Smart Agricultural Technology 15 (2026) 102437

Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

Smart cattle monitoring with UAV edge processing and resilient multi-channel alert delivery

Antonello Calabrò *, Eda Marchetti

CNR Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo" Via Giuseppe Moruzzi, 1-56124, Pisa, Italy

ARTICLE INFO

Keywords:
Smart livestock monitoring
UAV edge computing
LoRa
Satellite fallback
Resilient alert delivery
Precision livestock farming

ABSTRACT

Monitoring cattle in large grazing areas requires telemetry solutions that are energy-efficient on the animal side, responsive during field operations, and robust to intermittent connectivity . This paper presents a smart livestock-monitoring architecture that combines LoRa-enabled wearable collars, a UAV serving as both a mobile gateway and an edge monitoring node, and a farmer-side node that exposes a configurable runtime monitoring interface. The proposed system moves time-critical reasoning closer to the field through lightweight on-drone rule evaluation, while preserving richer inspection, persistence, and rule management on the ground. To improve alert availability under unstable connectivity, the architecture integrates a dependable dispatcher that prioritizes terrestrial communication and escalates critical alerts to Iridium Short Burst Data (SBD) when Wi-Fi or LTE/4G delivery is unavailable or cannot satisfy the required timeliness. Livestock-relevant conditions, including missing beacons, geofence violations, prolonged immobility, communication degradation, and delivery failures, are represented as explicit runtime rules. The paper reports a controlled, prototype-based evaluation that combines measured LoRa first-hop characterization, replayable mission scenarios, and direct timing observations along the RockBLOCK-Cloudloop-CONCERN integration path. In the considered mission profile, on-drone rule evaluation reduces detection latency by about 2 orders of magnitude compared to ground-centric processing, while the hybrid configuration with dependable dispatch delivers all critical alerts within the adopted latency budget during simulated terrestrial outages. These results indicate that combining explainable edge-side monitoring with resilient multi-channel alert delivery is a practical direction for livestock supervision in infrastructure-poor agricultural environments.

1. Introduction

The digital transformation of agriculture is increasingly driven by sensing, communication, and automation technologies that improve productivity, sustainability, and animal welfare. In livestock farming, precision livestock farming (PLF) technologies offer significant advantages, especially in intensive farming scenarios with large herd sizes, where managing individual animals can be challenging. Additionally, these technologies are also highly relevant in extensive and semi-extensive farming settings, where animals roam across vast areas, communication infrastructure is limited, and manual inspection can be costly and inconsistent. Low-power wide-area technologies such as LoRa are attractive in this context because they support energy-efficient telemetry from wearable collars, while unmanned aerial vehicles (UAVs) offer flexible mobile assets for data collection and field supervision.

While many livestock monitoring systems currently offer features like telemetry collection, animal localization, and alert generation, significant challenges still exist. These challenges primarily relate to the reliability and timeliness of alert delivery, especially in areas with limited infrastructure and intermittent connectivity. Additionally, there are issues with the effective presentation of monitoring information to farm operators and with integrating edge-side monitoring with resilient multi-channel communication mechanisms. Farmers are not only concerned with the availability of raw telemetry data; they also want to ensure that important operational conditions are reliably detected and communicated. Key examples of these conditions include prolonged immobility, missing beacon transmissions, geofence violations, and the assurance that critical alerts have reached the monitoring platform, especially when terrestrial connectivity is poor.

This paper addresses these challenges through a smart monitoring architecture that combines LoRa-based collars, UAV-assisted collection, and a configurable concern-oriented monitoring infrastructure. The proposed solution adopts a hybrid operational mode: critical rules can be evaluated on board the drone for low-latency reaction, while richer

* Corresponding author.
E-mail address: antonello.calabro@isti.cnr.it (A. Calabrò).

https://doi.org/10.1016/j.atech.2026.102437
Received 16 April 2026; Received in revised form 19 May 2026; Accepted 25 July 2026
Available online 1 August 2026

visualization, historical inspection, and rule management remain available on the farmer-side node. In parallel, a dependable dispatcher dynamically selects the most suitable communication channel for forwarding alerts and mission data, combining terrestrial-first delivery with time-bounded satellite escalation.

The design is motivated by four practical constraints. First, cattle monitoring often takes place in areas where continuous Internet connectivity and fixed gateway coverage cannot be assumed. As a consequence, ensuring that critical alerts are delivered reliably and within operationally acceptable time bounds becomes a central challenge during field operations. Second, wearable nodes mounted on animals must remain lightweight and energy efficient, which limits their local processing capabilities. Third, anomaly detection in farming scenarios should remain explainable and configurable by domain operators. Fourth, critical alerts should remain available to the farmer even when preferred terrestrial channels are temporarily unavailable.

Starting from these considerations, this paper makes the following contributions:

- it defines a UAV-assisted livestock-monitoring architecture in which lightweight collars, a drone-side edge node, and a farmer-side monitoring node cooperate under intermittent connectivity;
- it introduces a dependable multi-channel dispatch policy based on terrestrial-first delivery with time-bounded Iridium escalation for critical alerts;
- it details a working prototype integrating LoRa collars, on-drone monitoring, CONCERN: COmplex eveNt proCEssing monitoR iNfrastructure [1], local buffering, and satellite-backed fallback delivery;
- it reports a controlled prototype-based evaluation that combines measured LoRa first-hop characterization with model-assisted backhaul stress scenarios to assess detection latency, alert timeliness, and resilience under terrestrial outages.

Compared with prior UAV-LoRa agricultural systems, the proposed work places a stronger emphasis on the integration of edge-side livestock monitoring with dependable multi-channel alert delivery. In particular, the architecture is designed to preserve alert timeliness and operational availability under intermittent terrestrial connectivity, which is a recurrent challenge in infrastructure-poor farming environments.

2. Related work

The proposed work lies at the intersection of smart livestock monitoring, UAV-assisted agricultural sensing, resilient edge communication, and rule-based runtime monitoring.

LPWAN-based livestock monitoring. Recent reviews on precision livestock farming and cattle-monitoring technologies show that extensive and pasture-based systems impose strong constraints on sensing coverage, device autonomy, positioning accuracy, maintenance effort, and communication reliability [2–4]. In this context, LoRa and LoRaWAN are widely adopted in agriculture because they offer low power consumption and are well suited to sparse rural deployments [5–8]. In livestock scenarios, these technologies support wearable devices that periodically transmit identifiers, positions, and sensor readings [9]. Broader surveys on cattle wearables and biosensors also confirm the relevance of collar-based sensing for health, activity, and welfare monitoring, while highlighting recurring trade-offs among battery life, robustness, communication range, and deployment cost [10–12]. However, most proposed systems still assume static gateways and backend-centric analytics, with limited emphasis on how field alerts should be detected and delivered under intermittent connectivity.

UAV-assisted agricultural and livestock monitoring. UAVs have been extensively investigated for crop inspection, mapping, precision agriculture, and field surveillance [13–15]. In livestock settings, UAVs have also been used for animal detection, counting, identification, posture analysis, and feedlot observation through aerial imaging and deep-learning-based processing [16–18]. These studies demonstrate the operational value of aerial platforms for observation and localization, but they typically focus on sensing coverage, image acquisition, or computer-vision accuracy rather than on integrating telemetry-driven monitoring logic with resilient alert delivery.

UAVs as mobile gateways, edge nodes, and disconnected agricultural infrastructure. Recent work on UAV-assisted LoRaWAN, data-mule architectures, and edge-enabled drone gateways confirms the usefulness of local processing and flexible communication in infrastructure-poor environments [19–25]. More broadly, smart-agriculture architectures based on cloud–edge–device collaboration and seamless distributed IoT infrastructures show that moving computation closer to the field can improve responsiveness and reduce dependence on permanent backhaul connectivity [26,27]. Likewise, delay-tolerant and data-mule-based solutions highlight the practical relevance of operating under intermittent connectivity in remote agricultural areas [28]. Still, these contributions are usually assessed in terms of communication feasibility, collection efficiency, or architectural flexibility. Much less attention is paid to the operational question of how a field alert should be forwarded when Wi-Fi or LTE/4G coverage becomes unstable during a mission, especially when a bounded reaction time is required.

Runtime monitoring and event processing. Event-driven monitoring and rule-based processing help bridge the gap between raw telemetry and actionable knowledge [29–32]. In agriculture, rule engines and complex event processing have already been proposed to transform heterogeneous IoT observations into higher-level events and context-aware reactions [31]. The CONCERN infrastructure [1] provides concern specification, rule-driven analysis, and operator-facing dashboards. In this paper, these mechanisms are used as enabling technologies for a smart agriculture scenario in which monitoring logic must remain explainable, configurable by farm operators, and deployable, in part, at the edge.

Positioning of this paper. Existing livestock solutions predominantly assume fixed gateways, UAV agricultural systems often focus on sensing or aerial observation, and rule-based monitoring frameworks rarely address hybrid edge deployment with resilient multi-channel delivery. To the best of our knowledge, prior work does not explicitly combine LoRa-based cattle telemetry, UAV-mediated collection, configurable edge-side rule evaluation, end-to-end acknowledgment-aware dispatch, and dependable multi-channel alert delivery with terrestrial-first communication and time-bounded satellite escalation. The contribution of this work is precisely this integration, targeting infrastructure-poor livestock-monitoring missions in which timely and dependable alert availability matters as much as data collection itself.

3. Proposed architecture

Fig. 1 organizes the proposed solution around three main elements: the *Collar Node*, the *Drone Edge Node*, and the *Farmer Edge Node*. The architecture follows a hybrid deployment strategy in which low-latency monitoring and communication decisions are executed on the drone, while persistent storage, full visualization, and long-term analysis remain on farmer-side infrastructure. The figure also distinguishes the system's main operational flows, including LoRa telemetry collection, terrestrial backhaul communication, satellite fallback delivery, and deferred synchronization paths.

The *Collar Node* is a lightweight wearable device attached to the animal and designed to periodically broadcast LoRa beacons containing the animal identifier, a position estimate, body temperature, battery level, and activity indicators. The node is intentionally kept simple to preserve energy autonomy and reduce maintenance costs. The architecture does not prescribe a specific hardware platform: any collar capable of periodic LoRa beaconing with a configurable payload can serve as a Collar Node. This flexibility is important in livestock deployments, where cost, ruggedness, and autonomy are often more relevant than onboard computing capabilities.

