

REVIEW

Open access

Edge AI and TinyML for Affordable Precision Agriculture: A Framework-Oriented Review of Low-Power Intelligence, Deployment Constraints, and Farm-Scale Applications

Luca Ferraro^{1*}, Matteo Ricci¹, Giulia Moretti², Paolo Greco³

Abstract

Edge Artificial Intelligence and Tiny Machine Learning are increasingly discussed as approaches for moving selected analytical functions from remote cloud infrastructure to low-power devices located closer to agricultural data sources. Their potential relevance to precision agriculture lies in supporting local sensing, classification, monitoring, and decision-assistance functions where continuous connectivity, high-performance computing, or extensive data transmission may be impractical. This framework-oriented review examines Edge AI and TinyML through four connected dimensions: low-power intelligence, embedded farm systems, agricultural deployment constraints, and farm-scale applications and implementation barriers. The review distinguishes on-device inference and near-sensor intelligence from conventional agricultural Internet of Things sensing, generic wireless sensor networks, cloud-based artificial intelligence, and non-embedded machine-learning applications. It organizes the reviewed evidence around embedded hardware, model compression, memory, latency, energy use, sensor quality, connectivity, environmental exposure, calibration, maintenance, usability, affordability, local validation, cybersecurity, privacy, and human oversight. The synthesis indicates that compact inference can reduce reliance on continuous cloud access in appropriately designed systems, but miniaturized models alone do not establish accuracy, robustness, affordability, usability, sustainability, or deployment readiness. Farm-scale usefulness remains conditional on the quality of sensing systems, hardware-aware model design, environmental testing, lifecycle maintenance, locally relevant validation, and clear responsibility for system oversight. The main contribution is a framework that connects technical design choices with deployment conditions, implementation boundaries, and a research and practice agenda for evaluating low-power agricultural intelligence without converting prototype feasibility into unsupported operational claims.

Keywords Precision agriculture, Smart farming, Edge AI, TinyML, Embedded machine learning, Low-power intelligence

*Correspondence:

Luca Ferraro
luca.ferraro@unibo.it

¹ Department of Edge AI and TinyML for Agriculture, Faculty of Agricultural Sciences, University of Bologna, Bologna, Italy

² Department of Low-Power Intelligence and Deployment, Faculty of Agriculture, University of Turin, Turin, Italy

³ Department of Farm-Scale AI Applications, Faculty of Agriculture, Sapienza University of Rome, Rome, Italy

Introduction

Precision agriculture increasingly depends on connected systems that generate, transmit, process, and interpret data from crops, soils, machinery, weather conditions, water systems, and farm operations. Within these systems, analytical value is created not by isolated algorithms but through relationships among sensing devices, communication

infrastructure, data-management processes, decision interfaces, service providers, and farm users. Edge AI and TinyML therefore need to be understood as components within a broader agricultural data ecosystem rather than as independent substitutes for cloud computing, agronomic knowledge, or human decision-making [1].

Machine learning has been applied to agricultural classification, prediction, monitoring, and decision-support tasks using image, environmental, soil, livestock, crop, and water-related data. This breadth demonstrates the potential relevance of computational intelligence to farming, but it also creates substantial variation in data requirements, outcome definitions, validation practices, and practical risk. Reported predictive performance in an agricultural machine-learning study does not by itself show that the model can operate within constrained memory, energy, processing, sensor, or connectivity conditions, nor does it establish that embedded implementation will remain reliable under changing farm environments [2].

The affordability argument for Edge AI and TinyML is similarly more complex than reducing processor size or cloud usage. Digital agriculture operates within social and organizational settings shaped by skills, access to technical support, farm scale, market relationships, infrastructure, institutional capacity, and unequal control over data and digital services. A low-cost embedded device may remain inaccessible or unsustainable when installation, calibration, connectivity, maintenance, training, replacement, and interpretation requirements are included. Affordable precision agriculture must consequently be evaluated as a socio-technical proposition rather than a hardware-price category [3].

Edge AI and TinyML may assist selected agricultural functions, including monitoring and analytical support for irrigation, crop health, pest or disease conditions, and resource management. However, the agricultural significance of these functions depends on whether input data are sufficiently representative, outputs are interpretable, models remain valid under local conditions, and users can recognize uncertainty and system failure. This review therefore develops a framework-oriented synthesis that connects low-power intelligence and embedded farm systems with deployment constraints, farm-scale application conditions, implementation barriers, responsible-use boundaries, and research priorities rather than treating compact inference as evidence of operational effectiveness [4].

Edge AI and TinyML in Precision Agriculture

Edge AI refers broadly to the placement of artificial-intelligence processing nearer to the location where data are generated or used, including sensor nodes, embedded processors, local gateways, single-board computers, mobile platforms, or other edge devices. TinyML represents a more constrained subset of embedded machine learning in which models are designed to operate within limited processing, memory, storage, and energy resources, commonly at microcontroller scale. Its defining issue is therefore not simply small model size, but the combined feasibility of model execution, data preprocessing, runtime support, memory allocation, response time, and energy consumption within a constrained device environment [5].

TinyML also differs from conventional agricultural Internet of Things sensing. A sensor node that measures and transmits soil, weather, crop, water, or environmental variables is not necessarily an intelligent edge system. Near-sensor intelligence requires some local transformation, filtering, classification, anomaly detection, prediction, or decision logic before data are transferred to another computational layer. Frugal embedded intelligence consequently depends on coordinated design across sensors, processors, communication modules, power supplies, firmware, model architectures, and data pathways. Miniaturizing a model without evaluating these surrounding components may shift rather than resolve system constraints [6].

On-device inference can reduce the need to send every raw observation to a remote platform and may allow selected functions to continue during intermittent connectivity. It can also support lower communication latency and some forms of data minimization when only processed outputs are transmitted. These potential benefits are conditional, however, on device capability, model freshness, secure updating, failure detection, buffering, synchronization, and the continuing

validity of locally executed models. On-device processing does not remove privacy, cybersecurity, governance, or human-oversight requirements, and it does not establish that an automated output is agriculturally appropriate [7].

Edge AI in farming is best interpreted through a cloud-edge continuum rather than a strict choice between local and remote processing. Sensor nodes may perform basic preprocessing or event detection; gateways may aggregate information, coordinate communications, or execute larger models; and cloud services may support storage, model development, cross-site analysis, or controlled updating. The appropriate distribution of functions depends on data volume, response-time requirements, energy availability, connectivity conditions, privacy expectations, hardware resources, and the consequences of failure. Agricultural IoT architecture therefore provides the infrastructural context within which embedded intelligence must be evaluated [8].

Review Scope and Framework Orientation

The review covers Edge AI, TinyML, embedded machine learning, on-device inference, near-sensor intelligence, edge and fog computing where relevant, and their placement within agricultural IoT and cloud-edge systems. It includes literature concerning sensor nodes, gateways, constrained processors, low-power sensing, model optimization, farm-monitoring applications, environmental exposure, communications, maintenance, affordability, usability, validation, and responsible implementation. The agricultural scope includes field and controlled-environment applications while recognizing that sensing-to-decision systems consist of interconnected physical, computational, communication, and human-facing layers [9].

The framework does not treat every agricultural IoT system as Edge AI and does not classify every agricultural machine-learning model as TinyML. Conventional sensing without local analytical processing, cloud-only models without meaningful edge relevance, and wireless sensor networks used solely for data transport fall outside the central technical category, although they may provide important architectural or deployment evidence. The synthesis also distinguishes technical feasibility from implementation readiness by examining device, network, application, and support layers separately before considering how they interact within farm systems [10].

