EAI Endorsed Transactions
on Industrial Networks and Intelligent Systems
Review Article EAI.EU

IoT-Enabled Wireless Sensor Networks in Precision Agriculture: A Survey on Cross-Layer Resilience and Fault Management

Nuwan Jayawardene[1], Sulochana Sooriyaarachchi[1], Chandana Gamage[1]

[1]Department of Computer Science and Engineering, University of Moratuwa, Moratuwa, 10400, Western Province, Sri Lanka

Abstract

Precision Agriculture (PA) utilizes Wireless Sensor Networks (WSNs) to continuously monitor and collect data on both plants and machinery. Thereafter, data-driven proactive decision-making is used for crop protection, yield optimization, and other agriculture management tasks. These WSNs rely on clustering protocols to facilitate sensor node communications. However, the adverse, spatio-temporal nature of agricultural environments and the need for real-time operation pose severe challenges that manifest as physical and protocol-level network faults. This survey adopts a novel perspective of *Cross-Layer Resilience in Spatio-Temporal Agricultural IoT* to evaluate how hardware-level physical faults cascade into routing failures. We survey a range of clustering protocols and fault management frameworks devised to address these faults using proactive and reactive techniques, ultimately enabling sensor networks to maintain survivability and continuous quality of service.

Received on 05 January 2026; accepted on 15 June 2026; published on 22 June 2026

Keywords: Precision Agriculture, Wireless Sensor Networks, Clustering Protocols, Fault Management Frameworks

doi:10.4108/eetinis.132.11509

1. Introduction

Wireless Sensor Networks (WSN) constitute of number of independent sensing nodes communicating via wireless media with at least one data sink or Base Station (BS) for data aggregation [1]. WSNs are widely used in a variety of domains such as in smart homes, intelligent utility metering, precision agriculture (PA) and industrial automation for data gathering due to their versatility.

PA is a subdomain of Internet of Things (IoT) that adopts WSNs to gather and analyze data generated in an agricultural setting. PA enables proactive decision making that leads to an optimal harvest, reduction of human labor costs, and efficient use of agricultural inputs.

WSNs in PA are customized for each deployment due to the diversity of crops, cultivation styles and environments. The resulting systems cover data collection, diagnostics, data analysis, precision field operation and evaluation [2]. Network survivability is prioritized when deploying WSNs in PA with network lifetime and quality of service being key concerns.

Use cases of WSNs in PA are often in harsh environments and are exposed to a multitude of adverse scenarios when deployed in the field. These conditions can cause failures in different aspects of WSNs such as node-to-node communication, node neighborhood knowledge and sensor readings. WSN failures can result in severe losses if left unaddressed for even short periods as PA is heavily reliant on the continuous availability of data.

Clustering protocols are used to ensure consistent device communication during network faults while maintaining network longevity. These protocols focus on achieving network load balancing and effective node power management. Robust hardware construction is used to mitigate most hardware failures. However, hardware robustness is insufficient to achieve network survivability that has faults emerging as protocol

*Corresponding author. Email: nuwan.19@cse.mrt.ac.lk

failures. Fault management frameworks are used to address emerging faults upon detection. A variety of fault-tolerant techniques utilizing reactive and proactive measures are utilized for this purpose.

This survey adopts a novel perspective termed *Cross-Layer Resilience in Spatio-Temporal Agricultural IoT*. This perspective frames WSNs as dynamic systems where physical agricultural constraints such as vast spatial spread, seasonal temporal changes, and harsh weather conditions directly dictate both network topology and hardware degradation. By utilizing this lens, this paper evaluates clustering protocols and fault management frameworks based on their combined ability to prevent hardware-level physical faults from cascading into protocol-level routing failures, thereby ensuring robust network survivability.

Accordingly this survey explicitly defines the following research questions (RQs):

- **RQ1:** How do prevailing clustering protocols for WSNs in precision agriculture balance energy efficiency with network scalability in resource-constrained environments?
- **RQ2:** In what ways do the unique spatio-temporal and environmental constraints of agricultural deployments contribute to the manifestation of network faults?
- **RQ3:** Which fault management frameworks are most effective in isolating cross-layer fault propagation and maintaining continuous quality of service (QoS) in PA WSNs?

The paper is structured such that section 2 describes different types of WSN implementations in PA. Section 3 presents different fault types encountered by the WSNs. Section 4 reviews a range of clustering protocols used for inter-node communication. Section 5 provides an overview of group-based clustering protocols followed by section 6 describing a variety of chain-based protocols. Section 7 gives an overview of fault management fundamentals. Fault management frameworks based on said fundamentals are covered in section 8. An overall evaluation of clustering protocols and fault management frameworks is covered in section 9. The paper is concluded in section 10 with a summary.

2. Wireless Sensor Networks

This section discusses the architecture of WSNs in PA and real-world implementations of the architecture. Common features of WSN architectures include having spatially distributed sensor nodes and one or more network sinks. However, single network sink WSNs architectures are the most dominant. The network sink aggregates sensor data from nodes. Nodes get formed into a logical arrangement known as a topology for efficient communication with the network sink.

WSNs can be formed into centralized or distributed networks [3]. In centralized formation, the network is managed by the network sink, which acts as a central device responsible for tasks such as data processing and network management. In distributed formation, decision-making is performed locally by the nodes themselves without any reliance on a central device. Characteristics of distributed networks are self-organization and autonomous node behavior.

Topologies found in WSNs are logically separated into distinct layers, that form a hierarchical architecture as shown in figure 1. Each layer in the architecture has a specific purpose. Sensed data flow from the bottom-most layer to the top whereas topology management metadata for configuration and control flow from top to the bottom, with status metadata flowing in the opposite direction.

2.1. Architecture of WSNs in PA

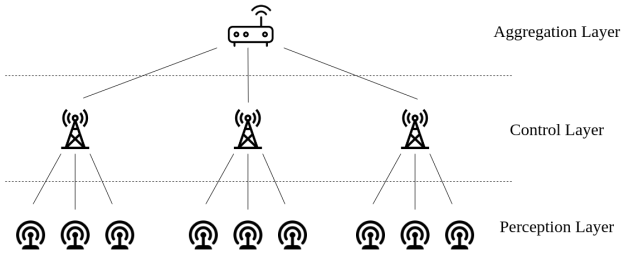

Figure 1. Generic hierarchical architecture for WSNs in PA

In figure 1, the layer referred to as the perception layer, consists of sensor nodes deployed in the field. These sensors include wireless communication hardware as well as basic processing and data storage capabilities. In a majority of WSNs, this layer consists of the most number of devices. These devices can either be heterogeneous or homogeneous in their capabilities. For management purposes, these devices can be logically grouped together into clusters as discussed in Section 4. This management functionality, known as clustering, is delegated to the control layer.

The control layer consists of devices either built specialized to manage the network in a heterogeneous scenario or elected by a certain process to manage the network in a homogeneous scenario. In the heterogeneous scenario, the devices could be equipped with more capable hardware, such as larger batteries and more processing power. These devices will be allocated to manage individual clusters, and they will be referred to as Cluster Heads (CH).

Data collected by the control layer is sent to the sink in the aggregation layer. Thereafter, the data can be

retrieved directly from the sink or uploaded to a remote server for network retrieval.

Many real-world WSN implementations in PA have been developed based on this hierarchical architecture. However, some aspects of the architecture are modified to address requirements in a particular PA use case.

2.2. WSN implementations in PA

Most often, WSNs for PA need to cover a large spatial region. However, in most WSNs, effective communication with the network sink is hindered by insufficient communication range. Such networks have mobile sinks or data haulers that traverse the agricultural land for data collection and transmission. The work of Khan et al. [4] and Gupta et al. [5] utilize a mobile ground vehicle as a sink to access a selected set of nodes (for example, a row of nodes in a network arranged in a grid pattern as shown in figure 2) in a cultivation area with each node in the row communicating with an associated column of sensor nodes embedded in the agricultural land. The mobile sink reduces the total energy usage of the network of nodes by reducing the transmission distance and also reduces connection failures. The ground vehicle is susceptible to the obstacles and dynamic elements presented by an agricultural setting, such as fallen branches and interference by humans or animals. This problem is solved by using Unmanned Aerial Vehicles (UAVs) as the data hauler platform.

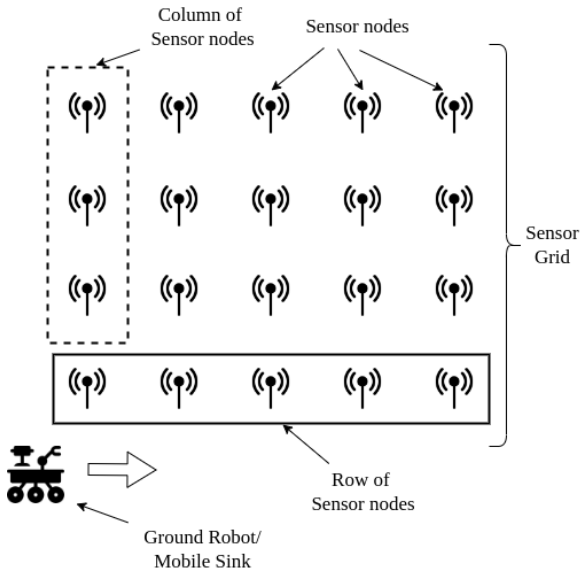

Figure 2. Nodes arranged in a grid with mobile ground sink

Valente et al. [6], Moribe et al. [7] and Shafi et al. [8] have opted for an UAV as a mobile sink. Their deployments consist of ground-based static sensor nodes in the perception layer and control layer while the UAV operates in the aggregation layer performing the role of a data hauler. Valente et al. [6] scattered their WSN in three separate zones with number of nodes covering each zone. To link up these separate zones a larger number of static sinks would be required when the geographical spread of zones is higher. In contrast, mobile sink overcomes geographical limitations, increases the coverage area and also reduces energy used by nodes as less power is needed to transmit data to the sink.

Node communication is another essential aspect of WSNs in PA. Commonly used protocols include ZigBee [9], LoRaWAN [10] and Z-Wave [11] at the physical and MAC layers. Wireless sensor node platforms integrate the transceiver hardware used for node communications, environmental sensing hardware, and a power source. Sensor node platforms commonly used in PA are MICAz, TelosB, and Imote2. These platforms come equipped with the Texas Instruments CC22420 transceiver and operate using AA batteries [12].

This spatio-temporal reality, where nodes must cover vast physical distances while weathering seasonal changes and unpredictable animal interference, serves as the primary catalyst for the cross-layer network deterioration discussed in subsequent sections.

3. Network Faults in WSN - [RQ2]

This section discusses the types of faults encountered by WSNs in PA environments.

3.1. Fault Taxonomy

WSN faults can be caused by a variety of reasons such as network design issues, incorrect network setup, and failure of hardware components during network lifetime [13]. Moridi et al. [13] and Muhammed et al. [14] have aggregated state-of-the-art research on WSN arrangements and classified fault types into multiple categories.