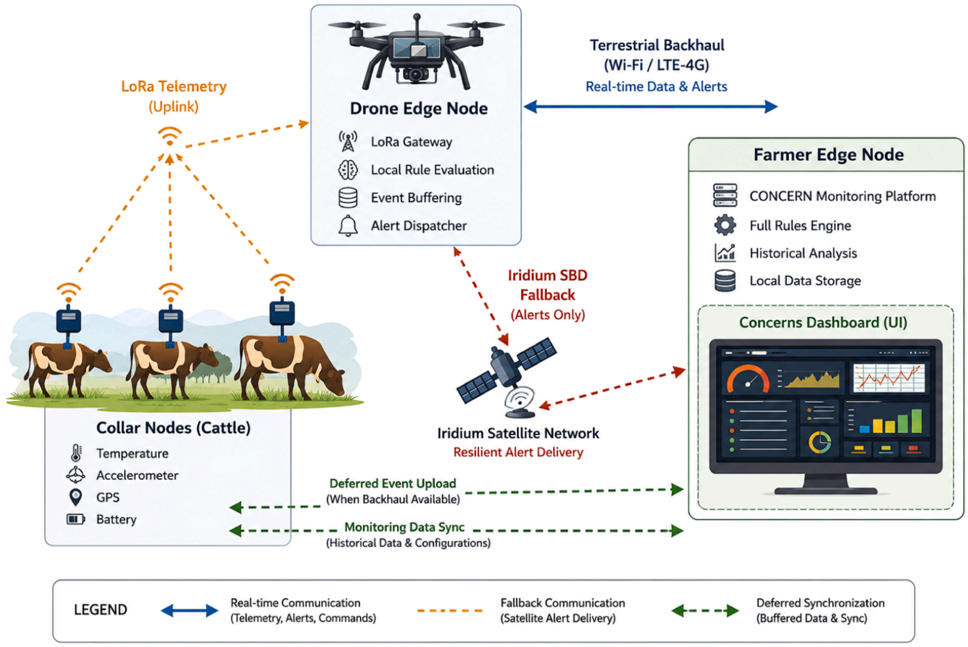

Fig. 1. Overview of the proposed livestock-monitoring architecture, showing LoRa telemetry collection, on-drone edge monitoring, terrestrial backhaul communication, satellite fallback delivery, and deferred synchronization flows among the Collar Nodes, Drone Edge Node, and Farmer Edge Node.

A relevant design aspect concerns the source of the animal position used by location-aware rules, such as geofence monitoring. The Collar Node may provide direct GPS coordinates when a positioning subsystem is available; otherwise, the Drone Edge Node contextualizes each LoRa transmission using its own GPS position, reception timestamp, and link-quality indicators. Therefore, geofence-related rules operate on the best position estimate available at runtime, while preserving traceability of whether that estimate originated from the collar or from drone-side contextualization.

The *Drone Edge Node* serves as both a mobile LoRa gateway and an edge monitoring node. During the mission, the UAV may operate either through pre-planned waypoint navigation or under manual operator control, depending on mission requirements and deployment conditions. In both cases, it primarily acts as a mobile LoRa gateway covering the monitored grazing area, rather than as a platform for active animal tracking. Livestock telemetry is therefore collected opportunistically through LoRa beacon reception during flight. The Drone Edge Node receives LoRa packets from the collars and enriches them with contextual information gathered on board, including reception timestamp, drone GPS position, mission identifier, signal-quality parameters, local connectivity status, and delivery constraints associated with the generated alert. Wi-Fi is used exclusively for short-range high-throughput data exchange when the UAV is close to the farm infrastructure, whereas LTE/4G and satellite communication support wider-area connectivity and resilient alert delivery. The drone hosts a lightweight CONCERN monitoring instance, previously tested on low-powered devices in [33], that evaluates a selected subset of rules that require timely responses. Representative examples include prolonged beacon absence, severe communication degradation, geofence violations, suspicious sensor values, and persistent alert-delivery failures.

In addition to local rule evaluation, the Drone Edge Node hosts a *Dependable Dispatcher*, a software component that manages channel selection and implements a terrestrial-first, time-bounded escalation policy. For each outgoing alert, the dispatcher first attempts delivery over the currently available terrestrial backhaul, preferring Wi-Fi when the UAV is close to the farmer-side station and LTE/4G otherwise. Each transmission creates an attempt record associated with an acknowledgment deadline. Delivery is considered successful only when the Farmer Edge Node returns an application-layer acknowledgment after message ingestion and persistence. If no end-to-end acknowledgment is received within a configurable timeout, the dispatcher can escalate by re-sending the alert through the Iridium SBD modem. Non-critical telemetry follows a store-carry-and-forward policy whenever no acceptable real-time channel is available. This distinction between real-time alert delivery and deferred synchronization is explicitly reflected in Fig. 1, where terrestrial backhaul communication, satellite fallback escalation, and buffered-data synchronization are represented as separate operational flows.

The *Farmer Edge Node* hosts the complete monitoring infrastructure. It can run on a Raspberry Pi or directly on the farmer's laptop and exposes the monitoring dashboard and operator interface for inspecting events, configuring rules, visualizing traces, and performing post-mission analysis. Compared with the drone-side instance, the farmer-side node offers persistent storage, a richer dashboard, and support for rule lifecycle management, historical comparison, and operator supervision. Fig. 2 shows examples of the operator dashboard and rule-management interface exposed by the Farmer Edge Node.

The proposed architecture supports four. In the first mode, LoRa telemetry is collected by the drone and forwarded to the Farmer Edge Node through Wi-Fi when the UAV is close to the farm infrastructure. In the second mode, LTE/4G is used as the primary terrestrial backhaul. In the third mode, when terrestrial connectivity is unavailable or does not satisfy the required timeliness for critical alerts, the drone can activate the Iridium satellite module. In the fourth mode, if no suitable real-time channel is available or the data are not latency critical, the drone stores events locally and synchronizes them later. This multi-mode behavior is central to the architecture because it supports livestock-monitoring missions in agricultural environments where persistent end-to-end connectivity cannot be guaranteed.

A key architectural design choice concerns where monitoring and dispatching logic should execute. A purely ground-centric solution would reduce onboard complexity but would depend on continuous backhaul availability. A purely drone-centric solution would minimize external dependencies but would place unnecessary computational and energy pressure on the aerial platform. For this reason, the selected design is hybrid: only a lightweight subset of rules and dispatch decisions

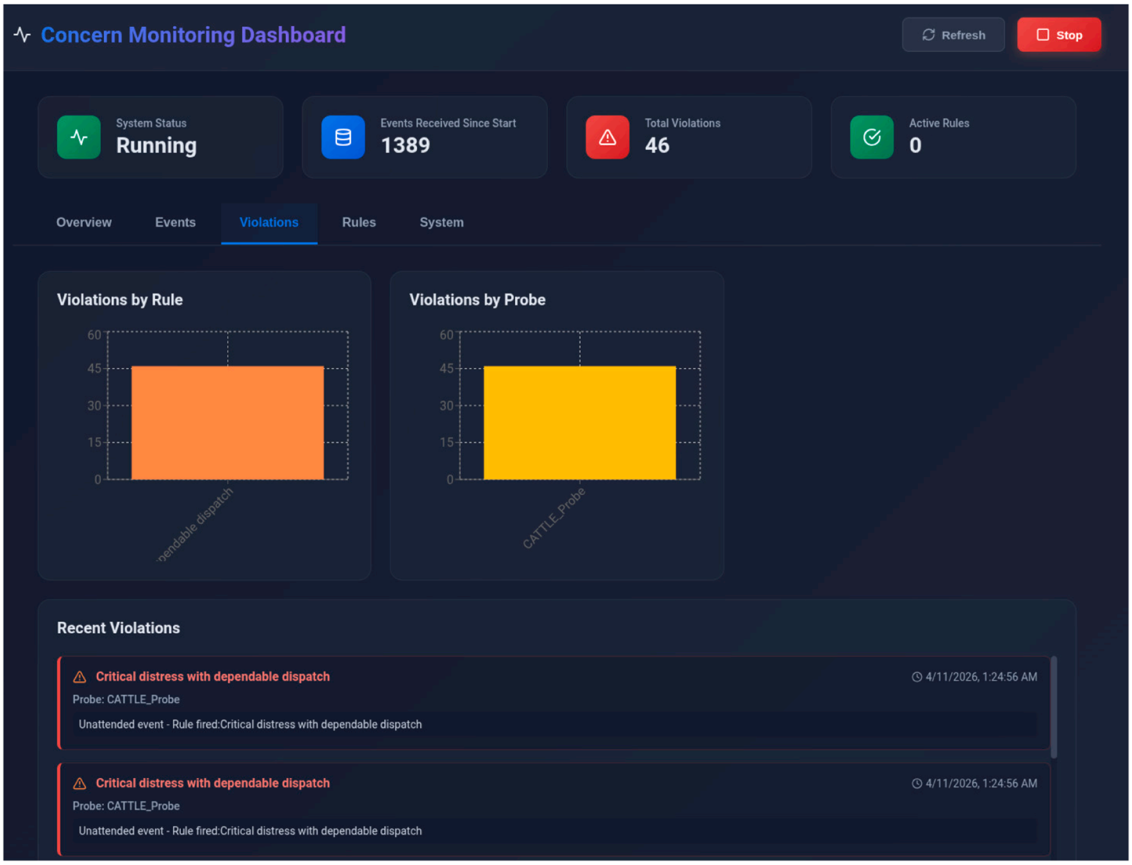

Fig. 2. CONCERN interface on the Farmer Edge Node for event inspection and violation tracking.

is evaluated on the drone, while the full analysis stack remains on the Farmer Edge Node. This arrangement balances latency, explainability, robustness, and resource consumption in a manner appropriate for practical smart-farming deployments.

From an operational perspective, the architecture explicitly supports the following concerns:

- *missing-beacon issue*: absence of transmissions from a known collar for longer than a configurable time window;
- *geofence issue*: evidence that an animal is outside a predefined safe region or moving toward its boundary;
- *immobility issue*: prolonged lack of motion, possibly combined with abnormal temperature values;
- *communication issue*: deteriorating RSSI/SNR, repeated packet loss, LTE outage, Wi-Fi unavailability, or satellite fallback activation;
- *delivery issue*: failure of a critical alert to reach the Farmer Edge Node within a required latency budget;
- *device-health issue*: low collar battery or unsustainable energy profile on wearable or drone nodes.

These concerns are represented as explicit runtime rules over event streams rather than as hard-coded control logic. This improves explainability, maintainability, and adaptability across different farms, herd management practices, and mission conditions.

4. Prototype implementation

The current prototype instantiates the proposed architecture using low-cost and widely available hardware components.The implementation is organized as a hybrid deployment in which the Drone Edge Node executes lightweight rule evaluation and dependable dispatching, while the Farmer Edge Node runs the full CONCERN stack. Table 1 summarizes the main hardware characteristics of the prototype components, with particular attention to constraints relevant to agricultural and UAV-assisted deployments.

