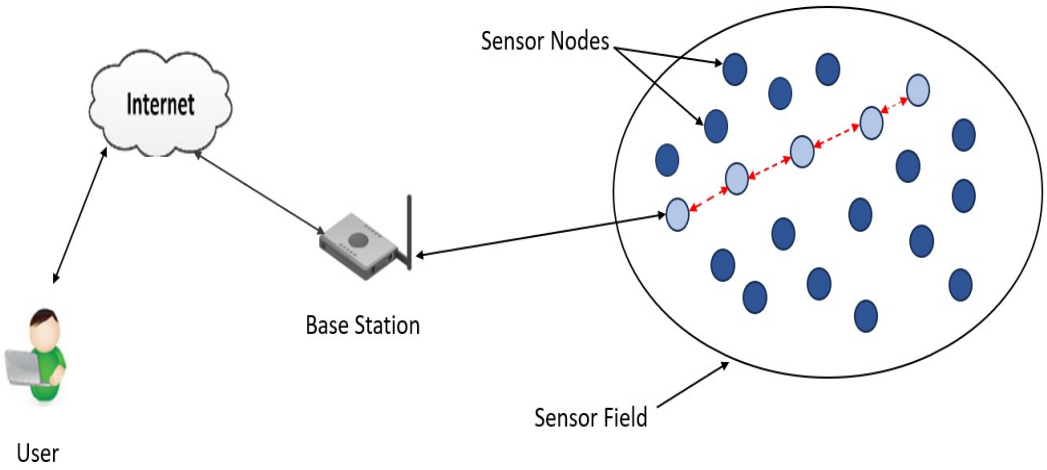


Fig. 1.3 Structure of wireless sensor network

The nodes can be deployed in locations that are not required to be pre-engineered or pre-determined, enabling their deployment in hard-to-reach areas. They can be fixed or mobile. This means that the protocols and algorithms need to be adapted to suit the WSN and should be capable of organizing themselves without human intervention. [4].

As mentioned in [13], WSN can be categorized into three distinct groups:

1. **Time-driven:** Nodes transmit data collected by sensors to the base station on a scheduled, periodic basis (e.g. soil moisture or temperature monitoring);

2. **Event-driven:** Nodes sent alerts upon the occurrence of specific events (e.g. fire detection);

3. **Query-driven:** Nodes exclusively transmit data to the Sink Node (SN) in response to a request query, as and when needed (e.g. storage room).

1.3 Wireless Sensor Network Applications

Numerous fields of application and usage scenarios are based on WSNs technology [7]. Many of these applications are already well-established and widely utilized. The main applications of WSNs can be classified as follows:

- Military applications;

- Health applications;

- Home applications;

- Industrial automation/applications;

- Environmental applications,

- Monitoring of civil structures.

All deployments of WSNs share a common feature, which is the utilization of spatially distributed sensor nodes. These nodes are typically implemented using low-power embedded systems and are equipped with sensors to measure specific physical quantities. The interconnection between these sensor nodes is determined by the particular requirements of the application. This interconnection can occur through wireless communication protocol such as near-field communication (e.g., Zigbee, Bluetooth Low Energy (BLE)) [62], or by utilizing low-power wide-area networks (LPWANs) like Long-Range Wide-Area Network (LoRaWAN).

In general, the majority of WSN deployments can be classified into one of two main application categories, determined by whether they provide continuous sensing (like environmental or process monitoring) or perform event detection (such as detecting forest fires or surveillance). While there are shared attributes between the two categories, such as network structure, notable differences exist in their respective requirements. This includes variations in anticipated operational lifespans, communication patterns, and the volume of data to be transmitted. Additionally, the specific application domains differ in terms of coverage area, intervals for data updates, and the environmental contexts within which the sensor nodes are situated.

This thesis centers its attention on monitoring applications that deliver continuous sensor data.

1.3.1 Military

Due to their adaptability in deployment and low-cost, sensors can be positioned in various locations, and their utilization becomes an advantage in battlefield scenarios. In contrast to conventional sensors, the destruction of specific nodes does not affect military operations. Numerous potential applications exist for sensor networks in military contexts, encompassing tasks such as monitoring friendly forces, conducting battlefield surveillance, identifying targets, assessing damage inflicted during combat, and detecting nuclear, biological, and chemical threats. Additionally, these networks are adopted in reconnaissance missions [4].

1.3.2 Health

WSNs have found adoption in various applications within the medical field, including continuous monitoring of patient's physiological data, location tracking, fall detection mechanisms, and even drug administration within hospital settings. The integration of sensors within the human body enhances monitoring by capturing vital signs and providing immediate feedback, ultimately improving patients' quality of life. However, the implementation of such embedded sensors is conditioned by some factors such as safety, reliability, low maintenance, and energy consumption[26].

In an innovative approach, Lee and Chung [71] introduced a smart shirt capable of capturing electrocardiogram (ECG) and acceleration signals (for detecting physical activities), enabling real-time health monitoring. This shirt incorporates wearable sensor nodes re-

sponsible for data acquisition, which is then transmitted to a central base station. Another noteworthy contribution outlined in [70] presents a system designed to remotely monitor the movements of elderly individuals using a set of motion sensors. This system empowers caregivers and healthcare professionals to identify unusual activity patterns in elders and respond promptly.

1.3.3 Home

Nowadays, people want to live in comfortable and convenient living spaces where WSN can play a crucial role in enhancing the quality of home environments. In home automation, nearly every household appliance, from microwaves and refrigerators to washing machines, can establish interactions among themselves and with external networks like the Internet. This capability extends to remotely activating or deactivating lights, as well as opening and closing doors from a distance. Consequently, users gain the ability to access and manage their domestic devices remotely.

Smart homes have the potential to revolutionize our lifestyles by creating an environment where the smart home acquires information and utilizes this data to enhance the well-being of its residents. In [53], a system is presented for remote-controlled home automation, enabling the remote management of devices, triggering alarms in the event of home intrusion or fires, and collecting data related to weather condition.

1.3.4 Structural Health Monitoring

WSNs are also increasingly used in the field of civil structure monitoring, particularly within systems for structural health monitoring (SHM). These systems are aimed at assessing the condition of corresponding civil structures. The central benefit of these systems lies in their capacity to transition from a reactive 'fix-and-fail' approach to a proactive 'predict-and-prevent' approach [69].

In the past, issues with civil structures, such as buildings or bridges, have typically been identified only after they become evident on the surface. By this stage, the extent of the damage has often grown to a point where noticeable consequences are unavoidable, like underground flooding caused by broken water pipes. Even more severe are situations in which civil constructions have collapsed due to structural flaws that remained undetected for a long period of time, as the case of the collapse of bridges [106].

Nowadays, SHM systems facilitate continuous surveillance of various assets, assessing their structural health state. These systems offer a way to identify structural changes even before they significantly affect the performance of the corresponding structures [64]. WSN-based SHMs are adopted widely in civil structures, including buildings [42], water pipelines [77], railway tracks [28], highways [73], and tunnels [134, 54]. The main examples of SHM systems based on WSN are implemented in bridges [98, 63, 57]. This is because bridges are more prone to developing issues due to vibrations, whether from strong winds, vehicles passing over them, or their specific architectural design.

1.3.5 Environmental

Environmental monitoring is one of the earliest applications that adopted WSN technology [88, 95]. This means the utilization of sensor networks to observe various environmental phenomena and physical conditions. Integrating a WSN into a natural environment facilitates the acquisition of long-term data at a previously unachievable scale, and resolution becomes feasible. This integration empowers applications to collect precise, localized measurements that would otherwise be challenging to gather. Consequently, numerous environmental applications have been suggested for implementation with WSN [129]. These applications include habitat monitoring, animal tracking, forest-fire detection, precision farming, and disaster relief.

Imagine a situation where a fire ignites within a forest. If a WSN is placed in the forest, it could promptly alert authorities even before the fire gains uncontrollable momentum as depicted in Figure 1.4. Precise details regarding the fire's location [92] can be swiftly determined. This prompt identification provides firefighters with an unparalleled edge, enabling them to reach the site before the fire escalates beyond control. Precision farming [114] is another field that can benefit from the advantages of WSN technology. In precision farming, spatial data analysis is essential to assess how crops react to diverse factors, including soil type [80]. The incorporation of sensor nodes strategically positioned throughout a field could furnish farmers with comprehensive soil analyses, aiding in the optimization of crop production and water management or potentially sending alerts when predetermined soil and crop conditions are met. The inherent nature of WSNs, tailored for autonomous operation, eliminates the need for continuous active physical monitoring.

Disaster relief case, exemplified by systems like the alert flood-detection mechanism [16], rely on distant field sensors to transmit real-time information to a central computer system.

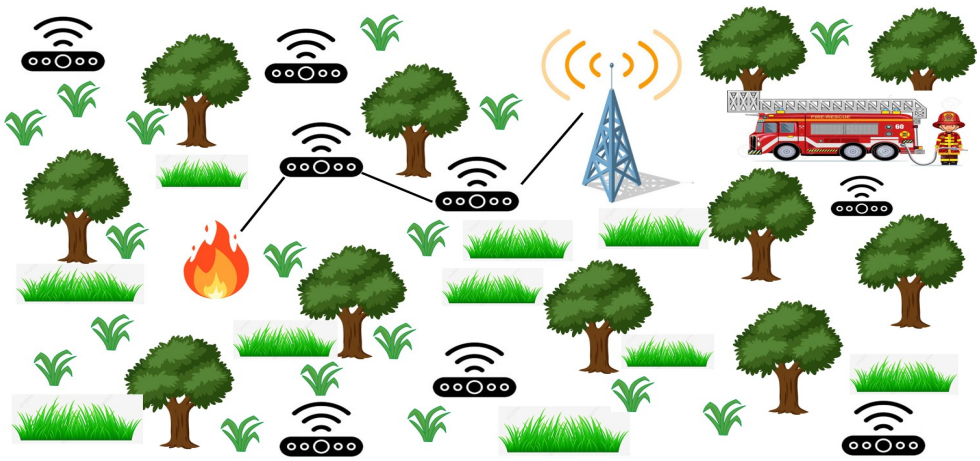


Fig. 1.4 Forest-fire monitoring application

In an alert setup, various sensor types are usually employed, including those for measuring rainfall, water levels, and weather conditions. Data from these distinct sensor groups are collected and subsequently transmitted to a central base station.

The most interesting application for our case study relies on precision farming.

1.4 Sensor Node

Sensor nodes sometimes also called motes are fundamental components in various sensing and monitoring systems, especially in the context of the Internet of Things (IoT) and WSNs. A sensor node typically consists of four key subsystems: sensing, computation, communication, and power supply. These wireless sensor node platforms face limitations in terms of power supply, computational capabilities, and storage [81]. The objectives in designing and developing wireless sensor nodes include achieving ultra-low power consumption, minimizing node cost, reducing physical size, and ensuring software and hardware reconfigurability. Consequently, the development of increasingly compact and cost-effective wireless sensor node platforms with ultra-low power consumption has become a hot research topic for the research community [23].

Various types of wireless sensor nodes and ad hoc network platforms exist due to the diversity in wireless networking requirements and applications. Nonetheless, they generally share a common foundational architecture. These platforms vary in terms of processing power, memory capacity, communication abilities,power supply and consumption, sensor

support, applications, and other factors. A typical wireless sensor node comprises sensing (sensor interfaces and sensors), computational elements (a microcontroller), communication modules, and power supply sub systems, as shown in Figure 1.5.

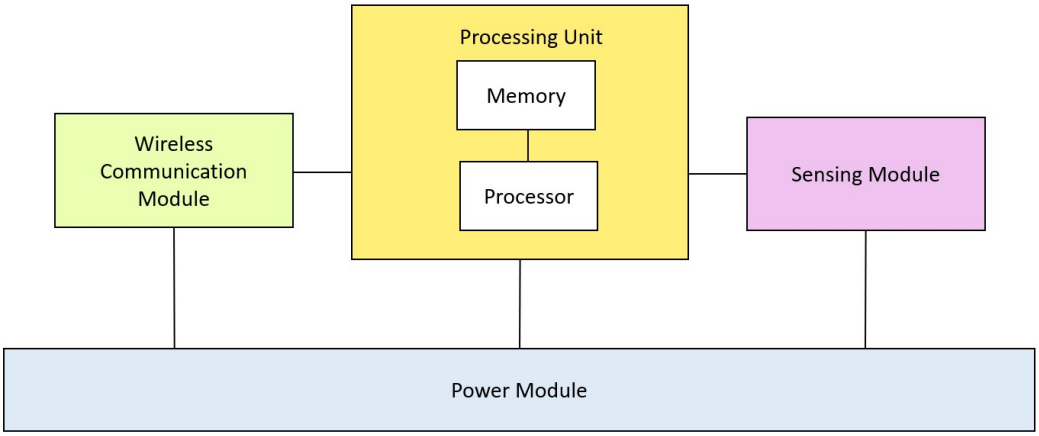


Fig. 1.5 Sensor architecture

1.4.1 Sensing Sub-System

Sensors convert physical phenomena into electrical signals [6]. Therefore, the sensing subsystem establishes a connection between the node and its surrounding environment. A variety of sensors designed for various physical quantities are adapted to meet the demands of different applications. These sensors differ in the quality of the provided measurements (i.e., resolution, accuracy, conversion time). Additionally, these sensors differ in how they provide the measurements to the subsequent processing unit. As a result, we can categorize sensors into two fundamental types: those that come as integrated solutions and those that produce an analog signal directly proportional to the measured physical quantity.

Integrated Sensors: These sensors are available as integrated solutions. Integrated sensors typically combine multiple components into a single unit. They are designed to provide a straightforward and ready-to-use solution for measuring specific physical quantities. Integrated sensors often come with built-in signal conditioning, processing, and sometimes digital communication capabilities. They are known for their convenience and ease of integration into various applications. These sensors often produce digital output, making them suitable for interfacing with microcontrollers and digital systems.

Analog Sensors (or Sensory Circuits): Analog sensors, on the other hand, output an analog signal that is proportional to the measured physical quantity. This means that the sensor's output voltage or current varies continuously with changes in the physical parameter it is measuring. Analog sensors typically require analog-to-digital converter (ADC) to convert their analog output into a digital format for further processing by microcontrollers or other digital devices. They offer flexibility in terms of signal processing and can be customized for specific applications.

1.4.2 Computation Sub-System

The microcontroller's responsibilities encompass processing signals from both the sensing sub-system and transceiver while coordinating various activities of other sensor mote components [85]. The primary objective of the central processing unit (CPU) core revolves around ensuring precise program execution. Consequently, the CPU should possess the capacity to execute tasks such as memory access, computation, external device coordination, and handling interruptions. In sensor motes, processors exhibit flexibility by operating in distinct modes like active, idle, and sleep modes to minimize unnecessary power consumption.

Regarding storage, the mote typically integrates flash memory and a digital (SD) card interface to furnish non-volatile memory for offline data acquisition. However, it's worth noting that the process of writing data to the SD card demands a substantial amount of power [31]. Furthermore, the electrically erasable programmable read-only memory (EEPROM) can be employed to store information for identifying registered sensor motes within the network [37]. Certain motes may incorporate an embedded micro-secure digital (SD) card interface to provide non-volatile memory for offline data capture. Nevertheless, it's important to highlight that writing data to the SD card can be power-intensive.

1.4.3 Communication Sub-System

The communication subsystem consists of a radio module or transceiver designed to enable the reception and transmission of signals to and from the wireless sensor motes, as well as to a central base station or sink [85]. Radio frequency (RF) communication proves ideal for WSNs due to its non-dependence on line-of-sight (LOS). Recent developments have made it possible to employ low-power radio modules that offer adjustable data rates and communication ranges, suitable to suit various application scenarios.

1.4.4 Power Supply Sub-System

The power supply sub-system typically comprises a battery and/or a supercapacitor used to store and provide power to the sensor node. In various application scenarios within WSNs, the expected lifespan of these sensor nodes can extend to several years. Replenishing or replacing batteries under such circumstances can prove costly, inconvenient, or even infeasible, particularly in extensive, remote, or hazardous environments. Consequently, designers have introduced ultra-low power and energy-efficient sensor nodes, along with the accompanying communication protocols and schedules for mote sleep/awake modes. Additionally, they may incorporate energy harvesting methods, such as capturing energy from sources like RF signals, solar power, vibrations, and more, to recharge the batteries [60].

1.5 Network Topology

WSNs can utilize a variety of different topologies. The choice of topology is determined by a set of factors, such as the frequency of data transmission, transmission distance, battery requirements, and so on. This section will delve into the three most commonly employed topologies in WSNs: Star, Mesh, and Cluster Tree. We will explore these topologies while assessing their reliability, energy efficiency, scalability, and data latency performance.

1.5.1 Star Topology

This topology is characterized by a single-hop communication system between the end nodes and the SN, as depicted in Figure 1.6. In this arrangement, all end nodes exclusively communicate with the SN and cannot directly communicate with one another. Furthermore, no alternative paths exist between the end nodes and the SN, rendering communication vulnerable to interruption if the path becomes obstructed. The SN's tasks involve gathering data from the end nodes and sending commands to them when necessary, and forwarding the gathered data to a higher-level system, such as the internet.

The star topology effectively reduces the chance of total network failure. This is because the failure of a single end node does not impact all other end nodes; it only affects the failed node. The energy efficiency of star networks hinges on the network's size and the distances between nodes. In scenarios characterized by small networks and short distances between end nodes, this topology is superior energy efficiency. End nodes can enter sleep mode when not actively transmitting, thereby conserving energy, and can awaken to sense and transmit data to the SN.

However, the addition of more nodes to the network intensifies the workload on the SN and raises the potential for collisions, resulting in diminished network performance. The scalability of the star topology becomes challenging when accommodating more than a few nodes. On the plus side, these networks exhibit minimal latency since there are no delays incurred at routers along the communication path [112].

This topology is particularly suitable for Personal Area Network (PAN) applications, as it is ideally suited for scenarios where the number of devices is typically limited, and communication ranges are short.

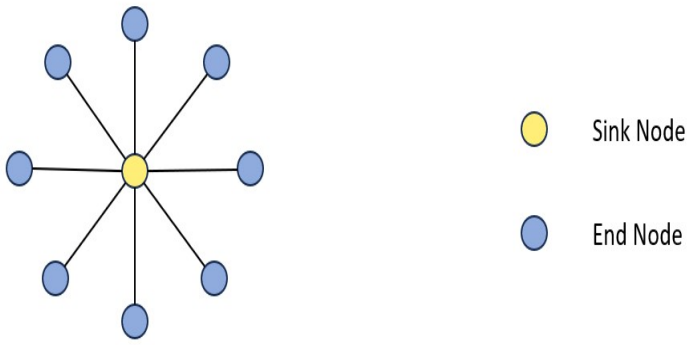


Fig. 1.6 Star topology

1.5.2 Mesh Topology

This topology involves a multi-hop communication system where a group of nodes, referred to as routers (as shown in Figure 1.7) establish connections with the SN. These nodes are designated as routers due to their routing capabilities. In this network, all nodes engage in communication with their neighboring nodes, relaying data from one node to the next until it reaches the SN. This propagation of data through the mesh network allows for network expansion. One notable advantage of the mesh topology is its robust fault tolerance, as the failure of a single node does not disrupt the network since multiple paths exist for data to reach the SN. When a node failure occurs, the network autonomously reconfigures itself. Additionally, each node not only senses data but also acts as a data relay, forwarding information to the SN.

This topology offers high reliability because each node has multiple pathways to reach the SN, which ensures redundancy. The multi-hop system significantly extends the network's coverage compared to a star topology, although, it has a higher power consumption rate. Nodes close to the SN tend to handle more traffic and deplete their energy reserves more quickly. Nevertheless, this topology offers excellent scalability since it is self-configuring and capable of dynamic reconfiguration. However, it may suffer from increased data latency, particularly as the number of nodes and the distances between them grow. Furthermore, nodes near the SN may experience higher data traffic loads, potentially leading to congestion issues [112].

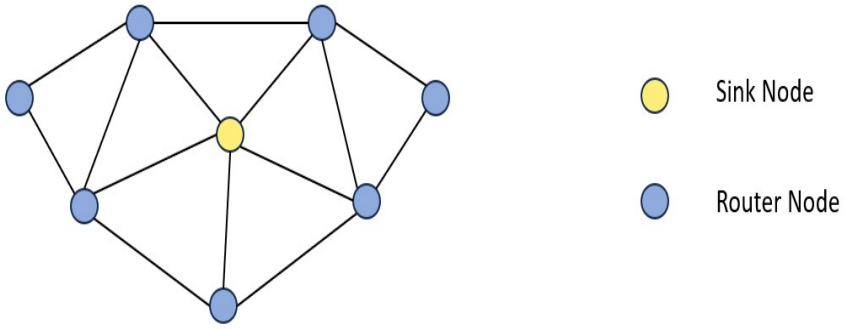


Fig. 1.7 Mesh topology

1.5.3 Cluster-Tree Topology

This topology leverages the advantages of both the Star and Mesh topologies, combining the efficiency of low power consumption and simplicity from the Star topology with the extended range capability of the Mesh topology. The Cluster-tree topology is rooted in the tree structure as shown in Figure 1.8, where nodes are logically grouped into clusters, each resembling a star-like subnetwork. In contrast to the Mesh topology, there is a single defined routing path between any pair of nodes in the Cluster-tree. This deterministic and time-efficient multi-hop communication relies on predetermined neighbor nodes for interactions. This topology offers the benefit of expanding the network's coverage range, resulting in enhanced scalability. When an end node needs to transmit information to the SN, it does so by sending the data through a router node. This configuration yields a relatively small-scale network with high reliability and low data latency.

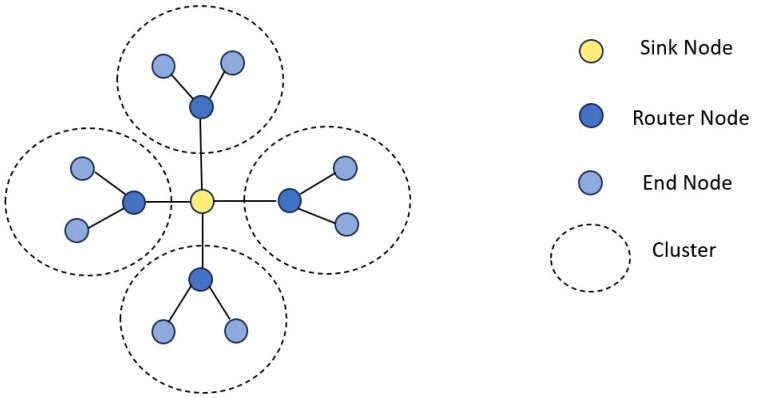


Fig. 1.8 Cluster topology

Table 1.1 provides a summary of the main features of the topologies discussed above.

1.6 Wireless Communication Protocols Applied in The Context of Smart Agriculture

WSNs use various communication protocols to facilitate the exchange of data and information between sensor nodes and the SN or gateway. These protocols are designed to address the unique challenges of WSNs, such as energy efficiency, scalability, and reliability. Wireless communication protocols can be categorized into two primary groups: long-range and short-range protocols, depending on their coverage capabilities. The selection of these protocols is influenced by the specific system conditions and requirements [19, 82]. Long-range network protocols have the capability to transmit data over extensive distances between devices. Examples of such protocols include LoRaWAN [19], Sigfox [67], and NB-IoT [89]. These protocols are used in smart agriculture systems where there's a need to connect devices across very large fields. In contrast, short-range network protocols like Wi-Fi [89] and ZigBee [127] enable data transfer over shorter distances, making them better suited for smaller-scale smart agriculture systems. Further detailed technical specifications for each of these protocols will be provided in the following sub-sections [10].

1.6.1 Zigbee

ZigBee is a Wireless Personal Area Network (WPAN) communication protocol based on the 802.15.4 standard. ZigBee emerged as a result of collaborative development efforts by

Table 1.1 Comparison of Star, Mesh, and Cluster-Tree Topologies

Feature	**Star Topology**	**Mesh Topology**	**Cluster-Tree Topology**
Structure	Centralized with one central node (hub).	Fully or partially connected network of nodes.	Hierarchical with parent-child relationships.
Communication Path	Through the central node.	Direct or multi-hop between nodes.	Multi-hop through parent and child nodes.
Fault Tolerance	Low (central node failure affects the network).	High (multiple paths for redundancy).	Moderate (failure of parent affects child nodes).
Scalability	Limited by the central node's capacity.	High (easily add nodes).	Moderate (depends on the hierarchical depth).
Energy Efficiency	Central node may consume more energy.	Varies (more energy for routing in multi-hop).	Moderate (energy depends on hierarchy).
Implementation Complexity	Simple and easy to implement.	Complex due to routing and path management.	Moderate (requires tree structure and maintenance).
Latency	Low (direct connection to central node).	Higher (due to multi-hop communication).	Moderate (depends on the tree level and hops).
Use Cases	Small networks (e.g., home automation).	Large, robust networks (e.g., industrial WSNs).	Medium-scale hierarchical networks (e.g., agriculture).
Cost	Low (fewer connections and devices needed).	High (more devices and connections required).	Moderate (based on the tree depth and nodes).
Data Traffic Handling	Centralized (bottlenecks possible).	Distributed (better for high traffic).	Hierarchical (optimized for specific levels).

the ZigBee Alliance, primarily known for its low cost and low power consumption. These attributes make ZigBee a suitable choice for applications in home automation and networking

applications [132]. ZigBee boasts reliability, cost-effectiveness, security, low power usage, and supports various topologies including star, peer-to-peer, and cluster tree. These noteworthy features have sparked the attention of the research community, particularly in the context of deploying WSN.

The effectiveness of the ZigBee protocol in the realm of smart agriculture has been demonstrated through a wide range of applications discussed in previous studies. Specifically, ZigBee is well-suited for low-power WSN configurations, which include multiple sensors as end nodes and a router for transmitting data over extended distances, as outlined in reference [33]. As an example, there are WSN designs based on ZigBee that have been developed to monitor soil temperature, and humidity levels, and to employ deep learning techniques for soil moisture content classification [123]. As ZigBee hardware is readily accessible in the market, it is frequently employed for rapid prototyping by interfacing with microcontroller boards [27] and single-board computers [66].

Although it is widely utilized in the field of smart agriculture, one notable disadvantage associated with ZigBee is its reported lack of a sleep-mode operation [12]. This situation potentially leading to inefficient power usage, which is a crucial consideration for IoT applications. Nonetheless, the transition between active and sleep modes can be effectively managed through proper programming and hardware configuration.
When considering the application of ZigBee in smart agriculture, the primary obstacle when it comes to employing ZigBee often lies in its inefficient power usage and the restricted capabilities in terms of range and data transmission speed.

Consequently, employing this protocol might not be advisable for scenarios involving expansive agricultural land and substantial inter-sensor node distances. In such cases, ZigBee is better suited to smaller-scale experimental test areas.

1.6.2 Wi-Fi

Wi-Fi emerged as one of the earliest and widely adopted wireless technologies, notably introduced through the IEEE 802.11 standard's release in 1997. It has gained recognition as the most promising alternative to the Global Positioning System (GPS) for indoor context-aware and location-based services. Primarily deployed in Wireless Local Area Networks (WLANs), it operates within the 2.4 GHz or 5 GHz frequency bands, offering a transmission range of up to 1 km and a minimum data rate of 150 kb/s.

Because Wi-Fi is readily accessible in both indoor and outdoor environments, it is often the primary choice for various IoT applications in the realm of smart agriculture. These applications encompass tasks such as gathering data and establishing cloud connections [8], as well as predicting conditions through data mining techniques [86]. Moreover, it has the flexibility to work in conjunction with other protocols within a smart agriculture system. As an example, in a hybrid monitoring architecture designed for smart agriculture environments, ZigBee is employed in channels requiring low data transmission, whereas Wi-Fi is employed for tasks like streaming surveillance video [48].

However, Wi-Fi's power consumption is notably high in the context of various IoT systems. making its utilization in all smart agriculture systems unfeasible, particularly in outdoor settings where meeting its power requirements is a formidable task. To address this issue, there have been studies focused on supplying external power through the deployment of photovoltaic solar panels in the field [124]. The suitability of incorporating solar panels was demonstrated through the creation of a hierarchical WSN structured like a tree. Within this arrangement, data concerning soil and environmental conditions collected at the network's terminal nodes were relayed to a central fertigation unit.
Furthermore, a reduction in power consumption among the nodes was achieved by adjusting the timing configuration for data acquisition and transmission intervals.

Despite being an older and well-established technology, the primary drawback of the Wi-Fi protocol remains its substantial power consumption. As a result, employing this protocol becomes nearly unfeasible when the hardware cannot be connected to main power line. Even though there have been attempts to integrate solar panels, this introduces additional expenses related to installation and maintenance costs.

