

MA-FedSense: Modality-Adaptive Federated Learning for Crop-Stress Detection Across Heterogeneous Farm Networks

Sahil Yousuf

Department of Physics, National Institute of Technology Srinagar, Srinagar 190006, Jammu and Kashmir, India

Abstract—Farms rarely operate with identical sensing infrastructure: some collect aerial RGB or near-infrared imagery, others rely on ground measurements, and only a subset may support additional stress-sensitive channels. This study presents MA-FedSense, a modality-adaptive federated framework designed for collaborative crop-stress modelling when clients possess different sensor subsets. The method separates modality-specific encoders from a shared fusion module and introduces MA-FedAvg, which aggregates each encoder only across clients that can train that modality while weighting the shared fusion update by local sample size and modality completeness. The contribution is implemented as a reproducible PyTorch/Flower-ready software package and evaluated as a completed methods and software-validation study. In the supplied environment, all 29 automated unit/integration tests passed in 4.29 s. A reference aggregation smoke test recovered the expected modality-specific weighted updates (soil = 5.0; gas =

10.0) and the completeness-weighted shared update (6.4). The default publication model contains 1,263,138 parameters, of which 743,778 remain trainable when only RGB and soil branches are available. The data interface enforces causal 288-step ground windows, plot-level partition integrity, native-modality checks, and immutable run archiving; the evaluation stack implements Macro-F1, AUROC, corrected ECE, threshold-round analysis, paired effects, and deterministic bootstrap confidence intervals. These results establish software correctness and reproducibility, not field predictive superiority. Real multi-farm crop-stress performance remains an external validation task.

Index Terms—federated learning; multimodal learning; precision agriculture; crop stress; sensor heterogeneity; software validation; reproducibility; modality-aware aggregation

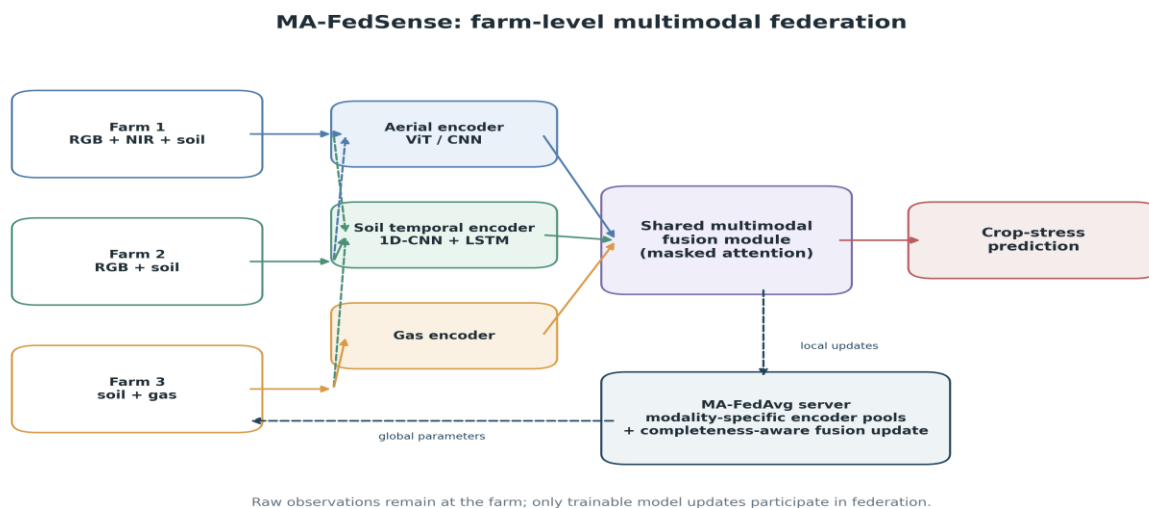


Figure 1. Overall MA-FedSense architecture. Farms retain raw observations, train only the modality branches they can support, and exchange trainable model updates with the aggregation server.

I. INTRODUCTION

Training a common AI model across different farms is not straightforward. Farm data are usually collected locally and may contain information that owners do not want to share with outside parties. At the same time, farms do not use identical sensing equipment. One farm may have RGB or multispectral drone images, another may depend mainly on soil-moisture and temperature sensors, while a better-equipped farm may also record electrical conductivity, pH, or plant-stress signals. This means that participating farms differ not only in the amount and distribution of their data, but also in the type of information available to them. A learning method designed around identical inputs at every site is therefore difficult to apply in practice.

Federated learning offers one way to train a shared model without directly pooling raw farm data. In Federated Averaging (FedAvg), each client trains locally and sends model updates to a coordinating server, where those updates are averaged [1]. McMahan et al. explicitly evaluated non-IID and unbalanced client data, so the limitation considered here is not simply that farms have different class distributions [1]. The harder case is structural: a farm without aerial imagery cannot train an image encoder, while a farm without a gas sensor cannot provide a meaningful update for a gas branch.

To address this problem, we propose **MA-FedSense**, a modality-adaptive federated framework for multimodal crop-stress detection. The model is divided into modality-specific encoders and a shared fusion component. Aerial images are handled by an image branch, while soil and other ground measurements are handled by temporal sensor branches. During federation, a modality encoder is averaged only across farms that actually possess that modality. The shared fusion module is then updated using information from all participating clients, with the proposed method accounting for both local sample size and modality availability.