The *Collar Node* is implemented as a LoRa-enabled wearable collar prototype for periodic beaconing. The node includes a microcontroller, a LoRa transceiver, a battery-powered subsystem, and optional sensors such as temperature and motion sensors. In the current version, the transmitted payload is intentionally compact to preserve energy autonomy and reduce channel occupancy. The beacon format includes at least the collar identifier, a sequence number, and optional sensor fields. This choice keeps the wearable device simple and consistent with the practical constraints of long-duration livestock monitoring. The compact payload design also helps reduce LoRa transmission airtime and the radio duty cycle, which are among the primary factors affecting energy consumption in battery-powered collar devices.

The *Drone Edge Node*, shown in Fig. 3, is implemented on a Raspberry Pi-based onboard platform equipped with a LoRa receiver, a GPS probe, local storage, Wi-Fi, LTE/4G connectivity, and an optional Iridium RockBLOCK 9603 modem. The implemented UAV platform supports waypoint-based navigation over the monitored grazing area, while telemetry collection relies on opportunistic reception of LoRa beacons during flight. UAV control remains logically separated from the communication backhaul used for telemetry forwarding and alert delivery. The node runs five main functions: a *LoRa acquisition module* that listens for collar transmissions and decodes incoming packets; a *context enrichment module* that attaches reception metadata such as timestamps, drone coordinates, signal-quality indicators, mission state, and channel-state information; a *lightweight monitoring module* derived from the CONCERN rule engine; a *synchronization manager* for local buffering and deferred uploads; and the *Dependable Delivery Dispatcher*, which selects the most

Table 1
Main hardware characteristics of the prototype components.

Component	Main characteristics
Collar Node	LoRa-enabled wearable prototype with compact beacon transmission; battery-powered (3.7 V / 1200 mAh Li-Po); estimated mass ≈ 85 g including enclosure and battery; IP67 enclosure protection
Drone Edge Node	Raspberry Pi-based platform; bare-board mass ≈ 350 g including cables, LoRa receiver, GPS, Wi-Fi, LTE/4G, and RockBLOCK 9603 Iridium modem (45g)
UAV Platform	Small multirotor UAV supporting the Drone Edge Node payload; nominal flight autonomy ≈ 28 min under standard operating conditions (eg, DJI AIR 2S or superior models)
Farmer Edge Node	Raspberry Pi 4 or laptop-class platform running the full CONCERN monitoring infrastructure and operator dashboard

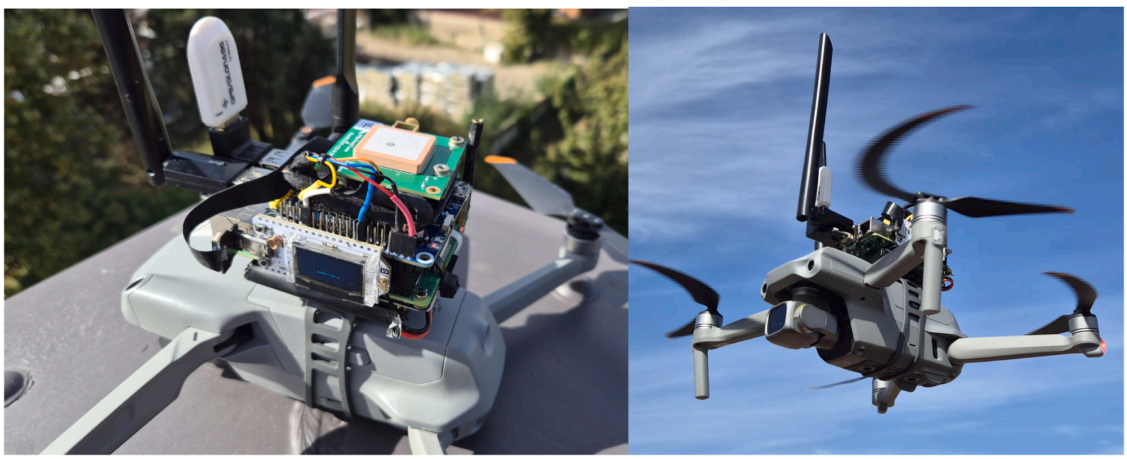

Fig. 3. Drone Edge Node prototype.

suitable uplink to the Farmer Edge Node. In this context, lightweight refers to the computational and energy footprint of the monitoring layer, which is designed to operate on resource-constrained onboard hardware compatible with the limited payload of small UAVs. In the current prototype, the drone-side monitoring instance runs on a Raspberry Pi 4B platform. Additional hardware characteristics of the system components are summarized in Table 1.

The *Farmer Edge Node* is deployed either on a Raspberry Pi at the farm or directly on the farmer's laptop. This node runs the full CONCERN stack, including the rule engine, the monitoring dashboard, persistence services, and the user-facing interface for rule configuration and trace inspection. The node stores both raw telemetry and derived monitoring outcomes, allowing post-mission analysis and comparison across missions. Each LoRa packet captured by the drone is translated into a normalized event representation.

A representative event schema includes the following fields:

⟨animalId, packetSeq, eventTime, sensorTemp, motionState, droneLat, droneLon, linkRSSI, linkSNR, missionId, wifiAvailable, wifiReliability, lteAvailable, lteReliability, iridiumAvailable, iridiumReliability, droneBatteryPct, backhaulState, sourceNode⟩

To support position-aware reasoning, the event model distinguishes between the *reported animal position*, directly transmitted by the wearable node, and the *observation context* generated by the drone at the time of packet reception. Some rules, such as geofence violations, can therefore operate on a precise collar-side position when available, while other rules rely on the last-known observation context derived from the drone trajectory and radio reception metadata. This choice allows the monitoring layer to remain applicable even when the wearable node is kept lightweight and does not continuously expose full positioning data.

For alerts generated by the rule engine, the system also uses a normalized dispatch context:

⟨alertId, animalId, severity, generatedAt, latencyBudgetMs, escalationTimeoutMs, minDeliveryProbability, requireEndToEndAck, selectedChannels, dispatchOutcome, retryCount, ackedAt⟩

This abstraction decouples rule evaluation from the original packet structure and enables the communication policy to be analyzed using the same event-based logic used for animal monitoring.

4.1. Rule-driven dependable dispatch

A relevant implementation aspect concerns how the system maps alert criticality to communication behavior under intermittent connectivity. Rather than selecting a single channel through a static rule, the *Dependable Delivery Dispatcher* implements a *terrestrial-first, time-bounded escalation policy*. For each outgoing alert, the dispatcher first attempts delivery over the currently available terrestrial backhaul, preferring Wi-Fi when the UAV is in proximity of the farmer-side station and LTE/4G otherwise. When both terrestrial channels are available, critical alerts may be dispatched redundantly to reduce delivery uncertainty. Each transmission attempt is recorded under a unique `messageId` and associated with an acknowledgment deadline.

A delivery attempt is considered *successful* only when the Farmer Edge Node returns an application-layer acknowledgment after the alert has been ingested and persisted by the CONCERN platform. Lower-level transport confirmations are treated as intermediate signals, but they are not counted as end-to-end delivery because the system ultimately cares about alert availability on the monitoring platform.

If no end-to-end acknowledgment is received within a configurable timeout (e.g., 10 s), the dispatcher escalates by transmitting the same alert through the Iridium SBD modem, while keeping terrestrial probing active in the background whenever possible. Escalation, therefore, does not permanently switch the system to satellite mode. Instead, the dispatcher continues Wi-Fi scanning, LTE registration, and link-state refreshes so that subsequent alerts can return to terrestrial delivery as soon as a cheaper, lower-latency path becomes available. To prevent excessive airtime consumption and duplicate flooding, the dispatcher enforces bounded retries, per-channel rate limits, and retransmission backoff for non-critical traffic, and it cancels pending attempts for a given message as soon as an end-to-end acknowledgment is received from any channel. On the receiver side, duplicate deliveries are handled through idempotent de-duplication based on the alert identifier.

This policy is reflected in the dispatch context used by the rule engine. In addition to alert severity, the context carries the latency budget, the minimum required end-to-end delivery probability, the availability and estimated reliability of each backhaul option, and the acknowledgment mode required by the alert class. For critical alerts, the reliability target applies to the *dispatch plan as a whole*, rather than to a single isolated channel. Consequently, a terrestrial-first attempt followed by time-bounded satellite escalation can satisfy a stringent reliability objective even if no individual terrestrial link alone would meet it.

```
rule "Critical distress with dependable dispatch"
salience 100
when
    $anchor : CattleTelemetryEvent( $animalId : animalId )
    Number( intValue >= 3 ) from accumulate( CattleTelemetryEvent(
        animalId == $animalId, this after[0s,10m] $anchor,
        motionState == "IMMOBILE", sensorTemp >= 39.5), count(1))
    not CattleTelemetryEvent(animalId == $animalId, this after[0s,10m] $anchor,
        motionState == "MOVING")
    $link : LinkStateEvent(this after[0s,30s] $anchor, droneBatteryPct >= 20)
then
    Alert alert = AlertFactory.createCriticalAlert($animalId,
        "Possible animal distress: prolonged immobility with fever",
        $anchor.getEventTime());
    DispatchContext ctx = new DispatchContext();
    ctx.setPriority(AlertPriority.CRITICAL); ctx.setEscalationTimeoutMs(10000L);
    ctx.setRequireEndToEndAck(true);
       ctx.setAllowParallelTerrestrialDispatch(true);
    ctx.setLatencyBudgetMs(20000L); ctx.setMinDeliveryProbability(0.995);
    ctx.setWifiAvailable($link.isWifiAvailable());
    ctx.setWifiReliability($link.getWifiReliability());
    ctx.setLteAvailable($link.isLteAvailable());
    ctx.setLteReliability($link.getLteReliability());
    ctx.setIridiumAvailable($link.isIridiumAvailable());
    ctx.setIridiumReliability($link.getIridiumReliability());
    ctx.setDroneBatteryPct($link.getDroneBatteryPct());
    dispatchReliably(alert, ctx);
end
```

Listing 1. Drools rule with dependable dispatch.

The rule in Listing 1 is representative because it couples anomaly detection with a dispatch policy that explicitly accounts for timeliness, fallback, and end-to-end acknowledgment semantics. For evaluation purposes, this enables assessment of both monitoring quality and delivery assurance through end-to-end latency, budget adherence, channel usage, and fallback activation. The monitoring rules exposed through the CONCERN infrastructure are intentionally configurable in order to reduce false-positive alerts under heterogeneous operational conditions. Representative tuning mechanisms include minimum-event accumulation thresholds, such as requiring multiple *motionState* readings within a temporal window rather than reacting to a single sample, configurable observation windows, severity levels, and operator-adjustable rule parameters for motion and temperature conditions. These mechanisms allow monitoring sensitivity to be adapted to different livestock-management scenarios without modifying the underlying monitoring infrastructure.