1.6.3 Sigfox

Sigfox is a wireless communication technology known for its utilization of ultra-narrowband cellular networks, resulting in the ability to transmit data at low rates. This cellular infrastructure extends the suitability of Sigfox for applications in rural areas, particularly for IoT (Internet of Things) [68]. Notably, one of the initial applications of the Sigfox network involved tracking livestock in mountainous grasslands during the summer season [78, 118].

With the help of Sigfox end-to-end communication, it becomes feasible to segment it into distinct layers, namely the end layer, backbone layer, and sensing layer. These layers can then interconnect and collaborate to deliver application services adopted for smart agriculture.

For example, this technology can be employed to monitor the levels of silos and tanks, measure the temperature of stored grain, protect remote farmhouses and outbuildings, secure gates and deter livestock thieves, enhance colony health through remote monitoring of beehives, and supervise food temperatures throughout the entire cold supply chain.

The application of Sigfox in smart agriculture is adopted in [29] through the implementation of water quality monitoring system. The system's architecture exploits two key Sigfox features: efficient operation with a low duty cycle and the use of ultra-narrow band communication. The first feature, characterized by minimal power consumption, enables the system to incorporate a solar panel for recharging a Li-Po battery, resulting in an off-grid monitoring solution. Additionally, the monitoring system benefits from Sigfox's bidirectional control mechanism and enhanced security features, both of which are considered.

The primary benefit of deploying Sigfox in a range of smart agriculture applications is attributed to its ability to operate without the need for additional infrastructure and its extensive coverage in most European Union countries. In a recently developed irrigation management system, both Sigfox and LoRaWAN were incorporated [34].
The system architecture was specifically structured for vertical communication with the cloud, signifying that the sensors exclusively communicate with a local gateway and do not engage in direct communication with each other. Local communication was established through LoRaWAN, while the data collected at the local gateway was subsequently transmitted to a cloud server via Sigfox.

Despite its exceptional low-power consumption capability, Sigfox exhibits a low data transmission speed. Moreover, it operates within unlicensed ISM band frequencies, which are susceptible to signal interference and may result in unreliable data transfer [35]. Such circumstances may prove unsuitable for agricultural applications in which efficient data transfer plays a pivotal role. For instance, an irrigation control system that relies on continuous monitoring of soil moisture requires timely access to pertinent information; otherwise, it may lead to situations of over- or under-irrigation [10].

1.6.4 Narrow Band IoT (NB-IoT)

NB-IoT is regarded as the latest radio access technology developed by the Third Generation Partnership Project (3GPP) to facilitate connectivity for IoT devices. It offers various deployment options with broad coverage capabilities and operates seamlessly alongside

Global System for Mobile Communications (GSM) and Long-Term Evolution (LTE) within licensed frequency bands at 900 MHz, utilizing a frequency bandwidth of 200 kHz. Noteworthy features include extended battery life, extensive coverage, cost-effectiveness, and robust network security [15].

Two practical applications of NB-IoT in smart agriculture involve tracking livestock and monitoring environmental conditions in greenhouses [107]. In the livestock tracking system, the architecture allows for real-time reporting of the animals' current location along with system information such as battery status. Location data can be obtained through networks or satellite systems and can also be used for geo-fencing purposes. In the greenhouse monitoring system, sensors collect and monitor various data points including water pH levels, light intensity, and energy consumption. The system can operate on battery power and incorporates a bidirectional communication scheme for firmware and software upgrades. NB-IoT's suitability for diverse smart agriculture applications is due to its expansive coverage, half-duplex communication, adaptable power consumption based on operational mode, and enhanced security features.

Another example of a smart agriculture system architecture utilizing NB-IoT involves the monitoring of paddy storage [90]. Pin sensors are employed to measure the humidity inside the paddy bags, and this information is then transmitted to a gateway via the NB-IoT protocol. Noteworthy is the system's capability to function for 48 hours on a 10,000 mAh lithium polymer battery, despite NB-IoT offering a higher data rate compared to Sigfox and LoRaWAN.

While NB IoT does provide a higher data rate compared to Sigfox and LoRaWAN it comes with the drawback of using frequency bands. This means that there are subscription costs associated with implementing such a system. Additionally, there may be a need, for hardware upgrades if there is already an existing LTE infrastructure, in place. In the context of agriculture, this can pose a limitation when the allocated budget is not sufficient to cover these expenses [10].

1.6.5 Long-Range Wireless Area Network (LoRaWAN)

LoRaWAN is a network layer protocol developed by the LoRa Alliance and operates on LoRa chips. It exploits the benefits of LoRa modulation to establish networks with a specific focus on the IoT paradigm. LoRaWAN employs a star topology wherein nodes gather sensor data and transmit it to gateways. These gateways then convert the data into

a format compatible with Internet Protocol (IP) for transmission to a remote application server via the Internet. Specifically engineered for IoT applications, LoRaWAN enables the connection of a huge number of sensors and devices across expansive networks.

LoRaWAN architecture can be segmented into two primary components: the frontend and the backend [82, 84]. The frontend comprises gateway elements and end-node devices, while the backend is responsible for essential functions such as ensuring security, storing data, and managing query scheduling. These crucial backend tasks are carried out by network servers [122].

LoRaWAN is widely employed in smart city applications that need extensive network coverage, but its usage is also growing in various other social contexts. In the last four years, investments in LoRaWAN infrastructure by governments and private operators have started yielding results in different countries. LoRaWAN is frequently employed in a range of Smart Agriculture applications, including applications like weather forecasting [36], irrigation control, and farm monitoring. One notable feature of LoRaWAN is its low power consumption, making it a highly attractive option for the implementation of smart agriculture systems. Through experimental testing with the integration of solar-based energy harvesting, it was demonstrated that the projected battery lifespan of a LoRaWAN unit can potentially reach six times that of a Wi-Fi unit and double that of a ZigBee unit [102].

An IoT architecture with a flexible number of layers, adapted to different application and computation scenarios, was introduced [135]. It was demonstrated that this adaptable structure is suitable for a wide range of smart systems, including those in the field of smart agriculture. Different smart agriculture scenarios were tested in experiments, and the results revealed that the LoRaWAN protocol is capable of efficiently handling message traffic at a rate of 1500 messages/minute. In another LoRaWAN-based smart agriculture architecture, there is a central middleware layer that enhances the system's scalability by allowing various modules to be attached to it. The primary reasons for selecting the LoRaWAN protocol were its ease of deployment and its modularity features [10]. Field experiments conducted over three months demonstrated reliable data transmission.

The key characteristics of the WSN protocols are summarized in Table 1.2, which highlights the most significant properties.

Table 1.2 Wireless communication protocols features

Communication Protocol	Frequency	Data rate	Range (coverage)	Power Consumption	Cost
Zigbee	2.4GHz	250Kbps	10-100m	Low	Low
Wi-Fi	2.4GHz	11-54 Mbps	20-100m	High	Low
Sigfox	908.42 MHz	10-1000 bps	3-10-50 km	Low	Meduim
NB-IoT	450 MHz – 3.5 GHz	250 Kbps	1 - 10 – 40 km	Low	High
LoRaWAN	433/868/780/915 MHz	0.3-100 kbps	5-15-20 km	Extremely low	Low

1.7 Challenges in Wireless Sensor Network

WSNs have emerged as a fundamental technology with applications ranging from environmental monitoring to healthcare and industrial automation. However, the successful deployment and operation of these networks come with a unique set of challenges. In this discussion, we delve into five significant challenges faced by WSNs. These challenges not only impact the performance and longevity of WSNs but also require innovative solutions to ensure the network's efficiency and effectiveness. We'll explore each of these challenges in detail to better understand the complexities and potential solutions in the world of WSNs.

Limited resources: the size of the sensors can vary from a few centimeters to a few cubic millimeters. Naturally, this limits the capabilities of the sensors in terms of computation, storage, and communication, despite remarkable advancements in miniaturization technology. Consequently, it is necessary for algorithms and protocols to use these resources very efficiently. In particular, energy is the most precious resource for the sensors. Typically, a sensor is equipped with a battery (e.g., AAA type) that can last for several hours in active mode and several months in standby mode. The entire system's lifespan depends on the individual sensors' energy levels. Indeed, as the sensors participate in routing, result aggregation, or network self-configuration, the unavailability of certain nodes can cause the malfunction of the entire system.

Dynamic environment: Throughout the lifespan of a WSN, environmental changes can occur. Nodes may be displaced due to strong winds, and large physical objects may be positioned in a way that interferes with communication between nodes. Faced with these

environmental constraints, the fault tolerance solution must be capable of reacting automatically to withstand these events.

The scale: The number of nodes deployed for a project can reach up to a million. Such a large number of nodes generates a significant amount of inter-node transmissions and requires that the sink be equipped with sufficient memory to store the received information. Therefore, WSN protocols and fault tolerance mechanisms must take into account the "large scale" factor of sensor networks.

Lack of reliability : Some sensors may generate errors or cease to function due to issues like energy depletion, physical problems, or interference. Radio transmission can be momentarily disrupted by electrical fields, and so on. Furthermore, the absence of physical security for such sensors and the vulnerable nature of radio communications are characteristics that increase the risk of failures in these networks. Another issue encountered when dealing with sensor data is their sometimes incomplete and noisy nature, as well as occasional redundancy. Indeed, unreliable communication links and environmental obstacles result in losses, collisions, and interference with the packets sent by the sensors. These incidents significantly degrade data quality.

Security: Security poses a significant challenge in WSNs because these networks are deployed in various applications, ranging from battlefield scenarios to surveillance, building monitoring, burglar alarms, and critical systems like airports and hospitals. Maintaining confidentiality is vital in sensor networks to safeguard data transmitted between sensor nodes within the network and between sensors and the base station, as failure to do so could result in eavesdropping on communications. In WSNs, each sensor node and the base station must possess the capability to verify the authenticity of received data, ensuring it originates from a trusted sender rather than an adversary who deceived legitimate nodes into accepting false data. The introduction of false data could significantly alter the network's predictive behavior. Data integrity is another key concern, as data should remain unchanged and reach the end user accurately. Various types of threats in sensor networks include spoofing and altering routing information, passive information gathering, node subversion, sinkhole attacks, sybil attacks, Denial of Service attacks, and jamming [110].

1.8 Wireless Sensor Network Reference Model

As a communication system, the design of WSNs also adheres to the layered model's network design abstraction. A network model or architecture serves as a structural framework employed to design a network. It offers a more comprehensive insight into the functioning of the network. The communication task of a communication system is segmented into multiple subtasks, with these associated subtasks are interconnected and organized into layers, where each layer is responsible for a specific aspect of the communication process. In the highest layer, which is where users directly interact with the system, the responsibilities primarily revolve around user interactions and are executed by application programs. These software packages utilize the services provided by other software packages in a lower layer, which in turn rely on the services of other packages, ultimately reaching the lowest layer where actual signals are transmitted between computers. These activities are structured into layers, with the top layer primarily encompassing software-related tasks, and as you descend through the layers, the focus gradually shifts towards hardware related activities.

In addition to the advantage of decomposing the communication task into manageable subtasks, layering makes it easy to incorporate new services as they become available. This occurs because layering creates well-defined interfaces between the layers. The different tasks of a layer are accomplished by protocols. The protocols supply communication services used for message exchange by higher-level entities, like application processes (managed by application protocols), as well as lower-level entities. While the functions of each layer are typically well-defined, there can be moments when the boundaries between them appear somewhat ambiguous.

Numerous network models have been established, with the two most popular ones being the 7 layer Open System Interconnection (OSI) model and the four layer Internet model. However, due to the unique characteristics of WSN, its architecture diverges somewhat from these standard models [9]. Its layered design model comprises five layers, as depicted in figure 1.9.
The application layer incorporates the functionalities of the application, presentation and session layers of the OSI model. The functions of the transport, network, data link, and physical layers are similar, to those of the OSI model. Here, we present a brief description of the functions of each of these layers:

The Application layer: It directly Interacts with users, performing tasks such as appli-

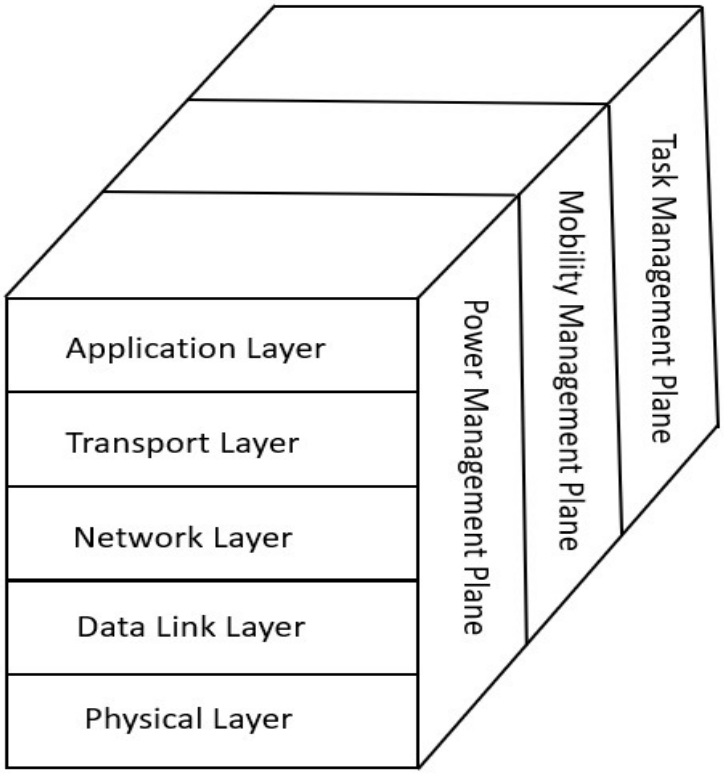


Fig. 1.9 Wireless sensor network architecture

cation processing, data aggregation, query handling, external database access, security, and more.

The Transport layer: It is responsible for tasks such as diffusion and gathering data, caching, and storage.

The Network layer: It is responsible for tasks such as adaptive topology management and routing

The Data Link layer: It performs tasks such, as managing channel sharing (MAC) keeping track of timing and ensuring locality.

The Physical layer: Its tasks involve activating communication channels and processing signals.

In addition to these layers, we have the three cross-layer planes mentioned above, which constitute the management protocols. These comprise power management, mobility management, and task management planes. They play a crucial role in managing the network and coordinating the sensors' operations to enhance the network's overall efficiency.

- **Power management**: manages the power usage of a sensor node.
- **Mobility management plane**: Detect the movement of sensor nodes. Nodes are capable of monitoring the movements and power levels of neighboring nodes for the purpose of power management and balance.
- **Task management plane**: Schedule the sensing tasks for a designated area and establish the status of each node, distinguishing between those that are inactive and those that are active.

1.9 Energy Comsuption Model of the Sensor Node

To present the energy consumption model of the sensor node, it is essential to model each individual node block. In this section, we discuss various operational modes of sensor nodes. After calculating the energy consumption for each mode, we obtain the energy consumption model. Figure 1.10 shows the possible working sequence of the sensor node, enabling the definition of various operating modes, which are controlled by the processing unit [17].
The energy consumption model is based on the following assumptions:

- The processing unit remains in the on-state throughout the operational sequence [117].
- Each phase of the sensor's operational sequence is characterized by a constant time duration.
- The radio module broadcasts a data packet at a constant transmission power level.
- This model does not account for local data storage but assumes the real-time transmission of all acquired data.

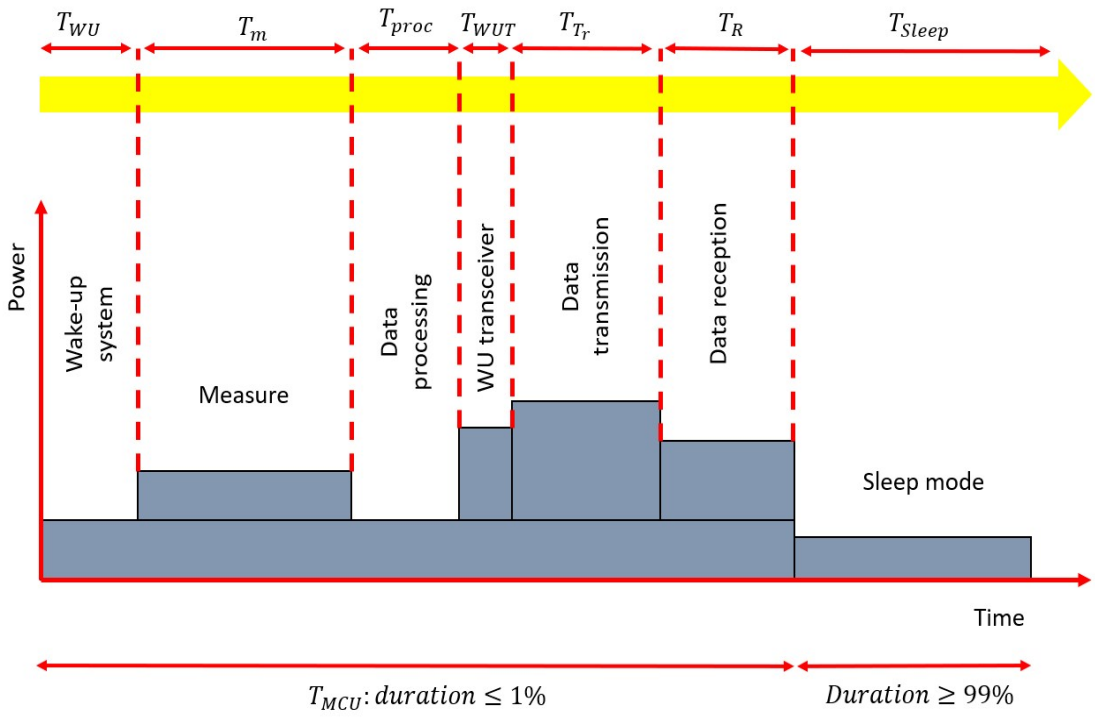


Fig. 1.10 Sensor working scenario

In this section, we define the different operating modes of the communicating sensor. One of the initial ways to approach this is to consider all elements active during a specific time period and inactive for the rest of the time cycle. Most of the time, the wireless sensor is in sleep mode. The energy consumption in this mode can have a substantial impact on the sensor's overall power consumption. Therefore, it is necessary to take into account the energy consumption while in sleep mode when developing our model.

The total energy E_{Total} consumed by the communicating sensor during one cycle is expressed by equation (1.1):

$$E_{Total} = E_{Sleep} + E_{Active} \tag{1.1}$$

where E_{Sleep} and E_{Active} represent the energy dissipated by the node in sleep mode and the total energy consumption during the active mode of the microcontroller, respectively. E_{Sleep} is given by:

$$E_{Sleep} = P_{Sleep} T_{Sleep} \tag{1.2}$$

where P_{Sleep} and T_{Sleep} denote the power consumption and the time duration in sleep mode, respectively.
The total energy consumption E_{Active} is computed as the sum of the energy consumption of each component within the sensor node. It is expressed as:

$$E_{Active} = E_{WU} + E_m + E_{proc} + E_{WUT} + E_{Tr} + E_R \tag{1.3}$$

where E_{WU}, E_m, E_{proc}, E_{WUT}, E_{Tr}, and E_R represent the energies consumed in the system during wake-up, data measurement, microcontroller processing, wake-up of the transceiver, transmission mode, and reception mode, respectively. Before doing measurements, the wake-up of the sensor node is managed by the processing unit. The energy consumption E_{WU} during the wake-up duration T_{WU} is expressed as:

$$E_{WU} = P_{ON}(f_{MCU})T_{WU} \tag{1.4}$$

Where $P_{ON}(fMcu)$ the power consumption of the microcontroller (which depends on the microcontroller frequency f_{MCU}), and the wake-up duration, T_{WU}, influence the overall consumed power. After the wake-up period, the sensor performs data measurements. Equation (1.5) describes the dissipated energy, E_m, during this measurement phase:

$$E_m = (P_{ON}(f_{MCU}) + P_m)T_m \tag{1.5}$$

Where P_m represents the power consumption, while T_m denotes the corresponding duration of measurements.

After taking measurements, the microcontroller moves on to processing the data. The time duration of $T_{proc}(f_{MCU})$ depends on the microcontroller's operating frequency, f_{MCU}, and the number of instructions, $N_{inst}(T_{proc}(f_{MCU}) = N_{inst}/f_{MCU})$. In equation (1.6), we determine the energy consumption by the processing unit, assuming one instruction per clock period:

$$E_{proc} = P_{ON}(f_{MCU})T_{proc}(f_{MCU}) \tag{1.6}$$

Then, the energy consumed E_{WUT} during the wake-up time T_{WUT} of the transceiver is expressed as:

$$E_{WUT} = (P_{ON}(f_{MCU}) + P_{WUT})T_{WUT} \tag{1.7}$$

where P_{WUT} denotes the power consumed during the wake-up of the transceiver. Subsequently, the energy consumed E_{Tr} during the transmit mode is described as:

$$E_{Tr} = (P_{ON}(f_{MCU}) + P_{Tr})T_{Tr} \tag{1.8}$$

P_{Tr}, representing the dissipated power during the transmit mode, combines with its time duration T_{Tr}, the latter is given by equation (1.9).

$$T_{Tr} = N_{bit}T_{bit} \tag{1.9}$$

where N_{bit} is the number of transmitted bits, and T_{bit} is the time it takes to transmit one bit.

$$E_R = (P_{ON}(f_{MCU}) + P_R)T_r \tag{1.10}$$

Where P_R and T_R represent, respectively, the power consumed during the reception mode and the time duration of reception.
Moreover, equation (1.11) describes how much energy the microcontroller E_{MCU} consumes when it is in the on-state:

$$E_{MCU} = P_{ON}(f_{MCU})T_{MCU}(f_{MCU}) \tag{1.11}$$

The time duration, $T_{MCU}(f_{MCU})$ of the microcontroller depends on its overall operating time in various modes and can be expressed as follows:

$$T_{MCU}(f_{MCU}) = T_{WU} + T_m + T_{proc}(f_{MCU}) + T_{WUT} + T_{Tr} + T_R \tag{1.12}$$

1.10 Conclusion

In conclusion, this chapter has delved into the pivotal role of WSNs within the domain of smart irrigation applications. It began by clarifying the pressing water challenges facing agriculture, emphasizing the need for efficient solutions to address these issues.

The integration of WSN technology represents a significant step forward in agricultural practices, offering real-time monitoring, precise control, and data-driven decision-making capabilities. Throughout this chapter, we have explored the architecture, components, and network topologies of WSNs, as well as their diverse applications. Moreover, the discussion extended to the communication protocols used in smart agriculture, highlighting their characteristics and suitability for different agricultural environments. We also addressed the

inherent challenges encountered in WSN deployments. Furthermore, an overview of the WSN reference model and an energy consumption model provided valuable insights into the design and optimization of WSNs for smart irrigation applications.

Overall, this chapter has underscored the transformative potential of WSNs in mitigating water challenges and enhancing agricultural sustainability. By leveraging technology to monitor and manage irrigation systems more effectively, WSN-enabled smart irrigation offers promising prospects for increasing efficiency, productivity, and resilience in agricultural practices.

Given these circumstances, improving water use efficiency in irrigated agriculture is essential. Effective water management involves using available resources to meet real-time crop irrigation needs, which requires careful monitoring and control of irrigation practices. Smart irrigation systems play a vital role in this context. By continuously monitoring the crop environment and adjusting water application, these systems ensure that soil moisture remains within specific thresholds, such as field capacity and the permanent wilting point. The importance of smart irrigation lies in its ability to optimize water use, thus contributing to sustainable agricultural practices and enhancing food security in a world facing increasing water scarcity.

Chapter 2

Fault Detection in Wireless Sensor Network: State of Art

WSNs have emerged as a critical component in various applications ranging from environmental monitoring to industrial automation. However, ensuring the reliable operation of these networks presents significant challenges, particularly in detecting and diagnosing faults. The ability to swiftly identify and address faults is paramount for maintaining the integrity and efficiency of WSNs.

This chapter delves into the state of the art in fault detection in WSNs. It begins with a discussion on the generality and motivations behind fault diagnosis in WSNs, emphasizing the importance of enhancing data reliability, optimizing bandwidth utilization, and prolonging network lifetime. From there, it explores fault classification and modeling, shedding light on the diverse methods employed for fault detection and diagnosis in WSNs.

A crucial aspect of fault detection lies in understanding and utilizing data correlations effectively. Categorizing fault detection approaches based on their utilization of correlations, redundancies, and attribute dependencies provides valuable insights into their efficacy and applicability. Furthermore, the chapter examines how fault detection techniques can be classified based on the system architecture and operation mode within a WSN, distinguishing between centralized, distributed, and hybrid architectures, as well as online and offline modes of operation.

The chapter also delves into the intricacies of fault detection and isolation, outlining the key steps involved in optimizing the utilization of sensor measurements for effective fault detection. Additionally, it discusses the unique challenges that arise in fault diagnosis in

WSNs, highlighting the limitations of conventional fault detection techniques and the need for tailored approaches to address the specific nature of sensor data.

2.1 Generality on Fault Diagnosis in Wireless Sensor Networks

The diagnostic process for sensor faults in WSNs is crucial for ensuring the network's reliability and performance. It involves detecting and identifying faulty sensors or sensor nodes that may affect the accuracy of data collected or disrupt communication within the network. The diagnostic techniques used in this context aim to quickly detect the problematic sensors, assess the severity of their faults, and isolate them from the network to prevent further issues. This process may include analyzing sensor data patterns, and monitoring communication anomalies. Effective sensor fault diagnostics not only improve data quality but also enable timely maintenance and replacement, ultimately contributing to the overall efficiency and longevity of the WSN.

In recent years, advancements in fault diagnosis technology have led to the emergence of various sensor fault diagnosis methods from diverse perspectives. Among these methods, the concept of redundancy stands out as a commonly utilized approach. Redundancy can be categorized into hardware redundancy and analysis redundancy, as illustrated in figure 2.1. Hardware redundancy involves the measurement of the same signal using multiple sensor devices. Typically, a linear combination of measurements obtained from redundant sensors is employed to derive estimates, and each measurement is compared to an estimate to detect any faults. On the other hand, redundancy analysis does not necessitate additional hardware. Therefore, it is often considered more cost-effective compared to hardware redundancy methods [72].

2.2 Motivations for Fault Diagnosis in Wireless Sensor Networks

In most applications, WSN nodes are required to operate autonomously in challenging environments, often without direct supervision. As a result, these nodes may become faulty or cease to function over time. Failure to promptly identify and address these faulty nodes can result in data inconsistencies, degradation of network performance, congestion in routing

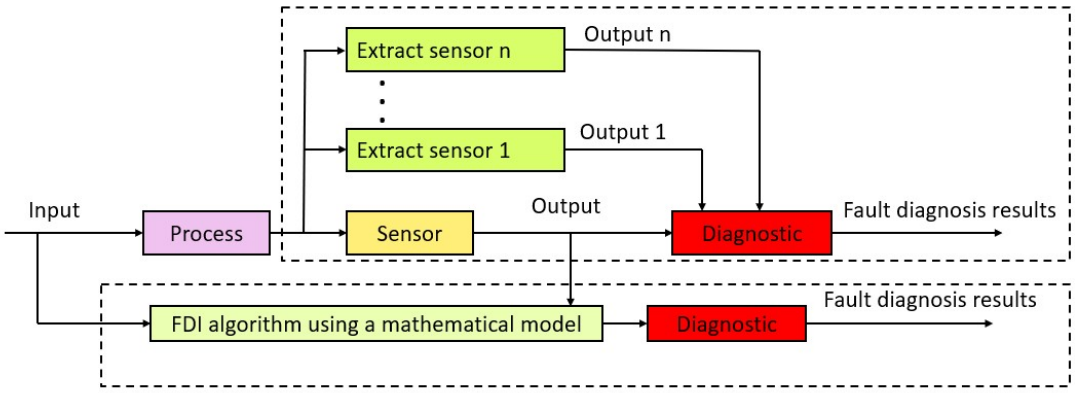


Fig. 2.1 The principles behind hardware redundancy and analytical redundancy

paths, and a decrease in overall network lifespan. Thus, the motivations for fault diagnosis in WSNs include the following objectives [131]:

1. **Increase data reliability**

   Sensor nodes are susceptible to various factors that render them faulty and unreliable, including hardware and software failures, environmental influences, and malicious attacks that manipulate nodes into producing erroneous data. When such faulty data is transmitted to a SN or Base Station (BS), it influences the accuracy of decision-making conducted by the BS. For example, in environments where networks are deployed to detect poisonous gases in chemical plants, the failure to detect gas leaks in time endangers worker safety.

   Fault diagnosis plays a crucial role in distinguishing between normal and faulty sensed data, enabling the elimination of erroneous data and mitigating adverse effects on the overall sensor network.

2. **Make Bandwidth Utilization Efficient**

   Bandwidth denotes the speed of data transfer, typically measured in bits per second. Given that WSNs operate within resource constraints, expanding bandwidth to accommodate node requirements poses challenges. Moreover, faulty data consumption further strains network bandwidth. This issue is exacerbated in multimedia sensor

networks due to the substantial bandwidth demand for multimedia data transmission. Additionally, algorithms employed in multimedia sensors necessitate sophisticated hardware and computational power for processing, resulting in computational energy dissipation comparable to communication energy consumption [128, 97].