The evidence base includes reviews, surveys, conceptual analyses, methodological contributions, and relevant experimental or prototype studies that illuminate at least one framework dimension. It excludes non-peer-reviewed technical promotion, product documentation, vendor claims, and unsupported operational prescriptions. The approach is framework-oriented rather than meta-analytic: its purpose is to organize design requirements, deployment constraints, application conditions, implementation barriers, and research priorities across heterogeneous evidence. The resulting framework is an analytical structure for interpreting low-power agricultural intelligence, not an operational guide or a claim that reviewed systems are validated for generalized farm use [11].

Low-Power Intelligence and Embedded Farm Systems

Agricultural TinyML studies illustrate how task requirements interact with embedded resource limits. Image-based crop-disease classification, for example, requires coordination among image acquisition, preprocessing, model architecture, compression, memory use, runtime compatibility, and inference latency. A model that fits within constrained hardware may support local screening where appropriately validated, but device compatibility does not establish diagnostic reliability across crops, disease stages, lighting conditions, backgrounds, locations, or seasons. Local datasets and hardware execution evidence therefore need to be interpreted as bounded feasibility evidence rather than universal field validation [12].

Embedded intelligence also extends beyond image classification. Multivariate greenhouse applications may use compact models to interpret environmental measurements or support bounded control-related classifications on constrained devices. Such systems link sensor quality, temporal sampling, preprocessing, model complexity, and output logic within a

single embedded pipeline. Their usefulness depends on whether environmental variables are measured reliably, whether the training conditions represent the intended greenhouse context, and whether failure states are separated from ordinary predictions. Compact model design can support microcontroller compatibility, but it cannot independently establish safe or effective environmental control [13].

Edge architectures may distribute computation across several hardware levels. Sensor nodes can perform low-level filtering or inference, mobile or fixed gateways can aggregate observations and execute more demanding functions, and cloud resources can support model development, storage, or controlled synchronization. UAV-assisted or gateway-mediated designs demonstrate how agricultural data and inference may move between sensing, edge, and remote layers according to energy, bandwidth, timing, and task requirements. This distribution can reduce selected communication burdens, although it also introduces dependencies involving coverage, synchronization, gateway availability, model consistency, and maintenance [14].

Hardware-aware design requires the model and runtime to be evaluated on the intended embedded platform rather than considered only as abstract software. Microcontrollers and other embedded processors differ in available memory, numerical support, peripheral interfaces, scheduling behavior, and compatible operations. Quantization, pruning, compact architectures, and other compression methods may reduce storage or processing demand, but theoretical reductions do not always produce equivalent device-level gains. Agricultural-object classification on constrained processors therefore illustrates the need to measure the complete inference pipeline and preserve sufficient operational headroom for firmware, sensor buffers, logging, communication, and future updates [15].

Low-power intelligence also depends on how often sensing, computation, and communication occur. Adaptive sampling may reduce unnecessary acquisition or radio activity when environmental conditions are stable, but it can also miss short-duration or spatially variable events when its assumptions are poorly matched to farm conditions. Energy assessment should consequently include sensing, preprocessing, inference, communication, idle operation, battery behavior, and solar input where relevant rather than reporting model computation alone [16]. Evaluation methodologies for TinyML should likewise combine predictive performance with memory, latency, energy, hardware compatibility, data representativeness, and failure analysis so that compact execution is not misinterpreted as reliable agricultural performance [17]. **Table 1** summarises low-power intelligence and embedded farm system requirements for Edge AI and TinyML in precision agriculture.

Table 1. Low-Power Intelligence and Embedded Farm System Requirements for Edge AI and TinyML in Precision Agriculture: On-Device Inference, Near-Sensor Intelligence, Microcontrollers, Sensor Nodes, Gateways, Model Compression, Quantization, Pruning, Energy Use, Memory Footprint, Latency, Offline Inference, and Performance Boundaries

Technical component	Core design question	Required evidence or input	Low-power function	Agricultural use context	Failure risk	Evaluation ne
Edge AI	Which analytical functions should occur locally, at a gateway, or remotely?	Data urgency, bandwidth, privacy, computational load, connectivity pattern, and consequences of delay	Moves selected processing nearer to data generation	Monitoring, event detection, local alerts, and bounded decision assistance	Poor task allocation may increase energy use, latency, or system fragility	Compare local gateway, and cloud pathways under equivalent task conditions

TinyML	Can the complete inference task operate within highly constrained resources?	Model, runtime, memory, operator support, preprocessing, latency, and energy requirements	Enables embedded inference on constrained processors	Sensor classification, disease screening, adaptive sampling, and local anomaly detection	A fitted model may remain inaccurate, outdated, or environmentally fragile	Joint predictive memory, timing, and energy assessment of intended hardware
Embedded machine learning	Is the learned function safely integrated with sensing, firmware, communications, and possible actuation?	Sensor pipeline, preprocessing logic, firmware architecture, output handling, and fallback states	Executes learned functions within embedded systems	Greenhouse monitoring, irrigation observation, environmental nodes, and compact classifiers	Pipeline mismatch or firmware failure may invalidate outputs	End-to-end architecture and hardware-in-the-loop testing
On-device inference	Which functions should remain available without continuous remote access?	Connectivity profile, model freshness, storage capacity, response-time need, and update pathway	Processes data locally and can reduce raw-data transfer	Remote or intermittently connected agricultural settings	Stale models, silent errors, or unsynchronised records may persist offline	Test disconnection, continued operation, reconnection, and synchronization
Near-sensor intelligence	What processing should occur directly beside the sensor?	Signal characteristics, event frequency, noise, calibration state, and transmission cost	Filters, compresses, or interprets observations before communication	Soil, weather, crop, water, image, and environmental sensing	Sensor noise or drift may be converted into confident local outputs	Compare raw signals, processed signals, and downstream decisions
Microcontrollers	Does the workload fit the processor's actual memory, arithmetic, timing, and peripheral limits?	RAM, flash, clock behavior, supported numerical precision, firmware size, and sensor interfaces	Provides low-power sensing, control, and inference	Distributed sensor nodes and compact detection systems	Memory overflow, unsupported operations, or missed timing deadlines	Physical-device profiling with full firmware and runtime

Single-board computers	Is additional processing capacity justified by increased power and system complexity?	Processor demand, operating-system overhead, thermal behavior, storage, and power supply	Supports larger edge workloads and gateway functions	Local aggregation, machine vision, and multi-node coordination	Higher energy use, software complexity, and thermal instability	Compare lifecycle burden and performance with smaller alternatives
Sensor nodes	Can sensing, inference, storage, and communication share one energy budget?	Sensor duty cycle, sampling schedule, radio use, inference frequency, and power source	Combines acquisition and local processing in a distributed unit	Soil, environmental, crop, water, and livestock-related monitoring where supported	Radio or sensing demand may dominate inference savings	Component-level and whole-node energy accounting
Gateway devices	Which functions require aggregation, protocol translation, buffering, or larger local models?	Node density, communication protocols, storage needs, model-management requirements, and coverage	Concentrates processing away from highly constrained nodes	Farm-level coordination, local dashboards, buffering, and edge-cloud synchronization	Gateway failure may interrupt multiple downstream services	Availability, fault recovery, coverage, and synchronization testing
Model compression	Which reductions preserve agriculturally relevant model behavior?	Baseline performance, target-device limits, representative data, model size, and operator compatibility	Reduces storage and computational requirements	Embedded vision, sensor classification, and local prediction	Compression may reduce performance for rare classes or local conditions	Compare compressed and uncompressed models across contexts and subgroups
Quantization	What numerical precision can be reduced without unacceptable degradation?	Calibration data, runtime support, supported operators, and task sensitivity	Lowers memory use and arithmetic cost	Microcontroller-based image and sensor inference	Quantization error or unsupported operations may change outputs	Device-level predictive, timing memory, and energy testing
Pruning	Can parameters or operations be removed in a form usable by the target runtime?	Sparsity structure, runtime support, model redundancy, and	May reduce storage or computation	Efficient embedded neural inference	Theoretical sparsity may not create actual acceleration or energy savings	Measure realized rather than nominal device benefits