To align with this survey's perspective of *Cross-Layer Resilience in Spatio-Temporal Agricultural IoT*, the standard taxonomy has been adapted and visually reorganized, as shown in Figure 3. The adapted taxonomy introduces a **Spatio-Temporal Catalyst** at the root, acknowledging that harsh environments, weather, and animal interference act as the primary triggers for network failures in PA. Furthermore, to demonstrate how physical damage cascades into logical failure, the taxonomy groups faults into two distinct structural layers:

- **Physical Layer (Hardware Degradation):** Encompasses system-centric and hardware component failures (e.g., battery depletion, sensor damage) directly caused by environmental exposure.

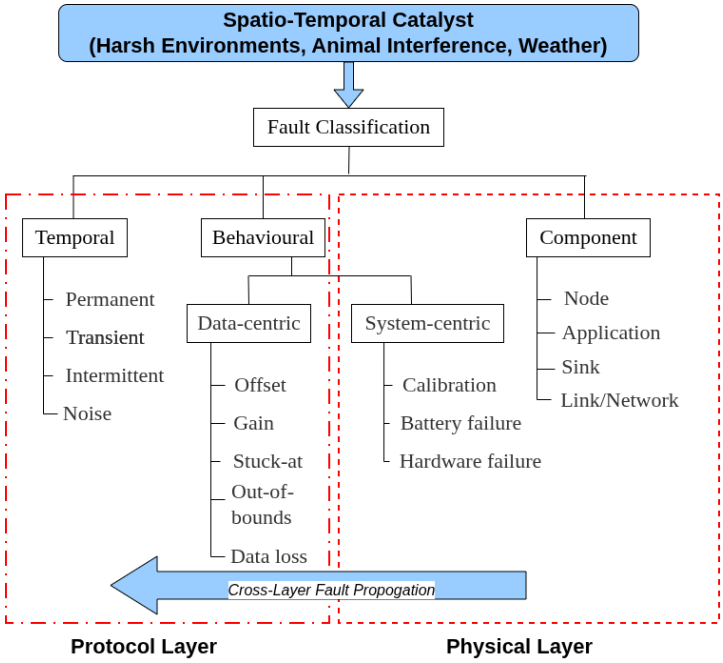

Figure 3. WSN fault taxonomy

- **Protocol Layer (Logical Failures):** Encompasses data-centric anomalies (e.g., gain, offset) and network/routing failures that manifest as a consequence of the underlying physical degradation.

The core challenge in maintaining network survivability lies in mitigating the **Cross-Layer Fault Propagation** from the physical layer to the protocol layer. An overview of each specific fault type within this framework is as follows:

Temporal faults. Faults categorized based on time duration are referred to as temporal faults. Temporal faults that result in the deactivation of nodes are permanent faults. Permanent faults are often the result of node failures. Due to this reason, they are also referred to as hard faults.

When a node outputs incorrect data while not being deactivated or disconnected, it is categorized as either a transient or an intermittent fault. Transient faults are temporary and could be the result of environmental changes. Intermittent faults develop over a longer duration and have a consistent frequency. It is possible for a transient fault to automatically get resolved or worsen to become an intermittent fault. The consistent frequency makes intermittent faults easier to recognize and diagnose compared to transient faults, which can be isolated events.

Behavioural faults. Anomalous behavior of a WSN component is categorized as a behavioral fault. Detection is based on the locality of the fault and is categorized as data-centric or system-centric. Behavioral faults can also be categorized as hard and soft faults based on their severity.

Data-centric faults are observed in sensed data across a time period [14]. They are also referred to as soft faults due to the lower severity and ease of diagnosis. The function $f(t)$ denotes the value that is sensed by the node n at a specific time instance t. $f(t)$ can be represented by the equation $\alpha + \beta x + \eta$, where α is an offset constant, β is a gain constant, x is the non-faulty sensor value at time t, and η represents the environmental noise in the data. In a real-world scenario, a node that is free from faults will display a sensed parameter of $f(t) = x + \eta$.

Offset faults occur when the readings of sensed data deviate from the average by an additive constant. This could be the result of miscalibrated sensor or a component failure. Offset faults can be represented as $f(t) = \alpha + x + \eta$ where α is the constant value that is added. Figure 4 represents offset fault on a time series.

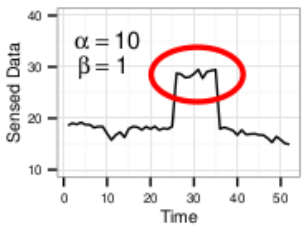

Figure 4. Offset fault time series [14]

When the rate of change of sensed data does not match expectations over an extended period, it is

classified as a gain fault. Gain faults are the result of the final value being a product of a constant value. They can be caused by a miscalibrated sensor or a failing component. Gain faults can be represented as $f(t) = \beta x + \eta$, where β is the multiplicand. Figure 5 showcases a gain fault on a time series.

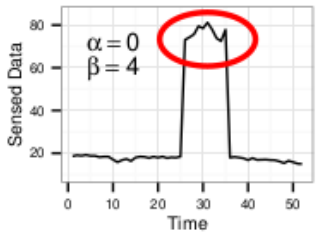

Figure 5. Gain fault time series [14]

When the sensed value does not vary over an extended period of time, it is a stuck-at fault. These faults can be intermittent, transient, or persistent. Stuck-at faults are represented as $f(t) = \alpha$ where α is the value that is sensed. These faults can be further classified as stuck-at-one or stuck-at-zero. This depends on whether the unvarying value is the maximum value sensed by the node or zero (no reading), respectively. Figure 6 represents a stuck-at fault on a time series.

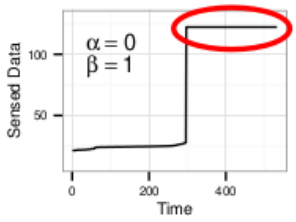

Figure 6. Stuck-at fault time series [14]

An out-of-bounds fault is identified when the sensed data lies beyond the thresholds set by the node specification. It can be modeled as $f(t) > \theta$ and $f(t) < \theta_1$ where θ and θ_1 are thresholds.

When sensed data is missing from a time series, that is categorized as a data-loss fault. Figure 7 represents data-loss fault on a time series.

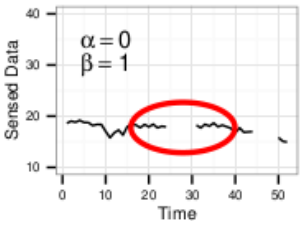

Figure 7. Data-loss fault time series [14]

System-centric faults can manifest as calibration failures, battery failures and hardware failures. Calibration errors can also give rise to data-centric faults such as gain and offset faults. If left unaddressed, it can lead to cross-layer fault propagation as shown in figure 8.

Hardware failures are commonly observed in sensor nodes due to constant exposure to harsh environmental conditions. Failures involve sensors, transmission antennas, built-in memory/storage and central processing units. Hardware faults are permanent and, depending on severity, will require a component replacement or replacement of the entire sensor node.

Battery failures are also a type of hardware failure that can manifest as sudden sensor node shutdowns and failure to transmit or receive messages. Tolles et al. [15] have concluded that most anomalous sensor readings are caused by battery failures. It will result in the manifestation of data-centric faults such as stuck-at faults and data-loss faults.

Considering the variety of faults with different severities that can be present in a WSN, several techniques are used to address each type. Fault management is used for this purpose.

Component faults. WSNs in PA are faced with dynamic events such as animal interference, weather phenomena, and human activities. This leads to network component failures during the lifetime of the WSN. Component faults can belong to node, network, sink, and application subcategories.

Nodes in a WSN are composed of hardware and software. Node hardware includes batteries, sensors, data processing electronics and signal transceivers. The software comes in the form of firmware that can receive data, collect sensor data, and transmit data. Even the most robustly built sensor nodes are still susceptible to faults, which can be especially harmful when the affected nodes are CHs.

Viewed through the lens of cross-layer resilience, agricultural WSNs are uniquely vulnerable to hardware-to-routing fault cascades as shown in figure 8. For example, severe weather (a spatio-temporal environmental constraint) might cause a slight battery failure (hardware layer), which leads to missed data transmissions (MAC layer), ultimately resulting in cluster disintegration and routing protocol failure (network layer). Therefore it is crucial to identify faults and address them at the point of origin.

As discussed in section 2, the network sink is a crucial component of a WSN, which is tasked with data aggregation, network management, and uploading data to a remote server. Failure of a sink can result in the complete loss of data collected over an extended period and the breakdown of communication with sensor nodes.

Wireless communication is used by nodes to transfer data, update network configurations, and coordinate node activity, and faults that hinder such processes are

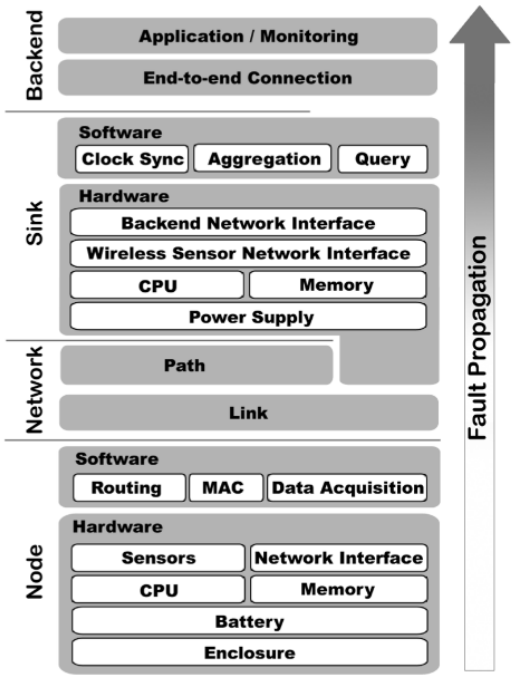

Figure 8. Fault propagation across hierarchical stack [16]

referred to as network faults. Routing protocols define how nodes communicate and how data is disseminated through the network [17]. A fault in the routing layer can create malformed packets, communication delays, and even data loss, and can be caused by radio interference, obstructions in the communication path, or even hardware failure. WSNs in PA adopt hierarchical-type routing, with clustering being the most dominant form.

4. Clustering Protocols - [RQ1]

Clustering is the process of grouping nodes in WSNs and it supports multiple objectives such as evenly distributing energy usage of a network among sensor nodes, deploying a mixed set of nodes with differing capabilities, application-driven placement of nodes in the physical environment [18] [19].

A leader node referred to as Cluster Head (CH) is allocated from each group to manage node activities within that cluster. CHs are responsible for transferring of sensed data received from non-cluster head nodes (NCH) to a data sink. Clustering operates at the perception and control layers shown in figure 1.

The primary methods of clustering are; group-based, chain-based and tree-based [20] as shown in figure 9.

In the context of cross-layer resilience, clustering protocols serve as the first line of defense against spatio-temporal agricultural constraints. By dynamically restructuring the topology, such as electing new CHs when existing ones fail due to environmental degradation, these protocols act as a buffer, preventing localized physical faults from causing total network failure. To reflect this, Figure 9 has been adapted to categorize these protocols based on their spatial defense mechanisms:

- **Group-Based Protocols (Localized Spatial Defense):** These protocols operate as single-hop networks within their clusters. They provide intra-cluster resilience, where the failure of one sensor node in a harsh micro-climate can be quickly mitigated by another node within the same localized group taking over its routing duties.
- **Chain-Based Protocols (Extended Spatial Defense):** These protocols operate via multi-hop mechanisms. They provide resilience over vast agricultural distances, ensuring that even if large geographic swathes of the network are damaged by weather or animal interference, data can still be routed through an extended, collaborative chain of remaining nodes.

