MQTT-Enabled Federated Self-Learning Minimal Learning Machine Classifier for Real-Time Hyperspectral Processing

Anna-Maria Raita[1,*], Hadrien Louichon [1,2], Ilkka Pölönen[1]

[1] Spectral Imaging Laboratory, Faculty of Information Technology,
University of Jyväskylä, Finland -(anna.m.raita, ilkka.polonen)@jyu.fi
[2] L'Institut Mines-Télécom, IMT Atlantique, France -hadrien.louichon@imt-atlantique.net

Keywords: MQTT communication, Federated learning, Self-learning Minimal Learning Machine, Hyperspectral imaging, Edge computing, Real-time processing

Abstract

Despite its potential in forestry, agriculture, environmental monitoring, safety surveillance, and defence, real-time hyperspectral imaging (HSI) remains challenging in practice because of the high dimensionality of the data and limited onboard computational resources. This work introduces a distributed HSI classification framework that integrates federated learning, a Self-learning Minimal Learning Machine classifier (SL-MLM), adaptive Kalman filter-based model fusion, and lightweight MQTT-based communication on Raspberry Pi edge devices and a laptop serving as the base station. Acting as local nodes, Raspberry Pis process HSI data row by row, update their models recursively, and only exchange compact model parameters and classification results with the base station. HSI data in its raw form remains local. The findings suggest that the proposed local learning workflow can be implemented on Raspberry Pi devices, and Kalman-based fusion improves stability and consistency in comparison to individual local models. The method is feasible in scenarios where the number of labelled data points is restricted, as the SL-MLM classifier can be initialized with a mere handful of class-specific reference points. The research demonstrates a feasible, low-cost approach to distributed embedded HSI classification and sensing.

1. Introduction

Real-time hyperspectral imaging (HSI) classification on unmanned aerial vehicles (UAVs) can support rapid decision-making in applications such as agriculture, forestry, defence and environmental monitoring. At the same time, practical deployment remains challenging because UAV platforms and embedded devices operate under strict resource constraints, while HSI data are high-dimensional and require efficient onboard processing.

Previous studies have indicated that Raspberry Pi platforms (Pi) can be used both in compact spectral imaging systems and in federated learning on resource-constrained edge devices. Lopez-Ruiz et al. (2017) presented a portable Pi–based multispectral imaging system in which the Pi served as the core platform for image acquisition and basic onboard processing. In distributed learning, Zhang et al. (2021) developed a federated learning platform for IoT that supports training on realistic Pi devices and uses MQTT as the communication backend, while also discussing practical constraints such as communication cost, memory usage, and end-to-end training efficiency. Do and Tran-Nguyen (2024) further showed that federated learning on Pi's can be extended even to large-scale ImageNet classification by using tailored local stochastic gradient descent and support vector machine models trained without data exchange. In that work, pretrained EfficientNetV2 was used as the feature extractor, whereas the federated stage relied on incremental local stochastic gradient descent and local support vector machine classifiers because of the limited device memory. These studies demonstrate that low-cost Pi hardware can support demanding distributed learning tasks when the models and training procedures are adapted to embedded constraints.

* Corresponding author

Our earlier work extended federated learning to real-time HSI classification by introducing a federated self-learning Minimal Learning Machine (SL-MLM) framework for UAV swarms, where onboard model training and prediction were combined with adaptive Kalman filter–based model fusion between UAVs (Raita-Hakola et al., 2025). The present study continues that line of work by moving the framework toward an explicit embedded implementation based on Pi devices and Message Queuing Telemetry Transport (MQTT)-based communication.

As a continuation of our earlier work on self-learning HSI classification (Raita-Hakola and Pölönen, 2022) and the federated SL-MLM classifier (Raita-Hakola et al., 2025), this study extends the framework to use MQTT. MQTT is a lightweight publish–subscribe communication protocol designed for reliable data exchange between distributed devices over constrained or unstable networks (Light, 2017). In the proposed implementation, Pi devices act as distributed learners that process HSI data row by row, update their local SL-MLM models using recursive least squares and anomaly-aware filtering, and transmit only compact model parameters rather than raw HSI data. A laptop acts as the base station and model fusion server, aggregating the incoming local models through an adaptive Kalman filter and returning the refined model to the devices. The computational burden remains moderate because the SL-MLM framework employs compact model updates rather than deep network retraining (de Souza Junior et al., 2015; Mesquita et al., 2017; Raita-Hakola et al., 2025). Although the current system utilises standard Internet Protocol-based communication, the same MQTT-based communication architecture could be extended to software-defined radio links in the future for infrastructure-free remote sensing scenarios (Hosseini et al., 2022).

The novelty of the present study lies in bringing together several previously separate directions into a single embedded framework. To the best of our knowledge, this is among the first studies to integrate HSI classification and federated learning within an explicit Pi–based embedded framework. In particular, the proposed system combines row-wise HSI classification, recursive local SL-MLM updating, adaptive Kalman filter–based model fusion, and MQTT-based communication within the same workflow.

We hypothesize that the proposed framework can be implemented as a coherent distributed system for HSI classification on low-cost edge hardware. In particular, we expect that Raspberry Pi devices are capable of executing the SL-MLM–based local learning workflow and exchanging compact model updates without transmitting raw HSI data. Such a design would reduce communication load and support a more privacy- and security-conscious operating principle, since the original HSI measurements would remain local to the devices. We further hypothesize that the classifier can be initialized using only a few class-specific reference points, making the approach practical in settings where labelled data are limited. Finally, we expect that federated Kalman filter–based fusion stabilizes learning across devices and improves the consistency of the shared classifier compared with individual local models. When confirmed, these properties would support the use of inexpensive edge devices for collaboratively building and refining a common HSI classifier and would provide a practical step toward distributed embedded spectral sensing systems, including future SDR-based mesh-networked implementations.

The paper is organised as follows. The methodology is presented step-by-step in Section 2, complete with equations, algorithms, data visualizations, and trial workflow. Results are shown in Section 3, discussion follows in Section 4, and conclusions are found in Section 5.

2. Material and Methods

2.1 From Minimal Learning Machine to Self-Learning Minimal Learning Machine

The Minimal Learning Machine (MLM) is a computationally efficient, supervised distance-based machine learning method. MLM learns the relationships between input-space and label-space distances, assuming a linear mapping between them (Mesquita et al., 2017). At the original MLM algorithm, the classification is solved via a quadratic optimisation problem, which is computationally demanding. However, by employing different algorithms to solve the labels, the method can be made computationally efficient for anomaly detection and classification using variation-based or nearest-neigh approaches, and it can be efficiently applied to highly dimensional data (Raita-Hakola and Pölönen, 2021; Mesquita et al., 2017; Raita-Hakola and Pölönen, 2022). Besides computational efficiency, MLM models utilise only a few optimisable hyperparameters. These are the number of reference points, the number of neighbours, the distance metrics, and the anomaly thresholds (Raita-Hakola, 2022).