		and target workload				
Knowledge distillation where supported	Can task-relevant behavior be transferred to a smaller model?	Suitable teacher model, representative data, student capacity, and independent evaluation	Produces a compact model from a larger reference model	Image or multimodal agricultural applications where evidence exists	The smaller model may inherit bias or local failure patterns	Independent validation of the compact model under intended conditions
Efficient neural networks	Is efficiency incorporated into model architecture from the outset?	Device-aware architecture, operation count, memory traffic, runtime compatibility, and task data	Reduces computation and storage demand	Crop, disease, pest, weed, or object recognition where supported	Efficiency on one processor may not transfer to another	Cross-device and cross-context benchmarking
Low-power sensing	Can sensing frequency or resolution be reduced without losing important agricultural variation?	Temporal dynamics, spatial variability, event duration, and agronomic relevance	Reduces sensing and communication energy	Soil and environmental monitoring	Under-sampling may miss transient or local events	Compare adaptive and fixed sensing against reference observations
Battery operation	Can the complete node remain available under realistic duty cycles and environmental conditions?	Battery characteristics, current profile, temperature, communication bursts, and servicing interval	Enables untethered operation	Remote and distributed farm nodes	Capacity loss or unexpected workload may cause premature shutdown	Long-duration monitoring or complete-node power demands
Solar energy where supported	Can harvested energy cover variable sensing, inference, and communication loads?	Irradiance, shading, storage, panel orientation, seasonal variation, and load profile	Supports recharging of autonomous nodes	Remote environmental and field monitoring	Weather variability, dust, shading, or battery ageing may create energy deficits	Seasonal energy balance and degradation evaluation
Latency	Is end-to-end response time suitable for the intended agricultural function?	Sensor acquisition time, preprocessing, inference, communication, interface delay,	Supports timely local processing	Alerts, monitoring, event detection, and bounded control assistance	Fast model inference may conceal slow sensing or communication	Measure complete sensing to-output timing

		and task tolerance				
Memory footprint	Does the full system fit with sufficient operational and update headroom?	Model, runtime, firmware, buffers, logs, communication stack, and update requirements	Limits storage and dynamic-memory demand	All constrained embedded applications	Model-only reporting may conceal system-level memory failure	Measure peak RAM and complete flash use on the target device
Offline inference	Which functions remain useful and safe during connectivity loss?	Local inputs, model version, buffering, failure logic, audit records, and synchronization rules	Maintains selected analytical functions without remote access	Intermittently connected agricultural environments	Local outputs may become stale, unreviewed, or inconsistent with remote records	Outage, recovery, resynchronization, and model-currency testing
No performance-guarantee boundary	What evidence is required before reliability or effectiveness can be claimed?	Multi-context validation, uncertainty analysis, failure-mode evidence, and relevant farm outcomes	Does not create a technical function; constrains interpretation	All Edge AI and TinyML applications	Prototype feasibility may be presented as operational performance	Prespecified local temporal, external, and system-level evaluation

Deployment Constraints in Agricultural Environments

Agricultural edge systems are exposed to deployment conditions that differ substantially from controlled laboratory environments. Outdoor or semi-outdoor platforms may need to coordinate low-power sensing, embedded processing, battery operation, solar energy harvesting where applicable, data storage, and communication through variable networks. Power autonomy cannot be inferred from processor efficiency alone because sensing frequency, radio transmission, gateway interaction, environmental conditions, and battery behaviour contribute to the total energy demand. Field-oriented environmental collectors demonstrate the importance of assessing the complete platform, including power supply, connectivity, sensor stability, calibration, enclosure, and maintenance requirements [18].

Energy efficiency must also be considered alongside communication reliability and cybersecurity. Agricultural wireless sensor networks can involve distributed, unattended nodes that communicate through constrained links and may be difficult to inspect physically. Energy-aware routing or reduced transmission may extend node operation, but security procedures, authentication, message integrity, and recovery mechanisms also require computational and communication resources. Treating security as a later software addition may therefore undermine both energy planning and system dependability, particularly when compromised or malfunctioning nodes influence downstream monitoring or control functions [19].

Local processing changes the flow of agricultural data but does not remove privacy or cybersecurity obligations. Edge devices may reduce the transmission of raw observations, yet they can still retain sensitive farm records, geolocated production information, imagery, operational patterns, user credentials, model parameters, and derived inferences. Devices, gateways, update channels, communication links, and user interfaces can each create attack or disclosure

pathways. Secure updating, access controls, authentication, data minimization, logging, and clearly defined retention practices remain relevant even when most inference occurs on the farm [20].

Safeguards must extend across the complete architecture because threats and failures can move between sensing, communications, processing, and application layers. Device heterogeneity, unattended installations, variable network quality, and inconsistent update practices complicate cross-layer protection [21]. Physical exposure adds further uncertainty: dust can foul sensors or ventilation openings; humidity and rain can affect electronics and measurements; heat can alter batteries, processors, and sensor responses; and equipment movement or vibration can damage connections. Protective enclosures and robust installation may reduce these risks, but their effectiveness requires testing under the intended environmental conditions rather than assumption from indoor operation [18].

Sensor drift, model drift, calibration changes, device failure, and maintenance gaps can gradually reduce system validity without producing an obvious technical fault. Embedded drift-compensation methods demonstrate that changes in sensor response can sometimes be detected or corrected locally, but evidence from one sensor domain cannot be transferred automatically to agricultural conditions without local evaluation [22]. Responsible deployment must also reflect the availability of local infrastructure, replacement components, technical skills, training, repair services, and human oversight, especially where digital interventions are introduced into resource-constrained settings [23]. These interactions mean that low-power intelligence, farm sensors, intermittent connectivity, environmental exposure, maintenance capacity, and validation boundaries must be evaluated as one deployment pathway rather than as isolated engineering variables.

Figure 1 illustrates the deployment architecture and constraint pathway for Edge AI and TinyML in agricultural environments, linking sensors, embedded inference, gateways, connectivity, power, environmental stressors, validation, and human oversight.