Group-based clustering employs controlling devices in the control layer to manage subgroups of sensor nodes within the perception layer. Some networks have specialized nodes that are equipped with batteries with higher capacities and transceivers with greater range. These networks with diverse nodes are known as heterogeneous group-based WSNs. In homogeneous group-based clustering protocols, where all the nodes are of similar capability, CHs can be elected through an algorithmic process. In heterogeneous group-based clustering the nodes with greater range and power are naturally selected as CHs.

Chain-based clustering does not involve heterogeneous nodes and therefore are able to algorithmically determine leader nodes and the optimal path for packet transfer.

5. Group-based Clustering Protocols - [RQ1]

This section discusses a set of hierarchical group-based clustering protocols utilized by WSNs and analyzes the protocols used in their formation. A visual representation of a group-based clustering protocol is given in figure 10 where each cluster has its own leader while the full network has a single sink. The selected set of protocols are some of the most widely cited in the domain.

5.1. Low Energy Adaptive Clustering Hierarchy (LEACH)

LEACH is the most commonly cited homogeneous group-based clustering protocol [21] and has introduced dynamic clustering via rotation of CH among

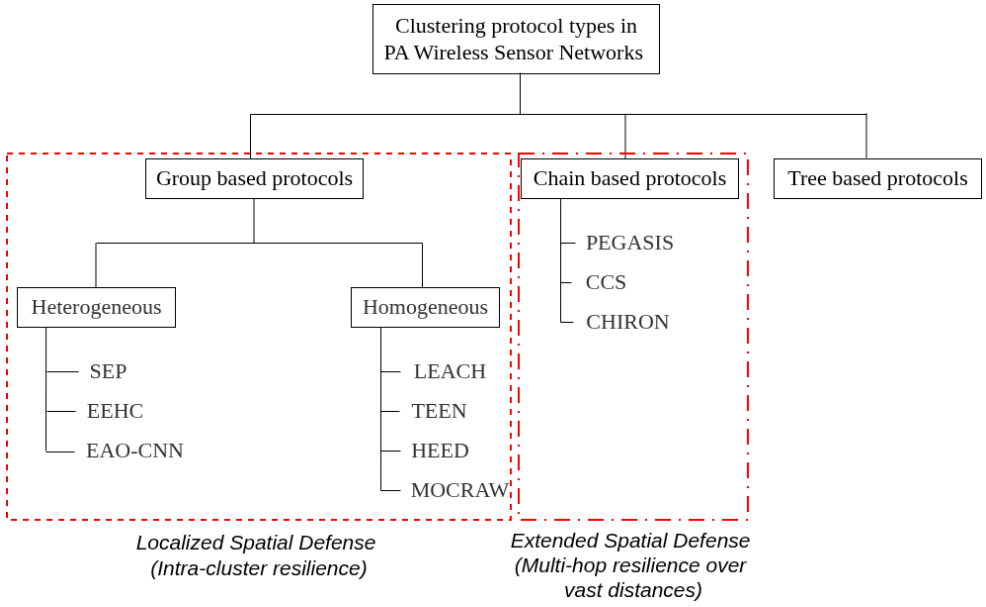

Figure 9. Clustering protocol taxonomy

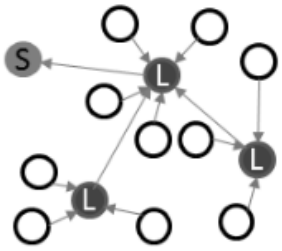

Figure 10. Group based clustering of WSN (S-Sink, L-Leader)

nodes in the network. CH rotation allows energy consumption to be balanced amongst sensor nodes, prolonging the lifetime of the network. LEACH uses the single-hop pattern to transmit data between devices. The protocol takes a decentralized approach to clustering by allowing nodes to self-elect as CHs by using an algorithm where nodes operate entirely independently. Decentralization increases scalability as there is no central backbone limiting the number of sensor nodes in the network.

The LEACH algorithm operates using a series of intervals. An interval is the duration within which all nodes in the network get elected as CH at least once. Each interval consists of sub-intervals that run on loops of two phases; setup and steady-state. This arrangement is visually represented in figure 11.

CHs are self-elected in the setup phase. Election is performed by having each node generate a random number N between zero and one which N is compared against a threshold value. The threshold value is generated using equation 1.

$$T(x) = \begin{cases} \dfrac{P}{1-P\left(r.mod\frac{1}{P}\right)}, & if\, x \in W \\ 0, & otherwise \end{cases} \quad (1)$$

where P is the percentage of CHs in the entire network, r is the sub-interval number and W is the set of sensor nodes not elected as CHs. Due to the random nature by which each node decides on its CH status, after each round there may be a number of CHs that is above or below the value P. Upon self-election, CHs broadcast their election to non-cluster head (NCH) nodes within range. NCH nodes determine which of the CHs are closest via the Received Signal Strenght Indicator (RSSI) value of the broadcasts and then they join the cluster lead by the nearest CH by replying to the broadcast message. The CH receives all acknowledgment messages from NCH nodes and creates a Time Division Multiple Access (TDMA) schedule that informs NCH nodes when to transmit data. This TDMA schedule is then broadcast to NCH nodes, concluding the setup phase.

During the steady-state phase, sensed data is sent to CHs by NCH nodes using the TDMA schedule. Finally, data collected by the CH is transmitted to the sink. The steady-state phase concludes after a predetermined time period. The network thereafter reverts back to the setup phase and restarts the process.

The primary limitation of LEACH is the random CH election process which does not consider parameters such as the residual energy of a node. This may lead to the election of nodes that are running low on battery which can result in network performance being

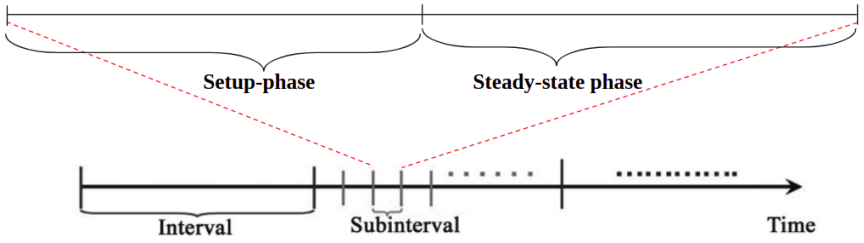

Figure 11. LEACH protocol operational phases

affected. Despite this limitation, the uncomplicated and straightforward design of LEACH has made it a foundation for more modern protocol variants since its introduction in [21].

5.2. Hybrid Energy Efficient Distributed (HEED) Protocol

HEED is another protocol that uses CH rotation. Compared to LEACH [21], it uses additional parameters such as residual energy and intra-cluster communication cost when electing CHs [22]. The secondary parameter is also used as a tiebreaker when two or more CHs are competing for a NCH node's connection. Similar to LEACH, nodes in HEED determine CH eligibility independently and uses a broadcast-response mechanism to connect to CHs.

The probability of a node getting elected as a CH, denoted by CH_{PR} is given in equation 2 where C_{PR} denotes the initial percentage of cluster heads among all n nodes, E_{max} is the maximum energy and E_{res} is residual energy.

$$CH_{PR} = \frac{E_{res}}{E_{max}} C_{PR} \tag{2}$$

Once the elected CH has performed data collection, it transmits data to the sink in a multi-hop arrangement via other CHs as shown in figure 12. This multi-hop arrangement enhances network scalability and coverage area at the cost of increased network overhead and energy loss in elected CHs. HEED has also been continuously developing since its introduction and has multiple variants focusing on different aspects of the protocol design [23].

5.3. Stable Election Protocol (SEP)

Smaragdakis et al. [24] shows that classical protocols such as LEACH cause networks to die prematurely when sensor nodes with varying energy capacities are present at the beginning. SEP addresses this by introducing what the authors refer to as heterogeneous nodes; standard and advanced, to the network. A node is categorized as advanced or standard depending on its relative energy level. A probability model is used to make this distinction.

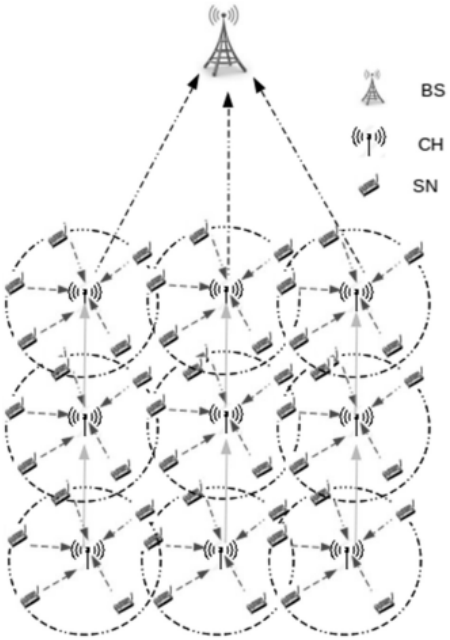

Figure 12. HEED cluster communication approach [23]

The probability equations for standard or normal nodes (P_{nrm}) is given in equation 3 and for advanced nodes (P_{adv}) in equation 4 where P_{opt} is the optimal CH election probability, a is the additional energy factor between advanced nodes and standard nodes, and m is the fraction of advanced nodes in the network.

$$P_{nrm} = \frac{P_{opt}}{(1 + a.m)} \tag{3}$$

$$P_{adv} = \frac{P_{opt}}{(1 + a.m)}(1 + a) \tag{4}$$

Similar to LEACH, the threshold generation for CH self-election is achieved using equation 1. It is adapted to generate threshold values for normal nodes and advanced nodes by substituting the P value with P_{adv} and P_{nrm} from equations 4 and 3 respectively.

SEP operates by having NCH nodes periodically communicate their remaining energy levels in messages sent to CHs during the handshaking process. The CHs then relay this information to the sink along with acquired field data. The energy level data would be aggregated by the sink to determine node classifications and the sink would broadcast the classification data back to the CHs which will update nodes in their respective clusters during the initial broadcast acknowledgment exchange.

The SEP is a centralized heterogeneous protocol as it relies on the sink for node classification. This creates a single point of failure which can affect network performance during a sink failure.

5.4. Energy Efficient Heterogeneous Clustered (EEHC) Scheme

EEHC builds upon the heterogeneous design of SEP and introduces an additional sensor type called super [25] along with normal and advanced. Similar to SEP, node energy levels are used for this classification. Further, it assigns a weight to the optimal CH probability P_{opt} calculated by dividing the initial energy of each node by the initial energy of a normal node. The clustering phase is similar to the setup phase of LEACH with the exception of the CH determination process. Weighted probabilities to be elected as CH of the three types of nodes as normal (P_{nrm}), advanced (P_{adv}) and super (P_{sup}) are calculated via equations 5, 6 and 7 respectively.

$$P_{nrm} = \frac{P_{opt}}{1 + m(\alpha + m_o\beta)} \tag{5}$$

$$P_{adv} = \frac{P_{opt}}{1 + m(\alpha + m_o\beta)}(1 + \alpha) \tag{6}$$

$$P_{sup} = \frac{P_{opt}}{1 + m(\alpha + m_o\beta)}(1 + \beta) \tag{7}$$

where m is the percentage of advanced and super nodes in the network making $(1 - m)$ the percentage of normal nodes; $m \cdot m_o$ is the percentage of super-nodes in the network, and $m(1 - m_o)$ is the percentage of advanced nodes in the network; and α and β are additional energy factors of advanced and super nodes.