At the core of the proposed framework is a self-learning method, which we modified from an MLM Nearest neighbours classifier (Raita-Hakola and Pölönen, 2022). The main difference is the learning strategy: where the classic MLM utilises whole training data X with selected reference points R to classify each new data point x_n, the SL-MLM uses only few reference data points R per class for initial training, and the model is updated recursively during the classification, which enables it to achieve high accuracy relatively fast (Raita-Hakola and Pölönen, 2022).

The primary goal of SL-MLM is to reduce the size of R, and use of training data X, as it is among the most significant aspect of the computational complexity of this method. At the original MLM, when the size of R approaches the size of X, the accuracy of the results improves (de Souza Junior et al., 2015). To gain higher accuracy using this strategy could be challenging for HSI and small devices, such as Pi's. However, by transforming the model into self-learning, utilising a Recursive Least Squares Algorithm(Romberg, 2016), we can reduce the computational complexity that is related to size of X and R (Raita-Hakola and Pölönen, 2022).

2.1.1 Minimal Learning Machine in HS data classification

When classifying HSI, the training set of spectra with d wavebands is $\mathbf{x}_i \in X \subset \mathbb{R}^d$ and $\mathbf{m}_k \in R$ is the intentionally selected sampled subset of the X. Correspondingly, $\mathbf{y}_i \in Y \subset \mathbb{R}$ are the labels of the training set and $\mathbf{t}_k \in T$ are the subset of the Y. The training set X consist of N samples, and subset R has K samples. Now, $d(\mathbf{x}_i, \mathbf{m}_k)$ and $\delta(\mathbf{y}_i, \mathbf{t}_k)$ are the linear mapping distances.

After the reference points are selected, we can define two matrices based on these distances $\mathbf{\Delta}_y \in \mathbb{R}^{N\times K}$ and $\mathbf{D}_x \in \mathbb{R}^{N\times K}$. By assuming the linear mapping between these two distance matrices, we have a linear model

$$\mathbf{\Delta}_y = \mathbf{D}_x\mathbf{B} + \mathbf{E}, \tag{1}$$

where $\mathbf{B}$ is coefficients and $\mathbf{E}$ is the residual. Coefficients $\mathbf{B}$ can be approximated using the ordinary least squares estimator (Mesquita et al., 2017)

$$\widehat{\mathbf{B}} = (\mathbf{D}_x^T\mathbf{D}_x)^{-1}\mathbf{D}_x^T\mathbf{\Delta}_y. \tag{2}$$

Now $\hat{\mathbf{B}}$ is a linear model between distances $\delta(\mathbf{y}_i, \mathbf{t}_k)$ and $d(\mathbf{x}_i, \mathbf{m}_k)$. Now, for the new spectrum, $\mathbf{x}_n$ the distance between its label y_n and set T is

$$\delta(y_n, T) = d(\mathbf{x}_n, R)\widehat{\mathbf{B}}. \tag{3}$$

Label y_n can be estimated by solving a quadratic optimisation problem

$$\min_{y_n} \sum_{k=1}^{K} \Big((y_n - \mathbf{t}_k)^T (y_n - \mathbf{t}_k) - \delta^2(y_n, T) \Big)^2. \tag{4}$$

The equation 4 is the computational challenge of the original MLM (Mesquita et al., 2017). If we are solving the labels with optimization methods, we can consider that the model $\widehat{\mathbf{B}}$ is an of generalization of the nearest neighbour classifier. Equation 3 gives us distances to the nearest label values of $\mathbf{x}_n$. For predicting the label of the $\mathbf{x}_n$, we need to perform argument sorting for $\delta(\mathbf{y}_n, T)$ and assign the closest label value from T. By selecting k the closest values and use majority voting of labels; we have similar results than with kNN. The detailed implementation is shown in algorithms 1 and 2 (Raita-Hakola, 2022; Mesquita et al., 2017) and visualised in Figure 1.

Algorithm 1: MLM training

Input: X, Y, k
Output: $\widehat{\mathbf{B}}$, R, T

1. Form subsets R and T; select randomly k reference points from X to R and their corresponding reference labels from Y to T.
2. Compute $\mathbf{D_x}$ distance matrix between R and X
3. Compute $\mathbf{\Delta_y}$ distance matrix between T and Y
4. Calculate $\widehat{\mathbf{B}} = (\mathbf{D_x^T D_x})^{-1}\mathbf{D_x^T \Delta_y}$.

Algorithm 2: MLM prediction

Input: R, T, $\widehat{\mathbf{B}}$, new data x_i
Output: distances $\delta(y_i, T)$

1. Compute $d(x_i, R)$
2. Compute $\delta(y_i, T) = d(x_i, R)\ \widehat{\mathbf{B}}$
3. Argsort $\delta(y_i, T)$
4. For a new data $\mathbf{x_i}$, assign n closest labels from labels T.

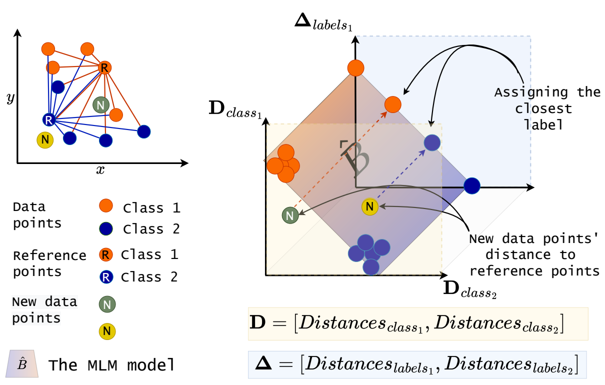

Figure 1. MLM classifier. Classifying new data points’ by their distances with linear mapping (Raita-Hakola, 2022).