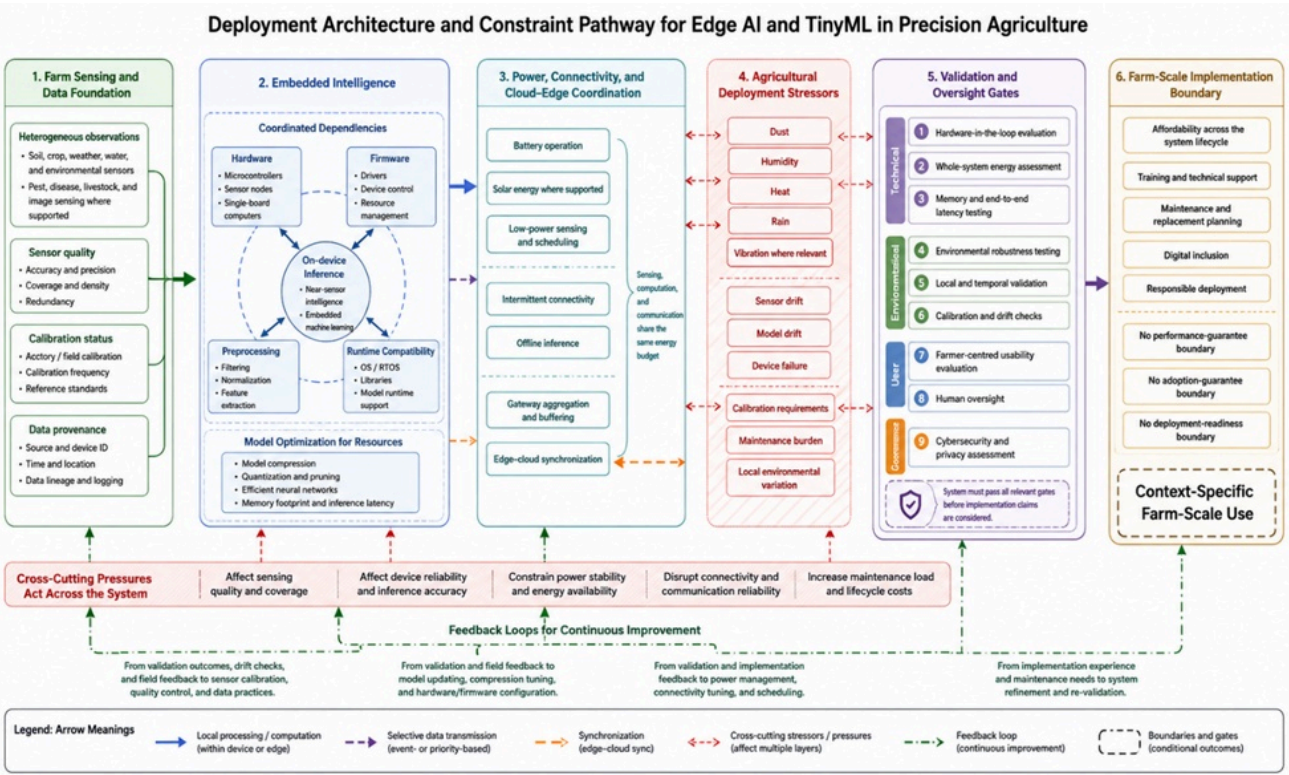


Figure 1. Deployment Architecture and Constraint Pathway for Edge AI and TinyML in Precision Agriculture

Alt-text description:

A conceptual architecture showing farm sensors feeding embedded Edge AI and TinyML devices, gateways, intermittent connectivity, power constraints, environmental stressors, calibration, validation, maintenance, human oversight, and deployment boundaries.

Farm-Scale Applications and Implementation Barriers

Farm-scale Edge AI applications can range from environmental monitoring to embedded decision assistance, but each application requires its own evidentiary boundary. A constrained crop-recommendation model may demonstrate that selected calculations can be executed locally without continuous cloud access, yet standalone inference does not establish that the recommendation is agronomically valid, economically beneficial, understandable to farmers, or appropriate under changing soils, climates, cultivars, and management practices [24]. Soil-moisture inference and related decision-support functions likewise depend on representative sensor placement, calibration, temporal sampling, and locally relevant outcome evidence rather than computational feasibility alone [14].

Crop monitoring, disease detection, pest recognition, and weed detection often depend on image or sensor data that vary with lighting, growth stage, background, species, symptom expression, occlusion, and field management. Reviews of weed-detection research show that dataset diversity, computational requirements, environmental variation, and cross-field generalization remain material barriers before embedded deployment can be inferred [25]. TinyML disease-classification studies provide useful evidence about model compression and constrained execution, but they do not establish autonomous diagnosis across farms or seasons [12]. Greenhouse models and UAV-assisted systems extend the application space to multivariate environmental monitoring and mobile edge architectures, although their outputs remain bounded by the conditions and tasks under which they were evaluated [13].

Irrigation monitoring illustrates the importance of combining edge processing with communication infrastructure and farmer-facing interfaces. Low-cost devices may collect or aggregate observations near irrigation equipment, while local or remote interfaces communicate system status to users. However, component affordability is only one part of the implementation problem. Coverage, installation, compatibility, data interpretation, interface design, troubleshooting, and continued technical support determine whether a monitoring architecture can remain usable after initial setup [26].

Livestock sensing, greenhouse management, mobile robotics, crop scouting, and farm decision support may also benefit from local processing where latency or connectivity constraints make continuous cloud access impractical. Nevertheless, tasks involving animals, moving equipment, environmental control, or treatment decisions introduce additional welfare, safety, and oversight concerns. Arable-farming evidence shows that IoT implementation is shaped by interoperability, connectivity, business models, maintenance, service availability, and the fit between technology and established work practices [27]. Edge intelligence cannot overcome these barriers merely by reducing model size.

Affordability should therefore include acquisition, installation, power supply, connectivity, calibration, training, software or model maintenance, repair, replacement, and access to competent support. Supply-side structures and service ecosystems can determine which farms obtain digital tools, who controls the associated data, and whether users remain dependent on external providers [28]. Farmer usability, digital inclusion, and local validation must be treated as evaluation requirements rather than assumed outcomes. A technically functional system may remain unused, misunderstood, or unsustainable, and neither short-term prototype operation nor local inference establishes adoption or farm-scale effectiveness. **Table 2** maps deployment constraints, farm-scale applications, and implementation barriers for Edge AI and TinyML in agricultural environments.

Table 2. Deployment Constraints, Farm-Scale Applications, and Implementation Barriers for Edge AI and TinyML in Agricultural Environments: Connectivity, Power, Battery Life, Memory, Processing, Sensor Quality, Calibration, Dust, Humidity, Heat, Rain, Device Failure, Maintenance, Affordability, Usability, Local Validation, and Deployment Boundaries

Application, constraint, or barrier	Core farm-scale question	Required data or system condition	Edge or TinyML function	Potential use	Implementation or failure risk	Evaluation or support need
Connectivity limitation	Can essential functions continue when remote communication is weak or unavailable?	Characterized coverage, outage duration, buffering capacity, and recovery logic	Local inference, buffering, and selective synchronization	Continued bounded monitoring or alerts	Missing, delayed, duplicated, or stale data	Disconnection, reconnection, and synchronization testing
Intermittent networks	Does the system recover correctly after repeated network interruption?	Reliable timestamps, local storage, conflict handling, and data-integrity checks	Store-and-forward operation	Remote field monitoring	Inconsistent records or delayed updates	Controlled outage and recovery evaluation
Power limitation	Which sensing, inference, communication, or idle processes dominate energy use?	Whole-node current profiles and realistic duty cycles	Adaptive sensing and low-power local processing	Untethered environmental monitoring	Premature shutdown or reduced sampling	Complete-system energy measurement
Battery life	Can the system remain available across temperature, ageing, and communication variation?	Battery characteristics, environmental conditions, and service interval	Supports autonomous operation	Distributed farm nodes	Capacity loss and unexpected maintenance	Long-duration field-oriented power assessment
Solar variability where supported	Can harvested energy cover seasonal and event-driven loads?	Irradiance, shading, storage, orientation, and load profile	Recharging and adaptive scheduling	Remote sensing installations	Weather, dust, shading, or ageing may create energy deficits	Seasonal energy-balance evaluation
Memory limitation	Does the complete software stack fit with sufficient operating headroom?	Model, firmware, runtime, buffers, logs, and update requirements	Compact model execution	Microcontroller-based applications	Overflow, crashes, or inability to update	Peak-RAM and full-flash assessment