Node election is achieved by passing a threshold value using the same equation as equation 1 where CH percentage value P can be replaced with the relevant CH election probability values for the three categories of nodes.

Similar to SEP, in EECH the node classification is done by the sink. This creates a single point-of-failure as sink nodes are vulnerable to failure.

5.5. Threshold-sensitive Energy Efficient Network (TEEN)

TEEN is a reactive clustering protocol for WSNs that react immediately to sudden environmental changes detected via sensed data [26]. In contrast to proactive protocols that get activated periodically to transmit collected sensed data, reactive protocols are best suited for data transmission in time-critical use cases such as temperature monitoring in an oil refinery which can have fatal results in a hazardous situation if readings are delayed.

TEEN uses a combination of soft threshold and strong threshold values to determine when data should be transmitted. The definitions for the threshold values are as follows;

- Soft Threshold is a minor deviation in the sensed value relative to the previously stored sensor data which triggers transmitter activation. To achieve this, nodes continuously monitor their environment to detect changes.
- Hard Threshold is an absolute value which if exceeded warrants data transmission to the CH.

The biggest downside of TEEN is its specialization as a reactive network. While it satisfies the requirement for a very specific set of use cases, it is not well-suited for most others that require proactive environmental monitoring.

5.6. Distance, Energy, and Density-Aware SOM Clustering-Based Routing (DEDSCR)

Recent advancements in 6G and IoT-enabled WSNs have driven the development of highly sophisticated clustering and routing techniques that leverage Machine Learning (ML). These approaches often synthesize multiple optimization methods to address the simultaneous challenges of cluster formation, CH selection, and optimal path routing under strict resource constraints.

A prominent example of this is the Distance, Energy, and Density-aware Self-Organizing Map Clustering-Based Routing (DEDSCR) scheme proposed by Le et al. [27]. DEDSCR utilizes a synergistic approach of unsupervised learning and meta-heuristic optimization to maximize network lifespan. The scheme applies the total sum of squared distance-based elbow (SDE) method to dynamically identify the optimal number of clusters, followed by a Self-Organizing Map (SOM) neural network to distribute and organize the sensor nodes evenly across the defined clusters.

For CH selection, DEDSCR implements a fuzzy logic model that evaluates the distance to the base station (BS) alongside the remaining energy of the nodes. Once the clusters and CHs are established, the fuzzy

model is used again to compute node priority values based on distance, possible energy expenditure, and local normalized density. These fuzzy outputs act as heuristic values for an Ant Colony Optimization (ACO) algorithm, which then determines the absolute optimal multi-hop routing paths from the CHs to the BS. This comprehensive consideration of distance, energy, and density (DED) using ML and fuzzy models ensures a highly balanced energy consumption rate, drastically outperforming traditional protocols like LEACH and K-means based approaches in large-scale deployments.

5.7. Meta-heuristic Optimized Cluster Head Selection-based Routing Algorithm (MOCRAW)

The Meta-heuristic Optimized Cluster-head selection-based Routing Algorithm for WSNs (MOCRAW) represents a significant evolution of earlier clustering protocols such as LEACH and HEED. While LEACH relies on randomized Cluster Head (CH) election and HEED introduces residual energy into the selection process, MOCRAW employs advanced meta-heuristic optimization to address their inherent limitations [28]. A major weakness of traditional group-based protocols is the "isolated node" or hot-spot problem, where uneven energy dissipation leads to nodes dying quickly and becoming disconnected from the network.

To resolve this, MOCRAW utilizes the Dragonfly Algorithm (DA), incorporating both Local Search Optimization (LSO) and Global Search Optimization (GSO) to provide loop-free and highly efficient routing. The protocol operates using two primary sub-processes: the Cluster Head Selection Algorithm (CHSA) and the Route Search Algorithm (RSA). The CHSA utilizes an Energy Level Matrix (ELM) that calculates optimal CH placement based on node density, residual energy, the distance between the CH and the Base Station (BS), and inter-cluster formation metrics. Simultaneously, the RSA discovers the most optimal inter-cluster paths from the source to the destination using Levy distribution.

By dynamically optimizing CH selection and routing paths, MOCRAW explicitly minimizes node energy consumption and ensures fast data transmission. This targeted optimization significantly extends overall network longevity, making MOCRAW highly applicable to Precision Agriculture (PA) WSN deployments.

5.8. Multi-Objective Optimized Clustering and Deep Learning Model

Building upon the concepts of node heterogeneity established by SEP and EEHC, recent advancements have focused on multi-objective clustering approaches tailored explicitly for the demanding environments of PA. A significant limitation of traditional protocols is that they often optimize a single metric, such as residual energy, without holistically addressing the complex interplay of factors that lead to premature network failure. To address this, Pandiyaraju et al. introduced a multi-objective clustering model that jointly optimizes energy utilization and data transmission paths specifically for PA [29].

This approach utilizes a multi-objective hybrid optimization technique called the Election-based Aquila Optimizer (EAO), which combines the Aquila Optimizer (AO) with the Election-Based Optimization Algorithm (EBOA). The CH selection process evaluates several factors simultaneously: energy consumption, delay, distance, communication overhead, inter-cluster distance, and intra-cluster distance. By evaluating this multi-objective fitness function, the network can avoid the single points of failure and bottlenecking issues common in purely heterogeneous networks like SEP.

Furthermore, this model integrates the newly developed EAO with an Optimized Convolutional Neural Network (O-CNN). Once a subset of nodes satisfies the multi-objective criteria, the O-CNN acts as a deep classifier to select the absolute best candidate for the CH role. This combination of meta-heuristic optimization for routing and deep learning for node classification allows the WSN to achieve superior performance metrics. Experimental evaluations demonstrated that this model achieved an impressive 99.23% classification accuracy, 99% packet delivery ratio, and a network lifetime of 98.24%, all while limiting maximum energy consumption to 50%. These results highlight the efficacy of combining multi-objective clustering with deep learning to maximize crop yields while minimizing energy expenditure in PA environments.

Table 1 lists a summary of the studied group-based protocols.

6. Chain-based Clustering Protocols - [RQ1]

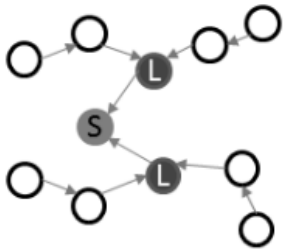

Figure 13. Chain-based clustering of WSN (S-Sink, L-Leader)

Hierarchical chain-based protocols utilize a collaborative strategy to pass packets. A visual representation of a WSN utilizing chain-based clustering is shown in figure 13. Nodes self-organize into logical chains that will allow packet forwarding to neighbor nodes in the chain until it reaches the intended recipient [30]. Some

Table 1. Summary of group-based clustering protocols

Proto.	CH election parameter			Hop pattern		Node Type	
	Random	Residual Energy	Comm. cost	Single	Multi	Homogeneous	Heterogeneous
LEACH [21]	✓			✓		✓	
HEED [22]		✓	✓		✓	✓	
SEP [24]		✓			✓		✓
EEHC [25]		✓			✓		✓
TEEN [26]		✓		✓		✓	
MOCRAW [28]		✓	✓		✓	✓	
EAO-CNN [29]		✓	✓		✓		✓

of the commonly cited chain-based protocols are as follows:

6.1. Power-efficient Gathering in Sensor Information Systems (PEGASIS)

PEGASIS is a multi-hop protocol in which nodes collaborate to form a chain to pass data to the sink [31]. It has been proposed as an improvement over LEACH and the centralized version of the LEACH protocol (LEACH-C) [32]. Chain formation is computed either centrally by the sink or independently by the nodes themselves. When independently determined, all nodes will be aware of the chain structure as they share global knowledge of the network state and utilize a greedy algorithm to identify the neighbor node to pass packets.

A chain is constructed starting from the network edge with each subsequent node identifying its closest neighbor using the RSSI value of received messages. A completed chain is shown in figure 14. During data transmission, each node merges its own data with the data received from the preceding neighbor in a process known as Data Fusion. For example in figure 14, data from node C_0 will be merged with data from C_1 and transferred to the sink via the chain leader; C_2. Leaders are rotated randomly to balance out the energy usage of the nodes.

A notable downside of PEGASIS is its reliance on a single chain for data transmission, which can form bottlenecks during packet delivery.

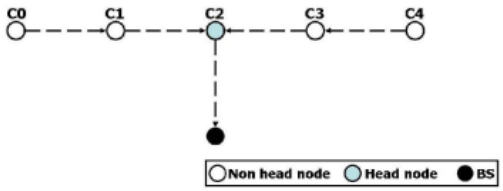

Figure 14. Message passing in PEGASIS [33]

6.2. Concentric Clustering Scheme (CCS)

CCS is a multi-hop protocol [33] proposed as an enhancement of PEGASIS. CCS considers the location of the sink to determine optimal leader nodes which reduces the formation of redundant chains that can take data further away from the sink as shown in figure 15.

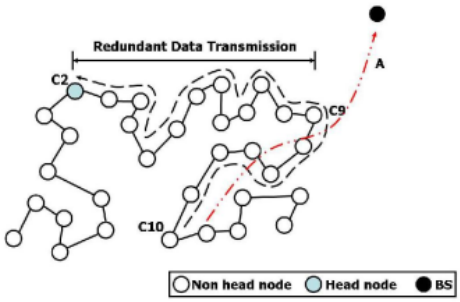

Figure 15. Redundant chain formation with inefficient message flow [33]

In the example network shown in figure 15, node C_2 is the selected leader node which will be passing a message from node C_{10} onto the sink. However, the optimal node for the role of leader node should be C_9 as it is closer to the sink. In protocols such as PEGASIS which is unaware of sink location, the section of the chain beyond C_9 is rendered redundant.

CCS divides the network into concentric arc-shaped levels as shown in figure 16 with a chain constructed in each using a greedy algorithm. Leader nodes are elected from each level to transmit data to the sink in a multi-hop arrangement which enables CCS to avoid redundant chains.

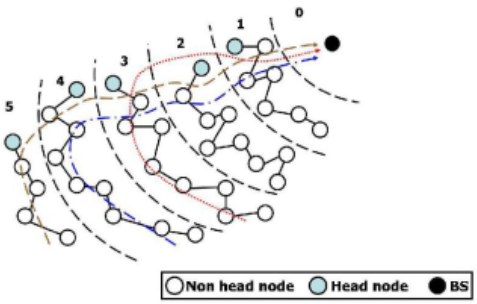

Figure 16. Multi layered CCS network [33]

However, this arrangement can result in the overuse of leader nodes closest to the sink. In addition to that,

not considering residual energy levels of nodes during the clustering setup is another notable downside of CCS.