The completed study makes four contributions:

- A farm-level multimodal federated formulation in which each farm is one client and clients may possess different sensing subsets.
- MA-FedAvg, an executable aggregation rule that averages each modality encoder only

across eligible clients and separately aggregates the shared fusion module.

- A reproducible data and protocol layer with plot-level Dirichlet partitions, deterministic M0-M3 modality masks, causal 72-hour ground windows, native-modality auditing, and immutable run identities.
- A completed software-validation package covering aggregation mathematics, missing-modality model behavior, field-data integrity, FedProx regularization, evaluation metrics, result archiving, and experiment-runner safeguards.

The scope of this paper is therefore precise: it establishes the method, implementation, and reproducibility of modality-adaptive federated learning for heterogeneous agricultural sensors. It does not claim crop-stress predictive superiority without a completed real-data federation.

II. RELATED WORK

McMahan et al. introduced FedAvg as a practical way to learn a shared model while keeping training data on local devices [1]. Each client performs local training and returns model updates instead of transferring its raw data. Their experiments already included unbalanced and non-IID data, so the weakness of FedAvg in the present study is not merely that farms have different data distributions. The harder problem is that farms may not even have the same input modalities. A farm with drone imagery can update an image encoder, whereas a farm without images cannot produce the same type of update for that branch.

Several methods have been proposed to make federated learning more reliable when clients differ from one another. FedProx adds a proximal term during local optimization and was designed for both statistical and systems heterogeneity [2]. SCAFFOLD uses control variates to reduce client drift caused by heterogeneous local objectives [3]. FedBN keeps batch-normalization statistics local, which helps when clients experience feature shifts caused by different scanners, sensors, or acquisition domains [4]. These methods are strong baselines for this study, but their main purpose is to handle differences in distributions, optimization, or feature statistics - not missing sensor types.

Federated learning has also been applied in agriculture. Bera et al. proposed FLAG, a federated framework for sustainable irrigation that combines local learning with an IoT and edge-oriented architecture [5]. Devaraj et al. developed RuralAI for tomato monitoring using integrated field sensing and hierarchical federated learning [6]. These studies show that federation is already relevant to agricultural networks. Their main focus, however, is not the aggregation of farms that possess different combinations of aerial and ground sensing branches. Work on multimodal federated learning comes closer to the problem considered here. Yuan et al. proposed MFedMC, which separates modality encoders from fusion components and jointly selects clients and modalities in heterogeneous multimodal networks [7]. Poudel et al. addressed missing modalities using a lightweight feature-imputation network and evaluated the approach on multimodal medical datasets [8]. These studies establish that modality incompleteness is a genuine federated-learning problem. Their evaluations, however, are not centered on the agricultural combination examined here: aerial crop imagery together with heterogeneous ground-sensor measurements distributed across farms.

The remaining gap is therefore narrower than claiming that federated learning has not been used in agriculture or that multimodal federated learning does not exist. Both already exist. The unresolved question addressed in this paper is whether a simple modality-aware aggregation rule can work effectively when agricultural clients have unequal sensing infrastructure. MA-FedSense addresses this setting by maintaining separate modality encoders, aggregating each encoder only across farms that possess the corresponding modality, and then combining their information through a shared fusion component.

III. PROBLEM FORMULATION

We consider a federated network containing K farms. Each farm acts as one client and keeps its own measurements locally. The important point is that farms do not necessarily collect the same kind of information. Let the full set of sensing modalities be

$$\mathcal{M} = \{\text{RGB}, \text{NIR}, \text{soil}, \text{gas}\}.$$

Farm k has only a subset of these modalities,

$$\mathcal{M}_k \subseteq \mathcal{M}.$$

For example, one farm may have RGB drone images and soil measurements,

$$\mathcal{M}_1 = \{\text{RGB}, \text{soil}\},$$

while another farm may also have NIR imagery and gas sensing,

$$\mathcal{M}_2 = \{\text{RGB}, \text{NIR}, \text{soil}, \text{gas}\}.$$

This difference is the main source of modality heterogeneity studied here. It is different from ordinary non-IID learning. Two farms can have different crop or soil distributions even when they use the same sensors. Here, some farms may be completely missing a sensing channel.

The learning task is crop-stress prediction. For the modalities available at farm k , the model produces

$$\hat{y}_k = f(x_k^{m_1}, x_k^{m_2}, \dots, x_k^{m_j}; \theta),$$

where $m_1, \dots, m_j \in \mathcal{M}_k$, $\hat{y}_k$ is the predicted crop-stress output, and θ contains the trainable parameters.

Instead of using one indivisible model, MA-FedSense separates the parameters into modality-specific encoders and a common fusion component:

$$\theta = \{\theta^1, \theta^2, \dots, \theta^{|\mathcal{M}|}, \phi\}.$$

Here, θ^m contains the parameters used to encode modality m , while ϕ represents the shared fusion and prediction layers. A farm without drone imagery cannot train the aerial-image encoder, and a farm without a gas sensor cannot provide a useful update for the gas branch. The proposed federated procedure therefore allows each client to train only the parts of the model supported by its own measurements.

