Zahra: Bulletin Big data, Data Science and Artificial Intelligence
Vol. 2, No. 2, [December 2023], pp. [98–109]
E-ISSN 2830-7984

Research Article Open Access (CC–BY-SA)

3 × 3 Window-Based Fusion of Spectral Statistics and Spatial Gradients for Random Forest Optimization in Landsat Land-Cover Classification

Dodi Siregar[1], Septiana Vratiwi[2], Tengku Mohd Diansyah[3], Erfan Babaee Tırkolaee[4*]

[1] *Computer Science Program, Faculty of Engineering and Computer Science, Universitas Harapan, Medan, Indonesia*
[2] *Faculty of Intelligent Computing, Universiti Malaysia Perlis (UniMAP), Malaysia*
[3] *Computer Science Program, Faculty of Engineering and Computer Science, Universitas Harapan, Medan, Indonesia*
[4] *İstinye Üniversitesi, Istanbul, Turki*

[1] *dodi.stth@gmail.com;* [2] *septiana@studentmail.unimap.edu.my;* [3] *dian.10.22@gmail.com;* [4] *erfan.babaee@engr.istinye.edu.tr*
** Corresponding author*

Abstract

This study examines 3 × 3 patch descriptors and Random Forest configuration search on the UCI Statlog Landsat Satellite benchmark. The official split contains 4,435 training and 2,000 test observations, each with 36 spectral values and one of six represented classes. The recorded comparison includes a 140-tree raw-feature baseline, a 140-tree untuned fusion model using 60 attributes, and a selected model labeled Optimized. The selection code specifies six feature representations, three-fold stratified training cross-validation, and a 36-trial budget with 12 startup trials. The winning feature type and parameter vector are not embedded in the archived evaluation, so they are not inferred from the title. Stored test outputs give accuracy of 90.85%, 91.40% and 91.20%, and macro-F1 of 89.37%, 89.77% and 89.94%, respectively. Optimized improves macro-F1 over Baseline by 0.57 percentage points and over Fusion_Untuned by 0.17 points, while the untuned fusion retains the highest accuracy. Class-4 recall increases from 63.51% to 69.19% relative to Baseline, with trade-offs in other classes. The evidence supports a small retrospective improvement in the primary macro-F1 endpoint, rather than uniform superiority or an isolated benefit of individual descriptors. Missing winning-model records, one seed, prior test inspection and absent geographic identifiers limit exact reproduction and independent confirmation.

Keywords: Landsat land-cover classification; Random Forest; spectral statistics; spatial gradients; feature fusion

1. Introduction

The growing demand for defensible land-cover information has made classification of multispectral satellite measurements an important research problem [1], [2]. Reviews of mapping products, classification architectures and training-data design show the breadth of available approaches and the dependence of reported performance on the input representation and evaluation setting [1], [2], [3], [4]. Landsat feature-relationship models and vegetation-index or texture-based studies illustrate how derived predictors can expose distinctions that are less accessible from individual spectral values [5], [6], [7]. Ensemble-learning research provides a methodological basis for studying Random Forest within this context [8]. Applications to mangrove mapping and Landsat time series further show that an ensemble classifier is part of a larger representation-and-validation pipeline [9], [10]. These contributions motivate an explicit comparison of raw and engineered patch inputs, rather than assuming that adding descriptors always improves every class. The present experiment uses archived 3 × 3 multispectral patch records. It compares a fixed raw-feature Random Forest, a fixed fusion model and a selected model under the same official UCI partition.

The methodological challenge is to distinguish feature construction from configuration selection and final assessment. Automated land-cover mapping and Landsat feature-selection studies use different combinations of

spectral, textural, ancillary and selected predictors [11], [12]; general feature-engineering and selection reviews likewise emphasize that the input definition is part of the learning procedure [13], [14] . Hyperparameter-search and automated-learning research requires an explicit parameter domain, objective, validation scheme and computational budget [15], [16], [17]. Explainable multisensor work provides a richer remote-sensing example, but does not verify this Statlog experiment or supply its missing model records [18]. Leakage and reproducibility studies warn that a development score is not an independent estimate of generalization [19], [20], [21], [22]. Benchmarking and validation research also makes the comparator and estimand material to interpreting a small difference [23], [24], [25]. These concerns are especially relevant when local spectral patches are available but scene identifiers are insufficient for reconstructing geographic blocks.