Heat	Can devices operate under direct sun and enclosure heating?	Thermal design and environmental monitoring	Temperature-aware scheduling where appropriate	Outdoor nodes and gateways	Thermal throttling, battery degradation, and sensor error	Worst-case temperature evaluation
Rain	Can installations resist precipitation, splash, and water ingress?	Enclosure integrity, cable protection, drainage, and mounting	No direct intelligence function	Outdoor monitoring	Service interruption or permanent damage	Rain and ingress testing
Device failure	Can failure be detected and handled without misleading downstream users?	Health monitoring, watchdogs, diagnostics, and safe fallback states	Fault detection where supported	Distributed sensor and gateway systems	Silent data loss or erroneous outputs	Fault-injection and recovery evaluation
Soil-moisture monitoring	Are measurements representative of local soil variability?	Calibrated sensors, appropriate placement, and temporal sampling	Local filtering, prediction, or anomaly detection	Irrigation-status awareness	Spatial bias, drift, and poor calibration	Site-specific validation
Irrigation monitoring	Can equipment and water status be observed across the intended area?	Reliable sensors, communication, and interpretable interfaces	Edge aggregation and local alerts	Pivot or distributed irrigation observation	Coverage gaps, interface burden, and maintenance	Operational and usability evaluation
Crop monitoring	Does the sensing method capture meaningful crop variation?	Representative local data across growth stages and environments	Embedded classification or event detection	Crop-status screening	Lighting, occlusion, and seasonal shift	Multi-site and temporal validation
Disease detection	Can symptoms be distinguished from visually similar conditions?	Locally relevant images and independent reference diagnosis	Embedded image classification	Preliminary screening	Misclassification, class imbalance, and transfer failure	Expert-confirmed local validation
Pest detection where supported	Are targets observable at the required scale and frequency?	Representative labels and reliable sensing conditions	Local event detection	Scouting support	Small targets, false alarms, and changing backgrounds	Field testing with confirmed labels

Weed detection where supported	Can weeds be separated from crops, residue, and soil across fields?	Diverse imagery and species labels	Efficient visual detection	Scouting or robotic perception support	Domain shift, occlusion, and computational demand	Cross-field and device-level testing
Livestock monitoring where supported	Are behavioural or environmental signals valid for the target animals and setting?	Reliable wearable, image, acoustic, or environmental data	Local classification and anomaly alerts	Behavioural or environmental screening	False alerts, device burden, and welfare concerns	Longitudinal and welfare-sensitive validation
Greenhouse monitoring where supported	Can compact models interpret multivariate climate data reliably?	Calibrated sensors, representative conditions, and safe output logic	Local multi-label inference	Microclimate monitoring or bounded control support	Sensor faults and narrow training conditions	Hardware-in-the-loop and fail-safe evaluation
Robotics where supported	Is edge perception timely and dependable for a bounded robotic function?	Synchronized sensing, localization, and independent safety controls	Low-latency local perception	Navigation, scouting, or target recognition	Dust, vibration, computational load, and human-machine hazards	System-level safety and field testing
Farm decision support	Are outputs understandable, relevant, reviewable, and accompanied by uncertainty?	Contextual data, validated models, and human-readable interfaces	Local recommendations or alerts	Timely assistance	Missing context, automation bias, and stale models	Agronomic and farmer-centred evaluation
Affordability	Is the complete lifecycle affordable for the intended farm context?	Acquisition, installation, energy, communication, maintenance, and support costs	May reduce selected computing or communication requirements	Wider access to distributed sensing	Hidden recurring costs and unequal service access	Context-specific total-cost assessment
Farmer usability	Can users interpret normal outputs, uncertainty, and failure states?	Appropriate interface, language, workflow fit, and support	Local alerts and simplified interaction	Faster access to information	Alert fatigue, confusion, and low trust	Participatory and longitudinal usability testing

Training and support	Who installs, updates, troubleshoots, and explains the system?	Training resources, service channels, and role allocation	Sustains embedded-system use	Routine operation	Skills gaps and provider dependence	Training and support evaluation
Maintenance	Can devices be cleaned, recalibrated, powered, and repaired locally?	Replaceable components, diagnostics, documentation, and supply access	Preserves sensing and inference quality	Long-term operation	Downtime, abandonment, or escalating lifecycle cost	Maintainability trials and service records
Digital inclusion	Are benefits and burdens distributed fairly across farm types and users?	Accessibility, language fit, infrastructure, and institutional support	Offline functions may broaden access in some contexts	Use under constrained connectivity	Exclusion through cost, skills, ownership, or service design	Distributional and accessibility assessment
Local validation	Does the system perform with local crops, soils, climates, and practices?	Representative local data and independent outcomes	Calibration or controlled adaptation	Context-sensitive inference	Failure of transfer from external datasets	External, temporal, and local validation
No adoption-guarantee boundary	Has sustained use and organizational fit been demonstrated?	Longitudinal user evidence	No technical function; constrains interpretation	All applications	Pilot use may be mistaken for acceptance	Continued-use and discontinuation studies
No farm-scale effectiveness guarantee boundary	Have agronomic, economic, or operational outcomes been evaluated appropriately?	Comparative field evidence and relevant outcome measures	No technical function; constrains interpretation	All applications	Technical metrics may be substituted for farm outcomes	Appropriately designed field evaluation
No deployment-readiness boundary	Has the complete system been assessed across sites, seasons, users, and failures?	Integrated technical, environmental, user, and governance evidence	No technical function; constrains interpretation	All systems	Prototype evidence may be generalized prematurely	Stage-gated multi-context assessment

Framework-Oriented Synthesis

The reviewed evidence indicates that Edge AI and TinyML should be assessed as interacting technical and socio-technical arrangements rather than as isolated compact models. Agriculture 4.0 research documents potential benefits

from connected sensing, automation, and analytics while also identifying financial, technical, organizational, interoperability, and skills barriers [29]. A framework for affordable precision agriculture must therefore connect model efficiency with hardware capacity, sensor quality, network conditions, service infrastructure, user capabilities, and the intended agricultural task.

No single cloud-edge configuration is universally appropriate. Local inference may reduce selected communication and latency requirements, gateways may support aggregation and coordination, and cloud resources may remain useful for storage, training, updating, and cross-site analysis. The distribution of these functions should follow local constraints and responsibility arrangements rather than an assumed linear transition from cloud systems to autonomous edge devices. Agriculture 4.0 pathways are diverse and context-dependent, and responsible implementation requires attention to who benefits, who bears risk, and which alternatives are excluded by a particular technical design [30].

The meaning of affordability also extends beyond computational efficiency. A system that uses low-cost or low-power hardware may still require substantial installation, calibration, connectivity, training, maintenance, replacement, and support. Its interface may not align with farm workflows, and its outputs may not be trusted or actionable. Digital agriculture must work for people and production while remaining attentive to wider environmental and social implications; technical capability alone cannot establish those outcomes [31].

Responsibility is distributed across model developers, device designers, data providers, system integrators, service organizations, advisers, farm managers, and users. Human oversight must therefore include clear authority to question, suspend, recalibrate, update, or reject system outputs. Trust should be grounded in demonstrated trustworthiness, transparent limitations, traceable data and model histories, understandable failure communication, and meaningful routes for contesting decisions rather than in generalized claims about artificial intelligence [32].