2.1.2 Self-Learning MLM To support continuous on-board adaptation, SL-MLM (Algorithm 3, Figure 2) extends the MLM classifier using Recursive Least Squares (RLS) algorithm (Romberg, 2016; Haykin, 2008). Where the classical MLM utilises whole training data X, reference points R and according labels, the SL-MLM needs only a few class-wise reference points R in its initial training. Instead of recomputing the regression from a large training dataset and reference points, RLS updates the model incrementally. For each new spectral sample, the distance matrices are updated, and the model parameters evolve as,

$$\hat{B}_i = \hat{B}_{i-1} + K_i(\Delta y_i - D_{x,i}\hat{B}_{i-1}), \tag{5}$$

where K_i is the RLS gain. This enables efficient real-time updates on resource-constrained devices. Since the SL-MLM classifier only uses new data and reference points, there is a chance that the model will be compromised by noisy or anomalous data in federated real-life systems that use multiple UAVs and their incoming pixel stream. Thus, we integrated an anomaly-aware mechanism into our federated SL-MLM approach.

Algorithm 3: SL-MLM with Recursive Least Squares (Raita-Hakola and Pölönen, 2022)

Input: X, R, Y, T
Initialize:
Calculate distance matrix $\mathbf{D}_{x,0}$
Calculate distance matrix $\mathbf{\Delta}_{y,0}$
Calculate $\mathbf{P}_0 = (\mathbf{D}_{x,0}^T \mathbf{D}_{x,0})^{-1}$
Calculate model $\widehat{\mathbf{B}}_{x,0} = \mathbf{P}_0 \mathbf{D}_{x,0}^T \mathbf{\Delta}_{y,0}$
for *i = 1,2,3...* **do**
  New data X_i, Y_i appears
  Calculate distance matrix $\mathbf{D}_{x,i}$
  Calculate distance matrix $\mathbf{\Delta}_{y,i}$
  Calculate $\mathbf{P}_{x,i} = \mathbf{P}_{i-1}\mathbf{D}_{x,i}^T$
  Calculate $\mathbf{P}_i = (\mathbf{P}_{i-1} - \mathbf{P}_{x,i}(\mathbf{I} + \mathbf{D}_{x,i}\mathbf{P}_{x,i})^{-1} \mathbf{D}_{x,i}\mathbf{P}_{i-1})$
  Calculate $\mathbf{K}_i = \mathbf{P}_i \mathbf{D}_{x,i}^T$
  Update model $\widehat{\mathbf{B}}_{x,i} = \widehat{\mathbf{B}}_{x,i-1} + \mathbf{K}_i(\mathbf{\Delta}_{y,i} - \mathbf{D}_{x,i}\widehat{\mathbf{B}}_{x,i-1})$
end for

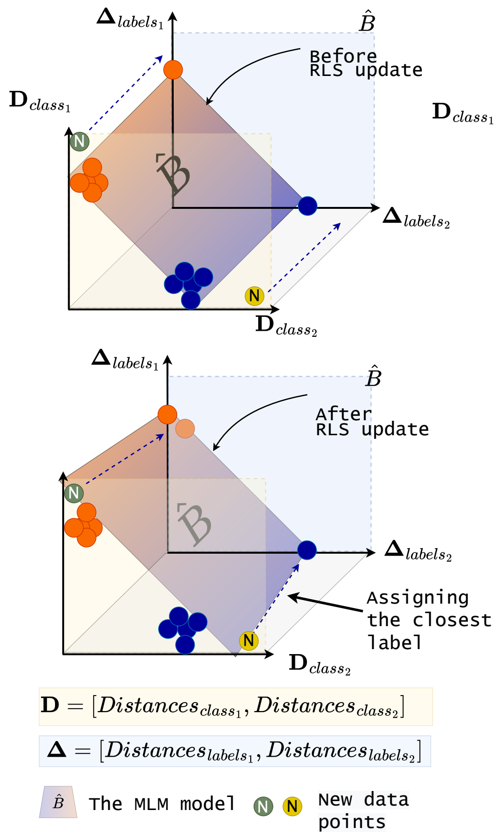

Figure 2. SL-MLM classifier. Classifying new data points’ with RLS updated linear mapping (Raita-Hakola, 2022).

2.1.3 Anomaly-Aware Learning To reduce the effect of unreliable training samples, SL-MLM incorporates an anomaly-detection mechanism based on the variance of predicted label distances (Raita-Hakola and Pölönen, 2021). In this approach, each classified pixel row is screened for anomalies using the variance measure

$$A_n = \mathrm{Var}(\delta(y_n, T)). \tag{6}$$

In each classified row, the recursive update is performed using only non-anomalous pixel spectra. More specifically, only spectra with variance within a predefined acceptance interval are retained, whereas spectra with excessively low or high variance are discarded to reduce the influence of noisy or atypical observations on self-learning.

2.1.4 Federated Learning with Kalman Fusion

Our federated learning framework implementation consists of Pi devices, which are simulating UAVs capturing and classifying HS data row-by-row, and communicating with a base station that visualises the classification results and updates the UAVs local models with a new global model. Each Pi maintains and updates its own SL-MLM model locally. Instead of transmitting raw HS data, Pi's send their row-wise classification results, regression coefficients $\hat{B}_i$ and covariance matrices P_i to the base station, minimizing communication requirements.

The base station fuses incoming models using an adaptive Kalman filter (Mehra, 1970). Treating each $(\hat{B}_i, P_i)$ pair as a noisy observation of the global model, the fusion follows the standard predict–update structure:

$$\begin{aligned} B_{\text{pred}} &= B_{\text{fuse}}, \\ P_{\text{pred}} &= P_{\text{fuse}} + Q, \\ K_i &= P_{\text{pred}}(P_{\text{pred}} + P_i)^{-1}, \\ B_{\text{fuse}} &= B_{\text{pred}} + K_i(\hat{B}_i - B_{\text{pred}}), \\ P_{\text{fuse}} &= (I - K_i)P_{\text{pred}} \end{aligned} \tag{7}$$

Adaptive process noise is updated, as

$$Q \leftarrow \|K_i\|_F I. \tag{8}$$

This ensures that the global model remains responsive to informative updates and avoids overconfidence. The fused model is redistributed to devices, enabling continual collaborative learning.

2.2 Federated learning using MQTT connection

In the proposed framework implementation, Message Queuing Telemetry Transport (MQTT) was employed as the communication middleware between distributed Pi devices and the coordinating base station. Algorithms 4 and 5 describe our framework's implementation and communication workflows.