6.3. Chain-based Hierarchical Routing Protocol (CHIRON)

CHIRON [34] is an enhancement of the Energy-Balanced Chain-cluster Routing Protocol (EBCRP) [35]. EBCRP divides the network into rectangular clusters with one chain being established within each cluster using a specialized scheme called the ladder algorithm. An example is shown in figure 17.

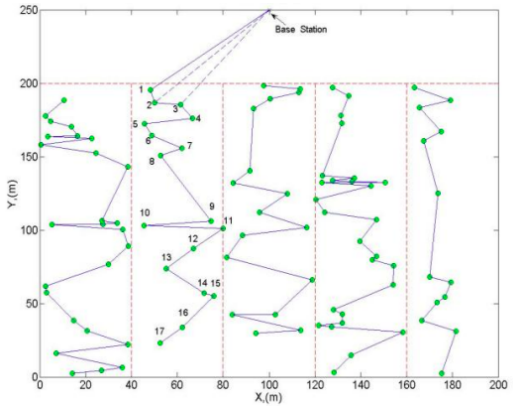

Figure 17. EBCRP network topology [35]

CHIRON utilizes BeamStar [36] technique for cluster formation with each cluster having an arc shape. The chain formation process is similar to that of PEGASIS where the node furthest from the sink initializes chain formation using the greedy algorithm to compute nearest neighbors. Residual energy level information is attached to transmitted packets during data fusion process. The node with the highest residual energy within a cluster is elected as the chain leader. Regular nodes send their packets along the chain to the chain leader during the data transmission phase. Chain leaders relay sensed data to the sink in a multi-hop, leader-by-leader approach starting from the furthest cluster. A CHIRON network consisting of six partitions is shown in figure 18.

A notable downside of the CHIRON protocol is its reliance on the network sink to compute and transmit chain formation data which can create a single point of failure.

A summary of the studied chain-based protocols is listed in Table 2.

While both group-based and chain-based clustering protocols establish a robust topological foundation for spatial defense, they primarily offer only baseline structural resilience. In the highly volatile spatio-temporal conditions of precision agriculture, relying solely on topological restructuring is insufficient to

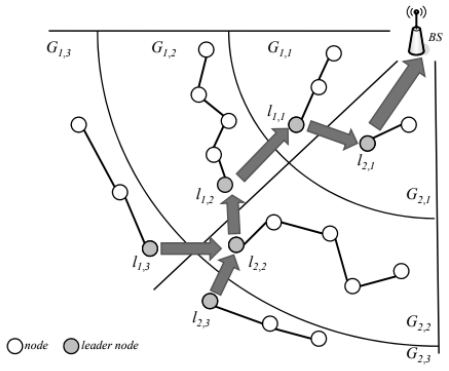

Figure 18. CHIRON network topology with data transmission [34]

prevent severe hardware damage from cascading into logical routing failures. Therefore, to ensure continuous network survivability against unpredictable environmental anomalies, WSNs must integrate dedicated, cross-layer fault management mechanisms, which are explored in the subsequent section.

7. Fault Management in WSNs - [RQ3]

Fault tolerance in a WSN is defined a s t he ability to handle faults and maintain optimal network performance [37]. This section discusses the stages of fault management in WSNs. An overview of each stage is as follows:

7.1. Fault Detection

Fault detection utilizes network monitoring to identify and locate faults. Network monitoring systems are of four types: passive, active, reactive, and proactive. Passive monitoring locates a fault, only once it has manifested, whereas Active monitoring uses real-time node data to identify faults. Proactive monitoring uses past fault data to predict future faults, and Reactive monitoring observes network behavior and takes action when faulty behavior is noticed [38].

Active monitoring has reduced detection latency since node status is checked regularly, but at the cost of network overhead and constant packet transmissions, which deplete the battery faster. Passive monitoring does not have an impact on network usage or node battery life, but it incurs a delay in identifying and isolating a fault. Proactive monitoring has the shortest delay in fault diagnosis as it aims to predict faults, but predictions can sometimes be incorrect, and the prediction models require extensive training and testing. Reactive methods are more accurate, but the diagnosis takes place after a fault has occurred.

Table 2. Summary of chain-based clustering protocols

Protocol	Node Position	Formation type		Chains per network		Leader selection	
		Unified	Partitioned	Single	Multiple	Random	Residual energy
PEGASIS [31]	✓	✓		✓		✓	
CCS [33]	✓		✓		✓	✓	
EBCRP [35]	✓		✓		✓		✓
CHIRON [34]	✓		✓		✓		✓

7.2. Fault Recovery

The fault recovery stage reestablishes network stability once a fault has been detected and isolated [39]. Fault recovery comprises of two steps:

1. Reconfiguration - updating network structure
2. Recovery - removing the effects of the fault

The recovery stage is categorized as backward and forward. Backward recovery returns a network back to the healthy state after a fault has occurred and is performed by consistently recording network state as checkpoints or logins [13]. Forward recovery employs redundancy by inserting backup nodes into a network, whereupon fault detection, a backup node assumes network activities. Forward recovery techniques can suffer from poor scalability when many nodes become faulty and can also be costly as multiple redundant nodes need to be added into a WSN.

Fault management frameworks are used to cover the multiple stages in a fault-tolerant system. [13].

8. Fault Management Frameworks - [RQ3]

This section discusses fault management frameworks used in WSNs based on a taxonomy built by combining the works of Moridi et al. [13] and Muhammed et al. [14].

To align with this survey's perspective of *Cross-Layer Resilience in Spatio-Temporal Agricultural IoT*, the taxonomy has been categorized based on the "Locus of Action": the specific layer at which a framework isolates and mitigates faults before they can propagate. As shown in Figure 19, fault management frameworks can be classified into three major groups: Centralized, Distributed, and Hybrid.

An analysis of each framework type, along with its capabilities, is as follows:

- **Centralized (Macro-Level Resilience):** These frameworks use a centrally operating network element, such as a sink, to manage faults across the entire network. They provide network-wide coordination, which is highly effective for identifying routing failures but struggles to isolate hardware degradation at the extreme edges of a vast agricultural deployment.
- **Distributed (Micro-Level Resilience):** These frameworks utilize multiple elements (individual nodes or clusters) to manage faults without central control. They excel at localized node and cluster isolation, quickly identifying hardware and data-centric anomalies caused by harsh micro-climates before they corrupt the routing layer.
- **Hybrid (Cross-Layer Resilience):** These frameworks combine both approaches. They utilize self-detection for micro-level physical anomalies and central coordination for macro-level routing failures, effectively providing multi-tiered isolation. This makes hybrid frameworks the pinnacle approach for handling the spatio-temporal challenges inherent to PA.

8.1. Centralized Fault Management Frameworks

In centralized fault management frameworks, the network sink maintains real-time data about the network state. Nodes are required to send periodic status updates to the sink which aggregates data for fault management operations.

Centralized fault management frameworks operate using either a centralized fault notification and mitigation scheme or a centralized detection and reporting mechanism. The fault notification and mitigation scheme utilizes a database to store node states and perform real-time fault detection using heuristic-based algorithms. Alternatively, collected data is used by a machine learning (ML) algorithm to predict fault occurrences. In contrast, centralized detection and reporting technique use scheduling for fault management purposes [13].

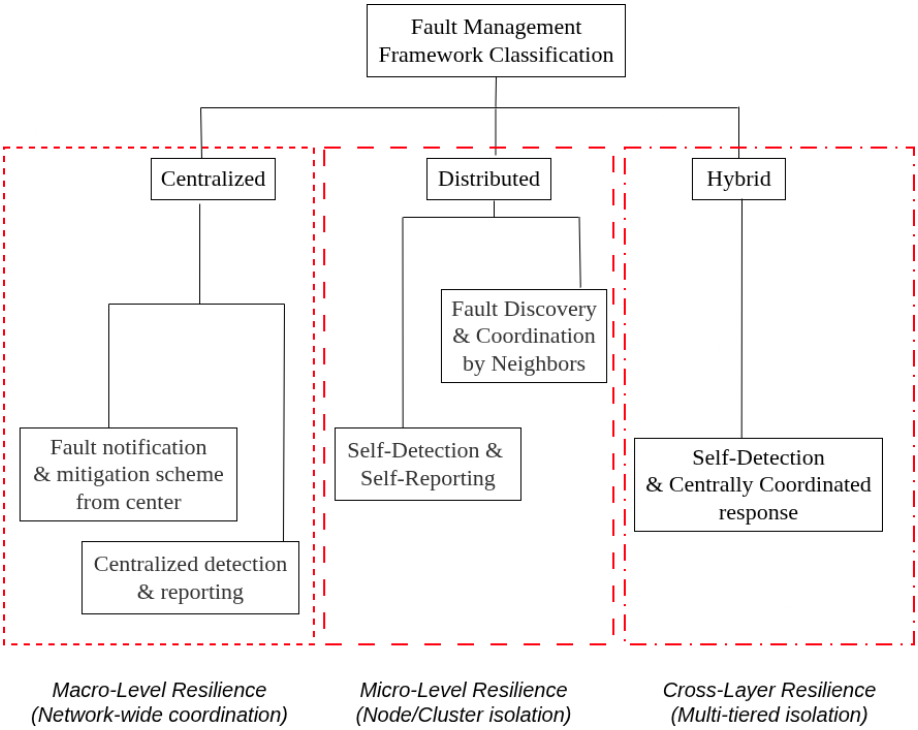

Figure 19. Fault Management Framework taxonomy

Centralized Fault Notification and Mitigation Scheme. The centralized fault notification and mitigation approach has a manager process running within the network sink. It gathers sensor node metadata along with sensed data. This metadata is compiled into a database that is continuously updated. The metadata includes node identity data, status data along with sensed data. The sink manager process monitors the database to identify predefined anomalous events.

The Sensor Network Management System (SNMS) described in [15] is a centralized WSN management framework that uses a database for fault notification and mitigation. It analyzes aggregated data and identifies causes for premature node death. A real-world experiment conducted by Tolle et al. [15] used SNMS on a multi-hop WSN with eighty nodes operating on the TinyOS software platform. Tolle et al. [15] focused on optimizing node resource usage when developing the framework. SNMS provides two primary management functions; a query system and event logging [15]. The query system supports continuous user-driven monitoring of known parameters such as node health data. Event logging handles post-event analysis and monitoring of unknown events via persistent data logs.

However, SNMS's notable downside is the added network overhead caused by continuous status updates requested by the central network manager. This results in rapid energy depletion of nodes and their eventual demise.

ML-based centralized fault management uses ML algorithms to analyze data aggregated at the network sink. ML approach is classified into three subcategories as supervised learning, unsupervised learning, and reinforcement learning. Supervised learning verifies data based on a pre-trained model. Learning methods include k-nearest neighbor, neutral networks and Bayesian statistics. Reinforcement learning attempts to understand patterns generated within a dynamic environment through trial and error. In unsupervised ML, learning is performed on unlabeled data to discover hidden patterns.

FT-CoWiseNets is a centralized fault tolerance framework that utilizes neural networks [16]. This framework has been designed based on six major requirements for business scenarios; extensibility, transparency, support for heterogeneity, fault identification, fault isolation, and automatic recovery. Souza et al. [16] have considered these requirements and designed the framework in three layers as device, middleware, and application.