IV. MA-FEDSENSE ARCHITECTURE

The model has three main parts: an aerial-image encoder, a ground-sensor temporal encoder, and a multimodal fusion module. The architecture is modular because the exact combination of inputs can change from one farm to another.

4.1 Aerial image encoder

RGB and NIR images collected from drones form the aerial branch. For a transformer-based implementation, an image is divided into smaller patches and projected into a common embedding space:

$$z_0 = [x_{\text{cls}}; x_p^1 E; x_p^2 E; \dots; x_p^N E] + E_{\text{pos}}.$$

Here, x_p^i is the flattened representation of image patch i , E is the learnable patch-projection matrix, x_{cls} is the classification token, and E_{pos} supplies positional information. In simple terms, the equation converts a large aerial image into a sequence of smaller learned representations that can later be fused with ground-sensor features.

The implemented publication model uses a lightweight ViT as the default aerial encoder because token outputs integrate naturally with attention-based fusion. A compact CNN is retained only as an optional computational ablation.

4.2 Ground-sensor temporal encoder

Ground sensors generate repeated measurements rather than images. The implemented field interface represents soil moisture, electrical conductivity, pH, and temperature, with gas measurements available only at selected sensing locations. At 15-minute sampling, a 72-hour input window contains 288 time steps.

For farm k , the time-series input is

$$X_k \in \mathbb{R}^{T \times C_k},$$

where C_k is the number of channels available at that farm. A one-dimensional convolution first identifies short-term patterns:

$$h_i^{(l)} = \rho \left(\sum_{c=1}^{c^{(l-1)}} \sum_{j=1}^q W_{c,j}^{(l)} x_{i+j,c}^{(l-1)} + b^{(l)} \right).$$

The convolution is followed by an LSTM so that longer temporal behaviour can be represented:

$$\begin{aligned} f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f), \\ i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i), \quad o_t \\ &= \sigma(W_o[h_{t-1}, x_t] + b_o), \\ \tilde{C}_t &= \tanh(W_c[h_{t-1}, x_t] + b_c), \\ C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \\ h_t &= o_t \odot \tanh(C_t). \end{aligned}$$

The two stages have different roles. The 1D-CNN detects short local changes, while the recurrent stage summarizes patterns that develop across a longer observation window.

4.3 Multimodal fusion

After the separate encoders process their inputs, their outputs are projected into a common representation space:

$$e_k^m = W_m h_k^m + b_m.$$

The available embeddings are collected as

$$Z_k = [e_k^{m_1}, e_k^{m_2}, \dots, e_k^{m_j}].$$

Only modalities present at farm k are included. Missing sensors are treated as unavailable information rather than replaced with arbitrary values. The fusion mechanism uses attention:

$$Q = Z_k W_Q, \quad K = Z_k W_K, \quad V = Z_k W_V,$$

$$A = \text{softmax} \left(\frac{QK^T}{\sqrt{d}} \right),$$

$$H_k = AV.$$

This allows one modality to change how another is interpreted. An unusual aerial pattern, for example, may receive different importance when ground measurements also show declining soil moisture. A masking mechanism prevents a missing modality from producing a false token.

Finally, the crop-stress prediction is obtained from the shared prediction head:

$$\hat{y}_k = g_\phi(H_k).$$

The architecture therefore separates how each sensor type is understood from how the available sensor information is combined. This separation makes modality-specific federation possible.

V. MODALITY-ADAPTIVE FEDERATED AVERAGING

Standard FedAvg combines participating client updates using a weighted average [1]. That is reasonable when every client trains the same parameter blocks. The situation is different in MA-FedSense. A farm with a gas sensor can update the gas encoder, but a farm without that sensor cannot. MA-FedAvg therefore checks which clients actually possess each modality and aggregates only eligible updates.