This study examines whether deterministic neighborhood statistics and spatial gradients, together with Random Forest configuration selection, change aggregate and class-wise classification performance. Spatial-validation studies caution that random or stratified partitions cannot establish transfer to independent geographic scenes [26], [27], [28], [29], [30]. Metric guidance further motivates reporting macro-F1, per-class recall and confusion patterns alongside overall accuracy [31], [32], [33], [34]. The available evaluation contains three models: Baseline, Fusion_Untuned and Optimized. The selection code confines its objective to the official training records, and the comparison reports the stored official-test outputs. The contribution is a traceable retrospective configuration comparison with explicit limitations, rather than a demonstrated decomposition of all feature and optimization effects. Internal learning curves are treated as diagnostics of the development data, consistent with the distinction between learning-curve analysis and independent validation [25], [35]. Calibration and deployment quality require additional evidence [36], [37], [38].

2. Method

2.1. Related Work

Recent reviews describe land-cover classification as a joint problem of representation, training observations and evaluation scope [1], [2], [3], [4]. Feature-relationship learning with Landsat imagery offers a direct example of representation-driven modeling [5]. These studies establish the remote-sensing context, but their map-level results are not performance estimates for Statlog. The present observations are archived local spectral patches, without a reconstructed scene map, and the target is the central-pixel class.

Vegetation indices, texture features and spectral–spatial representations have been investigated to improve class separation [6], [7]. Optical–radar fusion explores complementary sensor information under a different observation setting [39]. General feature-engineering and feature-selection reviews distinguish transformed predictors from changes to the learner or sampling design [13], [14]. In this experiment, fusion refers to deterministic summaries of one supplied 3 × 3 × 4 patch. It does not introduce a second sensor, new acquisition dates, GLCM textures, learned convolutional filters or geographic coordinates.

Zhang and colleagues review ensemble-learning algorithms across remote-sensing applications [8]. Purwanto and colleagues compare decision-tree and Random Forest approaches for mangrove mapping using spectral and ancillary inputs [9]. A recent ensemble framework further studies Landsat time-series land-cover classification [10]. These studies motivate examining both the forest configuration and the predictors supplied to it. Their tasks, imagery and evaluation units differ from the fixed Statlog benchmark, so their accuracy values cannot establish that one patch descriptor is preferable in the current comparison.

Automated land-cover mapping with Sentinel-2 combines spectral, textural and distance functions with ancillary information for training-data generation and Random Forest classification [11]. The Jiangsu coastal study uses Pearson correlation, SHAP and recursive feature elimination to examine a Landsat predictor set [12]. These are examples of defining and evaluating a representation, rather than evidence that the current descriptors are explanatory or that their importance is stable. No feature-importance stability analysis, SHAP explanation or recursive elimination is reported in the supplied Random Forest evaluation.

Bischl and colleagues systematize search objectives, algorithms, budgets and evaluation practice [15]. Automated-learning reviews, satellite-imagery optimization studies and empirical work on classification-tree tuning make configuration choices explicit parts of the comparison [16], [17], [40]. Explainable multisensor land-cover research illustrates additional choices outside this benchmark [18]. Multi-objective optimization studies also consider competing predictive, fairness and energy objectives [41], [42]. Here, the implemented search objective is

mean training-CV macro-F1. Runtime, energy and memory are not measured endpoints, and the three recorded models do not receive equal search budgets.