The device layer consists of sensor nodes gathering data, for example, from an agricultural environment as well as the communication protocol stack. The middleware layer provides a web interface and acts as the data aggregation point and it is divided into two sub-layers as platform abstraction layer and fault management layer. The platform abstraction sub-layer enables the WSN to support heterogeneous nodes. The fault management layer is tasked with fault detection

as well as recovery and includes four components; system state unit, system manager unit, fault diagnosis manager and fault recovery manager. The system state maintains a database of network entities and events. The fault diagnosis manager performs fault detection via neural networks and fault isolation via a rule engine. Thereafter fault recovery manager attempts to recover the network using a combination of approaches such as sensor value fusion and mapping. A high-level overview of these components is shown in Figure 20. FT-CoWiseNets also includes fault management in the device layer using neighbor co-operation and node self-management techniques.

Building on the principles of ML-based centralized fault management, Attarha et al. [40] proposed an Automated Fault Detection Framework specifically designed to address the subtle faultiness of sensors in PA applications. Unlike traditional methods that often only detect blatant point anomalies (e.g., extreme outliers or zero values), this framework focuses on identifying contextual or soft faults where a degraded sensor mimics normal behavior but produces inaccurate data over time.

A core component of this centralized framework is its systematic feature engineering technique. Instead of relying solely on raw data, the framework extracts inherent sensor data features (like the rate of change and correlations between co-located sensors) and weather station-based features (incorporating external data like humidity and temperature from nearby stations).

Centralized Detection and Reporting. Frameworks based on centralized detection and reporting consist of two phases; data gathering and time slot assignment [13]. In the data gathering phase, the central manager generates a tree structure connecting nodes, CHs and sink. Thereafter, in the time slot assignment phase, time slots are assigned to nodes for data transmissions.

FlexiMAC is a centralized fault management framework that utilizes centralized detection and reporting. Lee et al. [41] have described it as a self-healing protocol for periodic data-gathering use cases. FlexiMAC supports any network topology. The framework runs in two stages; setup phase and data gathering phase. The setup phase utilizes CSMA/CA protocol for packet transmissions. It then reverts to TDMA once slots have been allocated. This is similar in design to LEACH routing protocol discussed in section 4 [21]. The usefulness of FlexiMAC is its adaptive slot structure where slots are assigned by the central manager. This gives nodes the ability to modify their schedules without informing neighbor nodes. FlexiMAC introduces a separate Fault Tolerant-listening Slot (FTS) into its schedule where all nodes in the network are in listening mode. This allows nodes to configure themselves without prior knowledge of network state to accommodate a fault. The flexible slot structure enables FlexiMAC to be fault-tolerant and energy efficient without the need for topological constraints or additional hardware. However, requiring all nodes in the WSN to be configured to support the dynamic slot structure is a limitation of FlexiMAC.

8.2. Distributed Fault Management Frameworks

In this technique, fault management is performed via multiple devices in a WSN. This addresses scalability and single-point-of-failure issues presented by centralized frameworks. Distributed fault management framework design consists of managing devices such as CHs which control subsets of the WSN. These devices monitor, manipulate and communicate with each other to perform network management operations.

Distributed frameworks are divided into two groups based on the number of nodes participating in fault tolerance. The groups are Self-Detection and Self-Reporting as well as Fault discovery and Coordination by Neighbors. In the self-detection and self-reporting approach, the health status of a sensor node is determined by the sensor node itself whereas in Neighbor cooperation technique, neighboring nodes use spatial-temporal co-relation for error checking and fault detection.

Self-Detection and Self-Reporting. The Self-Detection and Self-Reporting approach performs fault detection by detecting and isolating faults at the source. This approach can be used in combination with neighbor cooperation. Eligibility for network participation is decided independently by each individual node.

Distributed Self-Fault Diagnosis (DSFD) is a distributed fault management system utilizing this approach [42]. DSFD is based on the modified three sigma edit test for detecting faults [43]. Unlike the standard three-sigma edit test, the modified three-sigma edit test does not require nodes to share fault status among neighbors. This results in reduced transmissions and energy consumption which prolongs network lifetime. It consists of an initialization phase and self-diagnosis phase. In the initialization phase, neighboring tables are generated based on data received from neighbor nodes. In the self-diagnosis phase, neighbor tables are analyzed. If no data is present it is regarded as a hard fault at the corresponding node.

The authors have observed that data-centric faults such as offset faults, gain faults and stuck-at faults are identified accurately by the framework.

Fault Discovery by Neighbors. Fault discovery by neighbors approach detects faults through data cross-verification from neighboring nodes. This approach can be further classified as majority voting and weighted majority voting based on node consensus. Nodes that

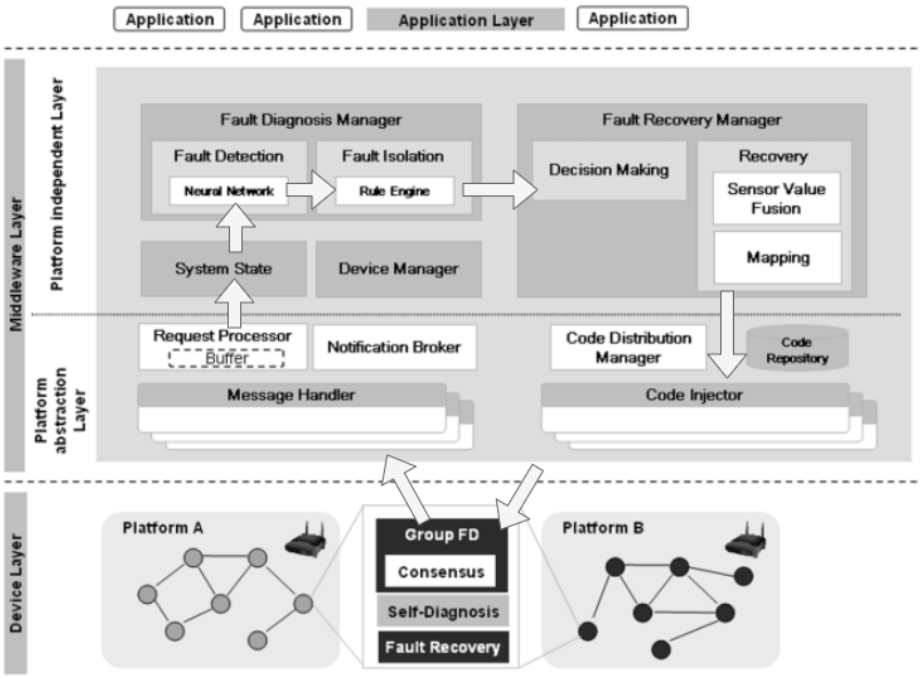

Figure 20. Components of FT-CoWiseNets framework

are voted out are rendered ineligible to participate in network activities.

Distributed Fault Identification (DFI) is a distributed fault management framework based on neighbor cooperation [44]. DFI predicts probable fault status by having every node in WSN share sensed data with neighbors. The framework has two phases; partial self-fault identification and self diagnosis phase. In partial self-fault identification phase, a node receive data from neighbors and compiles a neighbor table. Then data is compared against minimum sensed value and maximum sensed value as well as the deviation from mean to identify possible faults. In self diagnosis phase, majority voting is used to determine final fault status based on data exchanged with neighbors.

8.3. Hybrid Fault Management Frameworks

Hybrid fault management frameworks utilize a combination of self-detection and centrally coordinated approaches. These frameworks have a multi-tiered architecture to accommodate features of both types.

Fault Detection Scheme (FDS) is a hybrid fault management framework with a hierarchical network architecture [45]. FDS has been designed to detect large-scale, real-world phenomena such as forest fires and earthquakes even with unreliable data. FDS operates on a node level and a CH level. The node level utilizes a self-management approach where a Naive Bayes classifier is used to detect outlier data [46]. Non-faulty sensed data is considered to be in a range that can be split into intervals (classes) as $I = \{i_1, i_2, ..., i_n\}$. Initially, the classifier is inferred using maximum a posteriori (MAP) which along with Bayes rule is represented in equation 8.

$$i_{MAP} = argmax_{ij \in C} \frac{P(H_o|i_j)P(H_n|i_j)P(x_n|i_j)P(i_j)}{P(H_o, H_n, x_n)} \quad (8)$$

where $P(i_j)$ represents the probability of the sensed data from a specific sensor node falling within a predefined interval (class); $P(H_o|i_j)$ represents the probability of the last sensed data of a sensor node falling into a particular class, given that the current sensed data falls into a different class; $P(H_n|i_j)$ represents the conditional probability of a neighboring sensor node's last sensed data falling into a specific class, given that the sensed data has fallen into a different class, and $P(x_n|i_j)$ represents the probability of a neighboring sensor node's sensed data falling into a particular class, given that the sensed data has fallen into a different class.

Thereafter, nodes compare the class of sensed data with the class derived from equation 8. If classes are equal, the node is considered non-faulty ($report = 0$), otherwise, it is considered faulty ($report = 1$). Sensed data and associated reports are then transmitted to the CH which maintains two tables; a normal table and an anomaly table. Node values are compared against threshold values which represent the degree of similarity. Depending on the outcome, a node is classified as faulty or not faulty.

Ultimately, hybrid frameworks are particularly well-suited for maintaining cross-layer resilience in agricultural IoT. By utilizing self-detection for soft, node-level anomalies (micro-level) and centralized or CH-coordinated detection for routing and hardware failures (macro-level), these frameworks can isolate and mitigate faults across multiple layers before they cascade throughout the spatio-temporal deployment.

A summary of discussed fault management frameworks can be found in table 3.

9. Discussion

This section presents an evaluation on clustering protocols and fault management frameworks discussed in this paper as well as future directions and research challenges present in the domain.

9.1. Clustering Protocols

The two categories of clustering protocols that are widely used in PA applications were reviewed in this paper, namely group-based and chain-based protocols. Each category has its advantages and disadvantages which are compared in table 4.

Each protocol aims to improve upon a predecessor in key areas of performance such as network lifetime, fault tolerance, and scalability. However, the practicality of the proposed algorithms for field deployment raises additional issues. Factors affecting practicality include:

- Cost: An increase in the amount of equipment and components when designing clustering protocols will make it costlier and will render certain implementations impractical in a real-world scenario, making low-cost solutions more desirable.
- Maintenance: The system design should embrace simplicity to reduce the time and resources spent on maintenance.
- Non-extensibility: Designing a WSN to be technology-independent and interoperable between different components makes it easier to modify at present and further upgradable in the future.

It is often recommended to base future protocols on existing protocols that meet the above factors. The analysis summarized in table 6 demonstrates that, despite its age, the LEACH protocol fulfills most of the key requirements to serve as a suitable foundational protocol for researchers who are developing future clustering protocols.

However, traditional protocols such as LEACH and HEED often rely on probabilistic models or relatively straightforward heuristic parameters (like residual energy alone) for CH selection. While simple to implement, they can lead to uneven energy dissipation, sub-optimal routing paths, and the premature creation of isolated nodes, particularly in large-scale heterogeneous environments.

Conversely, Machine Learning-based protocols, such as DEDSCR, introduce significant computational complexity but offer vastly superior network longevity and load balancing. By employing advanced techniques like fuzzy logic and meta-heuristic algorithms to adaptively solve combined clustering and routing problems, these schemes natively address the core challenge of cross-layer resilience. They successfully decouple the hardware degradation caused by spatio-temporal extremes from the logical routing layer, ensuring continuous operation.