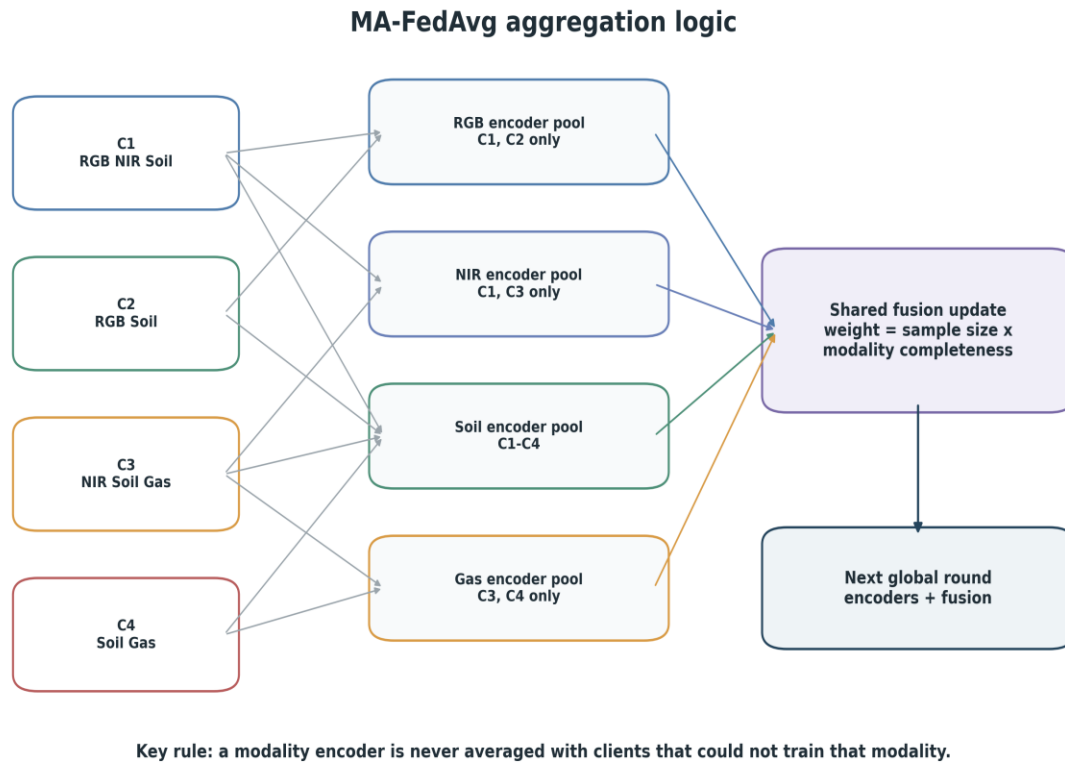


Figure 2. MA-FedAvg aggregation logic. Each modality encoder has an eligible client pool; the shared fusion block is aggregated separately using sample size and modality completeness.

5.1 Modality-specific encoder aggregation

For modality m , define the set of eligible farms as

$$\mathcal{K}_m = \{k: m \in \mathcal{M}_k\}.$$

Let n_k^m be the number of local training samples for modality m at farm k , and define

$$n^m = \sum_{k \in \mathcal{K}_m} n_k^m.$$

The global encoder for modality m is updated as

$$\theta_{t+1}^m = \sum_{k \in \mathcal{K}_m} \frac{n_k^m}{n^m} \theta_{t+1}^{m,k}.$$

The idea is simple. If only three farms have gas sensors, only those three farms take part in updating the gas encoder. Their contributions are weighted by the amount of gas-related training data they contain. This keeps the familiar sample-size weighting of FedAvg, but applies it only where the update is meaningful.

5.2 Modality-completeness weighting for the fusion module

The shared fusion component requires a different rule. Every participating farm can train the fusion module using the modalities available locally, but the amount of cross-modal information differs between farms. We define modality completeness as

$$\gamma_k = \frac{|\mathcal{M}_k|}{|\mathcal{M}|}.$$

The proposed aggregation weight is

$$\omega_k = \frac{\gamma_k n_k}{\sum_{j=1}^K \gamma_j n_j},$$

and the shared fusion parameters are updated by

$$\phi_{t+1} = \sum_{k=1}^K \omega_k \phi_{t+1}^k.$$

This gives greater influence to clients that contain both more training samples and a larger fraction of the available modalities. The reasoning is that these farms can observe a richer set of cross-modal relationships during local training. The weighting

remains a methodological design choice rather than an empirically proven advantage in the present software-validation study; the codebase exposes an ablation in which the completeness factor is set to one for all clients.

5.3 Local training procedure

At the beginning of communication round t , client k receives only the model components supported by its modalities:

$$\theta_t^k = \{\theta_t^m : m \in \mathcal{M}_k\} \cup \{\phi_t\}.$$

For E local epochs,

$$\theta_{t,e+1}^k = \theta_{t,e}^k - \eta \nabla \mathcal{L}_k(\theta_{t,e}^k),$$

where η is the learning rate and $\mathcal{L}_k$ is the local crop-stress loss. After training, the client returns only the parameters it actually updated. Each modality encoder is aggregated across $\mathcal{K}_m$, and the shared fusion module is updated using ω_k .

VI. SOFTWARE VALIDATION DESIGN

The present study evaluates whether MA-FedSense has been translated from its mathematical specification into software that behaves consistently under missing modalities, frozen client partitions, and strict result-archiving rules. Predictive accuracy on real farms is intentionally separated from software correctness: a method can be implemented correctly even before its agronomic effectiveness is established.

6.1 Validation objectives

- V1 - verify that modality-specific encoder aggregation and completeness-weighted fusion aggregation reproduce the defined MA-FedAvg equations.
- V2 - verify that the publication architecture accepts incomplete modality sets and freezes encoders that a client cannot train.
- V3 - verify deterministic plot-level partitioning, modality-mask generation, and protection against cross-client leakage.
- V4 - verify that the field-data interface constructs causal 72-hour windows without inventing missing sensor values or stress labels.
- V5 - verify that evaluation, threshold-round analysis, paired effect calculations, run archiving, and the experiment runner fail safely when required evidence is missing.

6.2 Verification environment and acceptance criteria
Local verification was executed with Python 3.13.5, PyTorch 2.10.0+cpu, NumPy 2.3.5, SciPy 1.17.0, and pandas 2.2.3. The package pins flwr[simulation] == 1.32.1 for the intended federated runtime. Flower itself was not installed in the supplied execution container, so the study distinguishes framework-independent software verification from an executed multi-client Flower federation. Acceptance required a fully passing automated test suite, successful Python bytecode compilation, deterministic protocol artifacts, and explicit rejection of unsupported or incomplete experimental states.