Leakage and common-pitfall studies describe how evaluation information can enter preprocessing or model selection and make a reported result difficult to interpret [19], [20], [21]. Reproducibility recommendations require sufficient records of the data, parameter choices and environment to reconstruct the final comparison [22]. Benchmark-selection and method-comparison research further warns that conclusions depend on the chosen tasks and comparators [23], [24]. These concerns motivate distinguishing a search specification from the particular winning configuration. The archived test summaries alone do not recover the selected feature type or complete Random Forest parameter vector.

Spatial+ and research on clustered samples distinguish geographic validation from ordinary random or stratified resampling [26], [27]. Studies of spatial block design, overly favorable remote-sensing accuracy assessment and geographically weighted error show why a single aggregate score does not describe all locations [28], [29], [30]. The code uses three shuffled stratified folds of the official training records, not geographic blocks. Although the records contain local 3 × 3 neighborhoods, their coordinates and scene identities are not available to reconstruct a new spatial holdout. Geographic transfer therefore remains outside the tested claim.

Multiclass assessment guidance distinguishes overall accuracy, macro averages, support-weighted summaries and class-level errors [31], [32]. Image-validation studies explain how an aggregate endpoint can conceal failures or create misleading comparisons [33], [34]. These contributions motivate retaining all three recorded Random Forest models and examining recall and F1 for each class. The selected objective is macro-F1, so a higher value need not coincide with the highest accuracy or weighted F1. Confusion matrices are used to describe the observed error redistribution, without inventing an edge-specific endpoint for which no boundary labels are supplied.

Probability calibration is distinct from correct class assignment [36], and class-conditional conformal mapping provides another approach to reliability in land-cover classification [37]. Deployment research documents costs and constraints beyond an accuracy table [38]. These studies define additional questions that are not answered by the present predictions. No calibrated probabilities, confidence intervals, inference latency, memory use or operational deployment assessment is reported. A forest vote or favorable class score is therefore not treated as verified confidence quality or evidence of deployment readiness.

Learning-curve research considers how predictive behavior changes with the number of training examples [35]; cross-validation theory clarifies that a reused development partition is not independent replication [25]. The recorded curves refit Baseline and Optimized at three fitting sizes within the official training data. They complement the test summaries but do not estimate variability over new seeds, scenes or complete search repetitions. Taken together, the literature supports explicit representation definitions and bounded model comparisons. The remaining gap here is a fully reproducible, component-matched assessment with archived winning settings and independent geographic confirmation.

2.2. Dataset

The experiment uses the UCI Statlog Landsat Satellite benchmark. The notebook loads 4,435 observations from sat.trn and 2,000 from sat.tst, retaining the 36 numerical input attributes. Each row represents four spectral channels at nine positions of a 3 × 3 patch, with the central-pixel label. The observed classes are 1, 2, 3, 4, 5 and 7. Table 1 specifies the data and the implemented input boundary.

Table 1. Dataset and official experimental partition

Dataset attribute	Study-specific protocol
Public source	UCI Statlog Landsat Satellite: official sat.trn and sat.tst
Research endpoint	Six classes: 1, 2, 3, 4, 5 and 7
Analysis unit	One 3 × 3 × 4 patch record; 36 raw attributes and the central-pixel class
Data partitions	4,435 official training / 2,000 official test; training-only 3-fold stratified CV
Feature construction	Deterministic row-local statistics and gradients; full representation has 60 attributes
Scope constraint	No coordinates or reconstructed scene; no claim of an independent geographic-block test

The code reshapes each 36-value row to a 3 × 3 × 4 array and computes features separately within that observation. This construction does not estimate a scaler, imputer or global feature-selection statistic from the test data. The same official test records are used for all three model summaries. Coordinates and scene identifiers

required to reconstruct independent geographic blocks are not provided to the workflow. Table 2 records the implemented development-to-test flow. Configuration selection uses three fixed StratifiedKFold splits of the 4,435 training records, with shuffle=True and random_state=42. Each candidate is fitted on two folds and scored on the third, and the objective is their mean macro-F1. The evaluation function refits the selected model on all official training records before computing the stored official-test summaries.