3. **Prolong Network Lifetime**

The lifetime of the network is closely tied to the energy levels of its nodes [59]. As nodes gradually deplete their energy reserves while executing network tasks and facilitating data transmission, they eventually reach a point of exhaustion. This depletion of battery power leads to deteriorating connectivity among nodes, resulting in their disconnection from the network and potential partitioning.

Fault diagnosis plays a crucial role in reducing faulty data transmission, thereby effectively lowering the energy consumption of the network. Additionally, the presence of dead nodes can result in inefficient energy usage. Furthermore, if a dead node serves as a routing node, it has the potential to induce routing loops, further exacerbating energy wastage within the network.

2.3 Definitions and Terminologies

To facilitate the understanding of the rest of this thesis, we introduce the following definitions [14]:

- **Anomaly:** An attribute that deviates from the principles of natural or logical laws.

- **Failure:** A failure signifies an enduring interruption in the sensor node's capability to carry out its essential task. This surpasses a regular failure, as it encompasses a total shutdown of the sensor node, resulting in a complete halt of its functions.

- **Fault:** A fault refers to any deviation or anomaly in the sensor's behavior from its expected or normal functioning, which may not necessarily result in physical failure or complete breakdown [45].

- **Outlier:** It is a data point, or a group of data points, that appears to differ significantly from the majority of the dataset. In the context of WSNs, outliers are described as measurements that notably diverge from the normal pattern of sensed data [20].

- **Perturbation:** Refers to any small or temporary change or disturbance in the sensor's environment or operating conditions that may affect its performance or output. It is defined as an unknown and uncontrolled input acting on a system, which can include variations in temperature, pressure, humidity, or other external factors that influence the sensor's ability to accurately measure or detect signals.

- **Residual or fault indicator:** It is a metric or signal used in various systems, particularly in engineering and statistical analysis, to quantify the difference between observed data and the predictions or expectations from a model or baseline. It serves as an indicator of the presence or magnitude of deviations, faults, or errors in the system being analyzed.

- **Threshold:** It refers to a predetermined limit or value used to assess the magnitude of residual errors. Residuals represent the differences between observed sensor measurements and the expected values predicted by a model. By comparing these residuals to a threshold, analysts can determine whether the deviations are within acceptable bounds or if they exceed a certain level, indicating potential sensor faults or anomalies.

- **Fault diagnosis:** It is the determination of the type, size, location and time of occurrence of a fault, it follows the detection of faults and includes isolation and identification.

- **Fault detection:** The purpose of the fault detection method is to identify the existence of defective nodes within the network and determining the moment when it occurred.

- **Fault isolation:** the malfunctioning sensor nodes are isolated from the network, while the nodes that are operating properly remain unchanged. The defective nodes are excluded from participating in the network's operations.

- **Fault identification:** This involves determining the size and temporal behavior of a fault.

- **Monitoring:** This is a continuous task, carried out in real time, of determining the state of a physical system, which consists of recording information and recognizing and indicating behavioral anomalies.

- **Network lifetime:** The duration over which the network carries out its assigned tasks effectively.

The aforementioned mechanisms play a crucial role in the fault diagnosis system, providing comprehensive insights into detected faults.

2.4 Fault Classification and Modeling

In general, sensor nodes located in the same place and time, which are free of faults, do not transmit identical values due to imperfections in the accuracy of their sensing units [11]. Sensor or data faults refer to instances where a sensor reports data that is inconsistent with the actual behavior of the phenomenon being observed.

Sensor nodes in WSN are prone to many failures, which can be categorized into three types[87]: Hardware failures, Software failures, and Communication failures.

Hardware failures : Occur as a result of issues with various sensor hardware components, such as the sensing unit, power unit, location unit, or processing unit.

Software failures : Arise from problems within the sensor programs.

Communication failures : Occur due to issues with the sensor transceiver, impacting the transmission of data [133].

In the literature, various classifications of fault types in WSNs can be found. Faults can be categorized based on their duration, underlying cause, or the behavior of the failed component. Figure 2.2 illustrates two primary aspects for classifying faults: the time-based aspect and the behavior-based aspect.

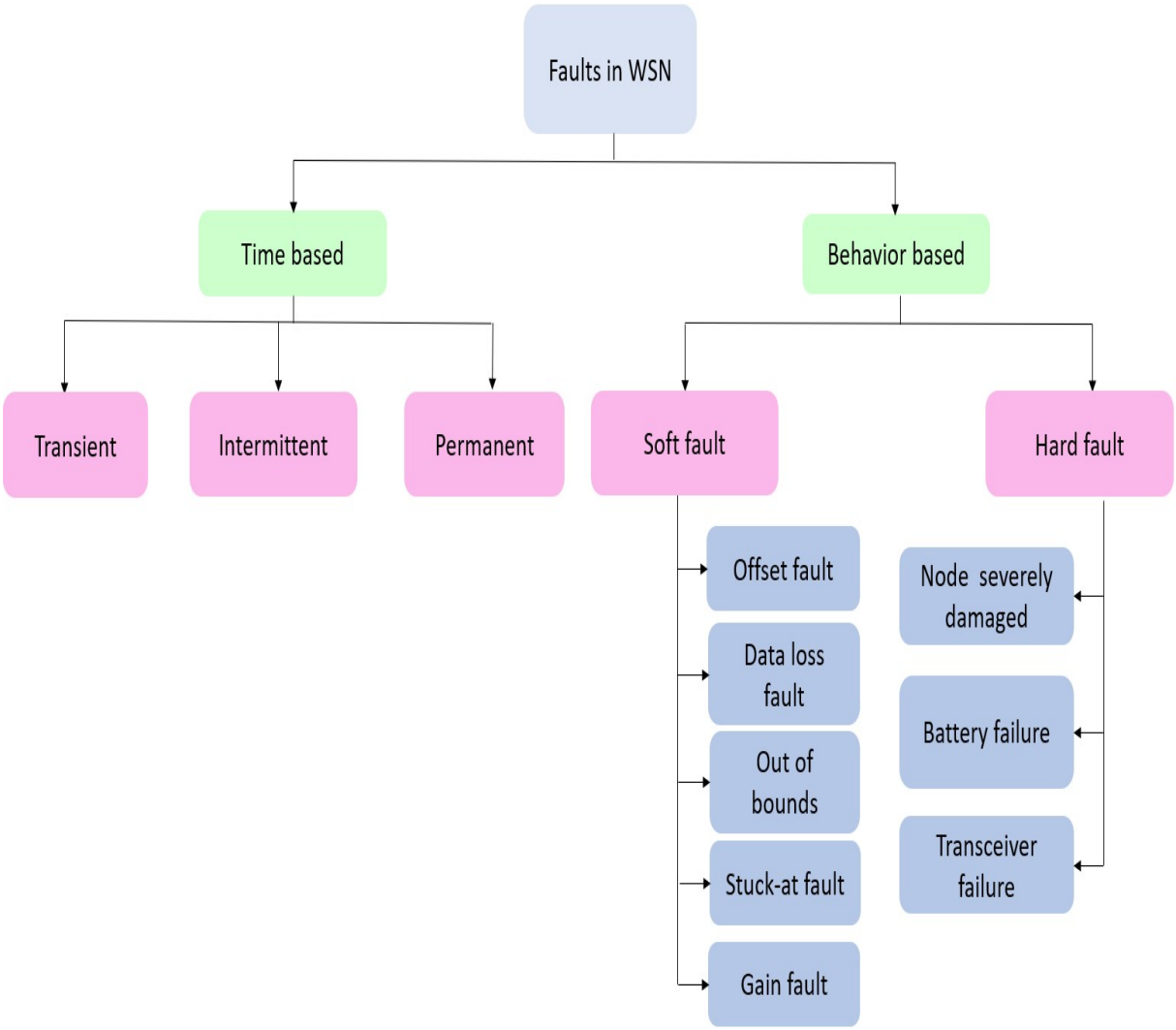


Fig. 2.2 Taxonomy of faults in wireless sensor network

The time-based aspect refers to the duration of the faults. Certain faults are temporary and occur for a specific period of time. Therefore, based on the duration, faults can be further classified into three categories: transient faults, intermittent faults, and permanent faults as shown in figure 2.3.

- **Permanent fault:** It occurs when a sensor consistently transmits incorrect data. These faults persist until a fault recovery procedure is performed to address the underlying issue and restore proper functionality.

- **Transient fault:** It is a temporary fault that is influenced by external factors originating from the sensor node's environment, such as network congestion or changing weather conditions. It is important to note that a transient fault does not necessarily indicate a faulty sensor node. Due to their accidental and short-lived nature, diagnosing and tracking transient faults can be relatively challenging.

- **Intermittent fault:** is related to the internal components of either software or hardware. In this case, the faulty sensor consistently transmits incorrect values but then alternates and sends the correct ones intermittently. These intermittent faults arise from internal issues and can be attributed to either software or hardware origins.

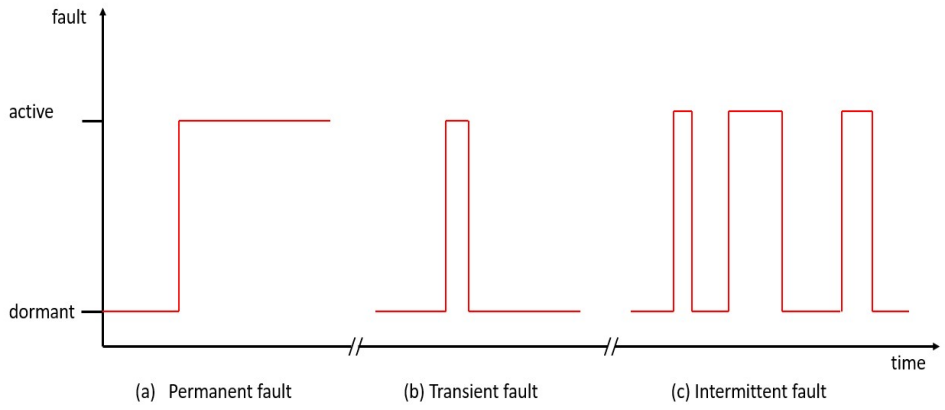


Fig. 2.3 Fault categorization based on their duration

Therefore, when considering the behavior-based aspect of faults, we can broadly classify them into two categories: soft faults and hard faults.

- **Hard fault:** Also known as a crash fault, refers to a situation where a sensor node loses its ability to establish communication with the rest of the network. This type of fault can occur due to various reasons, including a complete depletion of the battery, a faulty transceiver, or severe physical damage to the node.
  When a sensor node experiences a crash fault, it loses its internal state and becomes unable to participate in any in-network activities. These types of faults are considered natural faults that arise from natural phenomena without any human involvement.

- **Soft fault:** In contrast, soft fault occur when a sensor node continues to operate and communicate but exhibits altered behavior. For example, it may produce faulty data. Soft faults involve deviations in the expected behavior of the node while still maintaining its operational status.

From a data-centric perspective, soft faults in WSNs can also be categorized based on the data transmitted by the sensors. By considering the characteristics of the sensed data, faults can be identified and classified into various categories, including offset faults, gain faults, stuck-at-faults, out-of-bound faults, and data loss faults.

To represent these faults, let's consider the data generated by a sensor node as a time series, denoted as $d(n,t,y(t))$. Here, n represents the node ID, t denotes the specific time instance when the value was sensed, and $y(t)$ represents the value sensed by node n at time t. The function $y(t)$ can be modeled as $\alpha+\beta x+\eta$, where α is an additive constant known as the offset, β is a multiplicative constant called the gain, x represents the non-faulty sensor value at time t, and η represents the external noise present in the data.

In an ideal scenario, $y(t)$ would be equal to x. However, in real-world cases, a fault-free node will have $y(t) = x+\eta$ [87].

Offset fault

An offset fault occurs when the sensed data deviates from the expected values due to an additional constant being added. This can happen due to improper calibration of the sensor. In mathematical terms, the offset fault can be represented as $y(t) = \alpha+x+\eta$, where $y(t)$ is the measured value, α is the constant offset, x is the expected value, and η represents any noise or disturbance. Figure 2.4 illustrates an offset fault.

Gain fault

We say a gain fault is when the rate of change in sensed data deviates from the expected behavior for an extended duration as shown in figure 2.5. In this type of fault, a constant value is multiplied by the sensor data that is unaffected by the fault. Improper calibration of the sensors can contribute to the occurrence of a gain fault. Mathematically, a gain fault can be represented as $y(t) = \beta x+\eta$, where $y(t)$ is the measured value, β

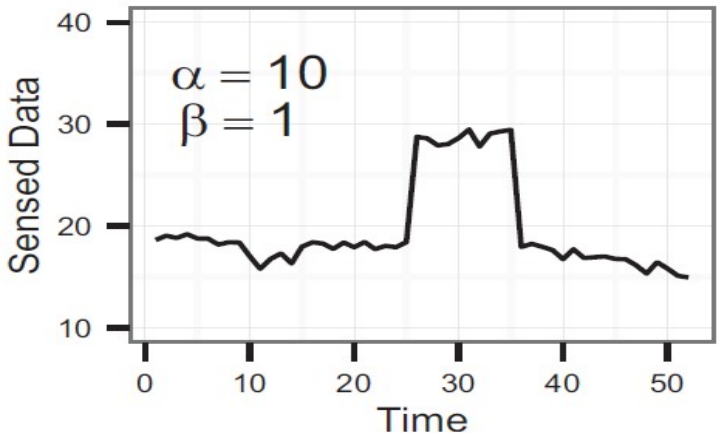


Fig. 2.4 Offset fault in WSN [87]

represents the constant value that multiplies the normal reading and η accounts for any accompanying noise or disturbances.

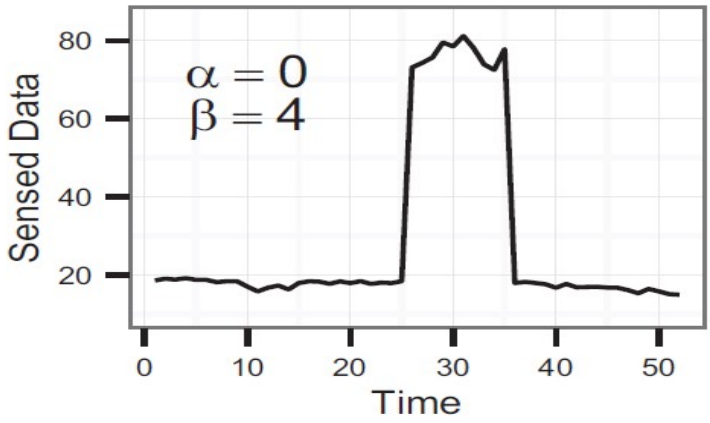


Fig. 2.5 Gain fault in WSN [87]

Stuck-at fault

A stuck-at fault refers to a situation where the data from a node remains constant, resulting in zero difference or variance within the data series as illustrated in figure 2.6. This fault can be categorized as either transient or persistent. In the context of modeling a stuck-at fault, it can be represented as $y(t) = \alpha$, where $y(t)$ is the measured value, α

represents the constant value that is sensed.

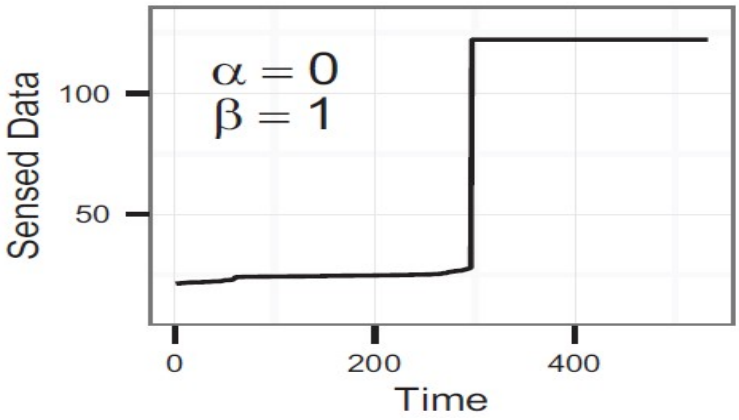


Fig. 2.6 Stuckat fault in WSN [87]

Out of bounds

An out-of-bound fault occurs when the sensed data exceeds the predefined thresholds set by the problem requirements. When a node exhibits this fault, it means that the measured value, denoted as y(t), is either greater than the upper threshold θ_1 or smaller than the lower threshold θ_2. These thresholds, θ_1 and θ_2 represent the limits specified by the application.

Data loss fault

A data loss fault refers to a situation where the sensed data is absent or missing from the time series of a specific node as illustrated in figure 2.7. if y(t)=ϕ, *and* $t > \tau$; where ϕ is the null set and τ is the maximum time required for sensing, then it is called data loss error.

2.5 Methods of Fault Detection and Diagnosis in WSN

In recent years, there has been a remarkable surge in the popularity and utilization of WSNs. This growth has led to the development of numerous approaches and

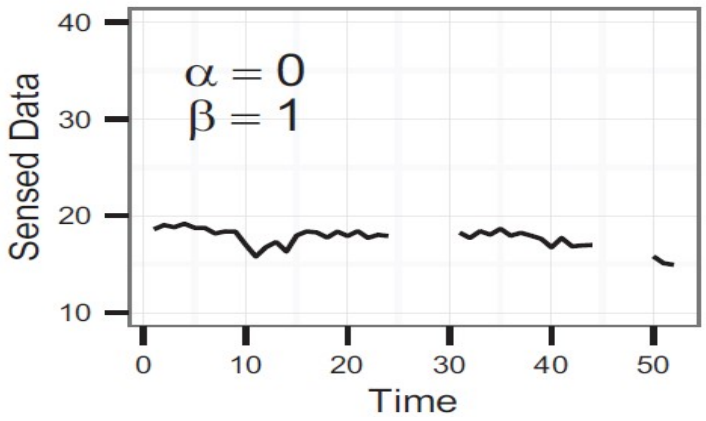


Fig. 2.7 Data loss fault in WSN [87]

strategies aimed at detecting various types of anomalies in diverse WSN applications. As the number of these approaches continues to increase, there arises a need for a comprehensive classification system.

To address this need, we have categorized sensor fault detection techniques in WSN into seven primary categories, as depicted in figure 2.8. These categories encompass a wide range of previous works documented in the literature. The seven categories are as follows[104]:

- Statistical-based;
- clustering-based;
- nearest neighbour-based;
- classification-based;
- artificial intelligence-based;
- spectral decomposition-based;
- hybrid-based.

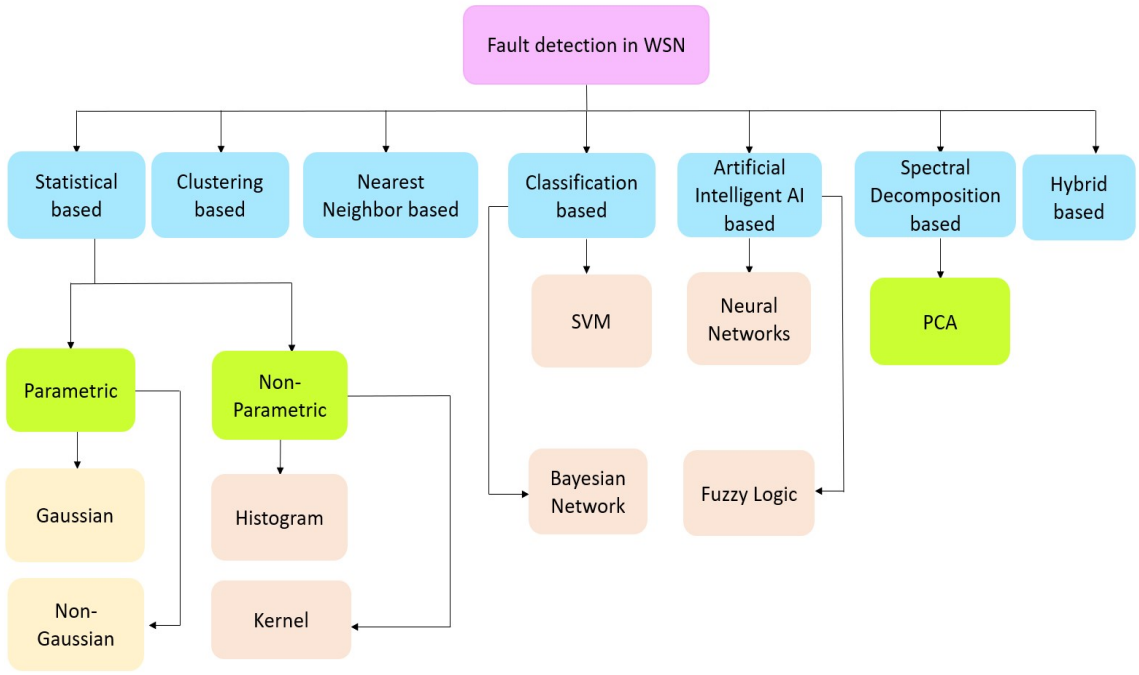


Fig. 2.8 Fault detection techniques for Wireless Sensor Networks

In the following subsequent sections, we will delve into a detailed description of each category, elucidating the underlying principles and methodologies employed in these techniques.

Statistical-Based Techniques

The initial techniques for sensor fault detection problems in WSN, we present those that rely on statistical approaches. These techniques involve assuming or estimating a statistical model, typically a probability distribution, which represents the underlying data distribution. They then assess how well the data values conform to this model.

If a data value exhibits a very low probability according to the generated model, it is deemed a fault. The statistical-based techniques can be further divided into two sub-categories[130]:

- **Parametric Approaches:** utilize models that depend on pre-existing knowledge about the distribution of the data, where the data is generated from a well-established model. In these cases, the distribution parameters are estimated using the available data. If a data value significantly deviates from the data model, it is

identified as a fault.

– **Non-parametric Approaches:** do not rely on any prior knowledge regarding the data distribution and instead construct their models solely based on the input data. The two prevalent nonparametric techniques are centered around histograms or kernel functions.
The Histogram technique [111] involves tallying and approximating the frequency of various data values by determining the potential occurrence of a given value. Additionally, this technique includes comparing a test data value with histogram categories to determine if it aligns with any of them. On the other hand, Kernel functions [96] compute the probability distribution function (PDF) for normal data values. If a new data value exhibits a low PDF, it is classified as an outlier. In the subsequent section, we provide a concise overview of recent studies that fall within this category.

Clustering-Based

Clustering techniques are commonly employed in the field of data mining to group data values that exhibit similar behavior into clusters as shown in figure 2.9. Faults are identified as data values that either do not belong to any cluster or belong to significantly smaller clusters compared to others. Detecting faults relies on the assumption that normal values are part of large clusters, while faults reside in small clusters or do not belong to any cluster at all.
These techniques utilize metrics, such as Euclidean distance, to measure the similarity between data values. Additionally, they employ inter-/intra-cluster distance thresholds and cluster width to determine clusters effectively. Selecting these parameters correctly is crucial for achieving efficient results. Moreover, clustering techniques not only gather similar data values into clusters but also merge some clusters to reduce communication costs.

Nearest Neighbour-Based Techniques

These techniques are commonly employed in data mining and machine learning to assess the significance of a data value based on its KNN (K-Nearest Neighbors). They utilize various established distance metrics to compute the distance or similarity measure between two data values. If a data value is found to be significantly distant from

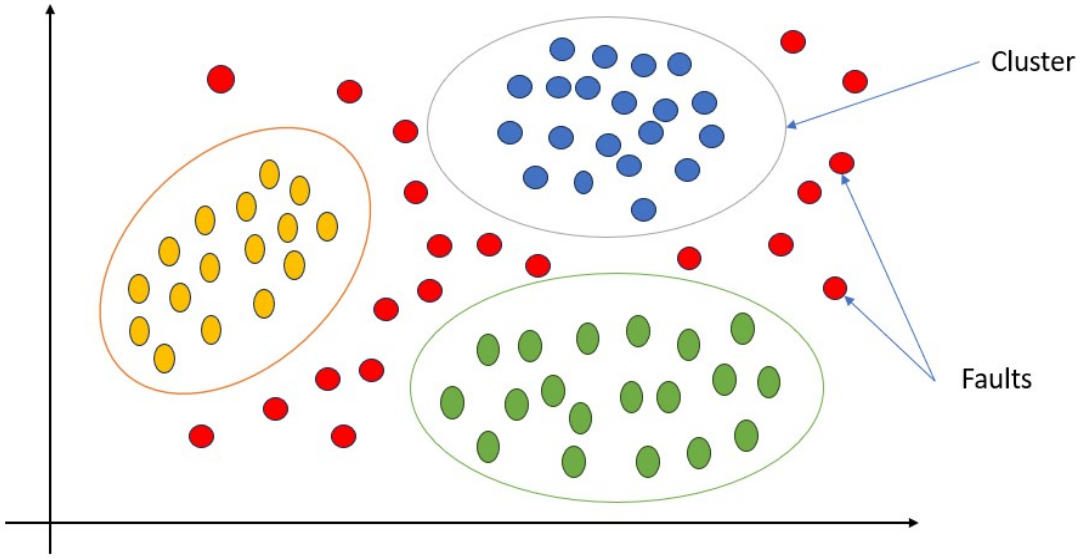


Fig. 2.9 The clustering-based fault detection method

its neighboring values, it is classified as an outlier. Numerous distance metrics exist for this purpose, such as:

- Euclidean distance, which is suitable for univariate data.
- Mahalanobis distance (MD), which is suitable for multivariate data.
- Hamming distance.

Classification-Based Techniques

Classification-based techniques involve the creation of a classification model, also known as a classifier, using certain data values as training samples. The classification process consists of two main phases: training and testing. During the training phase, the classifier learns from the provided dataset, while in the testing phase, the data values are categorized into either normal or abnormal (outlier) groups.

Moreover, when new data values arrive, it is necessary to conduct additional training to incorporate these fresh samples into the classifier. Figure 2.10 illustrates the Support

Vector Machine (SVM) classifier.

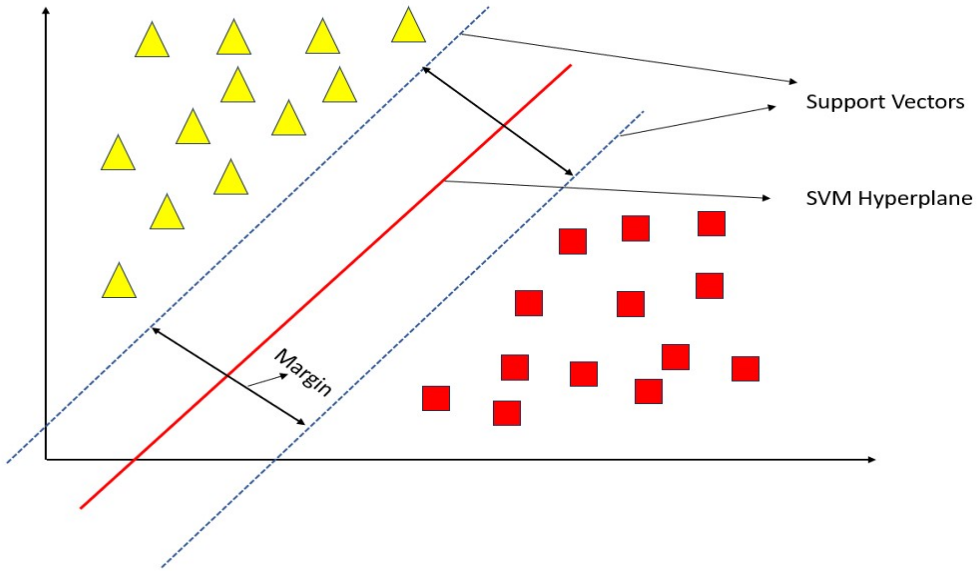


Fig. 2.10 SVM classifier

Artificial Intelligence Techniques

Artificial intelligence techniques have emerged as contemporary methods for detecting outliers in IoTs. Examples of these techniques include neural networks (NN) and fuzzy logic, both of which rely on decision-making theories. Neural networks serve as logical models that consider the entire dataset to provide a comprehensive perspective and make decisions. On the other hand, fuzzy logic employs distinct values such as (yes/no, right/wrong) to differentiate normal data values, offering valuable assistance in addressing outlier detection challenges.

Spectral Decomposition Techniques

In these techniques, the utilization of principal component analysis (PCA) serves to simplify and diminish the dimensions of the dataset. By doing so, PCA generates a new subset of data known as components, which are uncorrelated and retain the maximum variance. This process helps in identifying the normal behavior of the data

before detecting outliers. In particular, data values that are not well reconstructed or deviate significantly from the variability captured by the top principal components are considered outliers.

Hybrid Techniques

Hybrid techniques are emerging as a solution to overcome the limitations of previous approaches by combining multiple solutions. This integration allows for leveraging the strengths of different methods to achieve more favorable outcomes. By merging these techniques, we can attain improved results that address the drawbacks of individual approaches.