VII. FIELD-DATA INTERFACE AND PILOT INTEGRATION

MA-FedSense includes a strict field-data interface for synchronized aerial and ground observations. Each manifest row contains a sample identifier, plot identifier, timestamp, modality file paths, a pre-existing stress label, and a split designation. The loader does not generate labels, substitute random arrays, or interpret an all-zero tensor as a missing sensor.

Ground inputs follow the manuscript design of 72 hours at 15-minute cadence, giving 288 time steps. Soil windows have shape [288, 4] for moisture, electrical conductivity, pH, and temperature; gas windows have shape [288, 1] when genuine gas measurements are available. The final ground observation is constrained to occur at or before the associated aerial timestamp. Incomplete soil windows are rejected, while an unavailable gas modality remains absent rather than being zero-filled.

The field layer is therefore ready to ingest a paired pilot dataset, but no real pilot manifest or sensor arrays were supplied to the present validation environment. The results in this paper concern correctness of the interface and pipeline, not agronomic prediction from unseen field measurements.

VIII. HETEROGENEITY AND PARTITION PROTOCOL

Statistical heterogeneity is represented through deterministic group-level Dirichlet partitions with concentration values α in {10, 1, 0.5, 0.1}. The

independent partition unit is the plot (or another pre-specified spatial group), not an individual image patch or 15-minute window. Consequently, all observations derived from the same plot remain assigned to one pseudo-client, preventing cross-client leakage through repeated measurements of the same physical unit.

Modality heterogeneity is represented by deterministic M0-M3 masks. For the primary comparison universe, the core modalities are RGB, NIR, and soil; gas remains a supplementary channel because it is natively sparse in the pilot design. The freezer writes YAML artifacts containing client assignments, label counts, modality availability, and source fingerprints. A publication grid can therefore be recreated exactly for every seed without regenerating a partition during model comparison.

IX. COMPARATOR IMPLEMENTATION STATUS

The software package treats comparator availability as a methodological constraint rather than a naming convention. FedAvg [1], FedProx [2], and MA-FedAvg are implemented as runnable strategies. FedProx includes the required client-side proximal regularization around the round-start parameter snapshot. SCAFFOLD [3] is deliberately blocked until control variates and persistent state are implemented. FedBN [4] is also blocked on the current publication architecture because the model uses LayerNorm rather than BatchNorm, leaving no batch-normalization statistics to localize. The MFedMC-inspired comparator [7] remains blocked until its modality/client selection rule and communication budget are implemented faithfully. This guardrail prevents an important reproducibility failure: a run cannot be labelled with a sophisticated baseline name while actually executing ordinary FedAvg.

X. EVALUATION METRICS AND RESULT ARCHIVING

The evaluation layer implements binary accuracy, macro-F1, AUROC, expected calibration error (ECE), and threshold-reaching communication rounds. For the single-site pseudo-client setting, nonlinear metrics are computed from pooled held-out predictions rather than by averaging client-level AUROC or ECE values.

$$ECE = \sum_{b=1}^B \frac{|S_b|}{n} |\text{acc}(S_b) - \text{conf}(S_b)|$$

The corrected absolute calibration gap in the ECE expression is implemented directly in the test suite. The results pipeline also computes seed-paired differences, M0-to-M3 degradation, deterministic bootstrap confidence intervals, and R_{tau} , defined as the first round at which validation Macro-F1 reaches a pre-specified threshold. Runs that never reach the threshold remain recorded as not reached; no artificial terminal-round value is substituted.

Every completed run is associated with a deterministic identity and immutable JSON archive. Duplicate identities are rejected, checkpoint names are tied to the same identity, and final analysis refuses to impute missing experimental cells. These safeguards make the machine-readable archive, rather than terminal output or manual transcription, the source of record.

XI. SOFTWARE VERIFICATION RESULTS

The completed software-verification study produced a fully passing automated test suite. Running `python -m pytest -q` in the supplied package yielded 29 passed tests in 4.29 s. Python bytecode compilation of the `ma_fedsense` package also completed successfully. These are implementation results; they do not constitute crop-stress predictive performance.