Table 2. Experimental data flow and evaluation controls

Stage	Required handling
Stage A: input	Load the official files and retain the 36 raw values and six represented labels
Stage B: development	Use 3 fixed shuffled stratified folds of sat.trn with seed 42
Stage C: selection	Evaluate feature type and RF settings using mean training-CV macro-F1
Stage D: refit	Load the selected record and refit each final model on all 4,435 training cases
Stage E: report	Report the stored official-test outputs; identify the prior-test-inspection limitation

The code keeps official-test scores outside the search objective. However, the notebook notes prior inspection of historical test data, so the present comparison is retrospective and does not constitute a newly untouched confirmatory test. Reported endpoints are accuracy, macro-F1, weighted F1, balanced accuracy, class metrics, confusion matrices and internal learning curves. Edge-cover accuracy, feature-importance stability, calibration and inference cost were not evaluated.

2.3. Research Framework

Figure 1 presents the three recorded configurations under the same official partition. Baseline uses 36 raw attributes and a RandomForestClassifier with n_estimators=140, n_jobs=2 and random_state=42. Fusion_Untuned uses the same explicit forest settings with the 60-dimensional full representation. Optimized uses the feature type and parameter vector selected by the training-CV search and loaded from rf_best.json. The winning values are not embedded in the archived notebook.

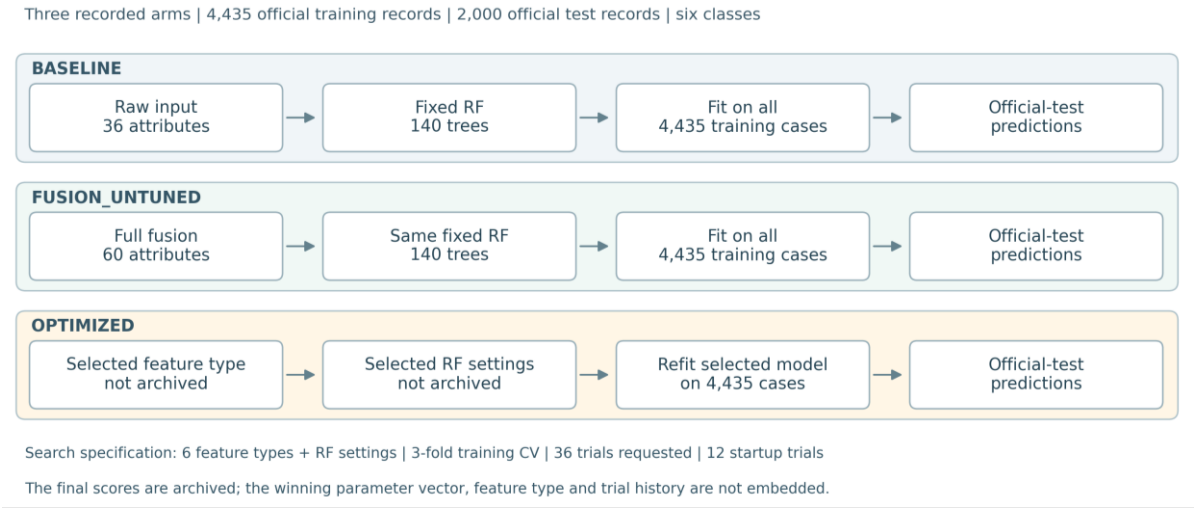


Figure 1. Recorded raw, untuned-fusion and selected Random Forest configurations

The raw-versus-untuned-fusion contrast changes the input representation while retaining the same explicit forest settings. The selected model can change both the representation and forest parameters because raw is one of the six search candidates. Its exact feature type and final dimension cannot be inferred from its Optimized label. The comparison therefore does not isolate a pure tuning effect or establish that the selected model necessarily uses every fused descriptor. Table 3 summarizes the comparison that is actually available. Statistics-only, gradients-only, a separately tuned raw-only forest and equal-budget searches for each fixed representation are not reported. Without these controls, the aggregate result cannot establish a synergistic interaction between statistics, gradients and optimization.