2.5.1 Data Correlations

Categorizing fault detection approaches is important and can be also distinguished based on their utilization of correlations, redundancies, and attribute dependencies in the input data[130]. This categorization becomes crucial because many of these approaches rely on assumptions about the nature or granularity of the data, specifically focusing on correlations between multiple instances. Four categories of correlations can be distinguished [55]:

1. **Temporal correlations**

   We examine sensor readings that are ordered in time and compare each reading to the ones before and after it. These correlations help us identify abnormal patterns in time, such as high variability in subsequent readings, no change in readings, gradual shifts, or readings that are outside the expected range [55].

2. **Spatial correlations**

   By considering the data gathered from neighboring nodes, we can detect variations in sensor readings for a particular node. In this process, taking into account the location information of the neighbors becomes advantageous. For indoor applications, this could involve utilizing a floor plan, while for outdoor deployments, geographical attributes would be considered [126].

3. **Spatio-temporal correlations**

   A combination of the aforementioned approaches is employed, where sensor readings are examined in both temporal and spatial contexts. This involves comparing the readings based on time and location, considering the data from the sensor node as well as its neighboring nodes. By utilizing this form of correlation, more complex anomalies can be identified, which may arise from factors such as sensor calibration issues, hardware malfunctions, design flaws, or unusual occurrences in the environment being monitored [126].

4. **Functional correlations**

   We utilize the relationships and dependencies among various attributes within the sensor data to identify combinations that are illogical or impossible. In order to achieve this, it is important to have additional information about the specific attributes and their interconnections, which provides the required context. For instance, considering physical laws, a temperature reading is associated with humidity and barometric pressure values. By incorporating this contextual knowledge, we can detect unreasonable or inconsistent combinations within the sensor data [130].

2.5.2 Model Structure

Fault detection techniques can also be classified based on the system architecture or organization of the fault detection process within a wireless sensor network and can classified into centralized, distributed and hybrid architecture [129].

- **Centralized Architecture:** In this architecture, the fault diagnosis process in the network is carried out by a single central node, which can either be a SN or a base station. This central node possesses superior computational power, storage capacity, and energy compared to the other sensor nodes in the network. It collects diagnostic information from the various sensor nodes within the network. Upon

receiving this data, the central node analyzes it and makes decisions regarding the fault status of the individual sensor nodes in the network. Figure 2.11 illustrates sensor fault diagnosis in centralized architecture.

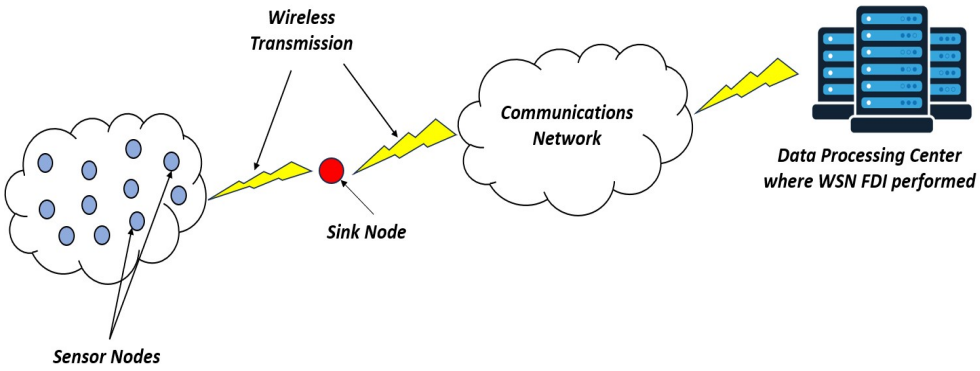


Fig. 2.11 Sensor fault diagnosis in centralized architecture

– **Distributed Architecture:** In a distributed architecture, the fault detection process is decentralized, and each sensor node in the network is responsible for its own fault detection. The sensors collect data and perform local analysis to detect faults. Communication between sensor nodes may occur to exchange information or coordinate the fault detection process.

– **Hybrid Architecture:** represents a synthesis of centralized and distributed methodologies, strategically designed to address the limitations observed in both individual approaches. Its primary objective is to optimize fault management within the network. This approach incorporates two crucial decision components: the SN and each sensor node. Employing a multi-tiered architecture, it organizes sensor nodes into cluster heads and cluster members. Within this framework, each cluster member transmits its diagnostic data to its respective cluster head. The cluster head then processes this information, ultimately determining the fault status of sensor nodes with the assistance of the base station.

2.5.3 Operation Mode

Fault detection techniques can be classified based on the type of data they analyze. Generally, there are two primary categories: those that utilize historical data and those that deal with real-time data. Historical data, sourced from files or databases, is typically available in its entirety or in large blocks. Conversely, real-time data, such as streaming data, is continuously generated and available as it happens. As a result, fault detection systems are commonly categorized as either online, where they process data in real-time, or offline, where they analyze historical data in batch mode.

- **Online Approaches:** Analyze the data in real-time by keeping monitoring ongoing events, (e.g., measurements). These methods are commonly referred to as streaming anomaly detection (SAD) since they handle incoming data as a continuous stream. As explained in [114], this processing can occur continuously, following the flow of data, or it can be based on discrete packets or batches of data.

- **Offline Approaches:** Conversely, analyze archived information to determine whether a fault has occurred. These offline detection methods typically integrate more advanced techniques and handle larger datasets.

2.6 Fault Detection and Isolation

Regardless of the FDI method employed, to optimize the utilization of information derived from sensors measurements, the FDI can be decomposed into the following three steps as shown in figure 2.12:

- **Generation of residuals:** In simpler terms, a residual is the difference between the measured output and the expected output. The magnitude of the residual signal obtained from the generator indicates whether a fault is present or not. Fault detection depends on evaluating these generated residuals.

- **Evaluation of the residuals:** Residuals are signals that provide information about the duration and occurrence of faults by comparing measurements to model calculations. These residuals are then compared to predefined limits to identify any deviations. Setting a threshold is crucial for detecting changes. During

normal operation, residuals are expected to be zero, but they deviate from zero when faults occur.

– **The decision:** It represents the final stage of the diagnostic process. This step enables the identification of faults, essentially pinpointing the root cause of anomalies within the network.

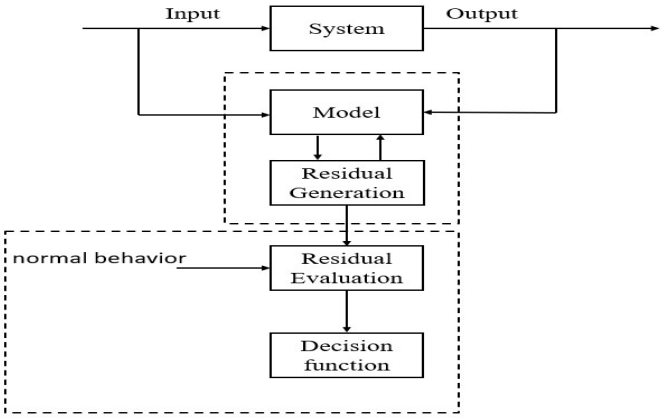


Fig. 2.12 Sensor fault detection and isolation process

2.7 Fault Diagnosis Challenges in Wireless Sensor Networks

Designing an appropriate fault detection technique becomes more challenging in sensor networks due to their specific nature and the characteristics of sensor data.

There are several reasons why conventional fault detection techniques may not be suitable for handling sensor data in WSN [130]:

– **Resource constraints:** pose a significant challenge in fault detection within WSNs. The sensor nodes used in WSNs are often low-cost and low-quality, which leads to stringent limitations in terms of resources such as energy, memory, computational capacity, and communication bandwidth. Traditional fault detection techniques have not adequately addressed the need for efficient resource utilization. These techniques tend to be computationally expensive and require

substantial memory for data analysis and storage. As a result, one of the key challenges in fault detection for WSNs is to minimize energy consumption while utilizing a reasonable amount of memory for storage and computational tasks.

- **High communication cost:** In WSN, the majority of the energy is consumed for radio communication rather than computation. Communication costs for sensor nodes are typically significantly higher than computation costs [4]. Consequently, a key challenge in fault detection for WSN lies in minimizing communication overhead to alleviate network traffic and extend the network's lifespan.

- **Distributed streaming data:** The sensor data in WSNs originates from various sources and can change dynamically. This means that the data streams may vary over time, making it challenging to analyze and detect faults effectively. Moreover, the underlying distribution of the streaming data, which describes how the data is spread or organized, is often unknown in advance. This lack of a priori knowledge adds complexity to fault detection tasks. Additionally, computing probabilities directly for analyzing the data can be difficult [38].
  Traditional fault detection techniques that rely on offline analysis of data may not be suitable for handling distributed stream data, as they typically assume a static dataset. Techniques that depend on prior knowledge of the data distribution are not well-suited for sensor data. Consequently, the challenge lies in finding ways to process distributed streaming data in real-time or online, taking into account the dynamic nature of the data and the absence of prior knowledge about its distribution.

- **The dynamic nature of network topology:** frequent communication failures, node mobility, and heterogeneity pose challenges for outlier detection in sensor networks. When deployed in unattended environments for extended periods, sensor networks are prone to experiencing changes in network topology and frequent communication failures. Furthermore, sensor nodes can relocate to different locations unpredictably and exhibit varying sensing and processing capabilities. They may even possess different types and quantities of sensors. These dynamic and heterogeneous characteristics add complexity to the design of effective outlier detection techniques for WSNs.

– **Large-scale deployments:** of sensor networks involve a significant number of sensor nodes, often reaching hundreds or even thousands. The primary challenge faced by traditional outlier detection techniques in such deployments is achieving a high detection rate while keeping the false alarm rate low. This necessitates the creation of a precise normal profile that accurately represents the expected behavior of sensor data [115]. Constructing such a profile becomes exceedingly difficult in the context of large-scale sensor network applications.

– **Identifying fault sources:** The sensor network is supposed to give us the basic information it picks up from the physical world and detects any events happening within the network. Detecting the cause of a fault in sensor data is challenging due to the limited resources and constantly changing environment of WSNs. Traditional methods for identifying outliers often fail to differentiate between errors and events, considering any unusual data point as an error, which leads to the loss of crucial hidden information about events. So, one of the challenges in detecting outliers in WSNs is figuring out where they come from and distinguishing between errors and events.

So, the key challenge for outlier detection techniques in WSNs is to ensure accurate results while keeping the resource usage of WSNs to a minimum

2.8 Limitations of Existing Approaches for Fault Detection in Wireless Sensor Networks

Despite significant advances, existing approaches for fault detection in WSNs face several limitations. Many fault detection algorithms require substantial energy for data processing and communication, rapidly depleting battery life [5]. Complex algorithms necessitating considerable computational resources may be unsuitable for resource-constrained sensor nodes, leading to inefficiencies or inability to execute on certain hardware [83]. Additionally, scalability issues arise as network size increases, resulting in performance degradation. Latency introduced by extensive data processing or communication is problematic for real-time applications. High false positive and false negative rates undermine reliability, and environmental noise and variations further contribute to inaccuracies, especially in dynamic and harsh environments. Methods heavily dependent on historical data can be problematic if such data is not

representative of current conditions or unavailable. Many approaches struggle to adapt to changing environmental conditions, affecting sensor readings and leading to incorrect fault detection. Algorithms assuming static network topologies may fail in dynamic environments where nodes frequently join, leave, or move. Poor data quality due to sensor noise, interference, or data corruption, and incomplete data from intermittent connectivity or node failures also pose challenges. Frequent calibration or maintenance required by some methods may not be practical for large-scale or remote deployments. Vulnerability to security threats such as spoofing, jamming, and denial-of-service attacks compromises the effectiveness of many fault detection approaches. Integrating fault detection mechanisms with existing WSN infrastructure and ensuring compatibility with various hardware and software platforms can be challenging. The cost of implementing and maintaining sophisticated fault detection systems may be prohibitive, particularly for large-scale or budget-constrained deployments. Machine learning-based approaches may require significant training time and may not adapt quickly to new fault types or changes in network behavior. Some methods introduce additional communication overhead for transmitting data required for fault detection, straining the network's bandwidth and increasing energy consumption. By acknowledging these limitations, this thesis aims to develop more efficient, robust, and scalable fault detection methods for WSNs.

2.9 Conclusion

In conclusion, fault detection in WSNs is a complex undertaking that requires a comprehensive understanding of the network architecture, data characteristics, and fault detection techniques. By leveraging data correlations, employing appropriate fault detection methods, and addressing the unique challenges posed by WSNs, it is possible to enhance the reliability and efficiency of these networks.

However, it is imperative to acknowledge the limitations and ongoing challenges in fault detection for fault diagnosis in WSNs. Conventional approaches may fall short in capturing fault-induced anomalies and may not always effectively leverage contextual information for accurate fault diagnosis. Moving forward, research efforts should focus on developing more robust fault detection techniques that incorporate spatial and temporal information, as well as node-level characteristics, to improve the reliability and effectiveness of fault diagnosis in WSNs. Through continued innovation

and collaboration, we can overcome these challenges and pave the way for more resilient and efficient WSNs in the future.

Chapter 3

Spatial Correlation-Based Fault Detection and Isolation for Smart Irrigation Systems in Wireless Sensor Networks

WSNs have become indispensable tools in various applications, including environmental monitoring, industrial automation, and precision agriculture. In the context of smart irrigation systems, WSNs play a crucial role in providing real-time data on soil moisture levels, temperature, and other environmental parameters essential for optimizing irrigation practices. However, the reliability of data collected by WSNs can be affected by sensor faults, which may arise due to hardware malfunctions, environmental factors, or communication errors.

Traditional fault detection approaches in WSNs often rely on hardware redundancy, where multiple sensor nodes are deployed to cross-validate data and detect inconsistencies. While effective, this approach comes with drawbacks such as increased hardware costs, higher energy consumption, and complexity in managing redundant nodes. Moreover, in large-scale deployments spanning vast agricultural landscapes, maintaining hardware redundancy becomes impractical and resource-intensive.

This chapter focuses on addressing these challenges, a novel approach to fault detection and isolation in WSNs develops, using correlation spatial rather than hardware

redundancy. This approach takes advantage of the inherent spatial correlations among sensor readings within the network. Fundamentally, neighboring sensor nodes in close proximity tend to exhibit similar measurements due to their shared environmental conditions. By analyzing these spatial correlations, it becomes possible to detect and isolate faulty sensors without the need for redundant hardware.

3.1 Soil Monitoring Background

The state of the soil plays a crucial role in agriculture, particularly in the context of smart irrigation systems. Effective monitoring of known parameters that characterize the soil is vital for smart irrigation systems, as it enables the precise determination of irrigation water quantities and optimization of crop production. The existing sensors designed for monitoring soil characteristics can utilize either the physical or chemical characteristics of the soil for measurements. Nevertheless, chemical sensors prove impractical for smart irrigation systems due to their dependency on human intervention for cleaning, calibration, and performing the measurement.
Therefore, an effective smart irrigation system typically incorporates sensors requiring minimal maintenance. Smart irrigation systems commonly concentrate on two key aspects for soil characterization.

The measurement of soil humidity serves to identify if crops are experiencing water stress due to insufficient water in the soil. High humidity levels can lead to root diseases and inefficient use of irrigation water. Conversely, low humidity may result in reduced crop yields and even plant death. Additionally, the water movement within the soil plays a crucial role in determining how nutrients reach the plants, whether through incorporation into irrigation water or other application methods.

Soil temperature experiments variations throughout the day and across seasons, influenced primarily by solar radiation and air temperature. Soil temperature plays a crucial role in shaping the chemical and physical characteristics of the soil. For instance, the soil's pH tends to rise within the temperature range of 25 to 39 °C [94]. Additionally, soil temperature impacts factors such as moisture content, salinity, soil structure, and aeration. The influence extends to the organisms and organic matter present in the soil. Notably, soil respiration increases between temperatures of 10 and 28 °C, micro-organisms and macro-organisms show heightened activity within the

range of 10 to 24 °C, and organic matter decomposes when temperatures range from 2 to 38 °C. Moreover, high soil temperatures lead to a decrease in the water uptake by plants, while low soil temperatures result in a reduction in nutrient uptake. Finally, root growth tends to improve under high soil temperatures, whereas low soil temperatures contribute to a decrease in root growth [39].

Measuring soil moisture and soil temperature only at the surface level is not sufficient, as plants require water to reach their root depth. The soil's moisture-holding capacity varies at different depths due to changes in soil type. For instance, in agricultural or park fields, the soil type may differ at various distances and depths, resulting in variations in the soil's water-holding capacity. As shown in figure 3.1, the soil has three distinct layers, topsoil, sub soil and bed rock [21] .

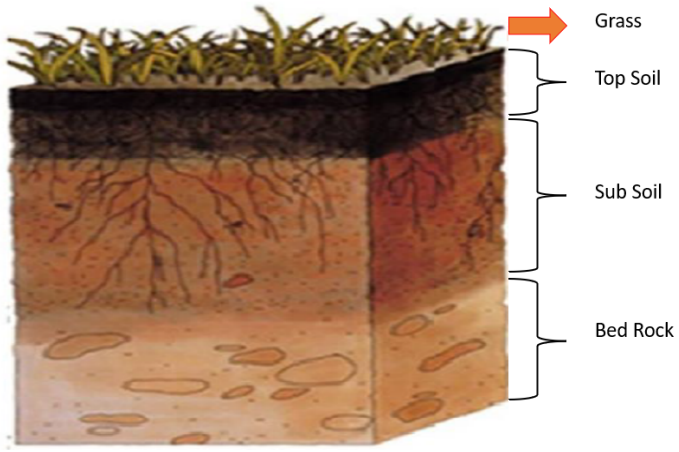


Fig. 3.1 Variance in soil type at different depths [21]

At the beginning of our study, it's really important to gather information to understand the land we want to monitor for smart irrigation. We do this in a few steps. First, we look at the geography, topography, and climate of the area to provide information about the size, location, and slope of the land, as well as weather conditions such as sunlight, temperature, wind, humidity, and pressure. this data can vary from one point to another within the same parcel.

Next, we check the soil by creating a soil map. This map helps us figure out different soil types and what crops can grow well there. We look at things like the soil's pH

levels, the condition of the ground, and if there are any harmful fungi present.

By studying these maps, we can divide the entire area into different zones. Each zone groups together places with similar properties in terms of geography, climate, and soil [30].

3.2 Methodology for Sensor Deployment

In designing the sensor deployment strategy for the smart irrigation system, several factors must be considered to ensure optimal data collection and fault detection capabilities. The methodology for sensor deployment involves the following steps:

1. **Zone Identification:** Begin by dividing the entire area into different zones based on similarities in geography, climate, and soil properties as shown in figure 3.2. This zoning helps in organizing the sensor deployment strategy effectively.
2. **Equivalence Zone Setup:** Within each equivalence zone, strategically install two sensor nodes. These nodes should be designed to measure the same soil parameter, such as soil temperature or soil moisture.
3. **Depth and Placement:** Ensure that both sensor nodes are placed at the same depth within the soil. This consistency in placement is crucial for accurate data collection and comparison.
4. **Redundant Measurement Strategy:** The deployment of two sensors measuring the same parameter at the same depth in each zone provides redundancy in measurements. This redundancy acts as a safeguard against potential sensor faults or inconsistencies.
5. **Calibration and Synchronization:** Before deployment, calibrate and synchronize the sensor nodes to ensure uniformity in measurements and data collection processes across all zones.
6. **Real-time Monitoring:** Implement real-time monitoring capabilities to continuously assess sensor data based on parity space analysis, enabling prompt identification and resolution of potential faults.

By incorporating these detailed steps into our methodology for sensor deployment, we can effectively leverage the parity space approach for fault detection and isolation in WSNs within our smart irrigation system [52].

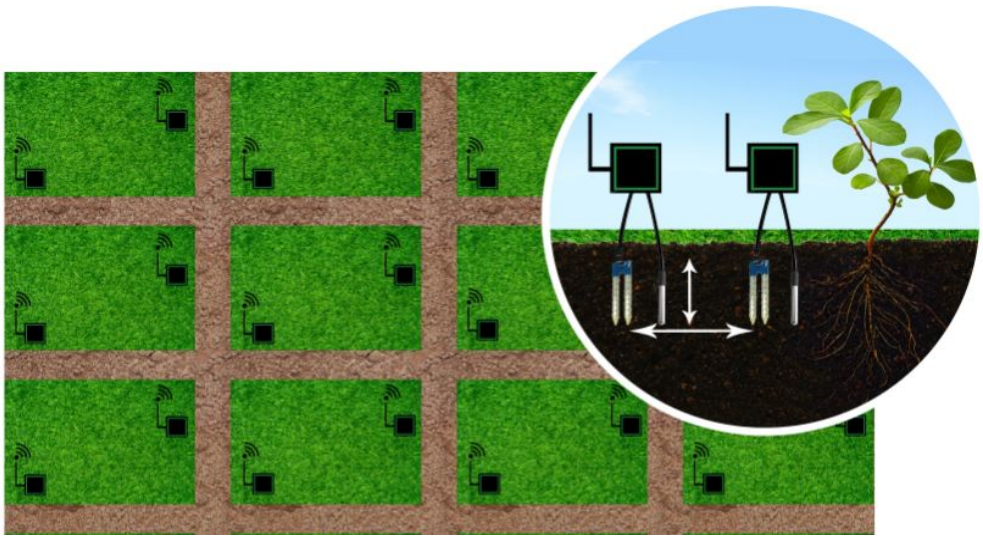


Fig. 3.2 Sensors nodes distribution

3.3 Parity Space

The parity space is a mathematical concept that is used in the field of FDI. The parity space refers to a space where its elements are represented by residuals, also known as parity vectors. In this context, the terms "residuals" and "parity vectors" are synonymous. The parity space can alternatively be termed a residual space. The equation that generates the residuals or parity vectors is called a parity relation or equation. In the context of FDI, the objective is to create a parity space and analyse its elements.

The residuals can be directly derived from redundant measurements when hardware redundancy is present. Let's consider the example of measuring an n-dimensional vector with m sensors. The measurement equation, in algebraic terms, is as follows [50]:

$$y(k) = Cx(k) + f(k) \tag{3.1}$$

Where $y(k)$ is an $mx1$ measurement vector, $x(k)$ is the $nx1$ state variable vector, C is an mxn measurement matrix, and $f(k)$ is the vector of sensor faults.
In this equation, the measurement vector $y(k)$ is a function of the state variable vector $x(k)$ and the measurement matrix C. The measurement matrix C maps the state variable vector $x(k)$ to the measurement vector $y(k)$. The vector of sensor faults $f(k)$ represents any errors or faults in the sensors that are used to measure the state variable vector

$x(k)$. By analyzing the residuals in the parity space, it is possible to detect and isolate faults in the system. This can help to improve the reliability and safety of the system.

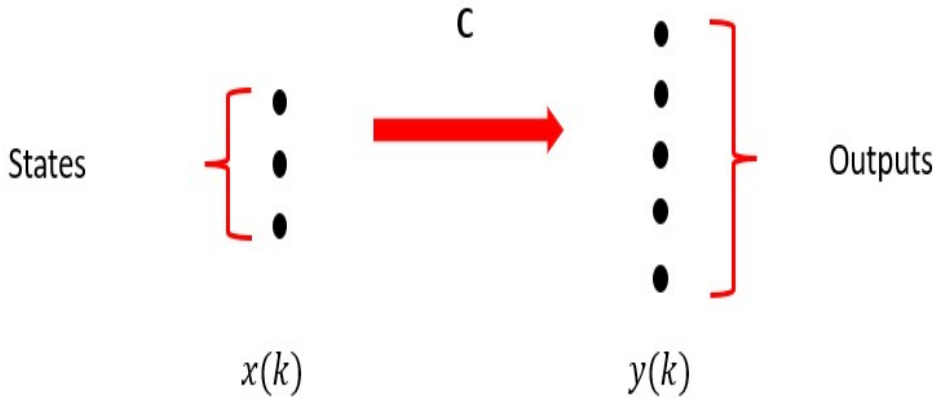


Fig. 3.3 Direct redundancy

Utilizing direct redundancy involves employing a greater number of sensors than the minimum required (e.g., two or more for scalar state variables and four or more for three-dimensional state variables), as depicted in figure 3.3. Consequently, in this case, the residual vector can be directly obtained by using the redundancy measurements. It is important to note that the dimension of $y(k)$ is greater than the dimension of $x(k)$.

$$\text{i.e} \qquad m > n \qquad \textit{and} \qquad \text{rank(C)=n} \tag{3.2}$$

For the purpose of FDI, the vector $y(k)$ can be assembled into a series of linearly independent parity equations, leading to the generation of the parity vector (residual):

$$r(k) = wy(k) \tag{3.3}$$

where r(k) is the $(m-n)x1$ residual vector which is also known as the parity vector. Ideally, the expectation is for the residual vector to have a zero value in the absence of faults.

$$r(k) = wy(k) = 0 \qquad \text{for the no-fault case} \tag{3.4}$$

This is known as a parity equation or parity relation. The $(m-n)$x m projection matrix W needs to be selected in a way that ensures the independence of the residual r from the system state x, achieving the following:

$$WC = 0 \tag{3.5}$$

This condition must be satisfied for the residuals to be null in the absence of faults and uncertainty. When this requirement is met, the residual becomes:

$$r(k) = w_1 f_1 + ... + w_m f_m \tag{3.6}$$

where w_i is the i^{th} column of W, f_i is the i^{th} element of $f(k)$.

Equation (3.6) indicates that the parity vector only has information about faults and is independent of state x(k). Figure 3.4 shows that the columns of W make up the parity space (or residual space), forming a kind of basis for this space. Another interesting thing is that if there is a problem with the i^{th} sensor, it makes the residual r grow in the direction of w_i. This means if there is an issue with the i^{th} measurement, it makes the norm of r bigger in the direction of w_i.

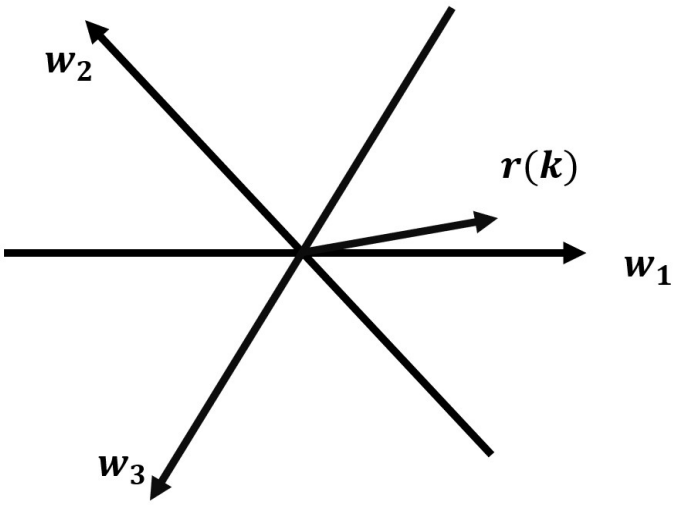


Fig. 3.4 Fault signature directions in the parity space

3.3.1 Fault Detection

In the context of fault detection, a decision function is used to determine whether a fault has occurred in the system. Using the notation presented in [25], a decision function is defined as:

$$DFD(k) = r(k)^T r(k) \tag{3.7}$$

If a fault occurs in the sensors, $DFD(k)$ will exceed a predefined threshold.

3.3.2 Fault Isolation

Fault isolation follows the fault detection and determines the location of fault. When a fault is detected, we need to locate the presence of the fault i.e. locate the faulty sensor. To isolate the faulty sensor we multiply the transpose of each column of the parity matrix (W) by the generated residual as the following expression:

$$FI_i(k) = w_i^T r(k) \qquad i = 1,2,...m. \tag{3.8}$$

For a given $r(k)$, a malfunctioning sensor is recognized by calculating the m values of $FI_i(k)$. If $FI_j(k)$ is the largest among these values, the sensor associated with $FI_j(k)$ is the one most likely to have experienced a fault.

From the perspective of parity space, the vectors represented by the columns of w establish m distinct directions for fault signatures (w_i, where $i = 1,2,...,m$). Once a fault is detected, it can be isolated by comparing the orientation of the parity vector with each of these signature directions. Furthermore, the fault isolation function $FI_i(k)$ quantifies the correlation between the residual vector and the fault signature directions [99].