Algorithm 4: Federated SL-MLM base station's operations

Initialise: Initialise first global model, and transfer it with reference data to base station and UAV's.
Input: Acquisition index i, UAV$_n$ local models $\widehat{\mathbf{B}}_{i,n}$, row-wise predictions, $P_{i,n}$, and anomaly scores $A_{i,n}$,
Output: Updated local fused model $\widehat{\mathbf{B}}_{\text{fuse,i}}$ and updated P_i
Start MQTT broadcasting: listen to topics UAV$_{i,n}$:
For each i**:**
1. Wait until each UAV$_n$ has published messages.
2. Extract row-wise i, $\widehat{\mathbf{B}}_{i,n}$, $P_{i,n}$, $A_{i,n}$ and prediction results.
3. Evaluate anomalies from $A_{i,t}$
if not all anomalies:
Update the global model with detections using $\widehat{\mathbf{B}}_{i,n}$, $P_{i,n}$ and Kalman Fusion.
Send updated model $\widehat{\mathbf{B}}_{\text{fuse,i}}$ and updated P_i to UAVs.
Continue listening $i + 1$.
else:
Continue listening $i + 1$ for non-anomalous detections.
Finally: Store predictions, i-wise models information and produce results analysis and visualisations.

Algorithm 5: Federated SL-MLM UAV's operations

Initialise: Initialise first global model, and transfer it with reference data to base station and UAV's. MQTT broker connection topic: base station.
Input: Updated local fused model $\widehat{\mathbf{B}}_{\text{fuse,i}}$ and updated P_i
Output: Acquisition index i, UAV$_n$ local models $\widehat{\mathbf{B}}_{i,n}$, row-wise predictions, $P_{i,n}$, and anomaly scores $A_{i,n}$,
At each UAV$_n$**:** start acquisition, analysis, and MQTT communication
For each new HS data row i**:**
1. Compute predictions $\widehat{Y}_{i,n}$ and anomalies $A_{i,n}$ using the current local model $\widehat{\mathbf{B}}_{i,n}$.
2. Evaluate anomalies, and update local $\widehat{\mathbf{B}}_{i,n}$ using RLS algorithm and non-anomalous predictions $\widehat{Y}_{i,n}$.
3. Send i, $\widehat{\mathbf{B}}_{i,n}$, $P_{i,n}$, $A_{i,n}$ and prediction results to base station
4. Wait until base station sends back updated $\widehat{\mathbf{B}}_{\text{fuse,i}}$ and P_i.
Continue
Finally: Stop acquisition.

MQTT is a lightweight publish–subscribe protocol originally developed for constrained and high-latency environments, enabling efficient transmission of small payloads through a broker–client architecture (Light, 2017). Its support for different Quality-of-Service (QoS) levels and asynchronous message exchange has made it suitable for distributed sensing and edge-computing applications (Light, 2017; Naik, 2017). In this work, MQTT provides the messaging layer used to coordinate real-time HSI classification while limiting unnecessary raw-data transfer between devices.

2.3 Data, trial parameters and source code

The experiments were implemented in Python using open-source libraries. Communication between the base station and the Pi nodes was established using the MQTT protocol over a mobile 4G Wi-Fi network. We used two Raspberry Pi 5 8 GB single-board computers (64-bit quad-core Arm Cortex-A76 processor running at 2.4GHz, with a basic Debian/GNU/Linux-based operating system) to emulate UAV nodes (Algorithm 5), while the base station was implemented on a relatively old Dell Latitude 7280 laptop running Ubuntu (Algorithm 4). Each Pi was connected to a keyboard, mouse, and a basic display during the trials. The experiments were conducted using the publicly available HSI Salinas A dataset (Graña et al., 2022).

Before use, the data were min–max normalised. The reference spectra R were randomly selected from six pixel columns extracted from both sides of the Salinas A scene. These columns were then removed from the remaining data, which was vertically divided into two subsets (Figure 3). To simulate a real-time acquisition scenario in which two UAVs scan different areas row by row using spectral line scanners, the two subsets were assigned separately to Pi devices, denoted as UAV1 and UAV2. During the experiments, the two Pi devices operated as independent UAV nodes, processing and classifying the incoming rows sequentially. A visualisation of the data pre-processing procedure is shown in Figure 3. The source code is available at (Raita et al., 2026).

The hyperparameters were optimised based on our previous studies and manual empirical optimization (Raita-Hakola, 2022). Euclidean distance was employed. The reference set R consisted of 30 samples in total, with 5 reference points selected for each of the six classes. Class prediction was based on nearest-reference voting with 5 neighbours in the present work.

The anomaly thresholds were empirically selected through trial and error and were subsequently fixed to

$$\tau_{\text{low}} = 0.95, \qquad \tau_{\text{high}} = 3.7. \tag{9}$$

Accordingly, a sample was flagged as anomalous whenever

$$A(\mathbf{x}) < \tau_{\text{low}} \quad \text{or} \quad A(\mathbf{x}) > \tau_{\text{high}}. \tag{10}$$

These thresholds were introduced as a lightweight safeguard against incorporating highly atypical spectra into the recursive update process.

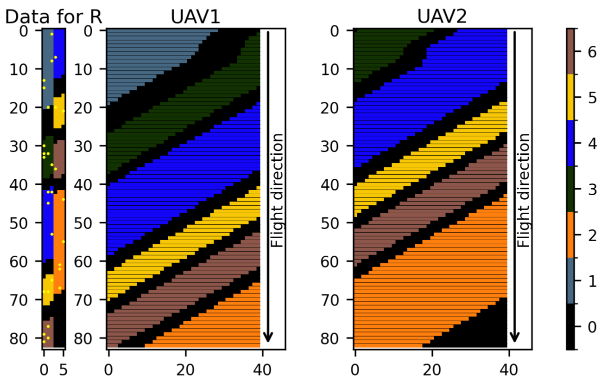

Figure 3. Ground truth and data handling visualisation. Left: The reference points R are drawn in yellow. Middle: After removing the data for R, the Salinas A scene was vertically divided. The models received HS data one line at a time, top-down, to simulate acquisition index i during “UAV flight”. Right: The colourbar represents ground truth classes, where 0 is non-labelled background.

3. Results

3.1 Federated learning performance

Visual inspection in Figure 4 shows progressive improvement in the full-image predictions of the fused base-station model. The initial model already captures the dominant spatial structure, but contains noticeable local errors and less accurate class boundaries. After update round 41, the classification map becomes more coherent, and by round 81 the result is substantially refined, with sharper boundaries and fewer misclassified pixels.