9.2. Fault Management Frameworks

Fault management frameworks were developed to address and mitigate faulty states in a network to maintain the quality of service. As shown in this review, these frameworks are classified into three types: centralized, distributed, and hybrid.

Centralized fault management keeps network status within the network sink, with all nodes sharing network metadata and sensed data that is then aggregated by the sink. This type of framework is straightforward in design and, with the added computing resources of the sink, can detect faults in advance for preventative measures to be taken. However, the sink-dependent design can cause processing delays and bottlenecks within the network, especially when scaling to a larger network. The dynamic environmental effects of PA scenarios, such as animal interference or unpredictable weather, can also cause the centralized sink to fail, affecting not just fault management capabilities but the entire network as a whole.

Distributed fault management detects faults close to the network edge, which eliminates the single-point-of-failure issue seen in centralized frameworks. It can also be more accurate as fault management takes place on a per-node or per-cluster basis. This serves well for PA use cases, as faults can be detected locally without relying on a central sink, which may not always be available. However, in distributed fault management frameworks based on neighbor cooperation, message exchange between neighboring nodes to detect faults can lead to increased energy consumption.

Hybrid fault management frameworks aim to have both a macro-level and a micro-level view of the network state, enabling multiple levels of fault detection and detection of multiple types of faults. This is more suited for PA scenarios due to the multiple levels of redundancy. Sensor-level soft-faults can be determined by the nodes themselves, while hard-faults can be detected by CHs or the network

Table 3. Summary of fault management frameworks

Framework	Category	Fault Detection	Fault Diagnosis	Fault Types	Network Type
SNMS	Fault notification and Mitigation scheme (Database)	Centralized	Passive	Hard	Homogenous
FT-CoWiseNets	Fault notification and Mitigation scheme (ML)	Centralized Self-Detect	Proactive	Hard	Heterogenous
FlexiMAC	Centralized detection	Centralized	Reactive	Hard	Homogenous
DSFD	Self-detection	Self-Detection	Proactive	Soft	Homogenous
DFI	Neighbor coordinat ion	Group-Detection	Active Proactive	Soft	Homogenous
FDS	Self detection and Neighbor coordination	Centralized Self-Detect Passive	Passive	Soft	Homogenous

Table 4. Advantages and disadvantages of routing protocol categories

Category	Advantages	Disadvantages
Group-based	• Arranges the network in a straightforward and simple regular topology. • Support either a network of heterogeneous nodes or network of homogeneous nodes, allowing for more versatility in PA scenarios.	• Single point of failure created by the central sink. • Reduced scalability due to the centralized sink. • Network area coverage limited by number of hops.
Chain-based	• No single point of failure. • Wider network area coverage due to multi-hop arrangement. • Multiple routes for packets to be sent to sink. • Full sink accessibility to all nodes.	• Longer chains result in delivery delays to the sink. • Queued-up packets result in bottlenecks. • Unresponsive nodes cause chain failures resulting in transmission delays. • Overuse of nodes closer to the sink resulting in premature death.

sink. The processes used for fault detection can also be adapted based on the layer in which the detection is occurring. This increased accuracy comes at the cost of data transfer delays, which is a notable downside of hybrid fault management frameworks. The increased processing undertaken by nodes and CHs to detect faults can also negatively impact the battery life of nodes and reduce the overall network lifetime.

9.3. Performance Metrics

Clustering protocols and fault tolerance frameworks utilize several metrics to quantitatively evaluate performance. Identifying shared performance metrics among them highlights the research directions that have been focused on over the years and what is expected of the state-of-the-art today.

Table 7 and Table 5 showcases a set of the highly cited papers that were identified during this study along with

Table 5. State-of-the-art fault tolerance frameworks

Name	Year	Performance Metrics
A self-adaptive and fault-tolerant routing algorithm for wireless sensor networks in microgrids [47]	2019	• Network lifetime • Data delivery ratio
Energy aware fault tolerant clustering with routing protocol for improved survivability in wireless sensor networks [48]	2022	• Network lifetime • Data delivery ratio • Throughput (kbps)
Self-healing and optimal fault tolerant routing in wireless sensor networks using genetical swarm optimization [49]	2022	• Energy efficiency • Data delivery ratio • Packet loss ratio
A fault tolerant optimal relay node selection algorithm for Wireless Sensor Networks using modified PSO [50]	2022	• Energy consumption • Data delivery ratio • Throughput (kbps)

Table 6. Comparison of reviewed routing protocols

Protocol	Base Protocol	Complexity	Scalability	Energy Usage	Packet Delivery Ratio
LEACH	-	Low	High	Low	Moderate
HEED	LEACH	Moderate	High	Low	Moderate
SEP	LEACH	Moderate	High	Moderate	Moderate
EEHC	SEP	High	Moderate	High	High
TEEN	LEACH	Moderate	Moderate	High	High
PEGASIS	LEACH	Moderate	Moderate	Moderate	Moderate
CCS	PEGASIS	Low	High	Low	Moderate
CHIRON	PEGASIS	High	High	Moderate	High
MOCRAW	LEACH	High	High	Low	High
EAO-CNN	-	High	High	Low	High

the performance metrics used. The observed metrics can be categorized based on energy efficiency and fault tolerance capability.

- *Energy Efficiency*: Observed in metrics such as number of alive nodes, dead nodes, and time of first node death. This indicates how well the network manages the finite amount of energy stored in all nodes to increase the network lifetime.
- *Fault Tolerance Capability*: Observed in metrics such as packet delivery ratio and packet loss ratio, it indicates how capable the WSN is at maintaining QoS despite the occurrence of faulty states.

According to the latest surveys in the field [55], these metrics confirm the research challenges that should be prioritized in future protocols and frameworks.

9.4. Future Works and Research Challenges

Clustering protocols that extend network lifetime by reducing energy consumption can be limited by practicality, and fault management frameworks, which address various network faults, can come at the cost of decreased quality of service. These limitations as well as other factors such as security and complexity indicate the presence of several open research problems to be addressed in the future, some of which are listed as follows:

Table 7. State-of-the-art LEACH variants

Name	Year	Performance Metrics
EDMHT-LEACH [51]	2017	• Number of alive nodes • Packet delivery ratio
EADCR: Energy Aware Distance Based Cluster Head Selection and Routing Protocol for Wireless Sensor Networks [52]	2020	• Number of alive nodes • Node death time vs round number
Improved node localization using K-means clustering for Wireless Sensor Networks [53]	2020	• Number of alive nodes • Energy consumption of overall network • Packet delivery ratio
Distance and energy-aware extended LEACH using secondary cluster head for wireless sensor networks [54]	2022	• Number of alive nodes • Average network lifetime duration • Energy consumption of overall network

- Cross-Layer Fault Tolerance: WSNs by design are intended to run with minimal human intervention. However, harsh agricultural environments inevitably cause physical component degradation that cascades into logical routing failures. Future protocols must move beyond isolated fault detection and integrate predictive, cross-layer ML models to proactively identify physical degradation before it induces a protocol-level fault.

- Complexity: Network protocol complexity informs node hardware complexity, which also has an impact on associated costs and maintainability. Therefore, low complexity should be encouraged when designing future clustering protocols.

- Spatio-Temporal Deployment Adaptation: WSN designs are inextricably linked to their spatio-temporal operating environments. Formulating generic protocols is insufficient due to the immense variability in spatial scales (e.g., land size, topography) and temporal shifts (e.g., seasonal climate changes, crop growth cycles). Future protocols must dynamically adapt their spatial defense mechanisms (intra-cluster vs. multi-hop) based on real-time environmental context to ensure optimal network survivability.

- Scalability: Sensor nodes in WSNs handle a significant amount of data, including sensor readings and network coordination, during a typical deployment. Unnecessary transmissions can increase overall power consumption and waste bandwidth, eventually leading to data loss. Future protocols may integrate intelligent load-balancing techniques to enhance scalability.

- Modulation: The clustering protocols and fault management frameworks discussed in this paper operate within the network layer and above. However, the capabilities of the lower physical and MAC layers are equally crucial for network performance. While current standards such as IEEE 802.15.4 [56] define several aspects of these low-level layers, future protocols and standards can consider advances in transceiver and battery technology to improve network performance further.

- Security: Constrained hardware resources often limit sensor node encryption and decryption capabilities, causing data in certain high-risk environments, such as power plants and nuclear reactors, to have sensitive data exposed. Factoring in security measures into future protocols and hardware designs will prove beneficial in such situations.

10. Conclusion

Wireless Sensor Networks (WSNs) form the critical data-gathering backbone of Precision Agriculture (PA), yet their deployment in harsh, dynamic environments makes them inherently prone to failures. This survey reviewed the literature to address the structural and functional challenges of maintaining QoS in PA WSNs through a novel theoretical lens of *Cross-Layer Resilience in Spatio-Temporal Agricultural IoT*.

We derived three primary conclusions corresponding to the research questions (RQs) posed in this survey:

- **Addressing RQ1:** Our analysis of clustering protocols reveals that while group-based protocols like LEACH provide robust, localized spatial defense (intra-cluster resilience), chain-based multi-hop protocols such as PEGASIS offer superior extended spatial defense across large agricultural grids. Advanced ML-based clustering protocols represent the optimal balance of energy efficiency and scalability.

- **Addressing RQ2:** We identified that the spatio-temporal extremes of agricultural environments accelerate physical component degradation. If not isolated, these hardware-level physical faults act as catalysts that cascade into logical, protocol-layer routing failures, leading to cross-layer network deterioration.

- **Addressing RQ3:** Our evaluation of fault management frameworks demonstrates that distributed, micro-level systems mitigate the single-point-of-failure risks inherent in centralized sinks. However, hybrid machine learning-based approaches are the pinnacle of cross-layer resilience, utilizing multi-tiered isolation to proactively sever the link between physical degradation and routing failure.

Ultimately, designing resilient PA WSNs requires a synergistic integration of adaptive spatial clustering and multi-tiered fault management to ensure the continuous survivability of precision agriculture systems.

11. Acknowledgement

This research was partially funded by the National Research Council (NRC) Sri Lanka under grant no. NRC-ID-19-040.

References

[1] Daanoune I, Abdennaceur B, Ballouk A. A comprehensive survey on LEACH-based clustering routing protocols in Wireless Sensor Networks. Ad Hoc Networks. 2021 4;114:102409.

[2] García L, Parra L, Jimenez JM, Lloret J, Mauri PV, Lorenz P. DronAway: A Proposal on the Use of Remote Sensing Drones as Mobile Gateway for WSN in Precision Agriculture. Applied Sciences 2020, Vol 10, Page 6668. 2020 9;10:6668. Available from: https://www.mdpi.com/2076-3417/10/19/6668/

[3] Carlos-Mancilla M, López-Mellado E, Siller M. Wireless sensor networks formation: Approaches and techniques. Journal of Sensors. 2016;2016.

[4] Khan THF, Kumar DS. Ambient crop field monitoring for improving context based agricultural by mobile sink in WSN. Journal of Ambient Intelligence and Humanized Computing 2019 11:4. 2019 1;11:1431-9.