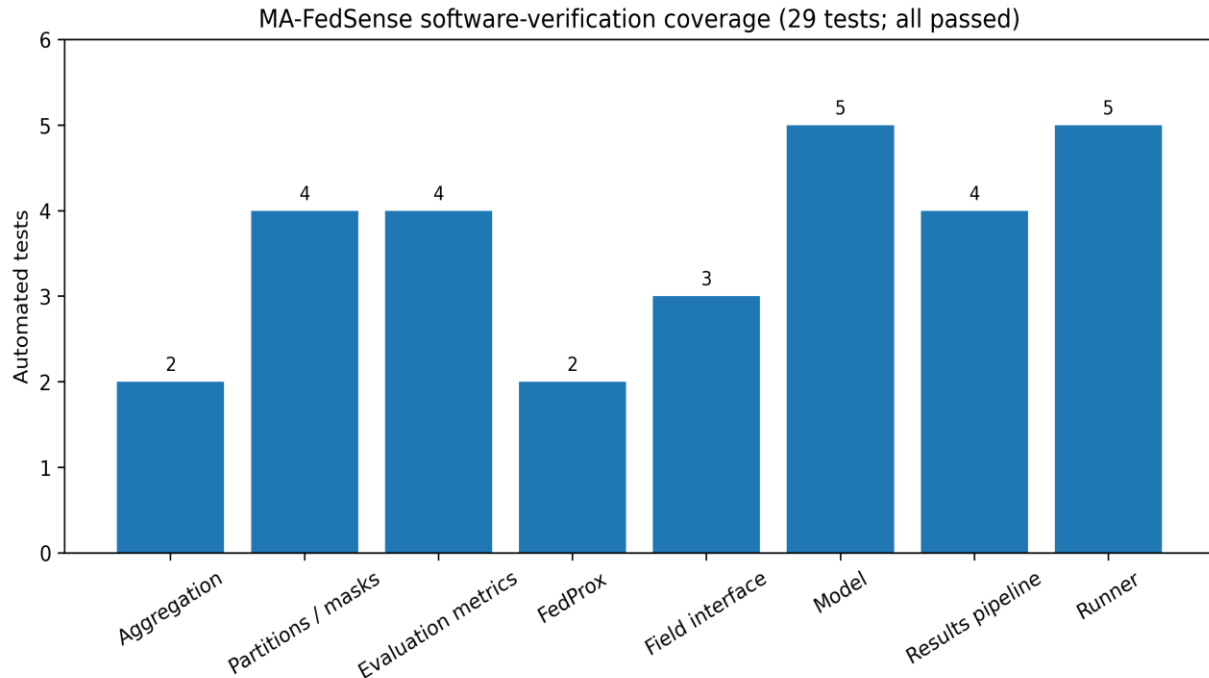


Figure 3. Distribution of the 29 automated verification tests across the main MA-FedSense software subsystems. All collected tests passed in the supplied environment.

Table 1. Automated software-verification summary.

Subsystem	Tests	Status	Primary invariant
MA-FedAvg aggregation	2 / 2	PASS	Modality-specific weighting; branch preservation
Partitions and masks	4 / 4	PASS	Determinism, exact coverage, group integrity, YAML round-trip
Evaluation metrics	4 / 4	PASS	AUROC, ECE, Macro-F1
FedProx objective	2 / 2	PASS	Reference snapshot and proximal term
Field-data interface	3 / 3	PASS	Native modalities, gas refusal, causal windows
Publication model	5 / 5	PASS	Missing modalities, prefix contract, encoder freezing
Results pipeline	4 / 4	PASS	R_tau, paired deltas, deterministic CI, duplicate rejection
Experiment runner	5 / 5	PASS	Method guards, frozen paths, archive identity, grid construction
Total	29 / 29	PASS	All automated tests passed

11.1 Aggregation correctness

The framework-independent MA-FedAvg smoke test passed. In the deterministic test fixture, the soil encoder pool aggregated to 5.0, the gas encoder pool

to 10.0, and the completeness-weighted shared fusion update to 6.400000095367432. These values are reference outputs constructed to verify the equations; they are not predictive scores.

Table 2. MA-FedAvg numerical smoke-test outputs.

Verification target	Observed reference output	Status
Soil encoder aggregation	5.0	PASS
Gas encoder aggregation	10.0	PASS
Completeness-weighted fusion aggregation	6.400000095367432	PASS
No-eligible-client branch preservation	Unmodified branch retained	PASS

11.2 Publication-architecture verification

The publication model combines separate RGB and NIR patch-transformer encoders, 1D-CNN-LSTM temporal encoders for soil and gas, a shared learned-query cross-attention fusion block, and a shared classifier. The default network contains 1,263,138

parameters. When a client has only RGB and soil, unavailable encoders are frozen and 743,778 parameters remain trainable. Tests confirmed that missing-modality forward passes are valid and that state-dictionary prefixes remain compatible with modality-specific aggregation.

Table 3. Publication-architecture verification.

Component	Implementation	Publication input	Parameter prefix	Status
RGB	Patch embedding + Transformer	Image tensor	encoders.rgb.*	PASS
NIR	Patch embedding + Transformer	Single-channel image	encoders.nir.*	PASS
Soil	1D-CNN + LSTM	[B, 288, 4]	encoders.soil.*	PASS
Gas	1D-CNN + LSTM	[B, 288, 1]	encoders.gas.*	PASS
Fusion	Learned-query cross-attention	Available modality tokens	fusion.*	PASS
Classifier	LayerNorm + MLP	Fused representation	classifier.*	PASS

11.3 Data-integrity and protocol verification

The partition and field-interface tests established four integrity properties. First, every independent spatial unit is assigned once and only once to a pseudo-client. Second, expanding a frozen plot assignment to its repeated windows never moves samples from the same plot to different clients. Third, causal ground windows use only observations at or before the aerial event. Fourth, missing modalities are represented as absent inputs; the loader refuses to invent gas data, interpolate incomplete soil windows, or synthesize stress labels. These constraints directly protect the

future empirical study from leakage and artificial multimodal pairing.