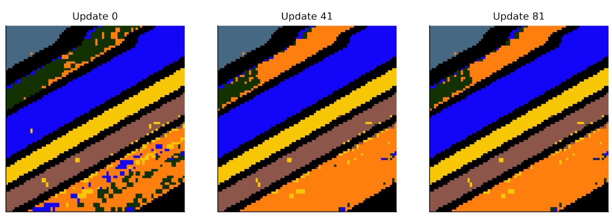

Figure 4. Evolution of full-test classification results at the base station during federated learning. From left to right: prediction using the initial model, prediction after update round 41, and prediction after update round 81. The sequence illustrates progressive refinement of class boundaries and reduction of misclassified regions as federated updates accumulate.

Row-wise predictions in Figures 5 and 6 further reflect the sequential acquisition process from top to bottom. Both UAV’s exhibit higher uncertainty and fragmented class assignments in early rows (<40), whereas later rows show improved spatial consistency and closer agreement with ground truth. This behaviour is consistent with the anomaly maps (Figure 7), where anomalies are concentrated in early rows and near class transitions, and decrease as more data are incorporated.

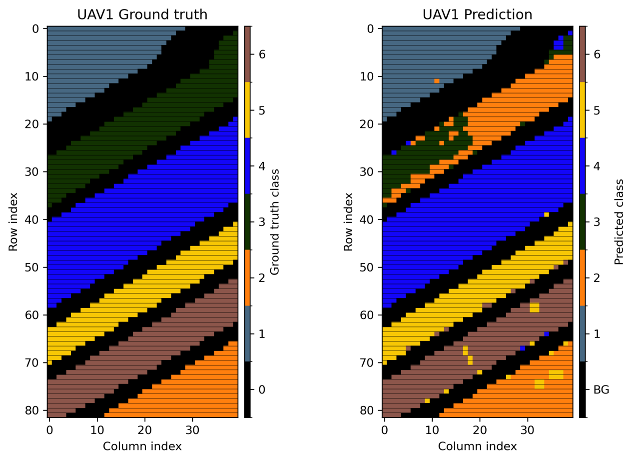

Figure 5. UAV1 row-wise learning: ground truth (left) and prediction (right). Early rows exhibit fragmented and noisy predictions, while later rows show improved spatial coherence and more accurate reconstruction of class boundaries.

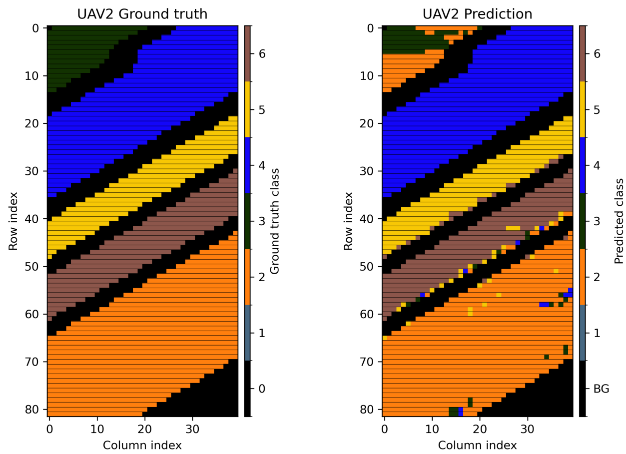

Figure 6. UAV2 row-wise learning: ground truth (left) and prediction (right). Predictions improve progressively from top to bottom, with later rows closely matching the ground truth and reduced classification noise.

The results demonstrate progressive spatial learning, where both local UAV models and the fused base-station model improve systematically as federated updates accumulate.

The evolution of classification performance over sequential updates seen in Figure 8 demonstrates the behaviour of the proposed federated learning framework. Initially, both local models (UAV1, UAV2) exhibit unstable performance, with UAV1 showing a pronounced degradation phase (accuracy dropping to 0.2) while UAV2 maintains moderate stability. This transient degradation corresponds to early-stage model adaptation under limited and biased local observations.

As updates accumulate, both local models converge toward

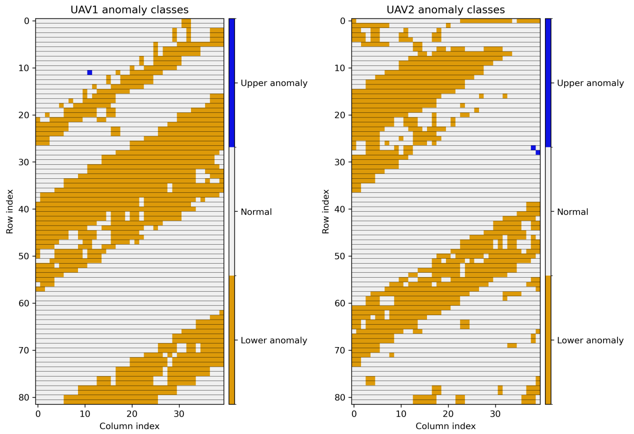

Figure 7. Row-wise anomaly detection during sequential learning for UAV1 (left) and UAV2 (right).

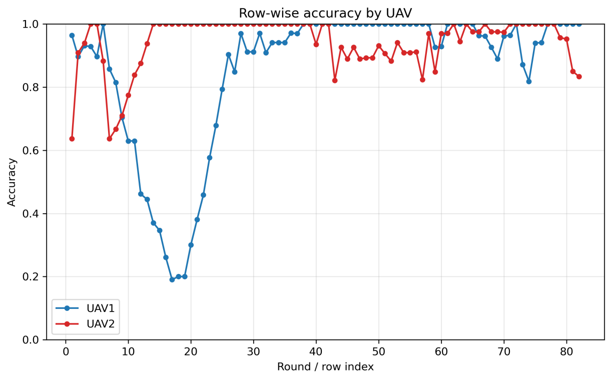

Figure 8. Classification accuracy of UAV1 and UAV2 over sequential updates (row-wise learning). UAV1 exhibits an initial instability phase followed by recovery, while UAV2 maintains more stable performance throughout.

high accuracy (over 0.87), indicating successful incremental learning from streaming HSI data. In the figure 9 the base station model obtained via Kalman fusion achieves faster stabilization and exhibits reduced variance compared to individual UAV models. This suggests that fusion effectively mitigates local noise and model drift inherent to distributed acquisition.

Full-image evaluation further confirms that the fused model achieves consistently high accuracy and weighted F1 scores across updates, indicating strong generalization beyond row-wise updates. The convergence plateau suggests that the model reaches a stable representation after approximately 20–30 updates.

The evolution of model coefficients $\hat{B}$ reveals rapid initial adaptation followed by stabilization. In the figure 10 the norm of parameters in model updates shows large early changes particularly for UAV1 followed by a sharp decay toward near-zero updates. Norm behaviour indicates convergence to a stable solution.