The synthesis consequently treats compact inference as one enabling layer within a larger evidence chain. TinyML evaluation requires multidimensional testing of prediction, hardware, energy, and latency limits [17]. Outdoor agricultural use adds power, connectivity, sensor, environmental, and maintenance dependencies [18]. Farm-scale implementation further requires usable interfaces, support capacity, local fit, and attention to unequal access [31]. Governance and responsibility remain necessary across the complete lifecycle [32]. **Table 3** presents a framework-oriented synthesis for Edge AI and TinyML in affordable precision agriculture.

Table 3. Framework-Oriented Synthesis for Edge AI and TinyML in Affordable Precision Agriculture: Low-Power Intelligence, Embedded Farm Systems, Cloud-Edge Trade-Offs, Model Compression, Energy Constraints, Connectivity, Environmental Robustness, Sensor Drift, Farm-Scale Applications, Affordability, Usability, Privacy, Cybersecurity, Local Validation, Responsible Deployment, and Research Priorities

Synthesis theme	Evidence pattern	Technical interpretation	Deployment interpretation	Implementation risk	Research or practice gap	Framework boundary
Low-power intelligence	Selected ML functions can be executed on constrained devices	Model architecture, runtime, memory, latency, and energy must be co-designed	Local processing is valuable only where task and context justify it	Compact execution may be mistaken for reliable intelligence	Comparable task–device evaluations	Low-power execution does not establish decision quality
Embedded farm systems	Sensing, processing, communications,	The complete pipeline	Failure in one component can	Model-only testing conceals	End-to-end and fault-oriented evaluation	The model is only one

	interfaces, and power are interdependent	determines feasibility	undermine the service	system dependencies		system component
Cloud-edge trade-off	Local, gateway, and cloud resources offer different advantages	Processing location should follow data, timing, privacy, and compute needs	Hybrid systems require synchronization and fallback logic	Over-centralization or over-localization may create fragility	Comparative cloud, gateway, and device studies	Edge and cloud are complementary
Model compression and accuracy	Compression can enable device fit while altering model behaviour	Quantization, pruning, distillation, and efficient architectures need device testing	Smaller models may remain locally invalid	Aggregate accuracy may hide loss for rare or local conditions	Compression studies using representative agricultural data	Compression does not preserve accuracy automatically
Energy constraints	Sensing, communication, inference, and idle operation share the energy budget	Whole-node profiling is required	Workload, temperature, battery age, and network state affect autonomy	Inference-only energy figures may exaggerate savings	Standard whole-system energy protocols	Low-power models do not prove long battery life
Latency constraints	Local processing may reduce some response delays	End-to-end timing includes sensing, preprocessing, inference, and communication	Required response time varies by application	Fast model execution may conceal slow system response	Application-specific end-to-end timing evidence	Low inference latency does not guarantee timely operation
Connectivity and offline inference	Local functions can continue during some network interruptions	Buffering, model freshness, and recovery logic are required	Intermittent operation needs graceful degradation	Offline outputs may become stale or unsynchronized	Outage and recovery evaluations	Offline inference does not remove network dependence
Environmental robustness	Agricultural devices face dust, moisture, heat, rain, and physical exposure	Enclosures, thermal design, and sensor protection affect performance	Long-term exposure can degrade availability and measurement quality	Indoor results may be generalized to field conditions	Repeatable environmental stress and ageing protocols	Robustness is limited to tested conditions
Sensor drift	Sensor response may change with ageing and exposure	Monitoring, recalibration, or compensation may be required	Drift increases maintenance and reference-measurement needs	Silent input bias can degrade predictions	Agricultural longitudinal drift datasets	Initial calibration does not prove stability

Model drift	Seasonal, environmental, and management changes may alter relationships	Model monitoring and controlled updating are needed	Labels may not be available for rapid drift detection	Outdated models may continue producing confident outputs	Multi-season and temporal validation	Offline models are not necessarily current
Farm-scale application diversity	Tasks range from scalar sensing to image analysis and decision support	Hardware and model requirements are task-specific	Risk and maintenance differ by application	Results from one task may be generalized to others	Application-specific benchmarks and field protocols	No universal TinyML configuration exists
Affordability	Lower computation or transmission may reduce selected expenses	Total architecture and lifecycle costs determine affordability	Installation, replacement, support, and connectivity remain material	Component price may conceal recurring costs	Total-cost evidence across farm contexts	TinyML does not guarantee affordability
Accessibility and digital inclusion	Local processing may help in constrained-connectivity settings	Accessible interfaces and support remain necessary	Benefits depend on skills, language, ownership, and service access	Technology may deepen existing inequalities	Inclusion studies across farm types and users	Offline operation does not resolve the digital divide
Maintenance and usability	Sustained value depends on understandable and serviceable systems	Diagnostics and interface design are core requirements	Local repair and calibration capacity affect availability	Functional systems may be abandoned or misused	Participatory usability and maintainability studies	Deployability is not sustained use
Privacy	Edge processing may reduce raw-data transmission	Data minimization, storage, access, and inference sensitivity still matter	Local devices may contain valuable or sensitive data	"Local data" claims may create false confidence	Data-flow and re-identification assessments	Edge processing does not guarantee privacy
Cybersecurity	Distributed devices and gateways create multiple attack surfaces	Secure boot, authentication, updates, and logging are required	Unattended devices may be difficult to inspect or patch	Compromise can affect data and downstream decisions	Agricultural edge threat models and recovery tests	Security cannot be permanently guaranteed
Local validation	Agricultural heterogeneity	Models require locally representative	Crop, soil, climate, and management	Benchmark performance may be	Multi-site, multi-season, and	Performance elsewhere cannot be

	limits transferability	and temporal evidence	variation affect usefulness	mistaken for generalization	external validation	assumed locally
Responsible deployment	Technical choices distribute authority, benefit, and risk	Oversight and traceability must be built into system design	Responsibility spans developers, deployers, advisers, and users	Accountability can become fragmented	Operational responsibility mapping and participatory assessment	Framework use does not certify responsibility
Research priorities	Evidence is fragmented across model, hardware, network, farm, and user studies	Integrated evaluation methods are needed	Field evidence should accompany efficiency claims	Selective reporting prevents comparison	Common multidimensional protocols	Proposed priorities are not validated solutions

Figure 2 presents the framework-oriented synthesis for Edge AI and TinyML in affordable precision agriculture, integrating low-power intelligence, embedded farm systems, deployment constraints, farm-scale applications, implementation barriers, responsible deployment, and research priorities.