[5] Gupta A, Gupta HP, Kumari P, Mishra R, Saraswat S, Dutta T. A Real-time Precision Agriculture Monitoring System using Mobile Sink in WSNs. International Symposium on Advanced Networks and Telecommunication Systems, ANTS. 2018 7;2018-December.

[6] Valente J, Sanz D, Barrientos A, del Cerro J, Ángela Ribeiro, Rossi C. An Air-Ground Wireless Sensor Network for Crop Monitoring. Sensors 2011, Vol 11, Pages 6088-6108. 2011 6;11:6088-108. Available from: https://www.mdpi.com/1424-8220/11/6/6088/

[7] Moribe T, Okada H, Kobayashl K, Katayama M. Combination of a wireless sensor network and drone using infrared thermometers for smart agriculture. CCNC 2018 - 2018 15th IEEE Annual Consumer Communications and Networking Conference. 2018 3;2018-January:1-2.

[8] Shafi U, Mumtaz R, García-Nieto J, Hassan SA, Zaidi SAR, Iqbal N. Precision Agriculture Techniques and Practices: From Considerations to Applications. Sensors 2019, Vol 19, Page 3796. 2019 9;19:3796. Available from: https://www.mdpi.com/1424-8220/19/17/3796.

[9] Farahani S. ZigBee and IEEE 802.15.4 Protocol Layers. ZigBee Wireless Networks and Transceivers. 2008 1:33-135.

[10] Yegin A, Kramp T, Dufour P, Gupta R, Soss R, Hersent O, et al. LoRaWAN protocol: specifications, security, and capabilities. LPWAN Technologies for IoT and M2M Applications. 2020 1:37-63.

[11] Badenhop CW, Graham SR, Ramsey BW, Mullins BE, Mailloux LO. The Z-Wave routing protocol and its security implications. Computers & Security. 2017 7;68:112-29.

[12] Ojha T, Misra S, Raghuwanshi NS. Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges. Computers and Electronics in Agriculture. 2015 10;118:66-84.

[13] Moridi E, Haghparast M, Hosseinzadeh M, Jassbi SJ. Fault management frameworks in wireless sensor networks: A survey. Computer Communications. 2020 4;155:205-26.

[14] Muhammed T, Shaikh RA. An analysis of fault detection strategies in wireless sensor networks. Journal of Network and Computer Applications. 2017 1;78:267-87.

[15] Tolle G, Polastre J, Szewczyk R, Culler D, Turner N, Tu K, et al. A macroscope in the redwoods. SenSys 2005 - Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems. 2005:51-63.

[16] Souza L, Vogt H. A Survey on Fault Tolerance in Wireless Sensor Networks. 2007.
[17] Shabbir N, Hassan SR, Shabbir N, Hassan SR. Routing Protocols for Wireless Sensor Networks (WSNs). Wireless Sensor Networks - Insights and Innovations. 2017 10. Available from: https://www.intechopen.com/state.item.idundefined/state.item.id.
[18] Shahraki A, Taherkordi A, Øystein Haugen, Eliassen F. Clustering objectives in wireless sensor networks: A survey and research direction analysis. Computer Networks. 2020 10;180:107376.
[19] Rawat P, Chauhan S. Clustering protocols in wireless sensor network: A survey, classification, issues, and future directions. Computer Science Review. 2021 5;40:100396.
[20] Boulhares I, Omari M. Hybridizing PEGASIS with LEACH-1R protocols in wireless sensor networks. Proceedings of 2016 8th International Conference on Modelling, Identification and Control, ICMIC 2016. 2017 1:1037-42.
[21] Heinzelman WR, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks. Proceedings of the Hawaii International Conference on System Sciences. 2000:223.
[22] Younis O, Fahmy S. HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Transactions on Mobile Computing. 2004 10;3:366-79.
[23] Ullah Z. A Survey on Hybrid, Energy Efficient and Distributed (HEED) Based Energy Efficient Clustering Protocols for Wireless Sensor Networks. Wireless Personal Communications. 2020 6;112:2685-713.
[24] Smaragdakis G, Matta I, Bestavros A. SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks. 2004 5. Available from: https://open.bu.edu/handle/2144/1548.
[25] Kumar D, Aseri TC, Patel RB. EEHC: Energy efficient heterogeneous clustered scheme for wireless sensor networks. Computer Communications. 2009 3;32:662-7.
[26] Manjeshwar A, Agrawal DP. TEEN: A routing protocol for enhanced efficiency in wireless sensor networks. Proceedings - 15th International Parallel and Distributed Processing Symposium, IPDPS 2001. 2001:2009-15.
[27] Le VT, Leone RD, Hoang T, Mau DO, Kim M, Vo NS. Distance-, Energy-, and Density-Aware SOM Clustering-Based Routing in Low-Power IoT-Enabled WSNs. IEEE Sensors Journal. 2025;25:42231-42. Available from: https://ieeexplore.ieee.org/abstract/document/11185311.
[28] Chaurasia S, Kumar K, Kumar N. MOCRAW: A Meta-heuristic Optimized Cluster head selection based Routing Algorithm for WSNs. Ad Hoc Networks. 2023 3;141:103079. Available from: https://www.sciencedirect.com/science/article/abs/pii/S1570870522002517.
[29] Pandiyaraju V, Ganapathy S, Mohith N, Kannan A. An optimal energy utilization model for precision agriculture in WSNs using multi-objective clustering and deep learning. Journal of King Saud University - Computer and Information Sciences. 2023 12;35:101803. Available from: https://www.sciencedirect.com/science/article/pii/S1319157823003579.
[30] Chan L, Chavez KG, Rudolph H, Hourani A. Hierarchical routing protocols for wireless sensor network: a compressive survey. Wireless Networks. 2020 7;26:3291-314.
[31] Lindsey S, Raghavendra C, Sivalingam KM. Data gathering algorithms in sensor networks using energy metrics. IEEE Transactions on Parallel and Distributed Systems. 2002 9;13:924-35.
[32] Heinzelman WB, Chandrakasan AP, Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications. 2002;1:660-70.
[33] Jung SM, Han YJ, Chung TM. The concentric clustering scheme for efficient energy consumption in the PEGASIS. International Conference on Advanced Communication Technology, ICACT. 2007;1:260-5.
[34] Chen KH, Huang JM, Hsiao CC. CHIRON: An energy-efficient Chain-Based Hierarchical Routing Protocol in wireless sensor networks. 2009 Wireless Telecommunications Symposium, WTS 2009. 2009.
[35] Xi-rong B, Shi Z, Ding-yu X, Zhi-tao Q. An energy-balanced chain-cluster routing protocol for wireless sensor networks. NSWCTC 2010 - The 2nd International Conference on Networks Security, Wireless Communications and Trusted Computing. 2010;2:79-84. EBCRP protocol.
[36] Mao S, Hou YT. BeamStar: An edge-based approach to routing in wireless sensor networks. IEEE Transactions on Mobile Computing. 2007 11;6:1284-96.
[37] Chouikhi S, Korbi IE, Ghamri-Doudane Y, Saidane LA. A survey on fault tolerance in small and large scale wireless sensor networks. Computer Communications. 2015 9;69:22-37.
[38] Zhang Z, Mehmood A, Shu L, Huo Z, Zhang Y, Mukherjee M. A survey on fault diagnosis in wireless sensor networks. IEEE Access. 2018 2;6:11349-64.
[39] Shih HC, Ho JH, Liao BY, Pan JS. Fault node recovery algorithm for a wireless sensor network. IEEE Sensors Journal. 2013;13:2683-9.
[40] Attarha S, Band S, Forster A. Automated Fault Detection Framework for Reliable Provision of IoT Applications in Agriculture. 2023 19th International Conference on the Design of Reliable Communication Networks, DRCN 2023. 2023. Available from: https://ieeexplore.ieee.org/abstract/document/10108238.
[41] Lee WL, Datta A, Cardell-Oliver R. FlexiMAC: A flexible TDMA-based MAC protocol for fault-tolerant and energy-efficient wireless sensor networks. Proceedings - 2006 IEEE International Conference on Networks, ICON 2006 - Networking-Challenges and Frontiers. 2006;2:337-42.
[42] Panda M, Khilar PM. Distributed self fault diagnosis algorithm for large scale wireless sensor networks using modified three sigma edit test. Ad Hoc Networks. 2015 2;25:170-84.
[43] Maronna RA, Martin RD, Yohai VJ, Salibián-Barrera M. Robust statistics: theory and methods (with R). John Wiley & Sons; 2019.

[44] Panda M, Khilar PM. Energy efficient distributed fault identification algorithm in wireless sensor networks. Journal of Computer Networks and Communications. 2014;2014.
[45] Titouna C, Aliouat M, Gueroui M. FDS: Fault Detection Scheme for Wireless Sensor Networks. Wireless Personal Communications. 2016 1;86:549-62.
[46] Zhang H. The optimality of naive Bayes. Aa. 2004;1(2):3.
[47] Rui L, Wang X, Zhang Y, Wang X, Qiu X. A self-adaptive and fault-tolerant routing algorithm for wireless sensor networks in microgrids. Future Generation Computer Systems. 2019 11;100:35-45.
[48] Mansour RF, Alsuhibany SA, Abdel-Khalek S, Alharbi R, Vaiyapuri T, Obaid AJ, et al. Energy aware fault tolerant clustering with routing protocol for improved survivability in wireless sensor networks. Computer Networks. 2022 7;212:109049.
[49] M S, Pillai AS, Anpalagan A. Self-healing and optimal fault tolerant routing in wireless sensor networks using genetical swarm optimization. Computer Networks. 2022 11;217:109359.
[50] Sreedevi P, Venkateswarlu DS. A fault tolerant optimal relay node selection algorithm for Wireless Sensor Networks using modified PSO. Pervasive and Mobile Computing. 2022 9;85:101642.
[51] Alnawafa E, Marghescu I. EDMHT-LEACH: Enhancing the performance of the DMHT-LEACH protocol for wireless sensor networks. 16th Networking in Education and Research RoEduNet International Conference, RoEduNet 2017 - Proceedings. 2017 11.
[52] Panchal A, Singh RK. EADCR: Energy Aware Distance Based Cluster Head Selection and Routing Protocol for Wireless Sensor Networks. https://doiorg/101142/S0218126621500638. 2020 9;30.
[53] Khediri SE, Fakhet W, Moulahi T, Khan R, Thaljaoui A, Kachouri A. Improved node localization using K-means clustering for Wireless Sensor Networks. Computer Science Review. 2020 8;37:100284.
[54] Hossan A, Akter S, Choudhury PK. Distance and energy aware extended LEACH using secondary cluster head for wireless sensor networks. Telematics and Informatics Reports. 2022 12;8:100029.
[55] Al-Sulaifanie AI, Al-Sulaifanie BK, Biswas S. Recent trends in clustering algorithms for wireless sensor networks: A comprehensive review. Computer Communications. 2022 7;191:395-424. Available from: https://www.sciencedirect.com/science/article/pii/S014036642200158X.
[56] Adams JT. An introduction to IEEE STD 802.15.4. IEEE Aerospace Conference Proceedings. 2006;2006.