4.2. Satellite webhook integration

A further implementation step was the integration of the Iridium fallback with the operational CONCERN backend. To this end, a dedicated REST entry point was added to the platform so that Cloudloop can notify CONCERN as soon as a mobile-originated RockBLOCK message is delivered by the satellite service. The current integration path therefore comprises the RockBLOCK modem on the drone, the Iridium network, the Cloudloop platform, an HTTP webhook destination, and the CONCERN REST API.

When the webhook is triggered, the backend decodes the hexadecimal payload, extracts the message identifier and transmission timestamp, computes the elapsed time between satellite transmission and Complex Event Processing (CEP) side reception, and creates a `SatelliteMessage` event that is inserted into the Drools engine and persisted by the platform. In the current prototype, the transported payload contains the monitoring information originally emitted by the LoRa Collar Node, including the collar-side data fields and the position associated with the transmitted message. The webhook metadata further exposes delivery-related information, including the RockBLOCK serial number, the MOMSN sequence number, and the satellite CEP indicator. This makes the satellite fallback observable within the same monitoring infrastructure used for livestock rules.

A dedicated rule named `satellite-message-received`, shown in Listing 2, was then introduced to log successful webhook deliveries at CEP level. This rule is intentionally simple: it confirms that the satellite message reached the application, reports the decoded payload, and records the measured application-visible latency. The same instrumentation was used in the direct timing campaign reported in Section 5.1.

```
rule "satellite-message-received"
    no-loop
    when
        $event: ConcernBaseEvent(this.getName() == "SatelliteMessage")
    then
        String serial = getSerial($event);
        int momsn = getMomsn($event);
        String payload = getPayload($event);
        double latency = getLatency($event);
        KieLauncher.printer("Satellite message #" + momsn + " from " + serial +
                            " | Latency: " + String.format("%.1f",
                            latency) + "s" | Payload: " + payload);
end
```

Listing 2. Drools rule for satellite notification.

A second relevant implementation aspect concerns synchronization semantics. Since agricultural missions may occur in areas without continuous terrestrial backhaul, the drone must support a store-carry-and-forward mode. In this mode, the drone maintains a local queue of raw events, derived alerts, and dispatch metadata, including transmission attempts, selected channels, first-send timestamp, retry counters, and acknowledgment status. Once a connection becomes available, the synchronization manager uploads pending data to the Farmer Edge Node, preserving temporal order, mission provenance, and alert identifiers for safe de-duplication. This mechanism keeps the system operational even when online monitoring is only partially feasible, while preserving traceability of both monitoring outcomes and communication behavior.

5. Prototype-based evaluation

The current stage of the work is assessed through a controlled, prototype-based evaluation in an open environment before moving to an extended farm campaign. The goal is not yet to claim full ecological validity, but to verify whether the proposed architecture satisfies the targeted monitoring and delivery properties under reproducible conditions. The prototype was exercised without live animals, while adopting open-area distances and communication conditions designed to approximate realistic cattle-monitoring scenarios.

The evaluation is organized around four research questions:

- **RQ1**: Can the hybrid monitoring architecture detect livestock-related concerns with sufficiently low latency for practical mission support?
- **RQ2**: Does dependable multi-channel dispatch improve end-to-end delivery of critical alerts compared with fixed communication policies?
- **RQ3**: What trade-off emerges between alert timeliness, delivery robustness, and communication cost when Iridium is introduced as a resilient fallback?
- **RQ4**: Is the system robust to intermittent terrestrial backhaul through a combination of local buffering, delayed synchronization, and satellite escalation?

To address these questions, the evaluation compares four deployment and dispatch configurations:

1. *Ground-centric monitoring*: rule evaluation is performed only on the Farmer Edge Node and the drone forwards telemetry without local reasoning;

2. *Drone-centric monitoring*: rule evaluation is performed only on the drone;
3. *Hybrid monitoring with fixed policy*: critical rules are evaluated on the drone, but alerts are forwarded using a static LTE-first strategy;
4. *Hybrid monitoring with dependable dispatch*: critical rules are evaluated on the drone and alerts are delivered through the Dependable Dispatcher using Wi-Fi, LTE/4G, Iridium, or deferred synchronization.

The experimental setup emulates a cattle-monitoring mission by replaying LoRa event traces generated from synthetic or semi-synthetic cattle trajectories and sensor readings. The replay engine injects controlled variations such as missing beacons, prolonged immobility, out-of-range temperature values, degraded RSSI/SNR, temporary Wi-Fi unavailability, LTE outages, and satellite-only fallback conditions. The considered mission profile assumes waypoint-based UAV coverage of the monitored area rather than active animal-following behavior; telemetry collection is therefore modeled as opportunistic LoRa beacon reception during flight. This controlled setup allows the same mission conditions to be reproduced across all deployment configurations.

The monitored metrics are the following:

- *detection latency*: time elapsed between the occurrence of a triggering condition and the generation of an alert;
- *alert completeness*: proportion of injected anomalies that are correctly detected;
- *end-to-end alert latency*: time elapsed between alert generation and alert availability on the Farmer Edge Node;
- *delivery success within budget*: proportion of critical alerts delivered within the adopted latency budget;
- *channel-selection distribution*: proportion of alerts delivered through Wi-Fi, LTE/4G, Iridium, or deferred synchronization;
- *buffered critical alerts*: number of critical alerts deferred because no acceptable real-time channel was available.

A representative experiment concerns the rule shown in Listing 1. In the considered mission, the same anomaly type is injected while the availability and quality of Wi-Fi, LTE/4G, and Iridium vary over time. This makes it possible to assess whether the dependable dispatcher can preserve alert timeliness under fluctuating connectivity while containing the use of costly channels. The experiment, therefore, links livestock-oriented anomaly detection with communication resilience.

The evaluation combines three evidence sources. The collar-to-drone hop is grounded in measured LoRa data collected on the prototype and then replayed across configurations. Wi-Fi and LTE/4G conditions are reproduced through controlled availability profiles and technology-specific latency/reliability models. In addition, the RockBLOCK-to-Cloudloop-to-Concern chain was directly exercised on the operational prototype, ensuring that the satellite fallback is supported by real, application-visible timing measurements rather than documentation alone.

The main limitation of the current evaluation is that, although the prototype is tested in open conditions with distances comparable to those in realistic deployments and the satellite path has been directly timed, the campaign does not involve live animals. As a consequence, the study cannot capture all field-related factors, such as animal behavioral variability, body-induced radio occlusions, herd dynamics, and UAV disturbances during actual missions. The present campaign should therefore be interpreted as a controlled, reproducible, prototype-level validation step aimed at assessing monitoring latency, dispatch behavior, and resilient alert delivery before future in vivo farm studies and larger, repeated operational campaigns.

5.1. Direct validation of the satellite integration

The Iridium fallback was validated in the real integration chain, which includes the RockBLOCK modem, the Iridium SBD service, the Cloudloop platform, the Cloudloop HTTP webhook destination, and the CONCERN REST endpoint. In this configuration, the webhook notifies CONCERN as soon as the mobile-originated satellite message reaches the service platform. The backend then decodes the message, computes the elapsed time between the satellite transmission timestamp and webhook reception at the CEP level, persists the event, and inserts a `SatelliteMessage` event into the rule engine.

The measured quantity is the application-visible delivery time from message transmission to CEP notification. In the current prototype, the payload transmitted over the satellite link includes monitoring information from the LoRa Collar Node, including collar-side data fields and the position associated with the transmitted message. The webhook metadata additionally exposes delivery-related fields such as the RockBLOCK serial number, MOMSN, transmission timestamp, and satellite CEP information. An example of the message received on the Cloudloop interface is shown in Fig. 4. This enables satellite deliveries to be observed through the same event-processing infrastructure used for livestock monitoring.

Across repeated trials in open-sky conditions, the application-visible satellite delay ranged from about 10 s in the best cases to about 30 s when the session required additional transmission attempts. The tests were executed in a site with favorable sky visibility, typically with three satellites visible, so the upper-end values should be interpreted as retransmission or session-completion effects rather than as severe line-of-sight blockage. These measurements confirm that the satellite fallback remains within a tens-of-seconds reaction window, which is substantially faster than minute-scale deferred synchronization and sufficient for the critical-alert class considered in this work.

6. Evaluation results

This section reports the results of the prototype-based evaluation described in Section 5. The campaign used a 10-minute emulated mission (600 s, 300 LoRa events at 2 s intervals), during which nine anomaly injections of varying severity were distributed along the timeline. Four configurations were compared: (i) Ground-centric monitoring, (ii) Drone-centric monitoring, (iii) Hybrid monitoring with a fixed LTE-first policy, and (iv) Hybrid monitoring with dependable multi-channel dispatch.

The connectivity profile was designed to stress-test backhaul resilience. Wi-Fi was available only within the first and last 60 s of the mission, simulating proximity to the farm. LTE coverage was deliberately removed between $t = 180$ s and $t = 420$ s, simulating a rural dead zone with only sporadic service. Iridium SBD remained available throughout as a resilient fallback. In the replay-based mission, the satellite segment is abstracted through the empirically grounded model introduced below, while Section 5.1 reports direct timing measurements on the real RockBLOCK-Cloudloop-Concern path. Unless otherwise stated, critical-alert timeliness in the replay campaign is assessed against a 20 s latency target.

6.1. LoRa link characterization

The LoRa measurements used in this evaluation were collected in an open-area campaign using three transmission distances. After filtering invalid acknowledgments (RSSI = 0 dBm) and likely retransmissions (MSG_Time > 5000 ms), the dataset comprises 561 valid samples. Table 2 summarizes the per-distance statistics.

The measured RSSI values range from −112 dBm to −56 dBm, with a median of approximately −72 dBm, well above the typical LoRa sensitivity threshold of −137 dBm [34]. The mean single-hop latency across all distances is 358.1 ms, which is consistent with the LoRa spreading factor and payload size used in the experiment. These empirical values were used as input to the mission replay, where each emulated cattle beacon samples RSSI and latency from the measured distributions.