Coefficient trajectories in Figure 11 demonstrate that only a subset of parameters undergoes significant variation, while most coefficients remain relatively stable. This suggests that the model effectively identifies a low-dimensional structure within the HSI feature space.

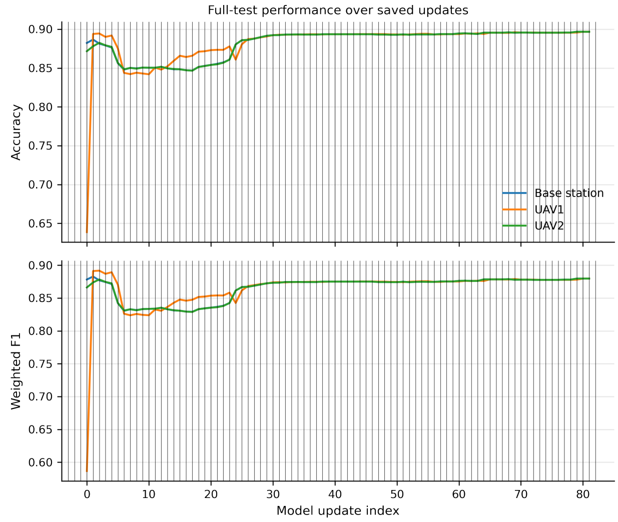

Figure 9. Full-test accuracy and weighted F1 score over updates for the base station (fused model) and UAV models. The fused model converges faster and exhibits reduced variance.

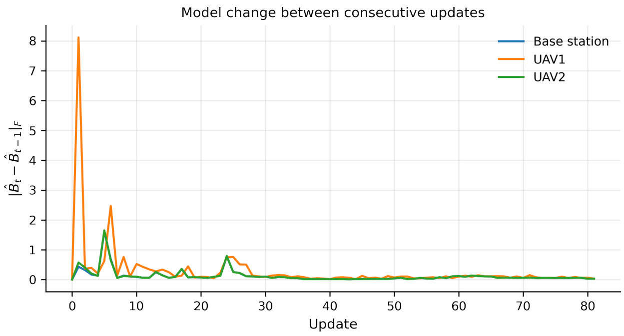

Figure 10. Norm of parameter changes $||\hat{B}_t - \hat{B}_{t-1}||_F$ over updates. Large initial updates are followed by rapid decay, indicating convergence.

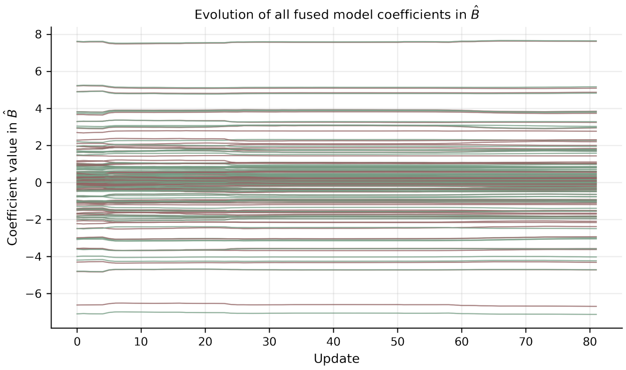

Figure 11. Trajectories of model coefficients in $\hat{B}$. Only a subset of coefficients exhibit significant variation, suggesting a low-dimensional effective representation.

The Kalman filter plays a central role in aggregating distributed updates. In the Figure 12 we see that trace of the fused covariance matrix decreases monotonically, indicating a steady reduction in model uncertainty. This is complemented by a significant decrease in the condition number, suggesting improved numerical stability of the covariance estimate.

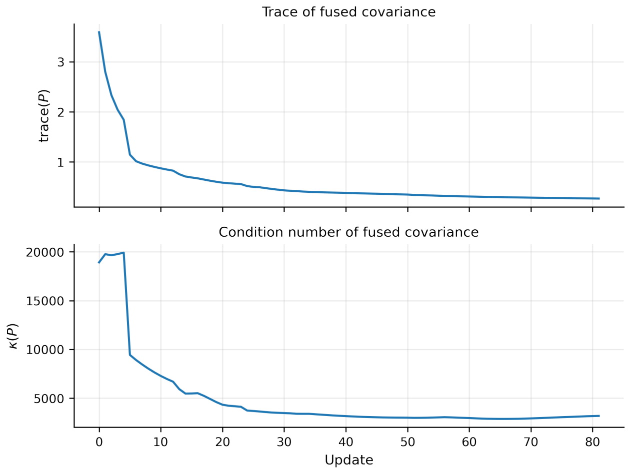

Figure 12. Evolution of covariance statistics in Kalman fusion: trace, condition number, Kalman gain norm, and log-determinant. These metrics indicate decreasing uncertainty and increasing numerical stability.

Figure 13 shows that the fusion process leads to a substantial contraction of the covariance volume, as evidenced by the overall decrease in $\log\det(P)$. After modest fluctuations in the early iterations, the covariance evolution becomes smooth and steadily decreasing, indicating stable uncertainty reduction. Meanwhile, the Kalman gain norm rapidly reaches a plateau, suggesting that the fusion dynamics quickly settle into a stable operating regime. These results support the numerical stability of the proposed fusion strategy and its ability to consistently integrate successive model updates.

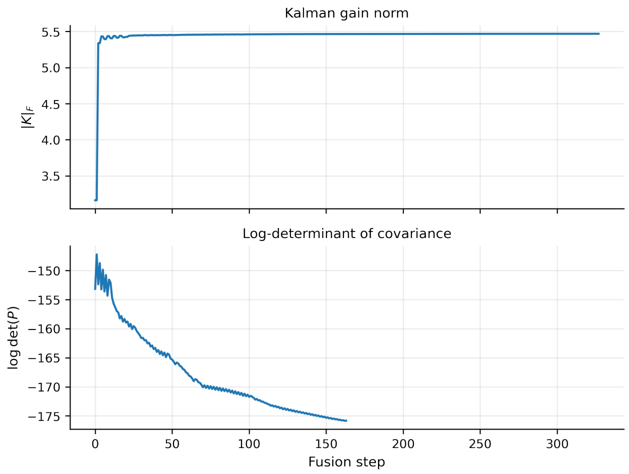

Figure 13. Kalman fusion diagnostics. Top: Frobenius norm of the Kalman gain, which rises rapidly and then stabilizes. Bottom: log-determinant of the covariance matrix, which decreases overall during fusion, indicating progressive reduction of uncertainty in the fused estimate.