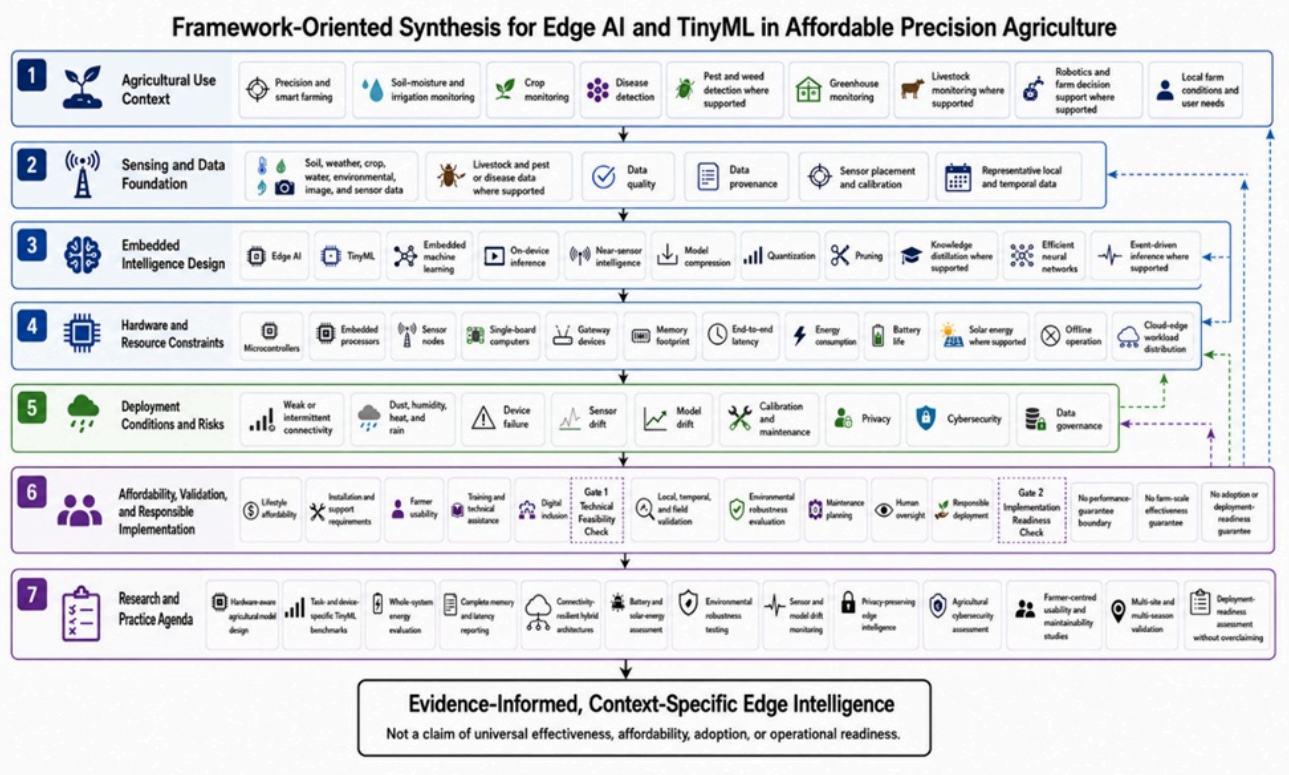


Figure 2. Framework-Oriented Synthesis for Edge AI and TinyML in Affordable Precision Agriculture

Alt-text description:

A conceptual synthesis framework linking Edge AI and TinyML design choices to embedded hardware, farm sensors, energy and connectivity constraints, environmental robustness, applications, affordability, usability, responsible deployment, validation, and research priorities.

Research and Practice Agenda

A first priority is hardware-aware agricultural model design. Models should be developed with the intended processor, runtime, sensor pipeline, preprocessing method, memory headroom, and update pathway in view rather than compressed only after training [15]. Agricultural TinyML benchmarks should report task definitions, dataset provenance, local relevance, operator compatibility, memory, latency, energy, and failure conditions. Multidimensional methodologies are needed because predictive accuracy alone cannot expose practical device and deployment limitations [17]. Energy protocols should measure sensing, inference, communication, and idle behaviour across realistic duty cycles [16], while memory and latency reporting should cover the complete firmware and sensor-to-output pathway rather than the model file alone [12].

A second priority concerns connectivity and environmental resilience. Hybrid architectures should be evaluated under weak, intermittent, and unavailable network conditions, including buffering, continued local operation, reconnection, synchronization, and model-version control [14]. Solar-powered and battery-operated sensing requires seasonal energy-balance evidence, degradation monitoring, and explicit maintenance assumptions [18]. Environmental testing should examine dust, humidity, heat, rain, contamination, enclosure ageing, and device failure under conditions relevant to the intended agricultural setting. Sensor and model drift monitoring should be incorporated into lifecycle design, but compensation methods must be validated with agricultural sensors and local reference data rather than transferred uncritically from other sensing domains [22].

A third priority is realistic evaluation of affordability, usability, inclusion, and maintenance. Low-cost deployment research should account for installation, communication, calibration, interfaces, support, replacement, and downtime rather than component price alone [26]. Farmer-centred usability studies should examine whether users understand outputs, uncertainty, alerts, connectivity loss, calibration needs, and failure states [31]. Maintenance planning should identify who cleans, repairs, recalibrates, powers, updates, and replaces devices and whether the required parts and skills are locally available [27]. Inclusion assessment should examine whether service models, language, ownership arrangements, infrastructure, or technical dependence exclude particular farms or users [23].

A fourth priority is integrated cybersecurity, privacy, responsibility, and farm-scale validation. Agricultural edge systems require threat modelling, authenticated updating, access control, incident recovery, and clear responsibility for compromised or malfunctioning devices [20]. Privacy-preserving edge intelligence should examine retained data, transmitted outputs, inference sensitivity, access rights, and the possibility that processed information remains commercially or personally revealing [21]. Responsible-deployment assessment should allocate oversight and accountability across developers, integrators, advisers, service providers, and farm users [32]. Farm-scale validation should combine technical, environmental, usability, maintenance, and relevant agricultural outcomes across appropriate sites and periods, while remaining explicit that even strong contextual evidence does not prove universal effectiveness or deployment readiness [29].

Conclusion

This framework-oriented review positions Edge AI and TinyML as potentially useful components of affordable precision agriculture when they are assessed within complete embedded farm systems rather than through miniaturized inference alone. The synthesis connects low-power intelligence, embedded hardware, sensor nodes, gateways, model compression, memory, latency, energy, connectivity, environmental exposure, calibration, maintenance, applications, usability, affordability, inclusion, cybersecurity, privacy, human oversight, and local validation. Edge AI and TinyML may reduce reliance on continuous cloud connectivity and support selected local analytical functions, but they do not independently guarantee accuracy, robustness, sustainability, farmer adoption, operational safety, or affordability. Their farm-scale relevance depends on hardware-aware design, whole-system energy and timing evaluation, representative sensor data, environmental robustness testing, maintainability, understandable interfaces, context-specific validation, and clearly allocated responsibility. The proposed framework therefore supports cautious comparison, evidence planning, and

research prioritization while preserving firm boundaries between embedded feasibility, deployment readiness, and demonstrated agricultural effectiveness.

Acknowledgements

None

Conflict of interest

None

Financial support

None

Ethics statement

None

Received: 26 Sep 2025

Revised: 03 Nov 2025

Accepted: 28 Nov 2025

Published online: 25 May 2026

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

References

Wolfer S, Ge L, Verdouw C, Bogaardt MJ. Big data in smart farming-a review. *Agric Syst.* 2017;153:69-80.
<https://doi.org/10.1016/j.agsy.2017.01.023>.

Liakos KG, Busato P, Moshou D, Pearson S, Bochtis D. Machine learning in agriculture: A review. *Sensors (Basel).* 2018;18(8):2674.
<https://doi.org/10.3390/s18082674>.

Klerkx L, Jakku E, Labarthe P. A review of social science on digital agriculture, smart farming and Agriculture 4.0: New contributions and a future research agenda. *NJAS Wageningen J Life Sci.* 2019;90-91:100315.
<https://doi.org/10.1016/j.njas.2019.100315>.

Talaviya T, Shah D, Patel N, Yagnik H, Shah M. Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides. *Artif Intell Agric.* 2020;4:58-73.
<https://doi.org/10.1016/j.aiia.2020.04.002>.

Ray PP. A review on TinyML: State-of-the-art and prospects. *J King Saud Univ Comput Inf Sci*. 2022;34(4):1595-623.
<https://doi.org/10.1016/j.jksuci.2021.11.019>.