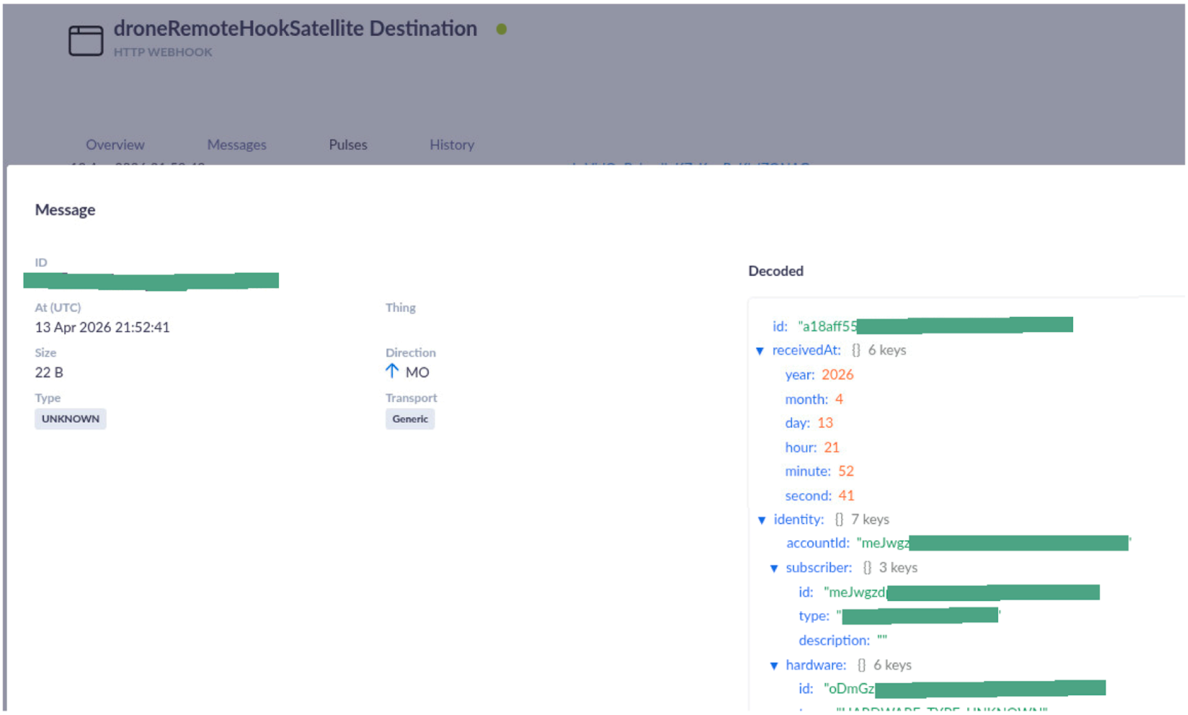

Fig. 4. Cloudloop HTTP destination configured to deliver RockBLOCK messages to the CONCERN webhook endpoint.

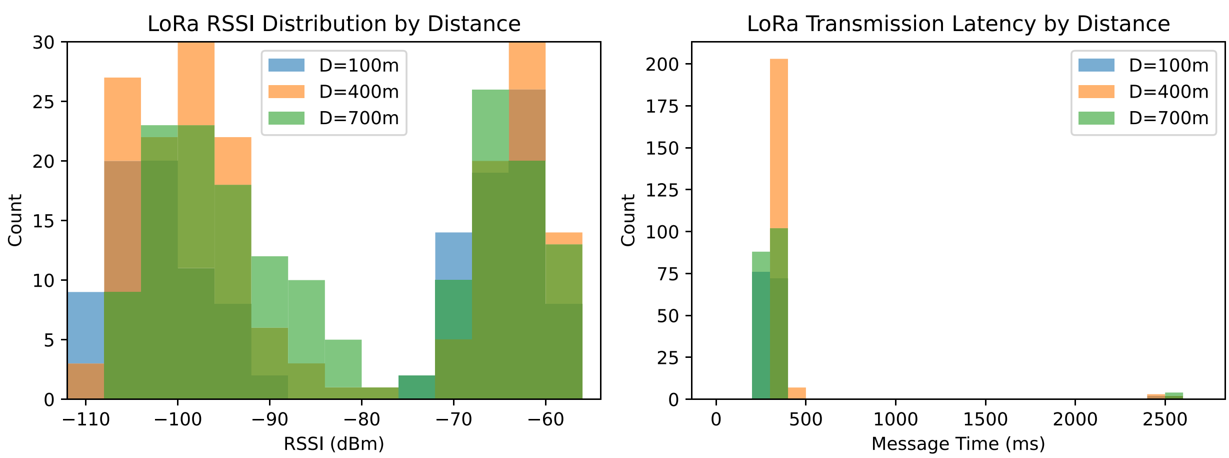

Fig. 5. LoRa RSSI distribution and transmission latency by distance, measured in open-area conditions.

Table 2
LoRa link characterization from open-area measurements.

Parameter	D = 100 m	D = 400 m	D = 700 m
Valid samples	152	215	194
RSSI range (dBm)	[−112, −56]	[−109, −58]	[−107, −57]
Mean MSG latency	304.1	446.5	302.6
Packet loss (%)	31.5	1.8	9.3

Fig. 5 shows the RSSI and transmission latency distributions grouped by distance. The unexpectedly higher loss observed at 100 m should not be interpreted as a monotonic distance effect; it is more plausibly explained by near-field effects, receiver saturation, or artifacts introduced by acknowledgment filtering and retransmission removal in the raw logs. Even with this anomaly, the overall characterization confirms that the collar-to-drone hop contributes sub-second latency and is not the dominant component of end-to-end alert delay.

6.2. Backhaul channel assumptions

The evaluation distinguishes between the first hop (collar-to-drone), which is grounded in the empirical LoRa measurements reported in Section 6.1, and the backhaul segment (drone-to-Farmer Edge Node), which is represented through a technology-specific channel model. Wi-Fi and LTE/4G are modeled through measurement-informed assumptions derived from the literature, whereas the Iridium SBD path is now anchored to the direct RockBLOCK-Cloudloop-Concern timing measurements reported in Section 5.1.

Wi-Fi is modeled as a low-latency, short-range channel available only in the vicinity of the farmer-side station, with latency distributed as $\mathcal{N}(12, 4^2)$ ms and nominal transport reliability of 0.98. LTE/4G is modeled as a coverage-dependent terrestrial channel with latency

Table 3
Backhaul channel parameters used in the evaluation.

Channel	Latency model	Transport reliability	Availability	Cost/msg
WiFi	$\mathcal{N}(12, 4^2)$ ms	0.98	Near-base only	Negligible
LTE/4G	$\mathcal{N}(55, 18^2)$ ms	0.96	Coverage-dependent	∼ $0.01
Iridium SBD	10–30 s empirical envelope	0.995	Global, sky-dependent	∼ $0.05
Deferred sync	30–180 s	1.00	Always available locally	Negligible

$\mathcal{N}(55, 18^2)$ ms and nominal transport reliability equal to 0.96 when service is available, consistently with recent rural and semi-rural measurements reported in the literature [35,36]. For Iridium SBD, the direct webhook-based integration tests reported in Section 5.1 showed application-visible delivery times of about 10–30 s from satellite transmission to CEP notification. The replay campaign, therefore, represents the satellite fallback through a bounded tens-of-seconds latency envelope anchored to those observations, while the reliability parameter remains conservatively configured at 0.995 under sky-visible conditions [37–39]. Deferred synchronization is modeled separately from the online channels as a store-carry-and-forward mechanism whose delivery time depends on outage duration rather than on instantaneous channel latency (Table 3).

Under this empirically grounded envelope, the satellite-assisted path is treated as a bounded fallback for critical alerts with a tens-of-seconds reaction window, while still remaining clearly slower and more expensive than terrestrial delivery. This choice is sufficient for comparing dispatch policies and is now supported by direct end-to-end timing measurements along the webhook-based integration path.

6.3. Direct RockBLOCK-to-CEP timing observations

Beyond the replay-based mission, the satellite path was exercised on the real integration chain connecting the RockBLOCK modem, the Iridium service, the Cloudloop platform, and the CONCERN webhook endpoint. In these trials, the application-visible delay was measured from the satellite transmission timestamp to the instant at which the webhook notification became available to the CEP layer. The direct measurements ranged from approximately 10 s in the best cases to approximately 30 s when additional transmission attempts were needed before completion.

These timings are operationally relevant because the transported message is not a synthetic ping: it contains the monitoring payload generated from the LoRa Collar Node, including the collar-originated information and the position associated with the transmitted message. Once received, the payload is decoded by the webhook endpoint and immediately inserted into the CEP engine, where the rule `satellite-message-received` confirms receipt at the application level. The result is a concrete validation that the satellite fallback used in the architecture is implemented and observable in the deployed prototype, rather than only represented in the replay model.

6.4. Detection latency (RQ1)

Table 4 reports the main evaluation metrics across the four configurations. All configurations detected the nine injected anomalies. The key difference lies in detection latency. Configurations with on-drone rule evaluation (Drone-centric, Hybrid-fixed, Hybrid-dependable) achieved mean detection latencies of 33.2 ms, 28.6 ms, and 27.5 ms, respectively, because rules are evaluated locally immediately upon event reception. In contrast, the Ground-centric configuration required forwarding telemetry before detection, resulting in a mean detection latency of 1364.4 ms, dominated by the forwarding path and ground-side processing.

The values in Table 4 show that local rule execution reduces mean detection latency by roughly two orders of magnitude with respect to ground-centric processing. For a livestock-monitoring system, this means that abnormal situations can be recognized while the UAV is still in the field, rather than waiting for full telemetry to be forwarded to the farmer-side station. The false-positive count is zero in all configurations because the replay-based anomaly injection uses known ground-truth conditions. This result should therefore not be interpreted as a statistical characterization of false-alarm behavior under real livestock-monitoring deployments.

Table 4
Summary of evaluation results across the four configurations. Nine anomalies were injected in the mission script.

Metric	Ground	Drone	Hybrid-F	Hybrid-D
Alert completeness	9/9	9/9	9/9	9/9
Mean det. latency (ms)	1364.4	33.2	28.6	27.5
Mean E2E latency (ms)	36,482	55,443	48,880	4234
Median E2E latency (ms)	1943	106	109	137
Delivery in budget	6/9	5/9	6/9	**9/9**
Buffered critical alerts	4	4	4	0
False positive alerts	0/9	0/9	0/9	0/9

6.5. Dependable dispatch effectiveness (RQ2)

The most significant finding concerns delivery success within the latency budget. The Hybrid-dependable configuration delivered all nine critical alerts within the adopted 20 s budget, compared with six out of nine for the Ground-centric and Hybrid-fixed configurations and five out of nine for the Drone-centric configuration. The improvement is attributable to the dispatcher's ability to escalate to Iridium SBD when Wi-Fi and LTE are unavailable, rather than deferring the alert to store-carry-and-forward.

Fig. 6 compares the delivery success ratios. The fixed-policy configurations suffered during the LTE dead zone ($t \in [180, 420]$ s), where four of the nine anomaly injections occurred. In these cases, alerts were deferred to local storage and synchronized only after terrestrial connectivity was restored, resulting in end-to-end latencies exceeding 30 s and often above 60 s. In contrast, the dependable dispatcher selected Iridium for four alerts during the outage period, avoiding deferred delivery and keeping satellite-assisted alert latency within the order of tens of seconds under the adopted model and the direct measurements reported for the real integration path.