The MLM formulation operates in a distance-based representation space, where each sample is described by its distances to a set of reference samples R. As a result, model parameters do not correspond to raw feature weights, but instead encode how relational distances influence class predictions. This induces a structured parameter space in which learning primarily adjusts geometric relationships between samples and reference points. Empirically, only a subset of coefficients exhibits significant updates, indicating that the model relies on a limited number of informative reference directions. This selective adaptation leads to rapid convergence, with early updates capturing the dominant class structure and later updates refining decision boundaries.

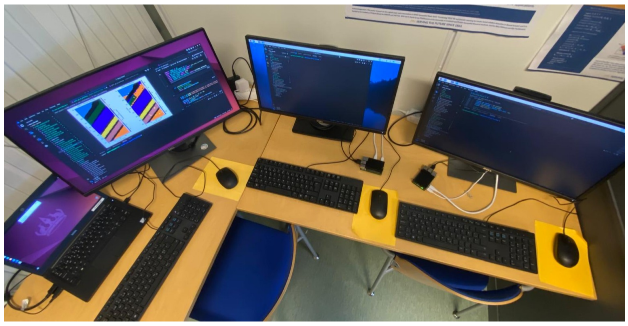

Figure 14. The proposed federated learning setup in action.

3.2 Communication

The proposed federated HSI learning framework was deployed in a distributed setup consisting of two UAV nodes (Raspberry Pis) and a base station (laptop) communicating via an MQTT broker over a wireless network, as shown in Figure 14. During the experiments, the devices were connected to screens, keyboards, and mice for practical operation and monitoring. The ability to measure setup performance and interpret results is limited by the fact that the setup was still more experimental than a fully integrated UAV-mounted implementation.

Empirical observations of system performance, communication, and latency were made with these limitations in mind. The system operated reliably under continuous communication, demonstrating stable bidirectional message exchange between the UAV nodes and the base station. The payloads of Algorithms 5 and 4 were successfully encoded, transferred, and decoded between the nodes and the base station, and each classification round was completed within seconds. No other devices were connected to the Wi-Fi network during the trials.

During hyperparameter optimization, approximately 60 experimental runs were conducted. The equipment was operated at two different locations: on the university campus and in a rural area 16 km from the city centre. The latter location’s mobile connectivity was empirically observed to be occasionally less reliable due to coverage limitations. There was occasional latency during the update cycle; however, it was neither systematic nor persistent. This was probably mostly due to network delays and message routing through a broker. However, the learning process remained consistent, and no divergence or degradation in classification performance was observed, despite these delays.

4. Discussion

The presented results demonstrate that the proposed framework successfully integrates federated learning, Kalman-based model fusion, and a distance-based minimal learning machine (MLM) into a coherent and operational system for HSI classification. A central observation is the stabilizing effect of Kalman filtering: while individual UAV models exhibit variability, the fused model converges more rapidly and maintains consistent performance. Variability exhibited particularly under limited and locally biased data. This behaviour follows from the probabilistic treatment of model updates, where contributions are weighted by their uncertainty, effectively suppressing unreliable estimates during early learning stages.

The initial degradation observed in individual UAV performance reflects the inherent challenges of decentralized and online learning, where limited spatial coverage leads to biased updates. The Kalman fusion mechanism mitigates this by integrating complementary information across UAVs. This is further supported by the evolution of uncertainty measures, where the monotonic decrease in covariance trace and log-determinant indicates stable information accumulation. However, the tendency toward near-singular covariance suggests increasing model confidence that may eventually become over-confident, motivating the use of regularization or covariance inflation in longer sequences.

The MLM formulation introduces a structured parameter space by operating in a relational, distance-based embedding defined with respect to representative samples R. Rather than relying directly on high-dimensional spectral features, the model encodes class structure through distances, leading to rapid convergence and reduced parameter variability. The observed sparsity in coefficient dynamics indicates that only a subset of parameters actively contributes to learning, suggesting that the dominant class geometry is captured early, with subsequent updates refining decision boundaries. Spatial results support this interpretation, showing progressively improved boundary delineation and increased spatial coherence through multi-UAV fusion.

From a computational perspective, MLM's and its updates cost is governed by the number of the reference points and the fused model state $\widehat{\mathbf{B}}_{\text{fuse}}$. In the MLM-based classification stage, the dominant operations arise from pairwise distance evaluation, linear projection, and neighbour ranking, yielding an overall complexity of

$$\mathcal{O}(mrp + mrc + mr\log r)\,,$$

where m denotes the number of classified samples, p the feature dimension, r the number of reference points, and c the output dimension of the model. The recursive least-squares update has complexity

$$\mathcal{O}\big(r^2q + rq^2 + q^3 + qrc\big)\,,$$

where q is the update block size; in the common case of scalar or very small batch updates, this reduces effectively to quadratic scaling with respect to r. In the Kalman fusion step, the fused state corresponds to the vectorized multi-output parameter matrix $\hat{\mathbf{B}} \in \mathbb{R}^{r\times c}$, giving state dimension $s = rc$. Consequently, the fusion update is dominated by dense covariance inversion and matrix multiplications, resulting in cubic complexity,

$$\mathcal{O}(s^3) = \mathcal{O}\big((rc)^3\big).$$

These results are consistent with the implemented code structure and indicate that the principal scalability bottlenecks arise from the reference set size and the dimensionality of the fused multi-output model.

In general, increasing the number of reference points is expected to improve robustness and class coverage, but this also increases computational cost in both classification and recursive updating. At the same time, the recursive learning framework is potentially vulnerable to model poisoning: if anomalous or corrupted spectra are incorporated into the update stream, model performance may deteriorate over time. For this reason, anomaly screening prior to recursive incorporation is important.

A practical strategy could be to estimate nominal score ranges using leave-one-out on the reference point set may be considered to estimate the nominal score range in the absence of a separate validation set. In such a procedure, each reference point is predicted using a model trained with the remaining samples, giving variance scores in range. This range could be used as a base for thresholds. In further studies also class-wise thresholds should be concerned. More advanced extensions could combine thresholding with adaptive reference-set maintenance, robust statistics, or uncertainty-aware update rules to improve resilience against corrupted observations while preserving online adaptability.

The experiments were conducted on a single benchmark dataset, which is relatively structured and exhibits high-class separability, which can be considered a limitation. In more heterogeneous environments, with increased spectral variability and intra-class diversity, the performance of the proposed approach may be affected. In such cases, a larger or more representative reference set R may be required to adequately capture class relationships. While random sampling of R proved sufficient in this study, more advanced selection strategies such as (Hakola and Pölönen, 2020) may further improve robustness and generalization.