3.4 Proposed FDI Algorithm

For each zone (i), we compute its residual. Two sensor nodes are used to measure the same quantity $x(k)$. The sensor outputs represent the state variable, with the measurement matrix C equal $[1 \quad 1]^T$. The mathematical model for each zone (i) is expressed

by the following equation [50]:

$$y_i(k) = \begin{bmatrix} y_i^1(k) \\ y_i^2(k) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} x(k) + \begin{bmatrix} f_i^1 \\ f_i^2 \end{bmatrix} \tag{3.9}$$

In zone (i), y_i^1 and y_i^2 represent the measurements from sensor 1 and sensor 2 respectively. Additionally, f_i^1 and f_i^2 denote the faults that could impact sensor node 1 and sensor node 2 respectively. As per the given relation, the parity vector w, orthogonal to C, is $W = [1 \quad -1]$, and the residual is calculated as follows:

$$r_i = y_i^1 - y_i^2 \tag{3.10}$$

To prevent false alarms, we have made adjustments to our residual. In the actual implementation, the residual value will not be zero during the normal operation of the sensor. Therefore, we define the fault detection threshold by a test. This means that the threshold is the maximum residual value that occurs during the healthy sensor operating mode [52].

Data: y_i^1, y_i^2 are the data received from sensor 1 and 2 respectively at an instant t to calculate the residual according to equation (4.10);

$r_i \leftarrow y_i^1 - y_i^2$;

if $|r_i| > h_d$ **then**

  /* if the residual higher than a threshold there is a fault detected and r_i stills in its value;

  $r_i \leftarrow y_i^1 - y_i^2$;

else

  /* if the residual lower than a threshold, r_i will take the value 0;

  $r_i \leftarrow 0$

end

/* if a fault is detected, fault isolation executed according to equation(3.8), The index of w_k that give the largest value in this multiplication $w_k^T r$, corresponds to the index of the faulty sensor. Where w_k is the k^{th} column of parity space matrix.

Algorithm 1: The proposed FDI algorithm

3.5 Laboratory Experiments

The effectiveness of the FDI algorithm has been validated through an experimental prototype depicted in figure 3.5. Monitoring soil temperature and soil moisture is crucial for regulating irrigation water levels [75]. In this experiment, two soil temperature sensors and two soil moisture sensors are embedded at the same depth within an iron frame containing soil [52]. An illustrative schematic of the experimental prototype is shown in figure 3.6.

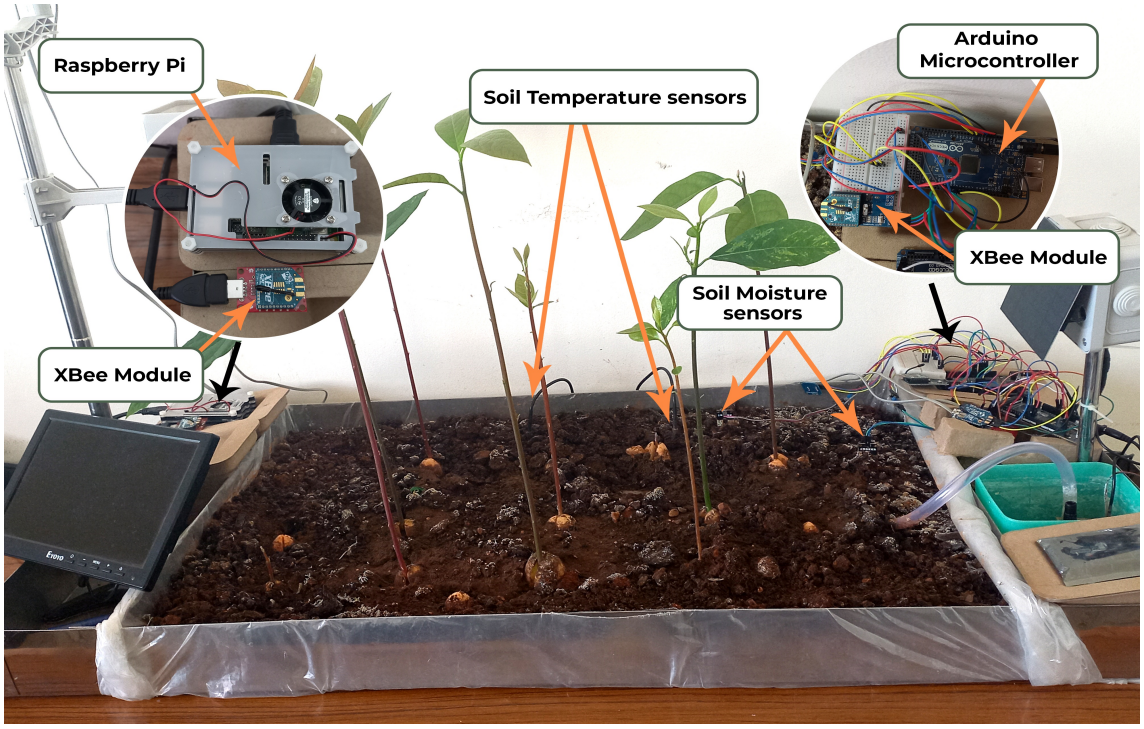


Fig. 3.5 Experimental prototype of the wireless sensor network deployed in soil for measuring soil temperature and soil moisture

The hardware used in this experimental prototype includes:

Soil Moisture Sensor:
The YL-69 soil moisture sensor has two probes that are inserted into the soil to measure its moisture content as shown in figure 3.6. The circuitry measures the resistance between the probes and converts it into a corresponding moisture level. Moist soil conducts electricity more effectively, resulting in lower resistance, while dry soil

exhibits higher resistance. The sensor includes output pins that can be connected to a microcontroller or other devices for data processing and analysis. YL-69 sensors typically operate within a voltage range of 3.3V to 5V, making them compatible with many microcontrollers and development boards.

Soil Temperature Sensor:
The DS18B20 is a digital temperature sensor belonging to the one-Wire family of sensors, which means it communicates with a microcontroller or other devices using a single data wire, as illustrated in the figure 3.6. The sensor is capable of measuring temperature in a range typically from -55°C to +125°C, making it suitable for a wide range of environments and applications, including soil temperature monitoring. It operates on a low power supply ranging from 3.0V to 5.5V, making it energy-efficient and suitable for battery-powered applications.

Data collected from the sensors are processed by an Arduino ADK. This microcontroller board, based on the ATmega2560, is specifically designed for interfacing with Android devices. Its USB host interface enables direct communication with Android smartphones and tablets, allowing for the development of Android-compatible hardware projects. Operating at 5V, the ADK provides 54 digital input/output pins, 16 analog inputs, and 4 UARTs, making it suitable for complex and interactive applications that integrate mobile devices.

The processed data is then transmitted to the local gateway via an XBee S2C Pro module, as illustrated in figure 3.6. This module, operating on the Zigbee protocol at 2.4 GHz, facilitates reliable, low-power wireless communication between the sensors and the gateway over short to medium distances. By integrating the XBee S2C Pro with the Arduino ADK and the soil moisture and temperature sensors, we enhance the system's flexibility and efficiency in collecting and managing field data for irrigation control.

The main gateway in this system is a Raspberry Pi, a credit-card sized, open-source computer board running Linux. Equipped with USB ports for peripherals like keyboards and mouse, and an HDMI port for video output, the Raspberry Pi, powered by the Broadcom BCM2835 chip, serves as the central hub for data processing and analysis.

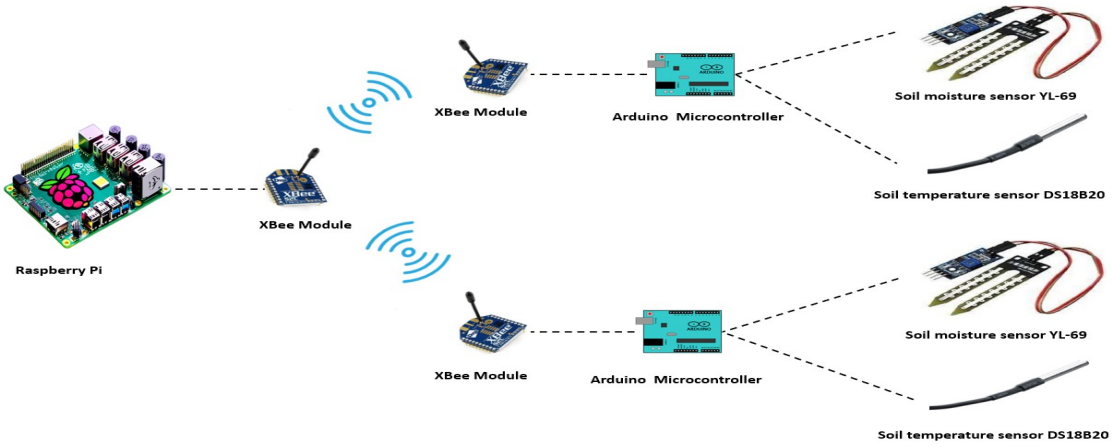


Fig. 3.6 Illustrative schematic of the experimental prototype for measuring soil temperature and soil moisture.

3.6 Results and Analysis

The sensors data are transmitted directly to the central processing unit for analysis. Our FDI algorithm is then applied to analyze the collected data. Before running the FDI algorithm, the sensors data are stored in their normal operating state for 5 hours with a sampling interval of ts=60s. This data collection process is illustrated in figure 3.7 for soil temperature and figure 3.8 for soil moisture. The purpose of this storage period is to establish fault detection thresholds based on the readings of the sensors in their healthy conditions [52].

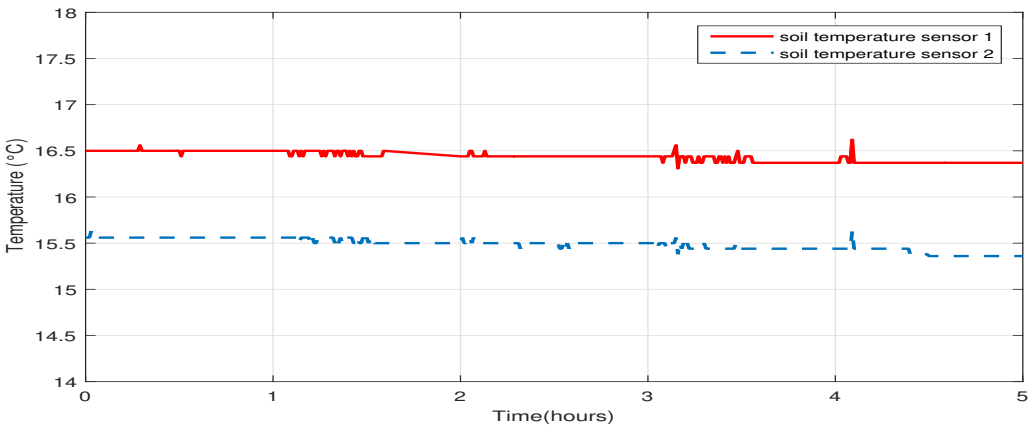


Fig. 3.7 Soil temperature recorded by the wireless sensor network in its healthy mode

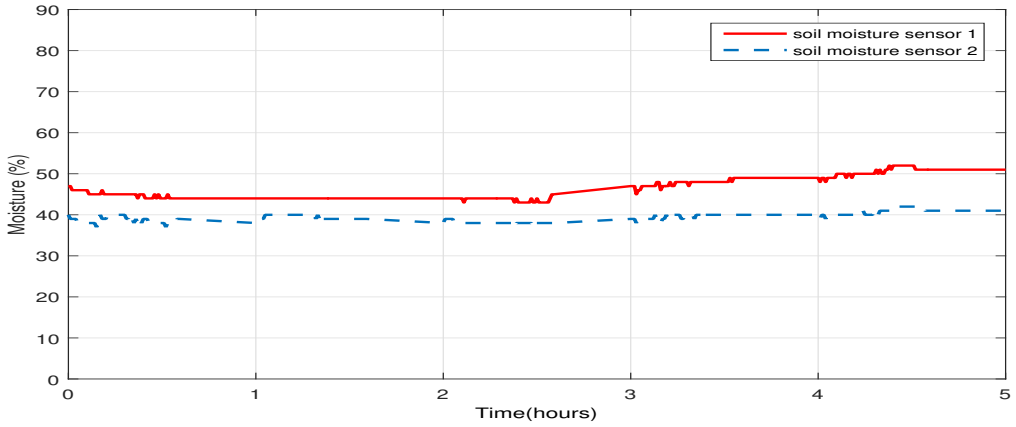


Fig. 3.8 Soil moisture measurement recorded by the wireless sensor network in its healthy mode

For each quantity $x(k)$, a residual is calculated in normal sensor operation scenarios, with the highest residual value serving as the fault detection threshold. In this study, the fault detection thresholds for soil temperature and soil moisture are set at 1°C and 11% respectively.

Following the threshold definition, abrupt sensors faults are introduced into each sensor, and a FDI algorithm is employed to detect and isolate these faults. In the initial test, sensors faults are introduced into soil temperature sensor 1 and soil moisture sensor 1 at t=2 hours. The evolution of residuals for soil temperature and soil moisture is depicted in figure 3.9 and figure 3.10. Notably, The introduction of faults in both sensors causes deviations in both residuals from zero. Our FDI algorithm promptly detectes these sensor faults immediately upon their occurrence.

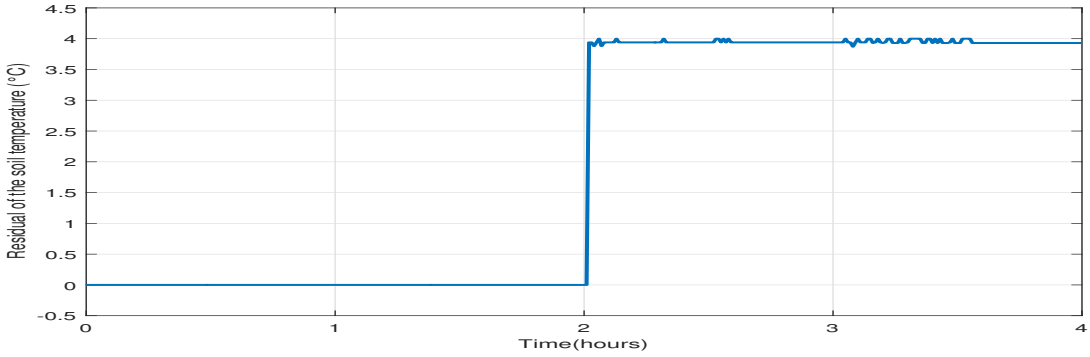


Fig. 3.9 Residual evolution of soil temperature

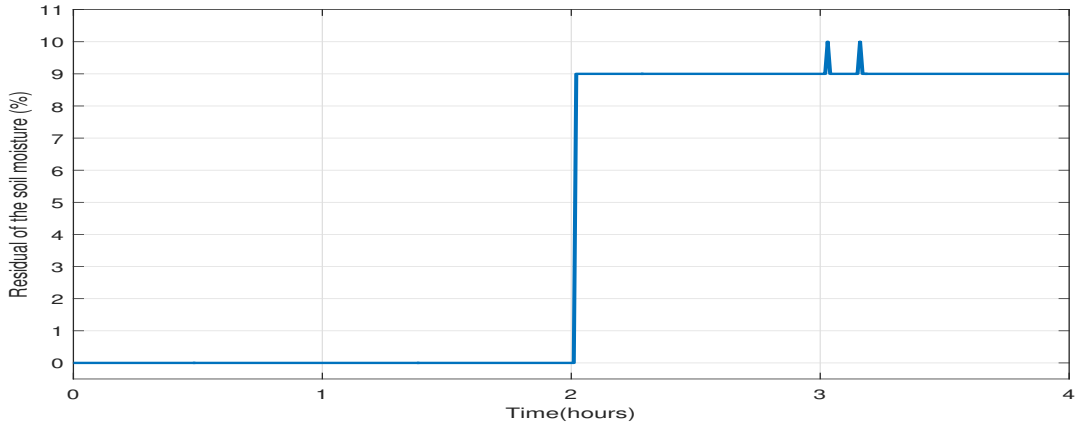


Fig. 3.10 Residual evolution of soil moisture

Following fault detection, our algorithm assesses the correlation between each residual and the columns of its respective parity matrix to isolate the faulty sensor, as detailed in subsection 3.3.2. In figure 3.11, FI_1 represents the correlation of the soil temperature residual with the first column of the parity matrix, while FI_2 corresponds to the correlation with the second column. And as can be seen, after fault detection (t=2 hours), FI_1 exhibits a higher magnitude than FI_2, indicating that the faulty sensor is soil temperature sensor 1. A similar analysis is conducted for soil moisture sensors in figure 3.12, where after fault detection, FI_1 displays a greater magnitude than FI_2, identifying the faulty sensor as soil moisture sensor 1.

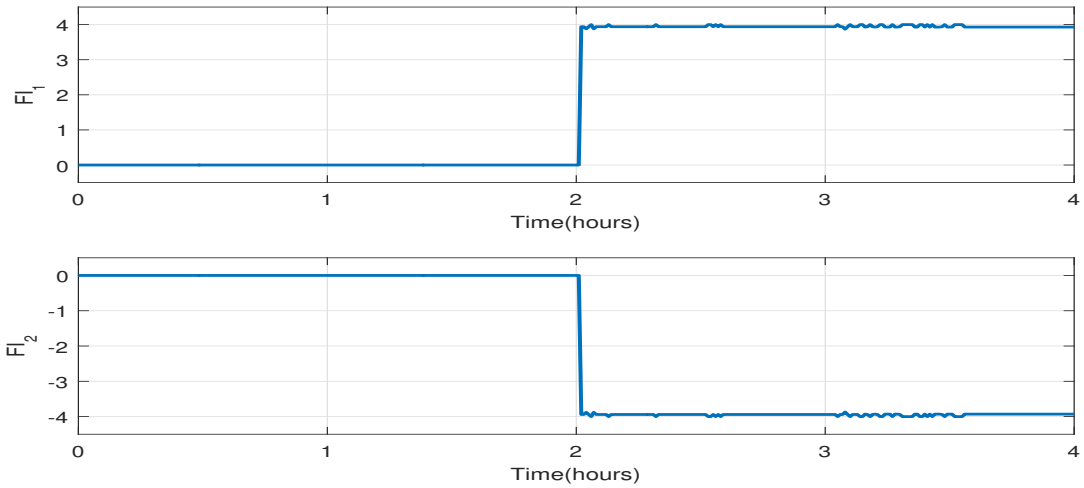


Fig. 3.11 Fault isolation process of soil temperature

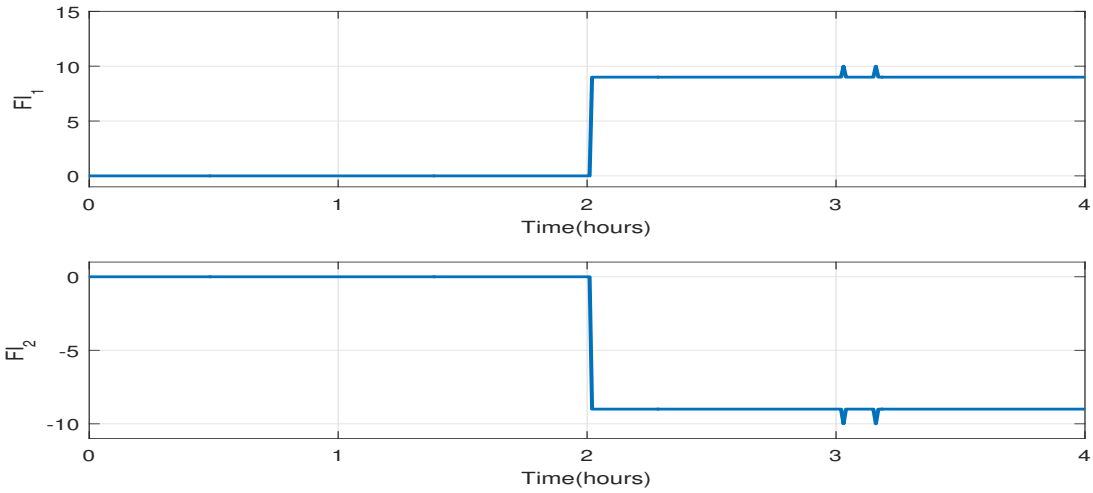


Fig. 3.12 Fault isolation process of soil moisture

During the second experiment, sensor faults were introduced into soil temperature sensor 2 and soil moisture sensor 2 at t=2 hours. The evolution of residuals for soil temperature and soil moisture is depicted in figure 3.13 and figure 3.14 respectively. Following the fault injection, both residuals exhibited deviations from zero, indicating the successful detection of faults within the sensor network.

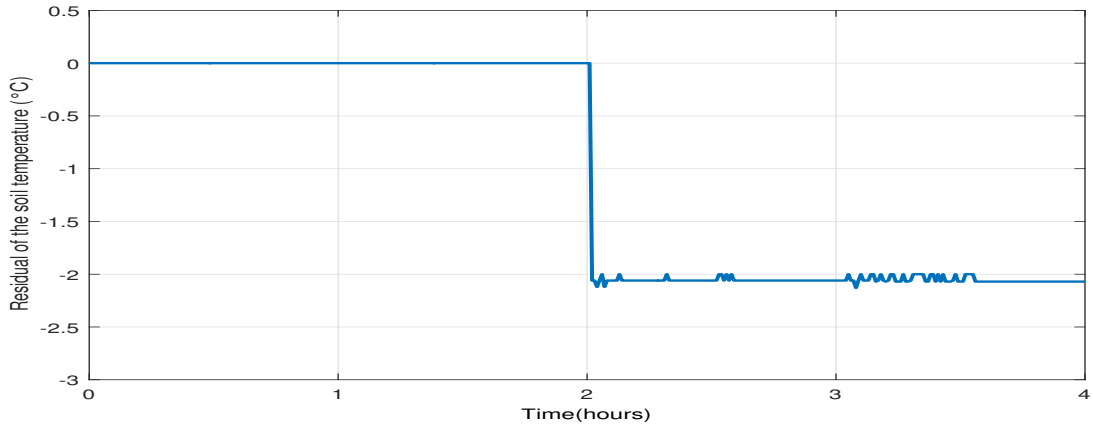


Fig. 3.13 Residual evolution of soil temperature

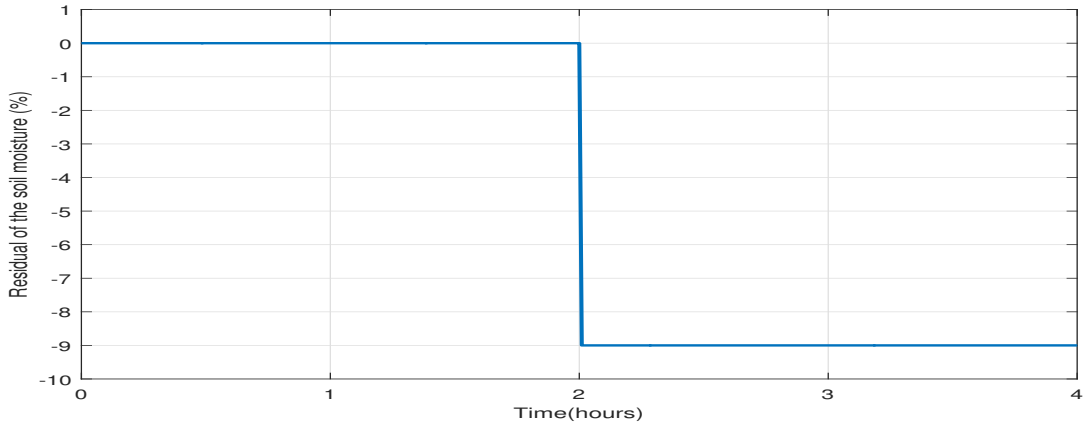


Fig. 3.14 Residual evolution of soil moisture

After detecting a fault, our algorithm assesses the correlation between each residual and its respective parity matrix columns to isolate the faulty sensor. In figure 3.15, FI_2 displayed a greater magnitude than FI_1, indicating that the faulty sensor is soil temperature sensor 2. Similarly, in the analysis of soil moisture sensors shown in figure 3.16, after fault detection, FI_2 exhibited a higher magnitude than FI_1, identifying the faulty sensor as soil moisture sensor 2.

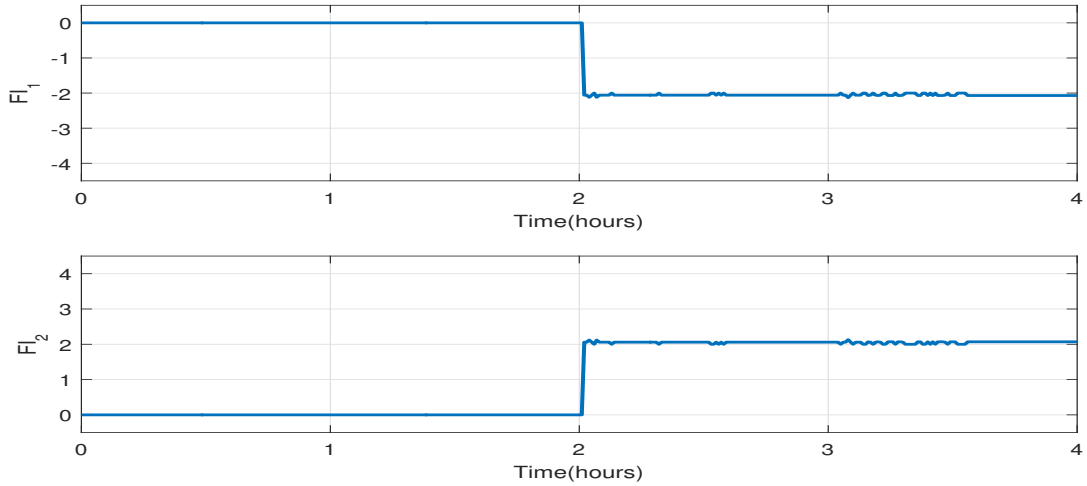


Fig. 3.15 Fault isolation process of soil temperature

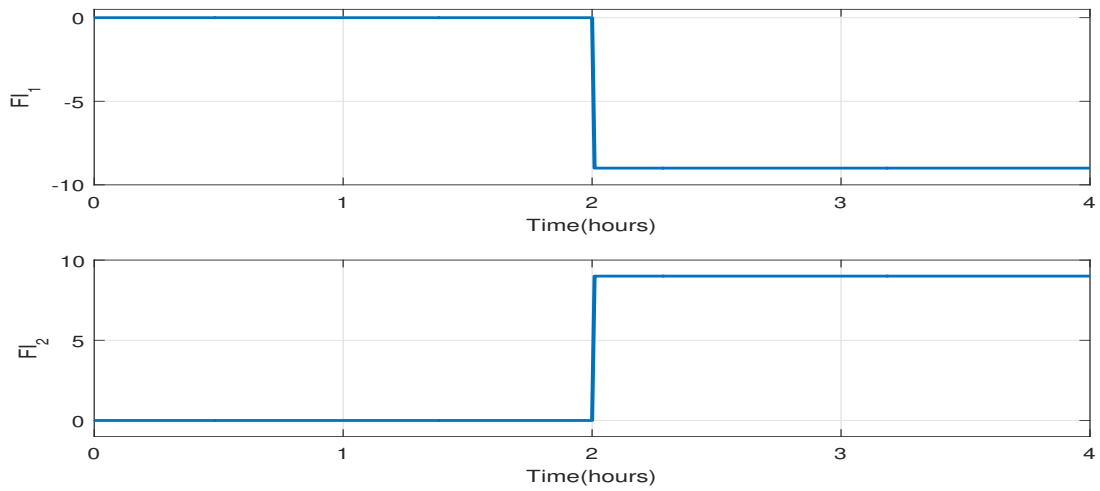


Fig. 3.16 Fault isolation process of soil moisture

3.7 Conclusion

In conclusion, this chapter has explored a novel approach to FDI in WSNs, focusing specifically on its application within smart irrigation systems. Traditional fault detection methods often rely on hardware redundancy, which can be costly and impractical, particularly in large-scale deployments.

The proposed methodology leverages spatial correlation among sensor readings to detect and isolate faulty sensors without the need for redundant hardware. By analyzing the inherent spatial correlations within the network, it becomes possible to identify sensor faults efficiently and effectively.

The chapter further delved into the significance of soil monitoring in agriculture, particularly in the context of smart irrigation systems. It highlighted the importance of measuring soil moisture and soil temperature for optimizing irrigation practices and maximizing crop production.

Moreover, the methodology for sensor deployment was outlined, emphasizing the importance of strategic placement, redundancy, calibration, and synchronization to ensure accurate data collection and fault detection capabilities.

The concept of parity space was introduced as a fundamental aspect of fault detection and isolation algorithms, providing insights into how residuals and parity vectors are utilized to detect and isolate sensor faults.

Laboratory experiments were conducted to validate the proposed fault detection and isolation algorithm. The results demonstrated the effectiveness and efficiency of the algorithm in promptly detecting and isolating sensor faults, thus ensuring the reliability and robustness of smart irrigation systems.

Chapter 4

Sensor Fault Detection in Wireless Sensor Network for Smart Irrigation Application using Kalman Filter

Soil moisture and soil temperature are fundamental parameters that significantly influence various agricultural and environmental processes [79]. Accurate and reliable measurements of these variables are crucial for understanding soil dynamics, optimizing agricultural practices, and predicting environmental changes. However, the integrity of such data can be compromised by sensor faults, leading to erroneous conclusions and suboptimal decision-making.