Sanchez-Iborra R, Skarmeta AF. TinyML-enabled frugal smart objects: Challenges and opportunities. *IEEE Circuits Syst Mag*. 2020;20(3):4-18.
<https://doi.org/10.1109/MCAS.2020.3005467>.

Heydari S, Mahmoud QH. Tiny machine learning and on-device inference: A survey of applications, challenges, and future directions. *Sensors (Basel)*. 2025;25(10):3191.
<https://doi.org/10.3390/s25103191>.

Elijah O, Rahman TA, Orikumhi I, Leow CY, Hindia MHDN. An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges. *IEEE Internet Things J*. 2018;5(5):3758-73.
<https://doi.org/10.1109/JIOT.2018.2844296>.

Tzounis A, Katsoulas N, Bartzanas T, Kittas C. Internet of Things in agriculture, recent advances and future challenges. *Biosyst Eng*. 2017;164:31-48.
<https://doi.org/10.1016/j.biosystemseng.2017.09.007>.

Ayaz M, Ammad-Uddin M, Sharif Z, Mansour A, Aggoune EHM. Internet-of-Things (IoT)-based smart agriculture: Toward making the fields talk. *IEEE Access*. 2019;7:129551-83.
<https://doi.org/10.1109/ACCESS.2019.2932609>.

Farooq MS, Riaz S, Abid A, Abid K, Naeem MA. A survey on the role of IoT in agriculture for the implementation of smart farming. *IEEE Access*. 2019;7:156237-71.
<https://doi.org/10.1109/ACCESS.2019.2949703>.

Gookyi DAN, Wulnye FA, Arthur EAE, Ahiadormey RK, Agyemang JO, Agyekum KOBO, et al. TinyML for smart agriculture: Comparative analysis of TinyML platforms and practical deployment for maize leaf disease identification. *Smart Agric Technol*. 2024;8:100490.
<https://doi.org/10.1016/j.atech.2024.100490>.

Ihoume I, Tadili R, Arbaoui N, Benchrif M, Idrissi A, Daoudi M. Developing a multi-label TinyML machine learning model for an active and optimized greenhouse microclimate control from multivariate sensed data. *Artif Intell Agric*. 2022;6:129-37.
<https://doi.org/10.1016/j.aiaa.2022.08.003>.

Hayajneh AM, Aldalameh SA, Alasali F, Al-Obiedollah H, Zaidi SAR, McLernon D. Tiny machine learning on the edge: A framework for transfer learning empowered unmanned aerial vehicle assisted smart farming. *IET Smart Cities*. 2024;6(1):10-26.
<https://doi.org/10.1049/smc2.12072>.

Donskoy D, Gvindjiliya V, Ivliev E. TinyML classification for agriculture objects with ESP32. *Digital*. 2025;5(4):48.
<https://doi.org/10.3390/digital5040048>.

Hinostroza A, Tarrillo J, Nuñez M. Smart energy management in agricultural wireless sensor nodes using TinyML-based adaptive sampling. *Sensors (Basel)*. 2026;26(7):2014.
<https://doi.org/10.3390/s26072014>.

Delnevo G, Mirri S, Prandi C, Manzoni P. An evaluation methodology to determine the actual limitations of a TinyML-based solution. *Internet Things*. 2023;22:100729.
<https://doi.org/10.1016/j.iot.2023.100729>.

Trînc EC, Niță V, Stolojescu-Crisan C, Ancuți C, Mihai RM, Sultănoiu CP. Cost-effective TinyML-ready design and field deployment of a solar-powered environmental monitoring data collector using LTE-M communication. *Appl Sci (Basel)*. 2026;16(7):3237.
<https://doi.org/10.3390/app16073237>.

Haseeb K, Ud Din I, Almogren A, Islam N. An energy efficient and secure IoT-based WSN framework: An application to smart agriculture. *Sensors (Basel)*. 2020;20(7):2081.
<https://doi.org/10.3390/s20072081>.

Gupta M, Abdelsalam M, Khorsandroo S, Mittal S. Security and privacy in smart farming: Challenges and opportunities. *IEEE Access*. 2020;8:34564-84.
<https://doi.org/10.1109/ACCESS.2020.2975142>.

Yang X, Shu L, Chen J, Ferrag MA, Wu J, Nurellari E, et al. A survey on smart agriculture: Development modes, technologies, and security and privacy challenges. *IEEE/CAA J Autom Sin*. 2021;8(2):273-302.
<https://doi.org/10.1109/JAS.2020.1003536>.

Krayden A, Avraham M, Ashkar H, Blank T, Stolyarova S, Nemirovsky Y. TinyML-based real-time drift compensation for gas sensors using spectral-temporal neural networks. *Chemosensors*. 2025;13(7):223.
<https://doi.org/10.3390/chemosensors13070223>.

McC Campbell M, Schumann C, Klerkx L. Good intentions in complex realities: Challenges for designing responsibly in digital agriculture in low-income countries. *Sociol Ruralis*. 2022;62(2):279-304.
<https://doi.org/10.1111/soru.12359>.

Baishya M, Dutta L. Tiny ML based crop recommendation system for precision agriculture 5.0. *Smart Agric Technol*. 2025;12:101247.
<https://doi.org/10.1016/j.atech.2025.101247>.

Pai DG, Kamath R, Balachandra M. Deep learning techniques for weed detection in agricultural environments: A comprehensive review. *IEEE Access*. 2024;12:113193-214.
<https://doi.org/10.1109/ACCESS.2024.3418454>.

Mateos Matilla D, Lozano Murciego Á, Jiménez-Bravo DM, Sales Mendes A, Leithardt VRQ. Low-cost edge computing devices and novel user interfaces for monitoring pivot irrigation systems based on Internet of Things and LoRaWAN technologies. *Biosyst Eng*. 2022;223:14-29.
<https://doi.org/10.1016/j.biosystemseng.2021.07.010>.

Villa-Henriksen A, Edwards GTC, Pesonen LA, Green O, Sørensen CAG. Internet of Things in arable farming: Implementation, applications, challenges and potential. *Biosyst Eng*. 2020;191:60-84.
<https://doi.org/10.1016/j.biosystemseng.2019.12.013>.

Birner R, Daum T, Pray C. Who drives the digital revolution in agriculture? A review of supply-side trends, players and challenges. *Appl Econ Perspect Policy*. 2021;43(4):1260-85.
<https://doi.org/10.1002/aep.13145>.

da Silveira F, Lermen FH, Amaral FG. An overview of Agriculture 4.0 development: Systematic review of descriptions, technologies, barriers, advantages, and disadvantages. *Comput Electron Agric*. 2021;189:106405.
<https://doi.org/10.1016/j.compag.2021.106405>.

Klerkx L, Rose DC. Dealing with the game-changing technologies of Agriculture 4.0: How do we manage diversity and responsibility in food system transition pathways? *Glob Food Sec*. 2020;24:100347.
<https://doi.org/10.1016/j.gfs.2019.100347>.

Rose DC, Wheeler R, Winter M, Lobley M, Chivers CA. Agriculture 4.0: Making it work for people, production, and the planet. *Land Use Policy*. 2021;100:104933.
<https://doi.org/10.1016/j.landusepol.2020.104933>.

Alexander CS, Yarborough M, Smith A. Who is responsible for 'responsible AI'? Navigating challenges to build trust in AI agriculture and food system technology. *Precis Agric*. 2024;25(1):146-85.

<https://doi.org/10.1007/s11119-023-10063-3>.