11.4 Metrics, archives, and runner verification

The evaluation tests reproduced exact binary AUROC behavior, a hand-calculated ECE example, and Macro-F1. The results pipeline correctly pairs experimental effects by seed, produces deterministic bootstrap intervals, preserves not-reached threshold rounds, and rejects duplicate run identities. Runner tests confirmed exact Cartesian construction of registered cells and the ability to freeze a complete protocol grid on a balanced 50-unit fixture.

11.5 Comparator readiness

Table 4. Comparator implementation and guard status.

Method	Implementation state	Runner status	Reason / note
FedAvg	Implemented	Runnable	Standard comparator
FedProx	Implemented	Runnable	Includes client-side proximal objective
MA-FedAvg	Implemented	Runnable	Primary proposed strategy
SCAFFOLD	Not yet implemented	Blocked	Requires client/server control variates

Method	Implementation state	Runner status	Reason / note
FedBN	Not methodologically active	Blocked	Current model has LayerNorm and no BatchNorm state
MFedMC-inspired	Not yet implemented	Blocked	Selection rule and communication budget not specified in code

The blocking behavior is itself a verification result: unsupported comparators cannot be silently substituted with another strategy and then reported under a misleading label.

XII. REPRODUCIBILITY AND EXECUTION WORKFLOW

The accompanying software package is organized so that raw observations, protocol artifacts, model

Implemented MA-FedSense reproducibility and data-integrity workflow

training, and manuscript statistics remain traceable. Field preprocessing creates causal NumPy windows and an auditable manifest; the protocol freezer creates deterministic partition and mask YAMLS; the experiment runner constructs exact method-by-alpha-by-regime-by-seed cells; the server archive writes immutable run records; and the results generator produces JSON, CSV, and Markdown outputs from archived evidence.



Every stage is deterministic or archived; missing modalities are never synthesized and incomplete final-run cells are never imputed.

Figure 4. Implemented MA-FedSense reproducibility and data-integrity workflow from field logs to manuscript-ready machine-readable outputs.

Core local verification commands are:

```
python -m pytest -q
python -m ma_fedsense.core_smoke
python -m compileall -q ma_fedsense
python -m ma_fedsense.freeze_protocol_grid --units-csv data/plot_units.csv --id-column plot_id --label-column stress_class --num-clients 10 --seeds 20260801 20260802 20260803 20260804 20260805 -
-alphas 10 1 0.5 0.1 --out configs/locked
python -m ma_fedsense.generate_results --log-dir outputs/final_runs --out-dir outputs/section11
```

XIII. PRIVACY AND COMMUNICATION SCOPE

The federated design keeps raw farm observations at the client during normal training, but plain federated

averaging does not by itself provide a formal differential-privacy guarantee. The current validated package therefore does not claim differential privacy. A clipping-and-noise extension can be evaluated separately using established DP-FL mechanisms [10,11].

The current transport also exchanges compatible complete model states and freezes or ignores unavailable branches locally. Consequently, this study does not claim communication-byte savings from modality-aware aggregation. Sparse branch transmission is a future engineering extension and must be measured before any communication-efficiency claim is made.

XIV. DISCUSSION

The central result of this study is not an accuracy number but a verified implementation. MA-FedSense now exists as an executable modular model, aggregation core, field-data interface, reproducibility layer, and result-processing pipeline whose critical invariants are covered by a 29-test suite. This distinction matters because a federated method for heterogeneous modalities can fail in subtle ways even when the high-level equations appear correct: an unavailable branch may accidentally receive gradients, repeated windows from one plot may leak across clients, a missing sensor may be represented by artificial zeros, or a baseline label may not match the strategy actually executed. The present package contains explicit tests or guards against each of these failure modes.

The modality-specific aggregation test provides direct evidence that the software follows the proposed mathematical mechanism. Separate parameter prefixes for RGB, NIR, soil, and gas make client eligibility observable at the server, while the fusion parameters remain structurally shared. The successful missing-modality forward and freezing tests show that the architecture can operate when a farm lacks one or more sensor branches without treating invented values as measurements.

The data-integrity layer is equally important for agricultural deployment. Repeated temporal windows and spatially adjacent images can create severe leakage if partitioning occurs at the row level. By freezing plot-level assignments before window expansion, MA-FedSense preserves the intended statistical unit. The causal 288-step extractor also prevents future ground observations from entering a sample associated with an earlier aerial event. These safeguards make subsequent predictive results more interpretable because the software constrains several common sources of optimistic bias before model fitting begins.

Comparator blocking demonstrates a related principle. SCAFFOLD, FedBN, and MFedMC are relevant literature comparators [3,4,7], but the current code refuses to report them until their algorithmic requirements are genuinely satisfied. This is preferable to an apparently larger benchmark table built from mislabeled runs. In particular, FedBN cannot be meaningfully distinguished on the present

LayerNorm architecture because there are no BatchNorm statistics to retain locally.

Taken together, the verification results support the methodological claim that MA-FedSense is reproducibly implementable for sensor-heterogeneous federated learning. They do not establish that MA-FedAvg improves crop-stress Macro-F1, AUROC, calibration, or convergence relative to competing methods. Those questions require the real paired pilot data and an executed federation and are separated from the claims of this completed software-validation study.