Fig. 7 shows the channel selection distribution for the Hybrid-dependable configuration. Wi-Fi was selected for two alerts generated near the base; LTE was used for three alerts within coverage; and Iridium was activated for the remaining four alerts during the terrestrial outage. No critical alerts were deferred.

6.6. Latency–reliability–cost trade-off (RQ3)

The introduction of Iridium as a resilient fallback reduced the mean end-to-end latency of the Hybrid-dependable configuration to 4234 ms, compared with 36482–55443 ms for the other configurations. The cost of this improvement is the per-message satellite fee (approximately $0.05 per SBD message). Over the simulated mission, four Iridium transmissions were required, corresponding to an estimated cost of $0.20. In practical livestock monitoring, this cost appears modest relative to the operational value of keeping critical alerts timely when terrestrial coverage is lost.

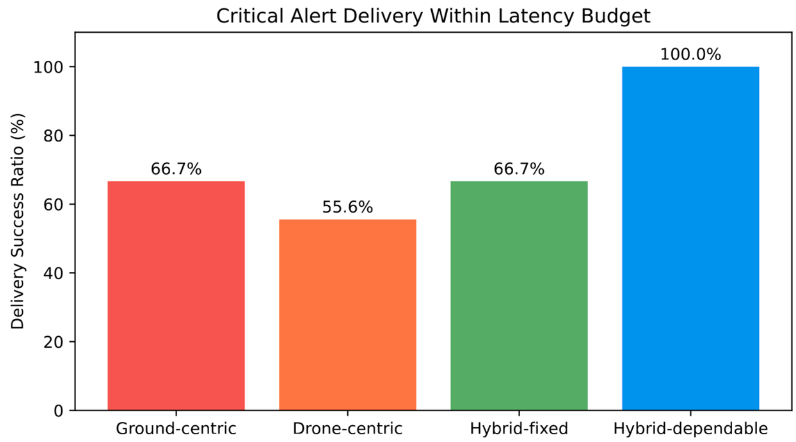

Fig. 6. Critical-alert delivery within the latency budget across configurations.

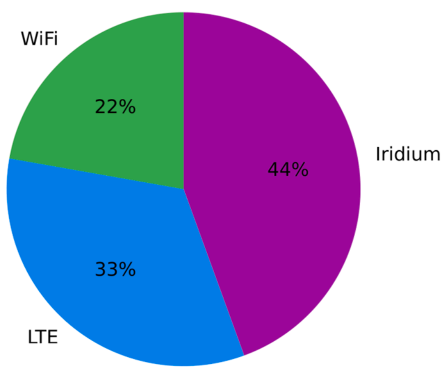

Fig. 7. Channel selection distribution for the Hybrid-dependable configuration.

The dispatcher uses satellite transmission only for critical alerts that cannot be served by terrestrial channels within the required timeliness constraints. Routine telemetry and non-critical information can still be deferred to synchronization, thereby avoiding unnecessary satellite costs. This behavior should therefore be interpreted as a policy-level trade-off: terrestrial communication remains the default low-cost path, while satellite escalation is activated only when required to preserve timely alert delivery. In large-scale or long-duration livestock-monitoring deployments, the cumulative number of satellite-assisted alerts may nevertheless increase depending on herd size, mission duration, environmental conditions, and rule sensitivity. For example, repeated connectivity outages combined with highly sensitive monitoring rules could lead to multiple satellite-assisted deliveries during a single operational mission. The proposed communication policy is therefore intentionally designed to keep Iridium usage sparse and event-driven, reserving satellite escalation for operationally relevant critical conditions. Although the present study does not provide a complete energy characterization of the system, the hardware configuration adopted in the prototype was selected to remain compatible with lightweight UAV-assisted deployments. As summarized in Table 1, the drone-side monitoring layer operates on a Raspberry Pi platform with limited computational and power requirements, while satellite escalation is intentionally kept sparse and event-driven to limit additional communication overhead. In practical deployments, the dominant energy constraint is expected to remain UAV propulsion rather than the monitoring logic itself. A more comprehensive energy characterization across extended operational missions remains part of future work.

6.7. Resilience to connectivity disruptions (RQ4)

Fig. 8 provides a timeline view of the mission, showing channel availability, LoRa signal quality, and alert dispatch events. The LTE dead zone between $t = 180$ and $t = 420$ s is clearly visible. During this interval, the fixed-policy configurations accumulated deferred alerts that were only delivered after reconnection at $t \approx 420$ s.

The Hybrid-dependable configuration maintained zero buffered critical alerts throughout the mission. In contrast, all other configurations accumulated three to four deferred alerts during the outage, with synchronization occurring only when LTE or Wi-Fi became available again. For configurations without satellite fallback, the worst-case end-to-end latency for a critical alert exceeded 120 s, The acceptable reaction window for livestock-monitoring alerts depends on the operational context and on the type of monitored condition [2,3]. In particular, events involving prolonged immobility, abnormal temperature conditions, or communication failures may require timely operator awareness to support rapid intervention. In this context, the adopted 20 s latency target is intended specifically for critical alerts requiring prompt delivery, rather than as a universally applicable requirement for all livestock-monitoring events.

This result is relevant because the gain does not arise solely from faster local monitoring. Rather, it derives from the combination of local alert generation, explicit criticality-aware dispatch, and the availability of a satellite-assisted fallback path when terrestrial connectivity becomes temporarily unavailable.

6.8. End-to-end latency analysis

Fig. 9 presents the mean and median end-to-end latencies on a logarithmic scale. The large gap between mean and median for the Ground-centric, Drone-centric, and Hybrid-fixed configurations reflects the bimodal latency distribution caused by deferred alerts: most alerts are delivered within a few hundred milliseconds, but deferred alerts inflate the mean by one to two orders of magnitude. The Hybrid-dependable configuration shows a much smaller mean–median gap (4234 ms vs. 137 ms), because satellite-assisted alerts incur delays of several to tens of seconds rather than the 30–180 s delays associated with deferred synchronization.

The mean value of the Hybrid-dependable configuration remains higher than the median because only a subset of alerts experiences satellite escalation. This is consistent with the intended policy semantics: the system is not designed to make all traffic uniformly fast, but to keep critical alerts within a bounded time window even under backhaul degradation.

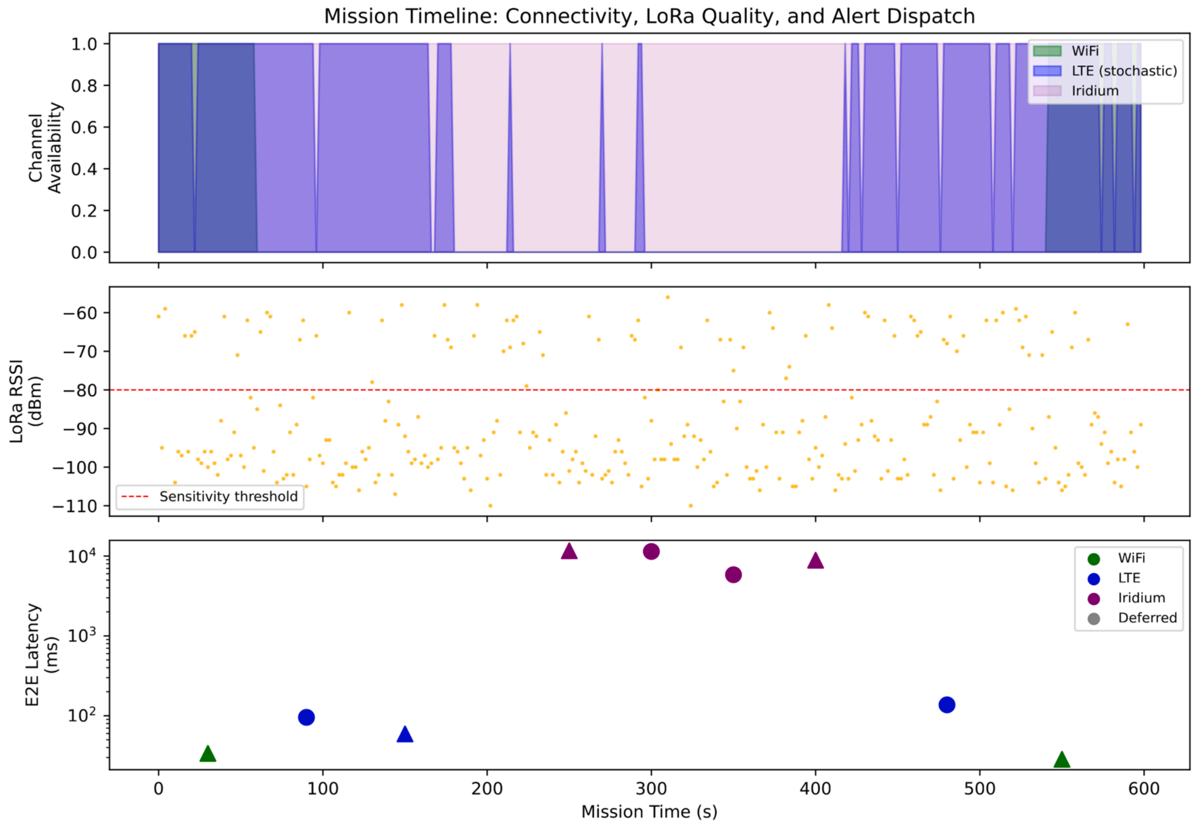

Fig. 8. Mission timeline showing channel availability, LoRa RSSI, and alert dispatch events for the Hybrid-dependable configuration. Triangles denote critical alerts; circles denote non-critical alerts.

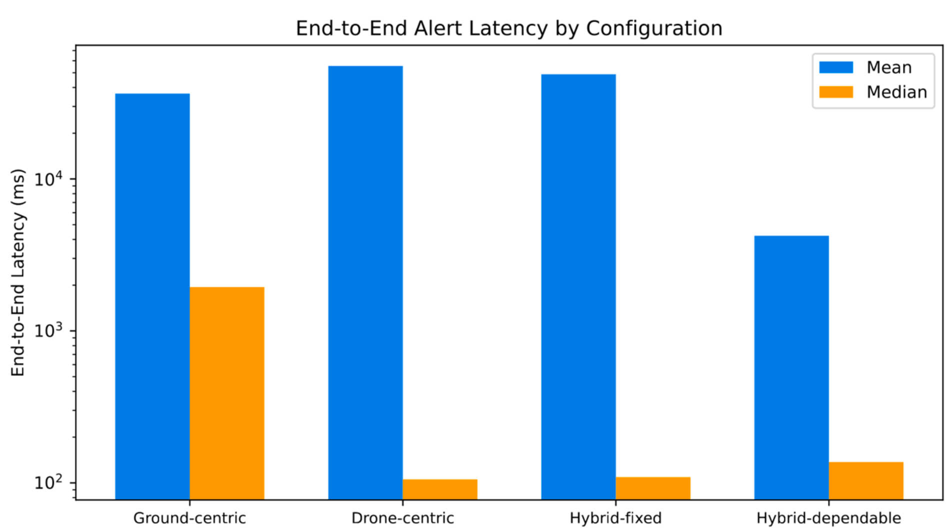

Fig. 9. Mean and median end-to-end alert latency (log scale).