In this chapter, we explore an innovative approach to sensor fault detection in WSN for soil monitoring systems, focusing on both soil moisture and soil temperature measurements. Our investigation revolves around exploiting the concept of temporal dependency in the time-series data of these variables to enhance fault detection capabilities.

Temporal dependency refers to the inherent relationship between successive measurements of a variable made at regular time intervals. In the context of soil monitoring, this temporal structure manifests as patterns or trends in the data over time. By leveraging this temporal context, we aim to develop robust analytical techniques capable of identifying anomalies indicative of sensor faults.

Our methodology centers on the integration of two powerful tools: the Kalman filter and autoregressive modeling. The Kalman filter, renowned for its ability to estimate the state of a dynamic system from noisy measurements, serves as the cornerstone

of our fault detection framework. By iteratively updating its estimates based on new observations and system dynamics, the Kalman filter provides a principled approach to distinguishing signals from noise and detecting faults in the data stream.

Complementing the Kalman filter, autoregressive modeling captures the temporal dependencies inherent in soil variables' dynamics. By modeling the relationship between current observations and past values of the variables, autoregressive models offer insights into the underlying temporal patterns of soil moisture and soil temperature fluctuations. This temporal context enriches the fault detection process, enabling the identification of anomalies that deviate from expected temporal behaviors.

Through the fusion of Kalman filtering and autoregressive modeling, we present a comprehensive methodology for sensor fault detection in WSN for soil monitoring systems, encompassing soil moisture and soil temperature measurements. By exploiting the notion of temporal dependency, we demonstrate how our approach enhances the reliability and accuracy of soil data, paving the way for more informed decision-making in agriculture, environmental management.

Finally, the chapter describes the practical implementation of the sensor fault detection algorithm within the framework of the IrrigSys project. This part includes a comprehensive overview of the system components, the hardware setup, the communication protocols, the network infrastructure, and the user interface developed for real-time monitoring. The experimental results underscore the algorithm's effectiveness in maintaining data accuracy and ensuring the reliability of the system.
By addressing sensor faults proactively, our approach ensures that smart irrigation systems operate optimally, contributing to the sustainability and productivity of modern agriculture.

4.1 Sensor Readings Prediction

Monitoring soil conditions, including variables like soil moisture and soil temperature, is critical for efficient water management. Accurate data from sensor nodes responsible for monitoring these parameters is indispensable for effective control of water usage in agricultural settings. To ensure data accuracy, we have devised an algorithm specifically tailored for analyzing the outputs from distributed sensor nodes, aiming to identify any potential faults.

In our observation of collected data, we have noted that soil moisture and soil temperature readings exhibit a non-random pattern and change gradually over time[58]. This suggests a temporal dependency among successive readings from the sensors [109]. To address this, we've chosen to utilize the Kalman filter algorithm for predicting sensor readings under normal operating conditions. The Kalman filter employs an Autoregressive (AR) model to construct the state transition process, taking into account the temporal dependencies among readings.

It's important to note that our algorithm assumes fault-free operation during its development phase. The core of our approach is depicted in Figure 4.1, showcasing the proposed sensor data prediction diagram [51].

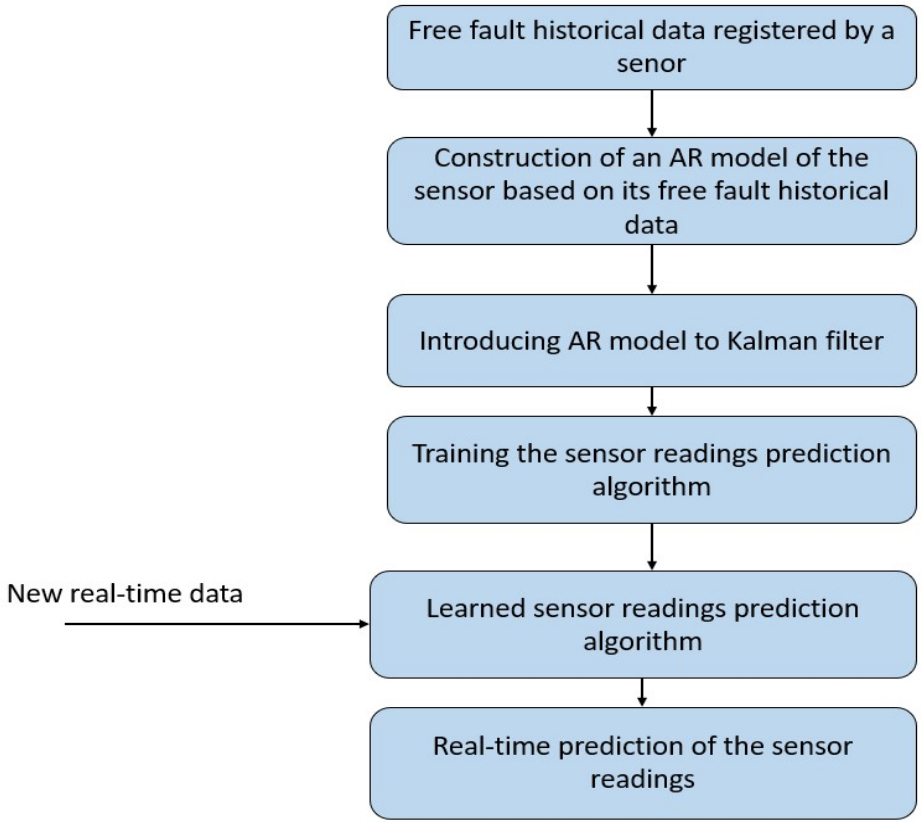


Fig. 4.1 Flowchart of the proposed scheme for Sensor readings prediction

4.2 Sensor Fault Detection using Kalman Filter

In this section, the fault detection scheme is presented, leveraging the advanced capabilities of the Kalman filter and an autoregressive model. The Kalman filter, combined with the AR model, serves as a powerful tool for state estimation in dynamic systems, offering precise predictions of sensor outputs while effectively dealing with measurement noise and uncertainties[51].
Central to the fault detection scheme is the comparison between predicted and measured sensor outputs as shown in Figure 4.2. In normal operating conditions, where sensor function correctly, the measurements closely align with the predicted values. However, in the presence of sensor faults, deviations between measured and predicted outputs become apparent. These deviations manifest as significant gap, exceeding predefined thresholds, triggering the fault detection mechanism.
The fault detection algorithm is seamlessly integrated into the WSN infrastructure, enabling continuous monitoring of sensor performance. Real-time feedback mechanisms facilitate proactive identification of sensor anomalies. Additionally, the fault detection scheme operates autonomously within the network, minimizing the need for manual intervention and ensuring robust performance in diverse environmental conditions.

To further understand the implementation and effectiveness of this scheme, the next subsections delve into the details of the AR model and the Kalman filter, elucidating their roles in enhancing fault detection capabilities.

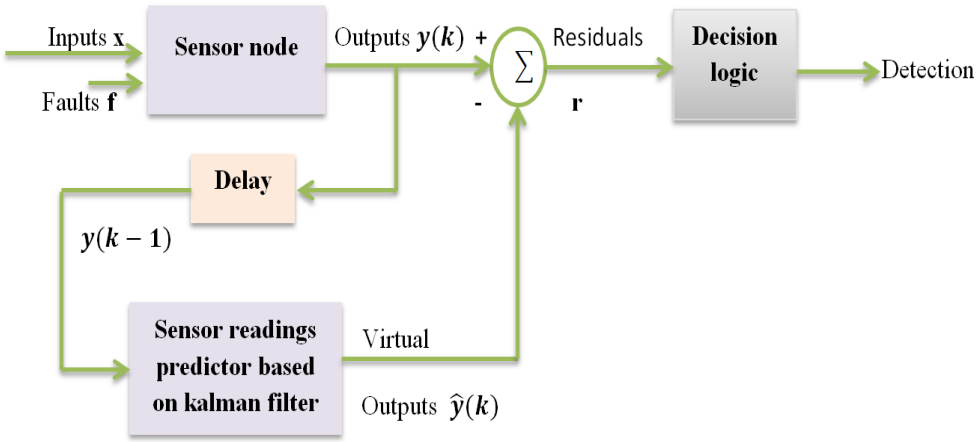


Fig. 4.2 Structural concept of the sensor fault detection

4.2.1 Autoregressive Model

An autoregressive model, often abbreviated as AR model, is a type of time series model used in statistics and econometrics. It's a model that predicts the next step in a sequence based on a linear combination of past observations.
In an AR model of order p, denoted as $AR(p)$, the value of the time series at a certain point depends linearly on its own previous values up to p time steps. Mathematically, an $AR(p)$ model can be expressed as:

$$X_t = C + \sum_{i=1}^{p} \phi_i X_{t-i} + \varepsilon_t \tag{4.1}$$

Where:

- X_t is the value of the time series at time t.
- C is a constant.
- $\phi_1, \phi_2, ..., \phi_p$ are the parameters of the model, representing the relationship between the current value and the previous values (detailed in Appendix A).
- ε_t is white noise, representing random error or disturbances at time t. The order p of the autoregressive model determines how many previous observations are used to predict the current value. Higher-order AR models consider more past observations and can capture more complex patterns in the data but may also be more difficult to estimate [18].

AR models are widely used in various fields such as finance, economics, signal processing, and climate modeling, among others, to analyze and forecast time series data.

4.2.2 Kalman Filter

The Kalman filter is a highly effective recursive algorithm used to accurately estimate the current state of a linear dynamic system based on a sequence of observed, possibly noisy, or inaccurate measurements. Kalman filter can also function as a predictor of future state values. The Kalman filter has been extensively used in engineering applications, such as control systems and target tracking, due to its straightforward form and minimal computational requirements. There exist some excellent literatures such

as [113] provide detailed explanations of the derivation and theoretical foundations of the Kalman filter. The filter operates by predicting the state $x \in \mathbb{R}^n$ of a discrete-time process represented by a linear stochastic difference equation:

$$x(k) = Ax(k-1) + W(k) \tag{4.2}$$

Where $x(k)$ is the predicted state at time instant k, A is the state transition matrix that relays $x(k-1)$ to $x(k)$, and $W(k)$ is the process noise that is assumed to be Gaussian with zero mean and covariance Q i.e $W(k) \sim \mathcal{N}(0, Q)$. The state transition process model is paired with a measurement equation that enables to observe the state and can be described as follows:

$$y(k) = Hx(k) + V(k) \tag{4.3}$$

Where $y(k) \in \mathbb{R}^m$ is the measurement vector at time k, H is the measurement matrix, and $V(k)$ is the measurement noise vector that is assumed to be Gaussian with zero mean and covariance R, i.e $V(k) \sim \mathcal{N}(0, R)$. Note that in different sources, the term "measurement" may also be denoted as "observation."

The Kalman filter plays a crucial role in generating estimates of $x(k)$ at time k, based on the initial estimate of $x(0)$, a series of measurements $y(1), y(2), ..., y(k)$, and the system information described by matrices A, H, Q and R. Note that the subscripts for these matrices are omitted here, assuming they remain constant over time, which is typical in most applications. While the covariance matrices are intended to reflect the noise statistics, in many practical scenarios, the true statistics of the noise are either unknown or non-Gaussian. Consequently, Q and R serve as tuning parameters, allowing users to adjust them to achieve desired performance levels.

The Kalman filter consists of two steps: the prediction and the update.

In the prediction step, the state estimate of the current time step $\hat{x}^-(k)$ is predicted based on the updated state estimate of the previous time step, and the predicted error covariance $P^-(k)$ is also computed in this prediction step. These are defined in equations 4.4 and 4.5, respectively.

In the update step, the Kalman gain $K_g(k)$ is computed first. Then, the updated state estimate $\hat{x}^+(k)$ is calculated by using the current measurement to correct the predicted state estimate, providing a more accurate state estimate for the current time step. After that, the Kalman filter calculates the updated error covariance $P^+(k)$.

In the following equations, the hat operator " ˆ " denotes an estimate of the state. Where, the superscripts "-" and "+" indicate predicted and updated state estimate, respectively.

Prediction:

Predicted state estimate:

$$\hat{x}^-(k|k-1) = A\hat{x}^+(k-1|k-1) \tag{4.4}$$

Predicted error covariance:

$$P^-(k|k-1) = AP^+(k-1|k-1)A^T + Q \tag{4.5}$$

Update:

Kalman gain :

$$K_g(k) = P^-(k|k)H^T[HP^-(k|k)H^T + R]^{-1} \tag{4.6}$$

Updated state estimate:

$$\hat{x}^+(k|k) = \hat{x}^-(k|k-1) + k_g(Y(k) - H\hat{x}^-(k|k-1)) \tag{4.7}$$

Updated error covariance:

$$P^+(k|k) = (1 - k_g H)P^-(k|k-1) \tag{4.8}$$

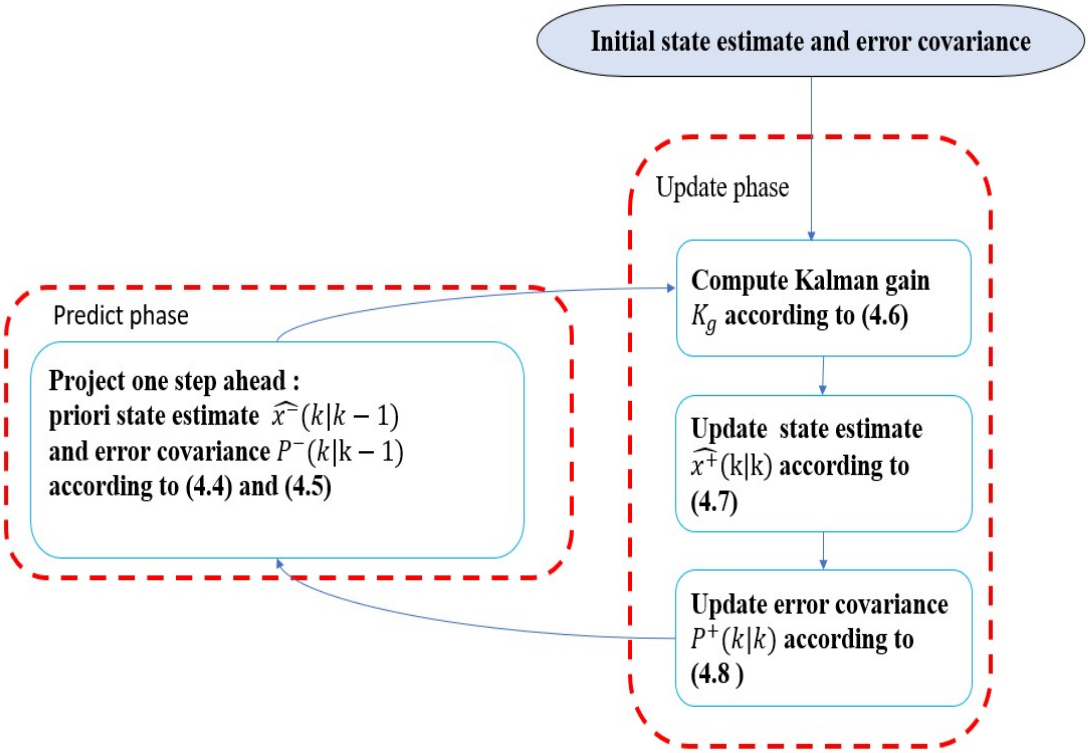


Fig. 4.3 Framework of the Kalman filter algorithm.

4.2.3 Residual Generation and Evaluation

In fault detection and diagnosis systems, the residual signal plays a crucial role as a fault indicator.
The residual signal, denoted as $r(k)$, is defined in equation 4.9 as the difference between the actual measurement $y(k)$ and its predicted value $\hat{y}(k)$ at time step k:

$$r(k) = y(k) - \hat{y}(k) \tag{4.9}$$

Where $r(k)$, $y(k)$, and $\hat{y}(k)$ are the residual vector, sensor measurement and its predicted output at time step k, respectively.

Under normal operating conditions, where the system is assumed to be fault-free and the predictions are accurate, the residual signal $r(k)$ should ideally be close to zero. This indicates that the actual measurements align closely with the predicted values.

However, if a fault occurs in the system, the residual signal will deviate from zero and exceed a predetermined threshold value. This deviation will indicate that there is a fault in the system [125].

The threshold value is set based on system characteristics and performance requirements. When the residual $r(k)$ exceeds this threshold, it serves as a clear indication that a fault or anomaly has occurred within the system being monitored.

In practical applications of fault detection and diagnosis systems, engineers use this residual signal $r(k)$ along with predefined thresholds to trigger alarms or initiate corrective actions.

4.2.4 Setting a Threshold

Following residual generation, fault detection relies on residual evaluation. Ideally, in the absence of faults, residuals should hover around zero, indicating a zero mean.Any deviation from this, termed a biased residual, indicates a fault. However, even under normal conditions, residuals may deviate from zero due to modeling inaccuracies, measurement noise, or external disturbances. Therefore, a non-zero threshold must be established, beyond which residual deviations suggest a potential fault.

Various methods exist to establish this threshold, either adaptively or with a fixed value. For instance, in [108], the threshold dynamically adjusts using recent data within a sliding window. Conversely, when residuals exhibit sufficient smoothness and perturbations are negligible, simpler approaches suffice. For example, a static threshold may be set, triggering a fault detection when surpassed.

Setting an appropriate threshold is paramount to swiftly detecting true faults while minimizing delays. Inaccurate thresholding can lead to false alarms caused by noise in the system. Conversely, overly conservative thresholds prolong detection times and reduce sensitivity. Thus, the threshold must strike a balance to ensure a desired Fault Detection Rate (FDR) or an acceptable False Alarm Rate (FAR) [2]. In this study, FAR is pegged at 1%, a widely accepted threshold. The threshold is determined based on

the statistical properties of the residual signal, as outlined in equation 4.10.

$$r_{th} = \mu + \rho\sigma \tag{4.10}$$

Where μ and σ are the mean and standard deviation of the residual respectively given in equations 4.11 and 4.12. ρ is the tolerance parameter that affects the false alarm probability. It is found that $\rho = 2.6$ works well in practice [40].

$$\mu = \frac{1}{m}\sum_{i=1}^{m}|r_i| \tag{4.11}$$

$$\sigma = \frac{1}{m}\sum_{i=1}^{m}(|r_i| - \mu)^2 \tag{4.12}$$

Where r_i is the residual that is generated under no sensor fault at time i

4.3 Experiments

Implementing a fault detection algorithm within a real-world application is a crucial step towards validating its efficacy and practical utility. In this section, we discuss the experimental implementation of our fault detection approach within the context of IrrigSys, a project constructed by our laboratory team to enhance water management in agricultural practices.

4.3.1 System Overview

IrrigSys is a comprehensive water management system designed to maximize efficiency and sustainability in agriculture. Developed collaboratively by our laboratory team, it comprises various interconnected components, including photovoltaic modules, weather stations, soil sensors, motor-pumping units, end devices, main gateways, cloud computing infrastructure, and application servers [44]. Together, these components work synergistically to optimize irrigation practices and promote effective water resource utilization [43].

Initially, we describe each component of IrrigSys and its role in contributing to the system's functionality:

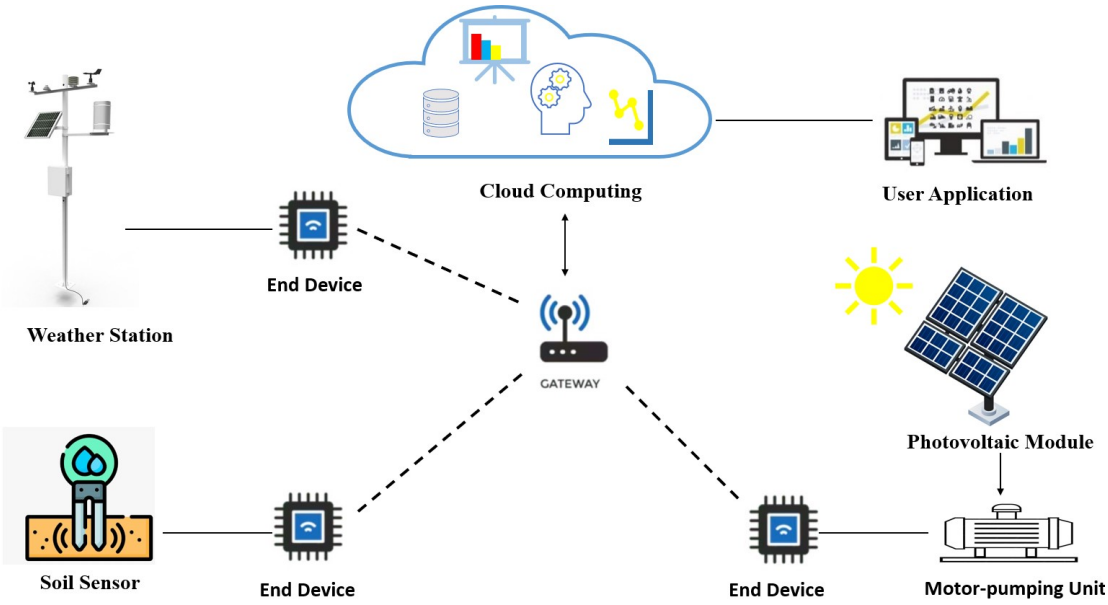


Fig. 4.4 Schematic of IrrigSys system

Photovoltaic Modules: Strategically positioned to harness solar energy and power various system components, reducing reliance on traditional energy sources.

Weather Station: Equipped with sensors to monitor meteorological parameters, providing real-time weather data essential for optimizing irrigation schedules.

Soil Sensors: Are devices strategically placed across agricultural fields to monitor various soil properties and moisture levels. These sensors play a crucial role in facilitating precise irrigation management by providing real-time data on soil conditions, allowing farmers to make informed decisions regarding watering schedules and optimizing crop growth.

Motor-Pumping Unit: Responsible for extracting and delivering water to the irrigation system, operating efficiently using electricity generated by photovoltaic modules.

End Devices: Local hubs for communication and control within specific field sections, facilitating localized monitoring and control of irrigation processes.

Main Gateway: Central hub for communication and coordination between system components, ensuring reliable data exchange and seamless operation.

Cloud Computing: Provides a platform for data storage, processing, and analysis, enabling advanced optimization of irrigation schedules and control instructions.

Application Server: Interface for end-users, such as farmers or system operators, to monitor and control the system remotely, empowering informed decision-making for efficient water management [43].

After thoroughly describing the components and functionality of the optimized irrigation system based on photovoltaic pumping, the next crucial step is to validate our algorithm practicality through its implementation.

4.3.2 Real-World Application of Fault Detection System

In our experimental setup, our objective is to seamlessly incorporate our fault detection algorithm into IrrigSys, thereby augmenting its capacity to detect and promptly address sensor malfunctions in real-time. Through this integration, we aim to enhance the system's reliability and efficiency, consequently achieving the goals of our laboratory's research.

This section outlines the practical implementation of our proposed fault detection system. Within our real prototype situated in the FSDM, an IoT-enabled Wireless Sensor Network (WSN) consisting of two sensor nodes is deployed directly into the soil, as depicted in Figure 4.5. For this experiment, we utilize a soil moisture sensor (ECH2O EC-5) and a soil temperature sensor (DS18B20). Each sensor is connected in series with a RAK811 WisNode LoRa Module [51].

The sensor nodes seamlessly transmit their measurements via the RAK811 WisNode LoRa Module to the main gateway. This gateway serves as a central hub, storing the received data in The Things Network (TTN) server. Subsequently, this data can be accessed by end-users through an application layer, facilitating real-time monitoring and analysis. Importantly, the real-time sensor fault detection mechanism is integrated into the user application using Node-Red.

With the real prototype in operation, we possess the capability to wirelessly receive data and perform comprehensive evaluations, as illustrated in Figure 4.6 and 4.7.



Fig. 4.5 The plot for typical installation at the faculty [43]

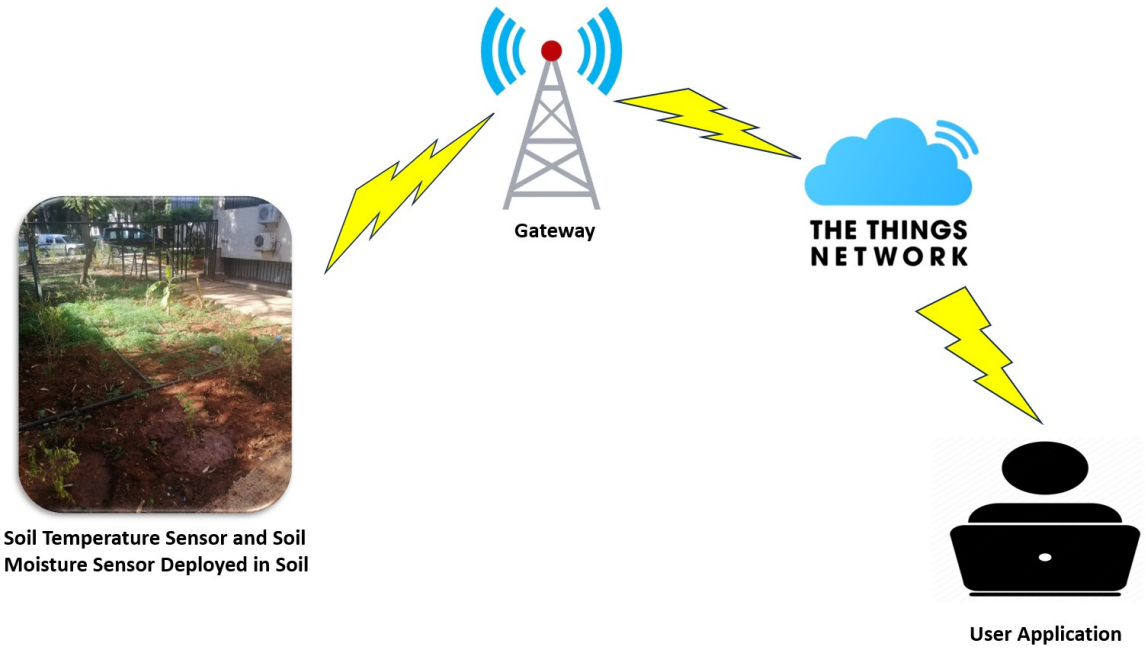


Fig. 4.6 Illustrative scheme of our laboratory experiment

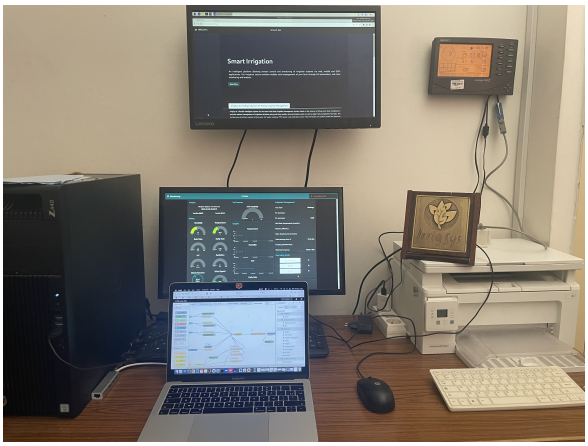


Fig. 4.7 Evaluation of data acquisition, transmission, and processing [43]

4.3.3 Hardware

Soil Moisture Sensor: The Decagon ECH2O EC-5 is a highly regarded soil moisture sensor used in various agricultural and environmental monitoring applications. This sensor is designed to provide accurate and reliable measurements of volumetric water content in soil, which is essential for understanding soil moisture dynamics and optimizing irrigation practices. One of the key features of the ECH2O EC-5 is its compact design and ease of integration into different systems. The sensor operates by measuring the dielectric constant of the soil, which correlates with the soil's water content. The EC-5 sensor is known for its robustness and durability, making it suitable for long-term field deployment in various environmental conditions. Its low power consumption is particularly advantageous for WSNs and remote monitoring applications, where energy efficiency is crucial. This sensor is commonly used in smart irrigation systems, research on soil-plant interactions, and other environmental monitoring projects. It helps researchers and practitioners monitor soil moisture levels with high precision, enabling better water management and contributing to sustainable agricultural practices. In Appendix B a table serves as a sensor manual, summarizing the essential informations to operate the device effectively.

Before deployment, we carefully calibrated the sensor to ensure its accuracy and reliability. Calibration involves configuring the sensor to match the soil properties and crop characteristics of our agricultural site. By ensuring accurate calibration before deployment, users can have confidence in the reliability and accuracy of soil moisture measurements obtained from the sensors.

We installed the sensor in the soil at a specific depth, strategically placed to measure moisture levels in the desired root zone.