XV. LIMITATIONS

- No real paired pilot manifest or field arrays were available in the present execution environment; therefore no agronomic predictive result is reported.
- Flower 1.32.1 is pinned by the package but was not installed in the supplied container, so an end-to-end Flower federation was not executed here.
- The automated tests establish software correctness for the tested invariants; they cannot prove generalization, robustness, fairness, or agronomic usefulness.
- Only FedAvg, FedProx, and MA-FedAvg are currently executable in the registered runner. SCAFFOLD and MFedMC remain implementation work, while FedBN requires a compatible BatchNorm architecture or removal from the final empirical benchmark.
- The current method does not provide a formal differential-privacy guarantee and does not implement sparse branch transmission, so privacy budgets and byte-level communication savings are not claimed.
- A single-site pseudo-client study is weaker evidence than a genuine multi-farm deployment with independently collected data.

XVI. EXTERNAL EMPIRICAL VALIDATION

The verified software is designed to support a subsequent empirical study rather than to replace one. A complete external validation should supply a paired field manifest with real labels, freeze plot-level partitions and M0-M3 masks for the registered

seeds, run each faithfully implemented comparator under identical data and evaluation schedules, and archive every completed cell. The primary crop-stress endpoints should be Macro-F1, AUROC, ECE, and R_{tau} , with seed-paired effects and uncertainty generated directly from the immutable archives.

Gas sensing should remain a supplementary analysis when native coverage is sparse. Multi-farm deployment should then test whether the pseudo-client findings transfer to genuine differences in equipment, crop distributions, management practices, connectivity, and sensor calibration. Because these measurements are not part of the current evidence base, no numerical claim from that future validation is anticipated or imputed in this paper.

XVII. CONCLUSION

This study completed the methodological design, software implementation, and local verification of MA-FedSense, a modality-adaptive federated framework for crop-stress modelling across farms with unequal sensing infrastructure. MA-FedAvg separates modality-specific encoders from a shared fusion component and restricts each encoder aggregation to clients that actually possess that modality.

The supplied implementation passed all 29 automated tests in the validation environment, reproduced the expected reference aggregation outputs, supported missing-modality inference and encoder freezing, enforced deterministic plot-level partitioning and causal 288-step field windows, and verified the metrics, archive, and runner safeguards needed for reproducible experiments. The publication model contains 1,263,138 parameters and retains 743,778 trainable parameters in an RGB-plus-soil client configuration.

These findings establish that MA-FedSense is an executable and reproducible method, not merely a conceptual proposal. They do not establish field predictive superiority, differential privacy, or communication savings, and those claims are deliberately excluded. The accompanying software and supplementary material provide the verified foundation required for a subsequent real-data, multi-client evaluation.

REFERENCES

- [1] McMahan, B., Moore, E., Ramage, D., Hampson, S., & Aguera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 54, 1273-1282.
- [2] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2.
- [3] Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., & Suresh, A. T. (2020). SCAFFOLD: Stochastic controlled averaging for federated learning. *Proceedings of the 37th International Conference on Machine Learning*, 119, 5132-5143.
- [4] Li, X., Jiang, M., Zhang, X., Kamp, M., & Dou, Q. (2021). FedBN: Federated learning on non-IID features via local batch normalization. *International Conference on Learning Representations*.
- [5] Bera, S., Dey, T., Mukherjee, A., & De, D. (2024). FLAG: Federated learning for sustainable irrigation in Agriculture 5.0. *IEEE Transactions on Consumer Electronics*, 70(1), 2303-2310.
<https://doi.org/10.1109/TCE.2024.3370373>
- [6] Devaraj, H., Sohail, S., Ooi, M., Li, B., Hudson, N., Baughman, M., Chard, K., Chard, R., Casella, E., Foster, I., & Rana, O. (2024). RuralAI in tomato farming: Integrated sensor system, distributed computing, and hierarchical federated learning for crop health monitoring. *IEEE Sensors Letters*, 8(5), 1-4, 5501604.
<https://doi.org/10.1109/LSSENS.2024.3384935>
- [7] Yuan, L., Han, D.-J., Wang, S., Upadhyay, D., & Brinton, C. G. (2026). Communication-efficient multimodal federated learning: Joint modality and client selection. *IEEE Transactions on Mobile Computing*. Advance online publication.
<https://doi.org/10.1109/TMC.2026.3673697>
- [8] Poudel, P., Chhetri, A., Gyawali, P., Leontidis, G., & Bhattarai, B. (2025). Multimodal

- federated learning with missing modalities through feature imputation network. *arXiv preprint arXiv:2505.20232*.
<https://doi.org/10.48550/arXiv.2505.20232>
- [9] Chiu, M. T., Xu, X., Wei, Y., Huang, Z., Schwing, A. G., Brunner, R., Khachatrian, H., Karapetyan, H., Dozier, I., Rose, G., Wilson, D., Tudor, A., Hovakimyan, N., Huang, T. S., & Shi, H. (2020). Agriculture-Vision: A large aerial image database for agricultural pattern analysis. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2828-2838.
- [10] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 308-318.
- [11] Andrew, G., Thakkar, O., McMahan, B., & Ramaswamy, S. (2021). Differentially private learning with adaptive clipping. *Advances in Neural Information Processing Systems*, 34.