6.9. *Additional mission-profile sensitivity analysis*

To provide preliminary evidence regarding the robustness of the proposed dispatch strategy under different operational conditions, additional replay-based mission profiles were considered beyond the primary evaluation scenario. These profiles do not constitute independent field campaigns, but controlled variations of the replay environment designed to stress the communication layer under different connectivity conditions.

Three representative profiles were analyzed: (i) a short LTE outage scenario (MP-A), (ii) a prolonged terrestrial outage scenario (MP-B), and (iii) an intermittent LTE-instability scenario with fluctuating terrestrial availability (MP-C). The replay engine preserved the same livestock-monitoring logic and anomaly-injection strategy adopted in the primary

Table 5
Additional replay-based mission-profile observations for the Hybrid-dependable configuration.

Profile	Connectivity condition	Delivery in budget	Iridium alerts	Buffered critical alerts
MP-A	Short LTE outage (60 s)	9/9	2	0
MP-B	Long LTE outage (300 s)	9/9	5	0
MP-C	Intermittent LTE instability	8/9	3	1

campaign while varying the temporal characteristics of backhaul degradation.

Table 5 summarizes the observed behavior of the dependable dispatch configuration under these additional profiles.

The results remain consistent with the experiment's main findings. In particular, the dependable dispatcher ensures timely delivery for most critical alerts even during prolonged or unstable terrestrial outages, while escalating satellite requests only when necessary to avoid deferred synchronization. The intermittent-connectivity profile (MP-C) also suggests that highly variable terrestrial conditions may still result in occasional budget violations despite resilient dispatching, highlighting the importance of additional large-scale operational campaigns for a more comprehensive statistical characterization.

6.10. Discussion

The results support the idea that hybrid monitoring with dependable multi-channel dispatch offers the best trade-off among the evaluated qualities for the considered mission profile. On-drone rule evaluation reduces detection latency by approximately 2 orders of magnitude compared to ground-centric processing, while the dependable dispatcher eliminates deferred critical alerts by exploiting a satellite fallback when terrestrial channels are unavailable.

The LoRa characterization confirms that the first communication hop contributes relatively modest latency (≈ 300–450 ms) and that the dominant factor in end-to-end alert timeliness is the backhaul segment. This observation reinforces the architectural decision to place a lightweight monitoring capability on the drone instead of delegating all reasoning to the ground. However, for extended operational missions spanning several hours, the dominant endurance constraint is expected to remain UAV propulsion rather than the onboard monitoring logic itself. In this context, prolonged coverage would likely require battery replacement strategies, professional drones, scheduled recharge intervals, or cooperative multi-UAV deployments, whereas the computational and communication overhead introduced by the monitoring infrastructure remains comparatively limited. At the same time, the current evaluation remains partly model-assisted for the backhaul. Wi-Fi and LTE/4G are represented through measurement-informed assumptions from the literature, whereas the Iridium path is now calibrated from direct webhook-based timing measurements on the actual RockBLOCK integration chain. The reported satellite-assisted results should therefore be interpreted as policy evidence supported by preliminary real-time observations, rather than as a complete statistical characterization of every operational condition.

Two limitations should be emphasized. First, the satellite timing evidence should be extended through larger, repeated RockBLOCK campaigns that cover different sky conditions, payload sizes, and operational contexts. Second, the current evaluation uses a single mission profile; additional profiles with different outage patterns, payload rates, and mission durations are needed to assess generalizability more thoroughly. These limitations are consistent with the threats to validity discussed in Section 6.12.

6.11. Scalability considerations

The current evaluation considers a moderate number of concurrent collars. To provide preliminary evidence on the scalability of the monitoring layer, we refer to the stress-testing results obtained with MOTEF [32], where the CONCERN rule engine sustained thousands of events per second with sub-millisecond per-event evaluation time on commodity hardware, and rule-set growth up to several hundred active rules introduced only linear overhead.

In the cattle-monitoring scenario, the expected event rate is bounded by the number of collars and their beaconing interval. With 100 collars transmitting every 2 s, the drone-side engine would process approximately 50 events/s, which is well within the throughput measured in [32]. The practical bottleneck for larger herds is therefore more likely to be LoRa channel capacity and collision probability than rule-evaluation overhead. A dedicated scalability campaign with multiple concurrent transmitters remains part of future work.

6.12. Threats to validity

The present evaluation is subject to several threats to validity.

Internal validity. The assessment is performed on a working prototype through controlled experiments and replayable scenarios, but without live animals. Although the considered distances and communication conditions approximate realistic livestock-monitoring settings, the absence of animal behavior removes several sources of variability, including irregular mobility, body-shadowing effects, collar orientation changes, and herd dynamics. As a consequence, telemetry generation and radio-link behavior may be more regular than in real farm deployments. Nevertheless, the controlled setup improves reproducibility across deployment configurations and allows communication and dispatch behaviors to be evaluated under comparable mission conditions.

Construct validity. The selected metrics capture relevant technical qualities of the proposed monitoring and dispatching architecture, including detection latency, alert timeliness, delivery success within budget, and resilience to connectivity disruptions. Likewise, the replay-based setup does not allow for a statistically representative estimate of false-positive alert rates under realistic conditions of animal behavioral variability and environmental noise. However, they do not directly measure higher-level farming outcomes, such as welfare improvements, reduced manual inspection effort, or long-term operational benefits across heterogeneous husbandry conditions. In addition, for the satellite path, transport completion and end-to-end platform delivery are not equivalent constructs: a successful SBD session only shows that the message has reached the service infrastructure, whereas the paper ultimately cares about alert availability after ingestion and persistence.

External validity. The prototype has not yet been validated in a full in vivo deployment. Real farms may introduce additional variability due to terrain morphology, vegetation, weather, herd dispersion, long-duration flights, and mission-specific operational constraints. The satellite path may also be affected by sky visibility, power-supply conditions, and integration details that are only partially represented in the current study. Transferability to production livestock-monitoring deployments must therefore be confirmed through field trials.

Reliability and reproducibility. Part of the evaluation relies on controlled anomaly injection, emulated backhaul outages, and replayable mission scripts. This improves repeatability, but it may under-represent the stochastic behavior of real wireless links and mission dynamics. In addition, the current results are based on a single mission profile and should not be over-generalized to longer or more heterogeneous operational scenarios.

Integration-path validity. The satellite fallback depends on a multi-stage path that includes the onboard modem, the Iridium SBD transport, the Ground Control delivery service, and the CONCERN ingestion endpoint. Official documentation states that HTTP webhook delivery is the preferred integration path, but also that the endpoint must return HTTP 200 within a short time window; otherwise, the message remains queued, and retries are restarted after one minute with exponential backoff [38]. This means that satellite availability alone does not guarantee end-to-end deadline satisfaction: failures in the application endpoint or in its acknowledgment logic can shift delivery from a second-scale path to a minute-scale retry schedule. Accordingly, any claim about critical-alert timeliness under satellite fallback should be interpreted with respect to the full integration chain, not only to the modem-side transmission outcome.

7. Conclusions

This paper presented a smart monitoring architecture for cattle farming in infrastructure-poor agricultural environments, where timely anomaly detection and dependable alert delivery cannot rely on continuous terrestrial connectivity. The proposed solution combines low-power LoRa-based collars, a UAV that acts as both a mobile gateway and an edge monitoring node, and a farmer-side platform that supports rule configuration, event inspection, and post-mission analysis. Its central contribution is the explicit integration of explainable rule-based livestock monitoring with a resilient multi-channel communication strategy.

The architecture adopts a hybrid design in which a lightweight subset of monitoring rules is evaluated directly on the drone, while the full monitoring stack remains available on the Farmer Edge Node. This choice allows the system to react in the field to operationally relevant conditions such as missing beacons, geofence issues, prolonged immobility, communication degradation, and alert-delivery failures, without moving unnecessary complexity onto the wearable device. In parallel, the dependable dispatcher implements terrestrial-first forwarding with time-bounded satellite escalation, ensuring critical alerts still reach the monitoring platform when Wi-Fi or LTE/4G connectivity becomes unavailable.

The prototype-based evaluation provides encouraging evidence for this design choice. In the considered mission profile, local rule execution on the drone reduced detection latency by approximately 2 orders of magnitude compared to a ground-centric solution. The results also showed that end-to-end alert timeliness is dominated not by the LoRa collar-to-drone hop, but by the backhaul strategy adopted after alert generation. Under terrestrial outages, the hybrid configuration with dependable dispatch delivered all critical alerts within the adopted latency budget, avoided deferred critical notifications, and achieved this behavior through selective satellite activation rather than pervasive use of costly fallback communication. In addition, the RockBLOCK–Cloudloop–CONCERN integration path was exercised directly on the operational prototype, confirming that the satellite-assisted path is not only modeled in replay experiments but also observable at the application level.

At the same time, the present study should be interpreted as a prototype-level validation rather than a full-farm demonstration. The evaluation does not yet include live animals, repeated campaigns across multiple mission profiles, or a complete characterization of energy overhead and long-duration operational behavior. For this reason, the current results support the feasibility and practical value of the proposed architecture, but not yet its full generalization to production livestock deployments.

Future work will extend the study along four directions: repeated in vivo farm trials with live animals; broader experimental campaigns with multiple outage patterns, mission durations, and herd scales; deeper characterization of the satellite-assisted path under different environmental and operational conditions; and energy-aware extensions of the dispatch policy, including cooperative multi-UAV monitoring for larger grazing areas. These steps will be necessary to move from strong prototype evidence to robust field validation in precision livestock farming.

CRediT authorship contribution statement

Antonello Calabrò: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Data curation, Conceptualization; **Eda Marchetti:** Writing – review & editing, Supervision, Methodology, Conceptualization, Investigation, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data contains position of private areas and should not be shared. Code is available on git.