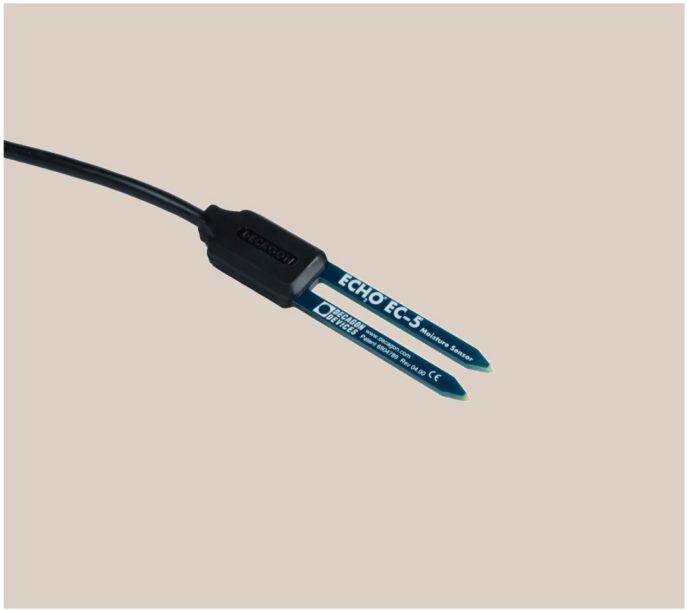


Fig. 4.8 Decagon's ECH2O EC-5 soil moisture [43]

Soil Temperature Sensor (DS18B20): The DS18B20 is a widely used digital soil temperature sensor that provides accurate and reliable temperature measurements. This sensor, manufactured by Maxim Integrated (now part of Analog Devices), is known for its precision and ease of integration into various environmental monitoring systems.

One of the standout features of the DS18B20 is its wide temperature range, from -55°C to +125°C, with an accuracy of ±0.5°C. This makes it suitable for a variety of applications, from agricultural research to environmental monitoring.

In agricultural applications, the DS18B20 helps monitor soil temperature to optimize planting conditions, manage crop growth, and improve irrigation strategies. It also plays a crucial role in research on soil-plant interactions, helping scientists understand how temperature affects soil dynamics and plant health.

4.3.4 Communication System

Our intelligent irrigation system utilizes the LoRaWAN (Long Range Wide Area Network) communication protocol, ensuring seamless and long-range data transmission.

By integrating LoRaWAN, our system establishes efficient and robust communication channels between IoT nodes, the gateway, and other system components.

The soil temperature and soil moisture sensors are integrated with LoRa nodes (Figure 4.9), which measure the respective parameters and transmit the data to the gateway. This setup allows for real-time monitoring of soil conditions.



Fig. 4.9 Installation of the LoRa node at the FSDM [43]

4.3.5 Network Infrastructure

LoRa gateway: The LoRa gateway is a critical component in the WSN for smart irrigation, serving as the intermediary between the field sensors and the central data processing system. The primary purpose of the LoRa gateway is to receive data transmitted from the sensors using the LoRaWAN protocol and forward this data to the server. This setup ensures efficient, long-range, and low-power communication, which is essential for remote agricultural applications where sensors are often spread across vast areas.(Figure 4.10).



Fig. 4.10 LoRa gateway [43]

The Things Network (TTN): In the implementation of the sensor fault detection algorithm for smart irrigation, the system's communication architecture utilizes The Things Network (TTN) server as a key component. TTN is an open-source, community-driven LoRaWAN network server that connects IoT devices to the internet through long-range, low-power wireless communication.

TTN acts as the intermediary between the wireless sensor nodes deployed in the irrigation field and the centralized processing system where data is analyzed for fault detection. The soil moisture and soil temperature sensors transmit their data wirelessly using LoRaWAN technology, which is particularly suited for applications that require long-range communication with minimal power consumption.

The sensor data, once transmitted through a gateway, is received by the TTN server, which performs several essential functions. It aggregates data from multiple sensor nodes, efficiently managing it before forwarding it to the designated application server for further processing. Additionally, TTN provides end-to-end encryption, ensuring that data is securely transmitted from the sensors to the application server, which is crucial in maintaining the integrity of sensor data in agricultural settings.

The server also facilitates seamless device management, including the registration, monitoring, and updating of devices. This simplifies the integration of new sensors

into the network and ensures consistent configurations across all devices. TTN's infrastructure is scalable, allowing for easy expansion of the sensor network as needed, and supports various deployment models, providing flexibility in system design [43].

In the context of fault detection, the TTN server plays a pivotal role in ensuring reliable communication between the sensor nodes and the analysis platform. By providing a stable and secure data transmission path, TTN minimizes the risk of communication failures that could be misinterpreted as sensor faults. The server's robust data handling capabilities support the real-time analysis necessary for effective fault detection, thereby enhancing the overall reliability and efficiency of the smart irrigation system.

4.3.6 Monitoring Dashboard

The Node-RED dashboard is a versatile and powerful tool used for real-time monitoring and control in various applications, including smart systems, IoT projects, and industrial automation. It provides a highly customizable visual interface where users can monitor data, track system performance, and interact with connected devices in an intuitive way. The dashboard's ability to display data through various types of charts, graphs, and gauges makes it ideal for real-time data analysis and decision-making (Figure 4.11). In the context of a WSN for smart irrigation, the Node-RED dashboard is specifically designed to facilitate fault detection and system monitoring. This dashboard is structured into two primary sections, each focused on a critical aspect of soil monitoring: soil moisture and soil temperature.

Each section of the dashboard presents key data points, including actual sensor readings (labeled as "Real Moisture" and "Real Temperature") alongside the corresponding predicted values. This layout enables users to visually compare the real-time sensor data with the predicted values, making it easier to detect any discrepancies or potential faults. Additionally, the dashboard features "Residual" graphs, which represent the difference between the actual sensor readings and the predicted values. These residuals are essential for fault detection, as significant deviations could indicate sensor malfunctions.

Overall, this Node-RED dashboard provides a clear and organized visual representation of the sensor data, allowing for quick identification of anomalies and facilitating timely interventions to ensure the accuracy and reliability of the smart irrigation system.

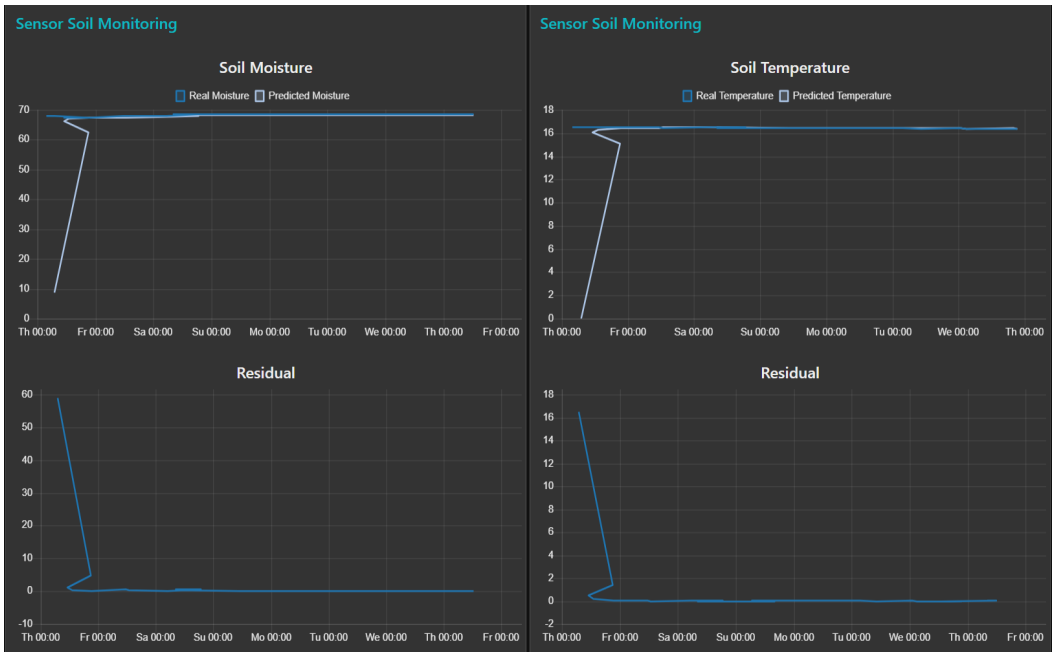


Fig. 4.11 The GUI of the real-time monitoring using Node-RED

4.4 Results and Analysis

Data Collection and Initial Setup

To evaluate the performance of our fault detection algorithm, we collected data from two sensors: a soil moisture sensor and a soil temperature sensor. The data collection spanned 200 minutes, with each sensor sampling data every minute, resulting in 200 data points per sensor. This dataset serves as the baseline for modeling the normal operational behavior of the sensors.

AR(1) Model Training

An autoregressive model of order 1 (AR(1)) with a constant term was chosen to predict the sensor readings based on their previous values. The AR(1) model was trained using the collected dataset, and the autoregressive coefficient (ϕ) and constant term (c) were determined for each sensor. The models for the soil moisture sensor and the soil temperature sensor were expressed as follows:

$$X_t^{Moisture} = C_{Moisture} + \phi_{Moisture} X_{t-1}^{Moisture} \tag{4.13}$$

$$X_t^{Temperature} = C_{Temperature} + \phi_{Temperature} X_{t-1}^{Temperature} \tag{4.14}$$

where $X_t^{(Sensor)}$ represents the reading from the sensor at time t, C_{Sensor} is the constant term, ϕ_{Sensor} is the autoregressive coefficient.

Prediction and Residual Calculation

Using the trained Kalman filter-based AR(1) models, we generate predictions for the sensor readings for soil moisture and soil temperature, as shown in Figure 4.12 and Figure 4.13.

The residuals, defined as the difference between the actual sensor readings and the predicted values, are then computed:

$$Residual_t^{Moisture} = Y_t^{Moisture} - Predicted_t^{Moisture} \tag{4.15}$$

$$Residual_t^{Temperature} = Y_t^{Temperature} - Predicted_t^{Temperature} \tag{4.16}$$

These residuals provide insights into the discrepancies between the expected and actual sensor behavior.

The residuals for both sensors are subjected to statistical analysis to characterize their behavior. The key statistical characteristics computed included the mean and standard deviation of the residuals.

Using the statistical characteristics of the residuals, we calculated the fault detection threshold for each sensor.

$$Threshold_{moisture} = \mu_{moisture} + 2.6\sigma_{moisture} \tag{4.17}$$

$$Threshold_{temperature} = \mu_{temperature} + 2.6\sigma_{temperature} \tag{4.18}$$

The parameters of the algorithm are summarized in Table 1.

Table 4.1 Parameters of the Kalman Filter-Based AR(1) Model for Sensor Data

Parameter	**Soil Temperature Sensor**	**Soil Moisture Sensor**
Coefficients of AR(1) Model	$\phi = 0.079$; $c = 15.193$	$\phi = 0.871$; $c = 8.756$
Q (Process Noise Covariance)	1.10^{-4}	1.10^{-2}
R (Measurement Noise Covariance)	0.1	0.001
Fault Detection Threshold	0.060	0.105

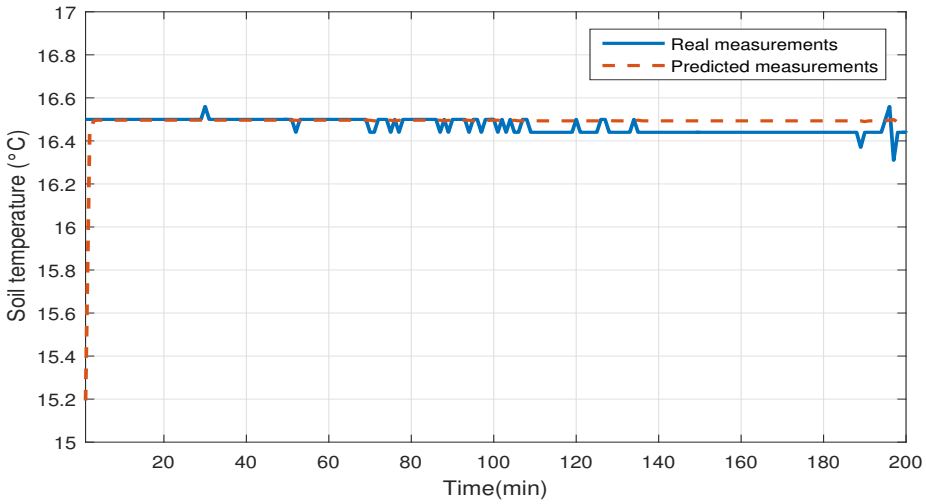


Fig. 4.12 Soil temperature measurements and predicted data in non-faulty mode

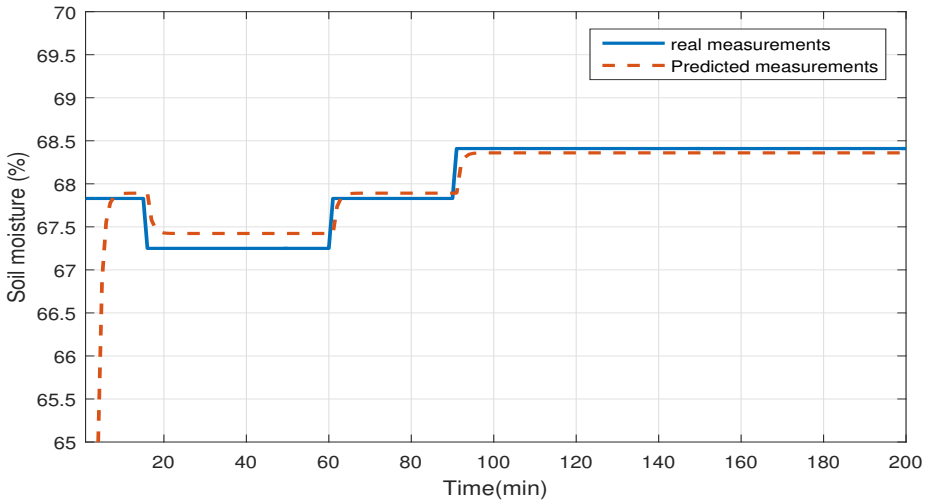


Fig. 4.13 Soil moisture measurements and predicted data in non-faulty mode

To validate the effectiveness of the fault detection algorithm, we introduced synthetic faults into the sensor readings and applied the trained Kalman filter-based AR(1) models to detect these anomalies.

At t=500 minutes, we introduce an abrupt fault into the soil temperature sensor. Following this fault injection, the actual soil temperature measurements deviate from the predicted data, as shown in Figure 4.14. This deviation causes the residual to exceed

the predefined threshold at the moment of injection, as illustrated in Figure 4.15. The fault detection algorithm successfully identified the anomaly at t=500min.

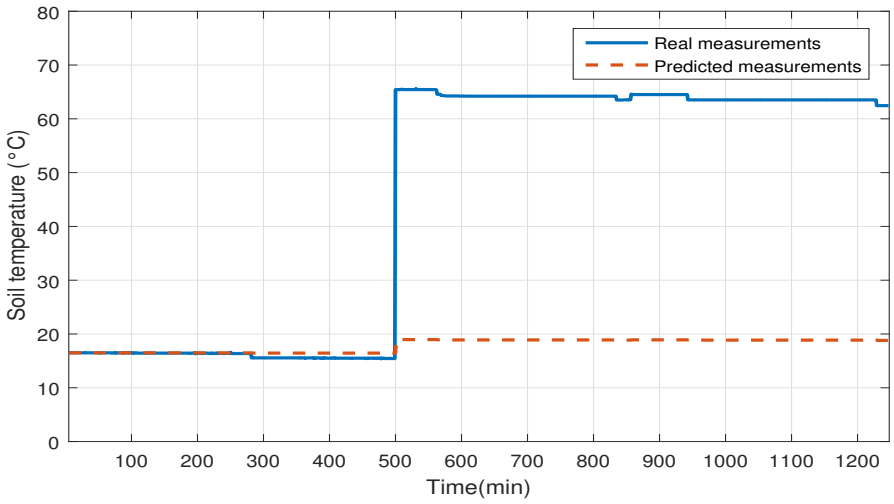


Fig. 4.14 Soil temperature measurements with its predicted data in faulty mode

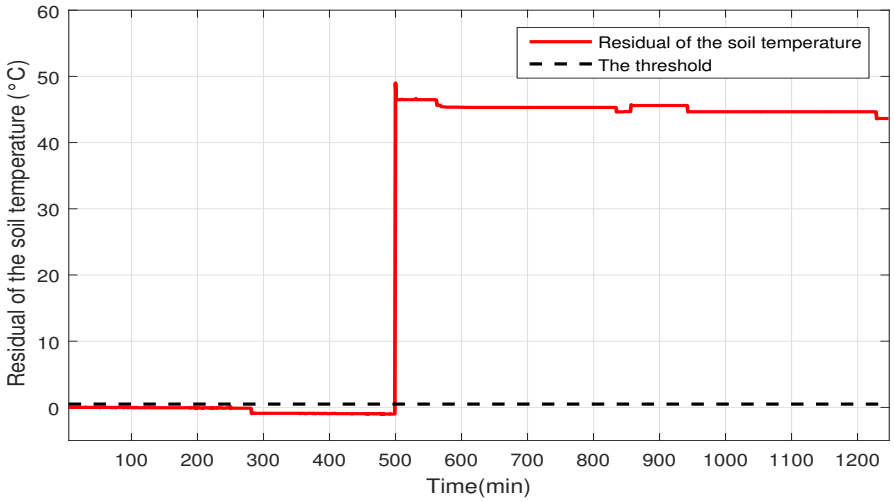


Fig. 4.15 Residuals of soil temperature sensor before and after fault introduction

At t=500 minutes, we also introduce an abrupt fault into the soil moisture sensor. Following the fault injection, the actual measurements of the soil moisture sensor deviate from the predicted data, as illustrated in Figure 4.16. This deviation causes the residuals to exceed the predefined threshold at the moment of injection, as shown in Figure 4.17.

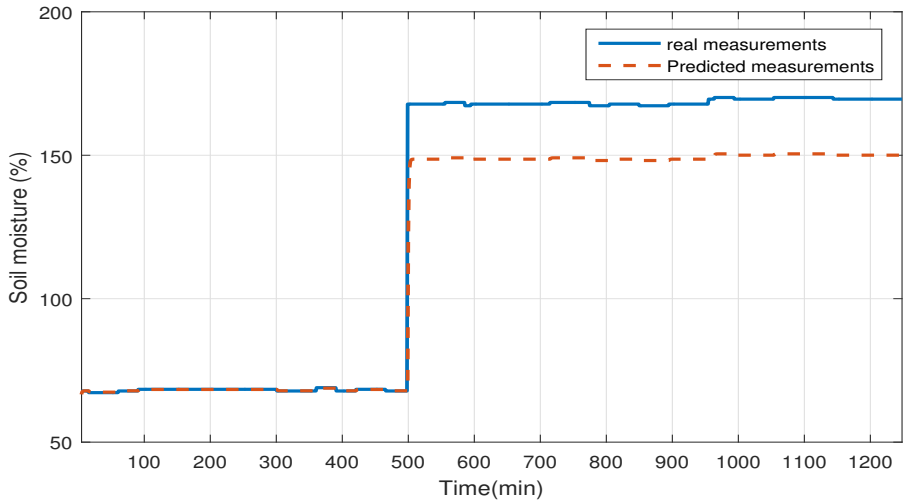


Fig. 4.16 Soil moisture measurements with its predicted data in faulty mode

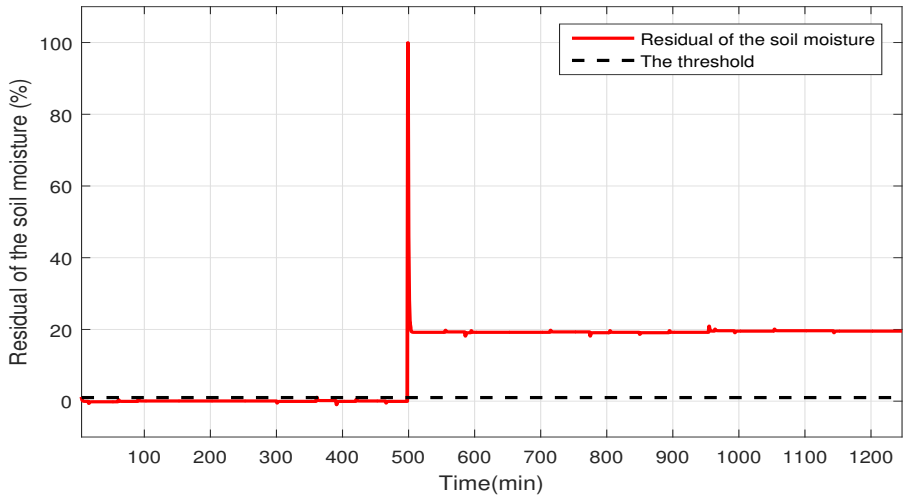


Fig. 4.17 Residuals of soil moisture sensor before and after fault introduction

The results demonstrate that the Kalman filter-based AR(1) model fault detection algorithm is capable of identifying anomalies in sensor readings with high accuracy. The statistical analysis of residuals and the calculated thresholds provided a reliable means to distinguish between normal and faulty sensor behavior.

4.5 Conclusion

In conclusion, the implementation of a sensor fault detection algorithm within a WSN for smart irrigation applications demonstrates significant promise in enhancing the reliability and efficiency of agricultural water management. The Kalman filter, employed in conjunction with an autoregressive model, provides a robust framework for predicting sensor readings and detecting anomalies. This approach ensures that faults are identified promptly, allowing for timely interventions and minimizing the adverse effects on irrigation decisions.

Our real-world implementation and experiments validate the efficacy of the proposed fault detection system. By maintaining accurate and reliable sensor data, the system supports optimal irrigation strategies, thereby contributing to sustainable agricultural practices. The insights gained from this study pave the way for further advancements in sensor fault detection algorithms, with potential applications extending beyond smart irrigation to other domains requiring precise environmental monitoring.

Conclusion and Future Work

WSNs have seen exponential adoption in recent years, revolutionizing various sectors through their ability to provide real-time data and optimize processes. In smart agriculture, WSNs enable continuous monitoring of soil and climate conditions, contributing to more precise irrigation management and improved crop productivity.

WSNs used in smart agriculture offer numerous benefits, but they are also prone to failures that can impact their reliability and effectiveness. These failures can be caused by various factors, including challenging environmental conditions such as excessive moisture, extreme temperatures, and exposure to the elements, which can damage the sensors and electronic components.

To ensure the reliability of a WSN in agriculture, it is crucial to guarantee that the measurements provided by the sensors are accurate and reliable. A fault detection and isolation system for failing sensors is essential to maintain the effectiveness of smart irrigation. This process ensures that irrigation is carried out optimally, providing the necessary amount of water to the crops without waste or deficit.

The research work presented in this thesis is part of this approach. It focuses on the development of systematic procedures for fault detection in WSNs for smart irrigation, using analytical redundancy methods.

In our research, we have emphasized the limited resources of sensor nodes by developing two innovative algorithms for detecting faults in a WSN, taking these constraints into account, unlike the most common approaches in the literature, which are often complex and energy-consuming. This conclusion will provide a concise summary of the key ideas discussed in the previous sections and offer some suggestions for future research.

We started by providing a comprehensive overview of the role of WSNs in smart irrigation, emphasizing the urgent water challenges in agriculture and the necessity for efficient solutions. The first chapter explored the architecture, components, and

network topologies of WSNs, along with their applications. It also discussed the communication protocols used in smart agriculture, underscoring their characteristics and suitability for different scenarios. Additionally, the chapter addressed the inherent challenges of deploying WSNs.

The complexities of fault detection in WSNs, examining various fault detection techniques and the importance of understanding network architecture and data characteristics are explored in the second chapter. It highlighted the limitations of conventional fault detection approaches and the need for innovative solutions that incorporate spatial and temporal information for accurate fault diagnosis. The chapter underscored the necessity for continued research and collaboration to overcome the challenges in fault detection and diagnosis in WSNs.

The third chapter introduced a novel approach to FDI in WSNs, specifically within the context of smart irrigation systems. The proposed methodology leveraged spatial correlation among sensor readings to detect and isolate faulty sensors without relying on redundant hardware. The chapter detailed the significance of soil monitoring in agriculture, particularly for optimizing irrigation practices. It outlined the methodology for sensor deployment, emphasizing strategic placement, redundancy, calibration, and synchronization. The chapter also introduced the concept of parity space as a fundamental aspect of fault detection and isolation algorithms.

In the fourth chapter and last chapter focused on the real-world implementation of a sensor fault detection algorithm using a Kalman filter in conjunction with an autoregressive model. This approach provided a robust framework for predicting sensor readings and detecting anomalies. The chapter presented the results of laboratory experiments, which demonstrated the effectiveness and efficiency of the proposed fault detection system. The real-world implementation validated the system's capability to maintain accurate and reliable sensor data, supporting optimal irrigation strategies and contributing to sustainable agricultural practices.

In conclusion, this thesis makes significant contributions to the field of smart irrigation by addressing the challenges of sensor reliability and fault detection in WSNs. The proposed methodologies and algorithms offer practical solutions for enhancing water use efficiency and ensuring the reliability and robustness of smart irrigation systems. This research provides valuable insights that open new avenues for advancements in sensor fault detection, with strong potential for application in other fields that require precise environmental monitoring. By integrating innovative FDI techniques and

robust sensor networks, this work supports the development of sustainable agricultural practices in the face of global water scarcity.

Future research directions include:

- This thesis addresses abrupt faults, future work will aim to develop algorithms capable of detecting a wider range of sensor faults.
- To evaluate the feasibility and performance of the proposed algorithms, the next step involves the real-time implementation of the WSN during an agricultural season. This will provide practical insights into the system's effectiveness and operational challenges.
- In this thesis, fault detection is achieved by predicting sensor measurements one step ahead. Although this prediction helps identify sensor faults, it cannot replace the actual measurement in case of a sensor fault. In future work, the focus will be on developing algorithms that predict sensor measurements and replace the real measurements with these predictions when a sensor node becomes faulty, ensuring uninterrupted data flow and decision-making.
- The current work employs spatial correlation in the first approach and temporal correlation in the second approach. Future efforts will focus on integrating both spatial and temporal correlations to enhance the accuracy and reliability of fault detection.

References

[1] Abedin, Z., Chowdhury, A. S., Hossain, M. S., Andersson, K., and Karim, R. (2017). An interoperable ip based wsn for smart irrigation system. In *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–5. IEEE.

[2] Abid, M., Chen, W., Ding, S. X., and Khan, A. Q. (2011). Optimal residual evaluation for nonlinear systems using post-filter and threshold. *International Journal of Control*, 84(3):526–539.

[3] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002a). A survey on sensor networks. *IEEE Communications magazine*, 40(8):102–114.

[4] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002b). Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422.

[5] Alansari, Z., Prasanth, A., and Belgaum, M. R. (2018). A comparison analysis of fault detection algorithms in wireless sensor networks. In *2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pages 1–6. IEEE.

[6] Algamili, A. S., Khir, M. H. M., Dennis, J. O., Ahmed, A. Y., Alabsi, S. S., Ba Hashwan, S. S., and Junaid, M. M. (2021). A review of actuation and sensing mechanisms in mems-based sensor devices. *Nanoscale research letters*, 16:1–21.

[7] Ali, A., Ming, Y., Chakraborty, S., and Iram, S. (2017). A comprehensive survey on real-time applications of wsn. *Future internet*, 9(4):77.

[8] Aliev, K., Jawaid, M. M., Narejo, S., Pasero, E., and Pulatov, A. (2018). Internet of plants application for smart agriculture. *International Journal of Advanced Computer Science and Applications*, 9(4).

[9] Alkhatib, A. A. A. and Baicher, G. S. (2012). Wireless sensor network architecture. In *International conference on computer networks and communication systems (ICNCS 2012)*, volume 35, pages 11–15.

[10] Avşar, E. and Mowla, M. N. (2022). Wireless communication protocols in smart agriculture: A review on applications, challenges and future trends. *Ad Hoc Networks*, page 102982.

[11] Azar, H. N. and Malazi, H. T. (2018). Decentralized detection of hybrid faults in mobile sensor nodes. *Simulation Modelling Practice and Theory*, 87:210–225.

[12] Banđur, Đ., Jakšić, B., Banđur, M., and Jović, S. (2019). An analysis of energy efficiency in wireless sensor networks (wsns) applied in smart agriculture. *Computers and electronics in agriculture*, 156:500–507.

[13] Barrenetxea, G., Ingelrest, F., Schaefer, G., Vetterli, M., Couach, O., and Parlange, M. (2008). Sensorscope: Out-of-the-box environmental monitoring. In *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*, pages 332–343. IEEE.

[14] Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M., and Schröder, J. (2006). *Diagnosis and fault-tolerant control*, volume 2. Springer.

[15] Boisguene, R., Tseng, S.-C., Huang, C.-W., and Lin, P. (2017). A survey on nb-iot downlink scheduling: Issues and potential solutions. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 547–551. IEEE.

[16] Bonnet, P., Gehrke, J., and Seshadri, P. (2000). Querying the physical world. *IEEE personal Communications*, 7(5):10–15.

[17] Bouguera, T., Diouris, J.-F., Chaillout, J.-J., Jaouadi, R., and Andrieux, G. (2018). Energy consumption model for sensor nodes based on lora and lorawan. *Sensors*, 18(7):2104.