Beyond algorithmic performance, the communication architecture plays a critical role in enabling distributed learning. The use of MQTT as a lightweight publish–subscribe middleware allows efficient exchange of compact model parameters, anomaly score and classification results$(\hat{B}, P, A, Y)$ while keeping high-dimensional HSI data local. Although the system functioned consistently across a wireless network, periods of inactivity were triggered by temporary latency introduced by a hard-coded semi-synchronous update scheme. From a learning perspective, such synchronization is not required. Both MLM updates and Kalman fusion are inherently incremental and can incorporate delayed information, suggesting that a fully asynchronous implementation would improve efficiency, scalability, and robustness to network variability.

As seen in Figure 14, the Raspberry Pi devices and the base station were connected to screens, keyboards, and mice. Consequently, we did not perform detailed Joule scope-based power measurements or more precise latency measurements because the additional I/O devices and related system activity would have influenced the results. Such measurements would require a more streamlined and controlled system setup in future work.

Looking forward, although the current system relies on standard Internet Protocol–based communication, the architecture can be extended to software-defined radio (SDR) platforms. In such a setting, MQTT messages can be transmitted over radio waveforms such as orthogonal frequency-division multiplexing (OFDM), quadrature phase-shift keying (QPSK), or low-power long-range modulation (LoRa) (Hosseini et al., 2022). This would enable operation in infrastructure-free environments and further enhance the applicability of the proposed framework in remote sensing scenarios.

The results demonstrate the feasibility of real-time federated HSI learning in a distributed edge environment, while highlighting opportunities for improving generalization, communication efficiency, and scalability through adaptive reference selection and asynchronous operation.

5. Conclusions

This work presented a distributed HSI learning framework that combines federated learning, a distance-based MLM, and adaptive Kalman filtering within a lightweight MQTT-based communication architecture. The trials were made using Raspberry Pi computers and a laptop.

The experiments indicate that Raspberry Pi devices can execute the SL-MLM–based local learning workflow and exchange compact model updates without transmitting raw HSI data. This reduces communication load and supports a more privacy- and security-conscious operating principle, since the original HSI measurements remain local to the devices. In addition, the classifier can be initialized using only a few class-specific reference points, making the approach practical in settings where labelled data are limited.

Federated Kalman filter–based fusion stabilizes learning across devices and improves the consistency of the shared classifier compared with individual local models. Overall, the results indicate that reliable classification can be achieved with a small number of labelled samples while maintaining computational and communication demands at a level suitable for low-cost edge devices such as UAV-mounted Raspberry Pi systems.

Together, these findings provide a practical step toward distributed embedded spectral sensing systems that may eventually operate over software-defined radio (SDR)–based mesh networks. At the same time, further studies with more diverse datasets, broader operating scenarios, and fully integrated real-UAV implementations are needed to assess generality and practical deployment.

6. Acknowledgements

This stydy has been supported by the Research Council of Finland under grant number 357382.

References

de Souza Junior, A. H., Corona, F., Barreto, G. A., Miche, Y., Lendasse, A., 2015. Minimal Learning Machine: A novel supervised distance-based approach for regression and classification. *Neurocomputing*, 164, 34-44.

Do, T., Tran-Nguyen, M., 2024. ImageNet classification with Raspberry Pis: federated learning algorithms of local classifiers. *International Journal of Web Information Systems*, 20(1), 48–65. https://doi.org/10.1108/IJWIS-03-2023-0057.

Graña, M., Veganzons, M., Ayerdi, B., 2022. Hyperspectral Remote Sensing Scenes. Data available in http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

Hakola, A.-M., Pölönen, I., 2020. Minimal learning machine in hyperspectral imaging classification. L. B. Santi, F. Bovolo, Emanuele (eds), *Image and Signal Processing for Remote Sensing XXVI*, 11533, 26.

Haykin, S. S., 2008. *Adaptive filter theory*. Pearson Education India.

Hosseini, S. Y., Chung, Y., Yuce, M. R., 2022. Software-Defined Radio: A Comprehensive Technical Review. *Sensors*, 22(7), 2577. https://www.mdpi.com/1424-8220/22/7/2577.

Light, R., 2017. Mosquitto: server and client implementation of MQTT. *Journal of Open Source Software*, 2(13), 265.

Lopez-Ruiz, N., Granados-Ortega, F., Carvajal, M. A., Martinez-Olmos, A., 2017. Portable multispectral imaging system based on Raspberry Pi. *Sensor Review*, 37(3), 322-329. https://doi.org/10.1108/SR-12-2016-0276.

Mehra, R., 1970. On the identification of variances and adaptive Kalman filtering. *IEEE Transactions on Automatic Control*, 15(2), 175–184.

Mesquita, D. P., Gomes, J. P., Souza Junior, A. H., 2017. Ensemble of Efficient Minimal Learning Machines for Classification and Regression. *Neural Processing Letters*, 46(3), 751–766.

Naik, N., 2017. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. *2017 IEEE International Systems Engineering Symposium*, 1–7.

Raita, A.-M., Louichon, H., Pölönen, I., 2026. Mqtt-enabled federated self-learning minimal learning machine framework. https://gitlab.jyu.fi/hsi/MLM/-/tree/42fdf7fcc17afcca1f770a1727f3e998c7ff18b9/. Accessed: 2026-03-31.

Raita-Hakola, A.-M., 2022. From sensors to machine vision systems: Exploring machine vision, computer vision and machine learning with hyperspectral imaging applications. Doctoral dissertation, University of Jyväskylä.

Raita-Hakola, A.-M., Kilpi-Chen, X., Pölönen, I., 2025. Smart Drones, Smarter Learning: Federated Self-learning Minimal Learning Machine Classifier for Real-Time Hyperspectral Image Classification. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.

Raita-Hakola, A.-M., Pölönen, I., 2022. Updating strategies for distance based classification model with recursive least squares. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3, 163–170.

Raita-Hakola, A.-M., Pölönen, I., 2021. Piecewise anomaly detection using minimal learning machine for hyperspectral images. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-3-2021, 89–96.

Romberg, J., 2016. Ece 6250 fall 2016 lecture notes. Lecture notes available from Georgia Tech University.

Zhang, T., He, C., Ma, T., Gao, L., Ma, M., Avestimehr, S., 2021. Federated learning for internet of things. *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, SenSys ’21, Association for Computing Machinery, New York, NY, USA, 413–419.