References

[1] A. Calabrò, E. Marchetti, P. Skrzypek, J. Marchel, CONCERN: A model-based monitoring infrastructure, Internet of Things 31 (2025) 101557. https://doi.org/10.1016/j.iot.2025.101557

[2] C. Aquilani, A. Confessore, R. Bozzi, F. Sirtori, C. Pugliese, Review: precision livestock farming technologies in pasture-based livestock systems, Animal 16 (1) (2022) 100429. https://doi.org/10.1016/j.animal.2021.100429

[3] B.C. Besler, P. Mojabi, Z. Lasemiimeni, J.E. Murphy, Z. Wang, R. Baker, J.M. Pearson, E.C. Fear, Scoping review of precision technologies for cattle monitoring, Smart Agric. Technol. 9 (2024) 100596. https://doi.org/10.1016/j.atech.2024.100596

[4] S. Montalván, P. Arcos, P. Sarzosa, R.A. Rocha, S.G. Yoo, Y. Kim, Technologies and solutions for cattle tracking: a review of the state of the art, Sensors 24 (19) (2024) 6486. https://doi.org/10.3390/s24196486

[5] G. Codeluppi, A. Cilfone, L. Davoli, G. Ferrari, LoRaFarM: a LoRaWAN-based smart farming modular IoT architecture, Sensors 20 (7) (2020) 2028. https://doi.org/10.3390/s20072028

[6] H. Lee, W. Jang, H.W. Yoon, D.-J. Kim, H. Jung, G. Choi, M. Lee, C. Weon, A. Smith, Feasibility of networking technology for smart farm: LoRa vs APRS, in: 2020 International Conference on Omni-layer Intelligent Systems (COINS), IEEE, 2020, pp. 1–6. https://doi.org/10.1109/COINS49042.2020.9191428

[7] H.W. Yoon, D.-J. Kim, M. Lee, C. Weon, A. Smith, L&M farm: a smart farm based on LoRa & MQTT, in: 2020 International Conference on Omni-layer Intelligent Systems (COINS), IEEE, 2020, pp. 1–6. https://doi.org/10.1109/COINS49042.2020.9191387

[8] S. Escolar, F. Rincón, X. del Toro, J. Barba, F.J. Villanueva, M.J. Santofimia, D. Villa, J.C. López, The PLATINO experience: a LoRa-based network of energy-harvesting devices for smart farming, in: 2019 XXXIV Conference on Design of Circuits and Integrated Systems (DCIS), IEEE, 2019, pp. 1–6. https://doi.org/10.1109/DCIS201949030.2019.8959848

[9] M. Behjati, A.B.M. Noh, H.A.H. Alobaidy, M.A. Zulkifley, R. Nordin, N.F. Abdullah, LoRa communications as an enabler for internet of drones towards large-scale livestock monitoring in rural farms, Sensors 21 (15) (2021) 5044. https://doi.org/10.3390/s21155044

[10] M. Alipio, M.L. Villena, Intelligent wearable devices and biosensors for monitoring cattle health conditions: a review and classification, Smart Health 27 (2023) 100369. https://doi.org/10.1016/j.smhl.2022.100369

[11] M. Lamanna, M. Bovo, D. Cavallini, Wearable collar technologies for dairy cows: a systematized review of the current applications and future innovations in precision livestock farming, Animals 15 (3) (2025) 458. https://doi.org/10.3390/ani15030458

[12] C.P. de Freitas, A. Selli, D.L. Pinto, A. Merlos-Ruiz, B.J.C. de Carvalho, R.V. Ventura, Applications of livestock monitoring devices and machine learning algorithms in animal production and reproduction: an overview, Anim. Reprod. 20 (2) (2023) e20230077. https://doi.org/10.1590/1984-3143-AR2023-0077

[13] D.C. Tsouros, S. Bibi, P. Sarigiannidis, A review on UAV-Based applications for precision agriculture, Information 10 (11) (2019) 349. https://doi.org/10.3390/info10110349

[14] A. Mukherjee, S. Misra, N.S. Raghuwanshi, A survey of unmanned aerial sensing solutions in precision agriculture, J. Netw. Comput. Appl. 148 (2019) 102461. https://doi.org/10.1016/j.jnca.2019.102461

[15] P.I. Radoglou-Grammatikis, P.G. Sarigiannidis, T. Lagkas, I.D. Moscholios, A compilation of UAV applications for precision agriculture, Comput. Netw. 172 (2020) 107148. https://doi.org/10.1016/j.comnet.2020.107148

[16] J.G.A. Barbedo, L.V. Koenigkan, T.T. Santos, P.M. Santos, A study on the detection of cattle in UAV images using deep learning, Sensors 19 (24) (2019) 5436. https://doi.org/10.3390/s19245436

[17] C.A. Mücher, S. Los, G.J. Franke, C. Kamphuis, Detection, identification and posture recognition of cattle with satellites, aerial photography and UAVs using deep learning techniques, Int. J. Remote Sens. 43 (7) (2022) 2377–2392. https://doi.org/10.1080/01431161.2022.2051634

[18] K.M. Bastos, J.P. Barcelos, G.F. Orioli, S.T. Nascimento, Use of drones for trough reading, animal counting, and production monitoring in feedlot systems, AgriEngineering 6 (4) (2024) 4460–4475. https://doi.org/10.3390/agriengineering6040253

[19] H. Zhao, W. Tang, S. Chen, A. Li, Y. Li, W. Cheng, Design and implementation of a novel UAV-assisted LoRaWAN network, Drones 8 (10) (2024) 520.

[20] G. Canello, S. Mignardi, K. Mikhaylov, C. Buratti, T. Hänninen, Data collection from LoRaWAN sensor network by UAV gateway: design, empirical results and dataset, in: 2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), IEEE, 2023, pp. 555–560.

[21] R. Xiong, C. Liang, H. Zhang, X. Xu, J. Luo, FlyingLoRa: towards energy efficient data collection in UAV-assisted LoRa networks, Comput. Netw. 220 (2023) 109511. https://doi.org/10.1016/j.comnet.2022.109511

[22] J. Gallego-Madrid, A. Molina-Zarca, R. Sánchez-Iborra, J. Bernal-Bernabé, J. Santa, P.M. Ruiz, A.F. Skarmeta-Gómez, Enhancing extensive and remote LoRa deployments through MEC-powered drone gateways, Sensors 20 (15) (2020) 4109.

[23] M.H.M. Ghazali, K. Teoh, W. Rahiman, A systematic review of real-time deployments of UAV-based LoRa communication network, IEEE Access 9 (2021) 124817–124830.

[24] Sapna, H. Gauttam, V. Chauhan, K.K. Pattanaik, A. Trivedi, H. Ghosh, A comprehensive review of edge computing empowered smart agriculture: trends, opportunities and future directions, Comput. Electron. Agric. 241 (2026) 111252.

[25] R. Gong, H. Zhang, G. Li, J. He, Edge computing-enabled smart agriculture: technical architectures, practical evolution, and bottleneck breakthroughs, Sensors 25 (17) (2025) 5302.

[26] P. Yu, F. Teng, W. Zhu, C. Shen, Z. Chen, J. Song, Cloud–edge–device collaborative computing in smart agriculture: architectures, applications, and future perspectives, Front. Plant Sci. 16 (2025). https://doi.org/10.3389/fpls.2025.1668545

[27] J. Morales-García, A. Bueno-Crespo, R. Martínez-España, F.J. García, S. Ros, J. Fernández-Pedauyé, J.M. Cecilia, SEPARATE: a tightly coupled, seamless IoT infrastructure for deploying AI algorithms in smart agriculture environments, Internet Things 22 (2023) 100734. https://doi.org/10.1016/j.iot.2023.100734

[28] R.C.M. Teixeira, C.B. Carvalho, C.T. Calafate, E. Mota, R.A. Fernandes, A.L. Printes, L.B.F. Nascimento, FloatingBlue: a delay tolerant networks-enabled internet of things architecture for remote areas combining data mules and low power communications, Sensors 24 (19) (2024) 6218. https://doi.org/10.3390/s24196218

[29] A. Bertolino, A. Calabrò, F. Lonetti, A. Sabetta, GLIMPSE: a generic and flexible monitoring infrastructure, in: Proceedings of the 13th European Workshop on Dependable Computing (EWDC '11), ACM, 2011, pp. 73–78. https://doi.org/10.1145/1978582.1978598

[30] P. Barsocchi, A. Calabrò, E. Ferro, C. Gennaro, E. Marchetti, C. Vairo, Boosting a low-cost smart home environment with usage and access control rules, Sensors 18 (6) (2018) 1886.

[31] B. Mazon-Olivo, D. Hernández-Rojas, J. Maza-Salinas, A. Pan, Rules engine and complex event processor in the context of internet of things for precision agriculture, Comput. Electron. Agric. 154 (2018) 347–360. https://doi.org/10.1016/j.compag.2018.09.013

[32] A. Calabrò, E. Marchetti, MOTEF: a testing framework for runtime monitoring infrastructures, IEEE Access 12 (2024) 38005–38016. https://doi.org/10.1109/ACCESS.2024.3376239

[33] A. Calabrò, E. Marchetti, Transponder: support for localizing distressed people through a flying drone network, Drones 8 (9) (2024). https://doi.org/10.3390/drones8090465

[34] Semtech Corporation, SX1276/77/78/79 – 137 MHz to 1020 MHz Low Power Long Range Transceiver, 2020. Datasheet, Rev. 7, May 2020.

[35] J. Braunfelds, G. Jakovels, I. Murans, A. Litvinenko, U. Senkans, R. Rumba, A. Onzuls, G. Valters, E. Lidere, E. Plone, Experimental study on LTE mobile network performance parameters for controlled drone flights, Sensors 24 (20) (2024) 6615. https://doi.org/10.3390/s24206615

[36] R. Mozny, P. Masek, M. Stusek, K. Molnar, M. Palenska, D. Moltchanov, J. Hosek, Experimental quality assessment of cellular networks and their utilization for UAV services, in: Proceedings of the IEEE Vehicular Technology Conference (VTC-Spring), Florence, Italy, 2023, pp. 1–6. https://doi.org/10.1109/VTC2023-Spring57618.2023.10199980

[37] Iridium Communications Inc., Iridium Short Burst Data (SBD) Service, 2024, Global network latency: 5–20 s; max MO message: 1960 bytes; max MT message: 1890 bytes. Accessed: April 2026. https://www.iridium.com/services/iridium-sbd/.

[38] Ground Control, Receiving MO Messages via HTTP Webhook, 2024, Ground Control Knowledgebase, last updated Oct. 28, 2024. Accessed: Apr. 11, 2026. https://docs.groundcontrol.com/iot/rockblock/web-services/receiving-mo-message.

[39] Ground Control, RockBLOCK FAQ, 2025, Ground Control Knowledgebase. Accessed: Apr. 11, 2026. https://docs.groundcontrol.com/iot/rockblock/support/faq.