[18] Brockwell, P. J. and Davis, R. A. (2002). *Introduction to time series and forecasting*. Springer.

[19] Centenaro, M., Vangelista, L., Zanella, A., and Zorzi, M. (2016). Long-range communications in unlicensed bands: The rising stars in the iot and smart city scenarios. *IEEE Wireless Communications*, 23(5):60–67.

[20] Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58.

[21] Chandrappa, V. Y., Ray, B., Ashwatha, N., and Shrestha, P. (2023). Spatiotemporal modeling to predict soil moisture for sustainable smart irrigation. *Internet of Things*, 21:100671.

[22] Chiarelli, D. D., D'Odorico, P., Müller, M. F., Mueller, N. D., Davis, K. F., Dell'Angelo, J., Penny, G., and Rulli, M. C. (2022). Competition for water induced by transnational land acquisitions for agriculture. *Nature Communications*, 13(1):505.

[23] Chuo, L.-X., Feng, Z., Kim, Y., Chiotellis, N., Yasuda, M., Miyoshi, S., Kawaminami, M., Grbic, A., Wentzloff, D., Blaauw, D., et al. (2019). Millimeter-scale node-to-node radio using a carrier frequency-interlocking if receiver for a fully integrated 4 x 4 x 4 mm³ wireless sensor node. *IEEE Journal of Solid-State Circuits*, 55(5):1128–1138.

[24] Connor, R., Koncagul, E., Tran, M., et al. (2021). The united nations world water development report 2021: valuing water: facts and figures.

[25] Daly, K. C., Gai, E., and Harrison, J. V. (1979). Generalized likelihood test for fdi in redundant sensor configurations. *Journal of Guidance and Control*, 2(1):9–17.

[26] Darwish, A. and Hassanien, A. E. (2011). Wearable and implantable wireless sensor network solutions for healthcare monitoring. *Sensors*, 11(6):5561–5595.

[27] de Oliveira, K. V., Castelli, H. M. E., Montebeller, S. J., and Avancini, T. G. P. (2017). Wireless sensor network for smart agriculture using zigbee protocol. In *2017 IEEE First Summer School on Smart Cities (S3C)*, pages 61–66. IEEE.

[28] Dhage, M. R. and Vemuru, S. (2017). Structural health monitoring of railway tracks using wsn. In *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, pages 1–5. IEEE.

[29] Di Gennaro, P., Lofú, D., Vitanio, D., Tedeschi, P., and Boccadoro, P. (2019). Waters: A sigfox-compliant prototype for water monitoring. *Internet Technology Letters*, 2(1):e74.

[30] Díaz, S. E., Pérez, J. C., Mateos, A. C., Marinescu, M.-C., and Guerra, B. B. (2011a). A novel methodology for the monitoring of the agricultural production process based on wireless sensor networks. *Computers and electronics in agriculture*, 76(2):252–265.

[31] Díaz, S. E., Pérez, J. C., and Muñoz, J. F. (2011b). Survey of the state-of-the-art in flash-based sensor nodes. In *Flash Memories*, pages 113–136. IntechOpen.

[32] Dolan, F., Lamontagne, J., Link, R., Hejazi, M., Reed, P., and Edmonds, J. (2021). Evaluating the economic impact of water scarcity in a changing world. *Nature communications*, 12(1):1–10.

[33] Farooq, M. S., Riaz, S., Abid, A., Abid, K., and Naeem, M. A. (2019). A survey on the role of iot in agriculture for the implementation of smart farming. *Ieee Access*, 7:156237–156271.

[34] Fernández-Ahumada, L. M., Ramírez-Faz, J., Torres-Romero, M., and López-Luque, R. (2019). Proposal for the design of monitoring and operating irrigation networks based on iot, cloud computing and free hardware technologies. *Sensors*, 19(10):2318.

[35] Foubert, B. and Mitton, N. (2020). Long-range wireless radio technologies: A survey. *Future internet*, 12(1):13.

[36] Fraga-Lamas, P., Celaya-Echarri, M., Azpilicueta, L., Lopez-Iturri, P., Falcone, F., and Fernández-Caramés, T. M. (2019). Design and empirical validation of a lorawan iot smart irrigation system. In *Proceedings*, volume 42, page 62. MDPI.

[37] Furtak, J., Zieliński, Z., and Chudzikiewicz, J. (2019). A framework for constructing a secure domain of sensor nodes. *Sensors*, 19(12):2797.

[38] Gaber, M. M. (2007). Data stream processing in sensor networks. *Learning from Data Streams: Processing Techniques in Sensor Networks*, pages 41–48.

[39] García, L., Parra, L., Jimenez, J. M., Parra, M., Lloret, J., Mauri, P. V., and Lorenz, P. (2021). Deployment strategies of soil monitoring wsn for precision agriculture irrigation scheduling in rural areas. *Sensors*, 21(5):1693.

[40] Gautam, S., Tamboli, P. K., Roy, K., Patankar, V. H., and Duttagupta, S. P. (2019). Sensors incipient fault detection and isolation of nuclear power plant using extended kalman filter and kullback–leibler divergence. *ISA transactions*, 92:180–190.

[41] Gerodimos, A., Maglaras, L., Ferrag, M. A., Ayres, N., and Kantzavelou, I. (2023). Iot: Communication protocols and security threats. *Internet of Things and Cyber-Physical Systems*.

[42] Giammarini, M., Isidori, D., Concettoni, E., Cristalli, C., Fioravanti, M., and Pieralisi, M. (2015). Design of wireless sensor network for real-time structural health monitoring. In *2015 IEEE 18th International Symposium on Design and Diagnostics of Electronic Circuits & Systems*, pages 107–110. IEEE.

[43] Hafian, A. (2023). *Design and Study of an Intelligent Irrigation System Based on Photovoltaic Pumping and Internet of Things*. PhD thesis, Sidi Mohamed Ben Abdellah University, Faculty of Sciences-Dhar El Mahraz, Fez, Morocco.

[44] Hafian, A., Benbrahim, M., and Kabbaj, M. N. (2023). Iot-based smart irrigation management system using real-time data. *International Journal of Electrical & Computer Engineering (2088-8708)*, 13(6).

[45] Hajiyev, C. and Caliskan, F. (2003). *Fault diagnosis and reconfiguration in flight control systems*, volume 2. Springer Science & Business Media.

[46] Hamami, L. and Nassereddine, B. (2020). Application of wireless sensor networks in the field of irrigation: A review. *Computers and Electronics in Agriculture*, 179:105782.

[47] Higginbottom, T. P., Adhikari, R., Dimova, R., Redicker, S., and Foster, T. (2021). Performance of large-scale irrigation projects in sub-saharan africa. *Nature Sustainability*, 4(6):501–508.

[48] HUANG, Y.-L., Jia-Yin, T., and SHANG, T.-Q. (2020). Design of monitoring system in smart agriculture environment. *DEStech Transactions on Social Science, Education and Human Science,(icesd)*.

[49] Jha, K., Doshi, A., Patel, P., and Shah, M. (2019). A comprehensive review on automation in agriculture using artificial intelligence. *Artificial Intelligence in Agriculture*, 2:1–12.

[50] Jihani, N., Kabbaj, M. N., and Benbrahim, M. (2021). Wireless sensor network fault detection and isolation for smart irrigation. In *International Conference on Digital Technologies and Applications*, pages 759–768. Springer.

[51] Jihani, N., Kabbaj, M. N., and Benbrahim, M. (2023a). Kalman filter based sensor fault detection in wireless sensor network for smart irrigation. *Results in Engineering*, 20:101395.

[52] Jihani, N., Kabbaj, M. N., and Benbrahim, M. (2023b). Sensor fault detection and isolation for smart irrigation wireless sensor network based on parity space. *International Journal of Electrical and Computer Engineering*, 13(2):1463.

[53] Jin, J.-s., Jin, J., Wang, Y.-h., Zhao, K., and Hu, J.-j. (2008). Development of remote-controlled home automation system with wireless sensor network. In *2008 Fifth IEEE International Symposium on Embedded Computing*, pages 169–173. IEEE.

[54] Joshi, J., Bagga, A., Reddy, A., Akhil, D., Munnangi, H., Nikhil, B., and Reddy, J. (2016). Structural health monitoring of earth air tunnel (eat) using wireless sensor network. In *2016 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*, pages 1–5. IEEE.

[55] Jurdak, R., Wang, X. R., Obst, O., and Valencia, P. (2011). Wireless sensor network anomalies: Diagnosis and detection strategies. In *Intelligence-Based Systems Engineering*, pages 309–325. Springer.

[56] Kallas, M., Honeine, P., Francis, C., and Amoud, H. (2013). Kernel autoregressive models using yule–walker equations. *Signal Processing*, 93(11):3053–3061.

[57] Kamble, P. R. and Vatti, R. A. (2015). Structural health monitoring of river bridges using wireless sensor networks. In *2015 International Conference on Pervasive Computing (ICPC)*, pages 1–5. IEEE.

[58] Kenda, K., Škrbec, J., and Škrjanc, M. (2013). Usage of the kalman filter for data cleaning of sensor data. In *Proceedings of the 16th International Multiconference Information Society–IS*, pages 172–175.

[59] Kerasiotis, F., Prayati, A., Antonopoulos, C., Koulamas, C., and Papadopoulos, G. (2010). Battery lifetime prediction model for a wsn platform. In *2010 Fourth International Conference on Sensor Technologies and Applications*, pages 525–530. IEEE.

[60] Khalifeh, A., Saadeh, M., Darabkh, K. A., and Nagaradjane, P. (2021). Radio frequency based wireless charging for unsupervised clustered wsn: System implementation and experimental evaluation. *Energies*, 14(7):1829.

[61] Khan, N., Ray, R. L., Sargani, G. R., Ihtisham, M., Khayyam, M., and Ismail, S. (2021). Current progress and future prospects of agriculture technology: Gateway to sustainable agriculture. *Sustainability*, 13(9):4883.

[62] Kharb, S. and Singhrova, A. (2018). Review of industrial standards for wireless sensor networks. In *Next-Generation Networks: Proceedings of CSI-2015*, pages 77–87. Springer.

[63] Khemapech, I., Sansrimahachai, W., and Toahchoodee, M. (2016). A real-time health monitoring and warning system for bridge structures. In *2016 IEEE Region 10 Conference (TENCON)*, pages 3010–3013. IEEE.

[64] Kim, S., Pakzad, S., Culler, D., Demmel, J., Fenves, G., Glaser, S., and Turon, M. (2007). Health monitoring of civil infrastructures using wireless sensor networks. In *Proceedings of the 6th international conference on Information processing in sensor networks*, pages 254–263.

[65] Kodali, R. K., Kuthada, M. S., and Borra, Y. K. Y. (2018). Lora based smart irrigation system. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, pages 1–5. IEEE.

[66] Krishna, K. L., Silver, O., Malende, W. F., and Anuradha, K. (2017). Internet of things application for implementation of smart agriculture system. In *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pages 54–59. IEEE.

[67] Lauridsen, M., Vejlgaard, B., Kovacs, I. Z., Nguyen, H., and Mogensen, P. (2017). Interference measurements in the european 868 mhz ism band with focus on lora and sigfox. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE.

[68] Lavric, A., Petrariu, A. I., and Popa, V. (2019). Long range sigfox communication protocol scalability analysis under large-scale, high-density conditions. *IEEE Access*, 7:35816–35825.

[69] Lee, J., Ghaffari, M., and Elmeligy, S. (2011). Self-maintenance and engineering immune systems: Towards smarter machines and manufacturing systems. *Annual Reviews in Control*, 35(1):111–122.

[70] Lee, S.-W., Kim, Y.-J., Lee, G.-S., Cho, B.-O., and Lee, N.-H. (2007). A remote behavioral monitoring system for elders living alone. In *2007 international conference on control, automation and systems*, pages 2725–2730. IEEE.

[71] Lee, Y.-D. and Chung, W.-Y. (2009). Wireless sensor network based wearable smart shirt for ubiquitous health and activity monitoring. *Sensors and Actuators B: Chemical*, 140(2):390–395.

[72] Li, D., Wang, Y., Wang, J., Wang, C., and Duan, Y. (2020). Recent advances in sensor fault diagnosis: A review. *Sensors and Actuators A: Physical*, 309:111990.

[73] Li, F. X., Islam, A. A., Jaroo, A. S., Hamid, H., Jalali, J., and Sammartino, M. (2015). Urban highway bridge structure health assessments using wireless sensor network. In *2015 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet)*, pages 75–77. IEEE.

[74] Liakos, K. G., Busato, P., Moshou, D., Pearson, S., and Bochtis, D. (2018). Machine learning in agriculture: A review. *Sensors*, 18(8):2674.

[75] Liao, X., Su, Z., Liu, G., Zotarelli, L., Cui, Y., and Snodgrass, C. (2016). Impact of soil moisture and temperature on potato production using seepage and center pivot irrigation. *Agricultural Water Management*, 165:230–236.

[76] Liu, X., Liu, W., Tang, Q., Liu, B., Wada, Y., and Yang, H. (2022). Global agricultural water scarcity assessment incorporating blue and green water availability under future climate change. *Earth's Future*, 10(4):e2021EF002567.

[77] Liu, Z. and Kleiner, Y. (2012). State-of-the-art review of technologies for pipe structural health monitoring. *IEEE Sensors Journal*, 12(6):1987–1992.

[78] Llaria, A., Terrasson, G., Arregui, H., and Hacala, A. (2015). Geolocation and monitoring platform for extensive farming in mountain pastures. In *2015 IEEE International Conference on Industrial Technology (ICIT)*, pages 2420–2425. IEEE.

[79] Lloret, J., Sendra, S., Garcia, L., and Jimenez, J. M. (2021). A wireless sensor network deployment for soil moisture monitoring in precision agriculture. *Sensors*, 21(21):7243.

[80] Locke, C., Carbone, G., Filippi, A., Sadler, E., Gerwig, B., Evans, D., et al. (2000). Using remote sensing and modeling to measure crop biophysical variability. In *5th International Conference on Precision Agriculture*.

[81] Lopez-Ardao, J. C., Rodriguez-Rubio, R. F., Suarez-Gonzalez, A., Rodriguez-Perez, M., and Sousa-Vieira, M. E. (2021). Current trends on green wireless sensor networks. *Sensors*, 21(13):4281.

[82] Mahmoud, M. S. and Mohamad, A. A. (2016). A study of efficient power consumption wireless communication techniques/modules for internet of things (iot) applications.

[83] Mazibuco, V. A., Nhung, N. P., and Linh, N. T. (2023). Fault detection in wireless sensor networks with deep neural networks. *Journal of Military Science and Technology,(CSCE7)*, pages 27–36.

[84] Migabo, E., Djouani, K., Kurien, A., and Olwal, T. (2017). A comparative survey study on lpwa networks: Lora and nb-iot. In *Proceedings of the Future Technologies Conference (FTC), Vancouver, BC, Canada*, pages 29–30.

[85] Misra, S., Roy, S. K., Roy, A., Obaidat, M. S., and Jha, A. (2019). Megan: Multipurpose energy-efficient, adaptable, and low-cost wireless sensor node for the internet of things. *IEEE Systems Journal*, 14(1):144–151.

[86] Muangprathub, J., Boonnam, N., Kajornkasirat, S., Lekbangpong, N., Wanichsombat, A., and Nillaor, P. (2019). Iot and agriculture data analysis for smart farm. *Computers and electronics in agriculture*, 156:467–474.

[87] Muhammed, T. and Shaikh, R. A. (2017). An analysis of fault detection strategies in wireless sensor networks. *Journal of Network and Computer Applications*, 78:267–287.

[88] Mujawar, T. and Deshmukh, L. (2019). Smart environment monitoring system using wired and wireless network: A comparative study. In *Atmospheric Air Pollution and Monitoring*. IntechOpen.

[89] Nair, K. K., Abu-Mahfouz, A. M., and Lefophane, S. (2019). Analysis of the narrow band internet of things (nb-iot) technology. In *2019 conference on information communications technology and society (ICTAS)*, pages 1–6. IEEE.

[90] Nakpong, N. and Nakjuatong, N. (2019). Precision farming via nb-iot technology. In *Proceedings of International Conferenceon Technological and Social Innovations 2019*.

[91] Nasiakou, A., Vavalis, M., and Zimeris, D. (2016). Smart energy for smart irrigation. *Computers and Electronics in Agriculture*, 129:74–83.

[92] Niculescu, D. and Nath, B. (2001). Ad hoc positioning system (aps). In *GLOBECOM'01. IEEE global telecommunications conference (Cat. No. 01CH37270)*, volume 5, pages 2926–2931. IEEE.

[93] Ojha, T., Misra, S., and Raghuwanshi, N. S. (2015). Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges. *Computers and electronics in agriculture*, 118:66–84.

[94] Onwuka, B. and Mang, B. (2018). Effects of soil temperature on some soil properties and plant growth. *Adv. Plants Agric. Res*, 8(1):34–37.

[95] Othman, M. F. and Shazali, K. (2012). Wireless sensor network applications: A study in environment monitoring system. *Procedia Engineering*, 41:1204–1210.

[96] Palpanas, T., Papadopoulos, D., Kalogeraki, V., and Gunopulos, D. (2003). Distributed deviation detection in sensor networks. *Acm sigmod record*, 32(4):77–82.

[97] Panigrahi, T., Panda, M., and Panda, G. (2016). Fault tolerant distributed estimation in wireless sensor networks. *Journal of Network and Computer Applications*, 69:27–39.

[98] Patil, P. K. and Patil, S. (2017). Review on structural health monitoring system using wsn for bridges. In *2017 International conference of electronics, communication and aerospace technology (ICECA)*, volume 1, pages 628–631. IEEE.

[99] Patton, R. J. and Chen, J. (1991). A review of parity space approaches to fault diagnosis. *IFAC Proceedings Volumes*, 24(6):65–81.

[100] Porkodi, R. and Bhuvaneswari, V. (2014). The internet of things (iot) applications and communication enabling technology standards: An overview. In *2014 International conference on intelligent computing applications*, pages 324–329. IEEE.

[101] Richter, B. D., Bartak, D., Caldwell, P., Davis, K. F., Debaere, P., Hoekstra, A. Y., Li, T., Marston, L., McManamay, R., Mekonnen, M. M., et al. (2020). Water scarcity and fish imperilment driven by beef production. *Nature Sustainability*, 3(4):319–328.

[102] Sadowski, S. and Spachos, P. (2020). Wireless technologies for smart agricultural monitoring using internet of things devices with energy harvesting capabilities. *Computers and Electronics in Agriculture*, 172:105338.

[103] Salman, T. (2015). Internet of things protocols and standards. *M. Of END, Affairs*.

[104] Samara, M. A., Bennis, I., Abouaissa, A., and Lorenz, P. (2022). A survey of outlier detection techniques in iot: review and classification. *Journal of Sensor and Actuator Networks*, 11(1):4.

[105] Samie, F., Bauer, L., and Henkel, J. (2016). Iot technologies for embedded computing: A survey. In *Proceedings of the Eleventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, pages 1–10.

[106] Scheer, J. (2011). *Failed bridges: case studies, causes and consequences*. John Wiley & Sons.

[107] Sciddurlo, G., Petrosino, A., Quadrini, M., Roseti, C., Striccoli, D., Zampognaro, F., Luglio, M., Perticaroli, S., Mosca, A., Lombardi, F., et al. (2021). Looking at nb-iot over leo satellite systems: Design and evaluation of a service-oriented solution. *IEEE Internet of Things Journal*, 9(16):14952–14964.

[108] Serdio, F., Lughofer, E., Pichler, K., Buchegger, T., Pichler, M., and Efendic, H. (2014). Fault detection in multi-sensor networks based on multivariate time-series models and orthogonal transformations. *Information Fusion*, 20:272–291.

[109] Sharma, A. B., Golubchik, L., and Govindan, R. (2010). Sensor faults: Detection methods and prevalence in real-world datasets. *ACM Transactions on Sensor Networks (TOSN)*, 6(3):1–39.

[110] Sharma, S., Bansal, R. K., and Bansal, S. (2013). Issues and challenges in wireless sensor networks. In *2013 international conference on machine intelligence and research advancement*, pages 58–62. IEEE.

[111] Sheng, B., Li, Q., Mao, W., and Jin, W. (2007). Outlier detection in sensor networks. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 219–228.

[112] Shrestha, A. and Xing, L. (2007). A performance comparison of different topologies for wireless sensor networks. In *2007 IEEE Conference on Technologies for Homeland Security*, pages 280–285. IEEE.

[113] Simon, D. (2006). *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons.

[114] Sudduth, K. A. (1999). Engineering technologies for precision farming. In *International seminar on agricultural mechanization technology for precision farming*, pages 5–27. Citeseer.

[115] Tan, P.-N. (2006). Knowledge discovery from sensor data. *Sensors Magazine*.

[116] Tantalaki, N., Souravlas, S., and Roumeliotis, M. (2019). Data-driven decision making in precision agriculture: The rise of big data in agricultural systems. *Journal of agricultural & food information*, 20(4):344–380.

[117] Terrasson, G., Llaria, A., and Briand, R. (2014). System level dimensioning of low power biomedical body sensor networks. In *2014 IEEE Faible Tension Faible Consommation*, pages 1–4. IEEE.

[118] Terrasson, G., Llaria, A., Marra, A., and Voaden, S. (2016). Accelerometer based solution for precision livestock farming: geolocation enhancement and animal activity identification. In *IOP Conference Series: Materials Science and Engineering*, volume 138, page 012004. IOP Publishing.

[119] Tubaishat, M. and Madria, S. (2003). Sensor networks: an overview. *IEEE potentials*, 22(2):20–23.

[120] Ungureanu, N., Vlăduț, V., and Voicu, G. (2020). Water scarcity and wastewater reuse in crop irrigation. *Sustainability*, 12(21):9055.

[121] Van Es, H. and Woodard, J. (2017). Innovation in agriculture and food systems in the digital age. *The global innovation index*, 99.

[122] Vangelista, L., Zanella, A., and Zorzi, M. (2015). Long-range iot technologies: The dawn of lora™. In *Future Access Enablers for Ubiquitous and Intelligent Infrastructures: First International Conference, FABULOUS 2015, Ohrid, Republic of Macedonia, September 23-25, 2015. Revised Selected Papers 1*, pages 51–58. Springer.

[123] Vincent, D. R., Deepa, N., Elavarasan, D., Srinivasan, K., Chauhdary, S. H., and Iwendi, C. (2019). Sensors driven ai-based agriculture recommendation model for assessing land suitability. *Sensors*, 19(17):3667.

[124] Visconti, P., de Fazio, R., Velázquez, R., Del-Valle-Soto, C., and Giannoccaro, N. I. (2020). Development of sensors-based agri-food traceability system remotely managed by a software platform for optimized farm management. *Sensors*, 20(13):3632.

[125] Wang, L., Wu, L., Guan, Y., and Wang, G. (2015). Online sensor fault detection based on an improved strong tracking filter. *Sensors*, 15(2):4578–4591.

[126] Widhalm, D. (2022). *Sensor Node Fault Detection in Wireless Sensor Networks: An Immune-inspired Approach*. PhD thesis, Wien.

[127] Xiaojing, Z. and Yuanguai, L. (2012). Zigbee implementation in intelligent agriculture based on internet of things. In *2nd International Conference on Electronic & Mechanical Engineering and Information Technology*, pages 1842–1846. Atlantis Press.

[128] Xu, H., Zhang, R., Lin, C., and Ma, Y. (2016). Novel approach of fault diagnosis in wireless sensor networks node based on rough set and neural network model. *International Journal of Future Generation Communication and Networking*, 9(4):1–16.

[129] Yick, J., Mukherjee, B., and Ghosal, D. (2008). Wireless sensor network survey. *Computer networks*, 52(12):2292–2330.

[130] Zhang, Y., Meratnia, N., and Havinga, P. (2010). Outlier detection techniques for wireless sensor networks: A survey. *IEEE communications surveys & tutorials*, 12(2):159–170.

[131] Zhang, Z., Mehmood, A., Shu, L., Huo, Z., Zhang, Y., and Mukherjee, M. (2018). A survey on fault diagnosis in wireless sensor networks. *IEEE Access*, 6:11349–11364.

[132] Zhihua, S. (2016). Design of smart home system based on zigbee. In *2016 International Conference on Robots & Intelligent System (ICRIS)*, pages 167–170. IEEE.

[133] Zidi, S., Moulahi, T., and Alaya, B. (2017). Fault detection in wireless sensor networks through svm classifier. *IEEE Sensors Journal*, 18(1):340–347.

[134] Zrelli, A. and Ezzedine, T. (2017). Localization of damage using wireless sensor networks for tunnel health monitoring. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1161–1165. IEEE.

[135] Zyrianoff, I., Heideker, A., Silva, D., Kleinschmidt, J., Soininen, J.-P., Salmon Cinotti, T., and Kamienski, C. (2019). Architecting and deploying iot smart applications: A performance–oriented approach. *Sensors*, 20(1):84.

Appendix A

Calculation of Autoregressive (AR) Model Coefficients

In this appendix, we provide the steps to calculate the coefficients of the autoregressive (AR) model, which is referenced in the thesis.

A.1 Overview of the AR Model

An autoregressive (AR) model predicts the value of a time series based on its previous values. The general form of an AR model of order p is:

$$X_t = C + \sum_{i=1}^{p} \phi_i X_{t-i} + \varepsilon_t \tag{A.1}$$

Where:

- X_t is the value of the current value.
- $X_{t-1}, X_{t-2}, ..., X_{t-p}$ are the past value.
- C is a constant.
- $\phi_1, \phi_2, ..., \phi_p$ are the model coefficients.
- ε_t is white noise error term.

  The parameters $\phi_1, \phi_2, ..., \phi_p$ are directly connected with the covariance function of the process. One can therefore determine these parameters from the autocorrelation function.

A.2 Steps to Calculate AR Coefficients

There are several methods to calculate the coefficients $\phi_1, \phi_2, \ldots, \phi_p$, including:

a. **Yule-Walker Equations**

Let the data be centered, thus let μ be the expectation of X_t, namely,

$$\mu = \mathbb{E}[X_t] \tag{A.2}$$

where $\mathbb{E}[\cdot]$ is the expectation. If we apply the expectation on each side of A.1, we get that $(1 - \sum_{i=1}^{p} \phi_i)\mu = \mathbb{E}[\varepsilon_t]$. For any positive lag τ, we can evaluate the autocorrelation function of each time series. Let r be the empirical counterpart of the autocorrelation function of the time series, then $r(\tau) = \sum_{i=1}^{p} \phi_i r(\tau - i)$, for any lag $\tau \geq 1$. Since the The autocorrelation function is even, i.e., $r(-\tau) = r(\tau)$, we obtain the matrix form of the Yule-Walker equations:

$$r = R\phi,$$

where $r = [r(1)\, r(2) \cdots r(p)]^\top$, $\phi = [\phi_1\, \phi_2 \cdots \phi_p]^\top$, and

$$R = \begin{bmatrix} r(0) & r(1) & \cdots & r(p-1) \\ r(1) & r(0) & \cdots & r(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(p-1) & r(p-2) & \cdots & r(0) \end{bmatrix}.$$

Assuming that the $p \times p$ symmetric matrix R is invertible, the coefficients α are estimated by

$$\phi = R^{-1} r.$$

Once the coefficients are estimated, the AR model can be applied to predict future samples, with

$$X_k = \sum_{i=1}^{p} \phi_i X_{k-i}.$$

While this technique is easy to implement, it is not adapted for nonlinear systems [56].

b. **Least Squares Estimation**

Alternatively, the AR coefficients can be estimated using least squares, where we minimize the sum of squared residuals:

$$\min \sum_{t=1}^{n} \left(X_t - \sum_{i=1}^{p} \phi_i X_{t-i} \right)^2$$

This can be solved using numerical methods or standard regression techniques.

Appendix B

Technical Datasheet of Decagon's ECH2O EC-5 Soil Moisture Sensor

Measurement Time	10 ms (milliseconds)
Accuracy	At least 0.03 m^3/m^3 for all soils, up to 8 dS/m With soil-specific calibration: ± 0.02 m^3/m^3 (±2%)
Resolution	0.001 m^3/m^3 VWC in mineral soils, 0.25% in growing media
Power Requirements	2.5VDC - 3.6VDC @ 10mA
Output	10-40% of excitation voltage (250-1000mV at 2500mV excitation)
Operating Environment	-40 to +60°C
Range of Measurement	0 to saturation
Sensor Dimensions	8.9 cm x 1.8 cm x 0.7 cm
Cable Length	5 m standard, custom lengths or extension cables available
Connector Types	3.5 mm plug or stripped and tinned lead wires
Datalogger Compatibility (not exclusive)	**Decagon:** Em5b, Em50 (and radio-enabled versions) **Campbell Scientific:** CR10X, 21X, 23X, CR1000, CR3000, etc.