Table 3. Recorded configurations and limits of the comparison

Component	Comparative specification
Baseline	Raw 36 attributes; RF with 140 trees, n_jobs=2 and seed 42
Fusion_Untuned	Full 60-attribute representation; same explicit 140-tree RF settings
Controlled factors	Same official partition, class labels, seed and reported test metrics
Optimized / missing record	Training-selected feature type and RF settings; exact winner fields not archived
Comparison scope	No statistics-only, gradients-only, tuned raw-only or equal-budget component study

Table 3 supports two descriptive contrasts: adding the full descriptor block without tuning, and comparing the recorded selected model with the fixed alternatives. Because the second contrast can alter several design choices simultaneously, its score difference is a result for the combined selected configuration. It is not an independently measured contribution of TPE, one gradient definition or one statistical summary.

2.4. Proposed Method

The feature generator computes four-channel neighborhood mean, population standard deviation, range, center-versus-surround contrast and horizontal/vertical differences from each 3 × 3 patch. The full representation appends six four-value blocks to the original 36 values, giving 60 attributes. Baseline retains 36. Figure 2 shows the feature construction and training-only search boundary.

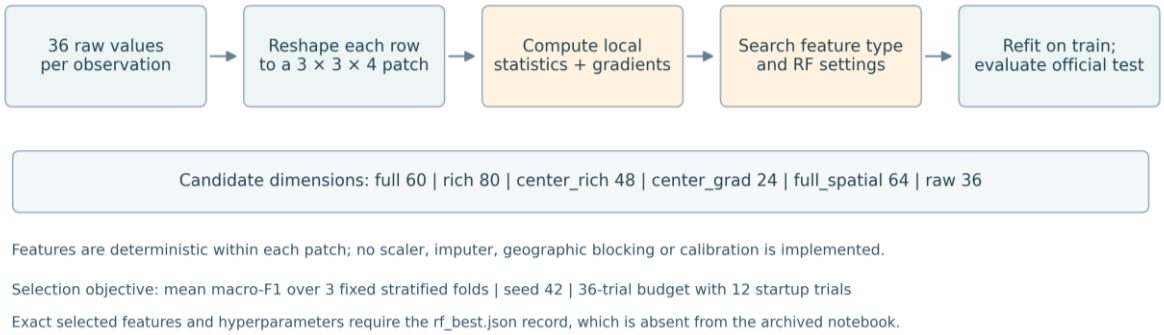


Figure 2. Deterministic patch features and training-only configuration selection

For channel b, mean and standard deviation summarize the nine patch values, and range is their maximum minus minimum. Center-versus-surround contrast is the central value minus the mean of the other eight positions. The horizontal difference is the mean of the right column minus the left column, and the vertical difference is the mean of the bottom row minus the top row. Thus full concatenates raw36, mean4, std4, range4, contrast4, dx4 and dy4. These differences are local numeric descriptors, not measured classification accuracy at a geographic boundary.

For one observation, let X_{rcb} be the patch value at row r, column c and band b, with $r, c \in \{1,2,3\}$ and $b \in \{1,2,3,4\}$. The central band value is $c_b = X_{22b}$. Equation (1) defines the mean, population standard deviation and range computed over the nine values in each band.

$$\begin{aligned} \mu_b &= \tfrac{1}{9}\textstyle\sum_{r=1}^{3}\sum_{c=1}^{3} X_{rcb}\,, \\ \sigma_b &= \sqrt{\tfrac{1}{9}\textstyle\sum_{r=1}^{3}\sum_{c=1}^{3}(X_{rcb}-\mu_b)^2}\,, \\ R_b &= \max_{r,c} X_{rcb} - \min_{r,c} X_{rcb}. \end{aligned} \tag{1}$$

Let $\mathcal{N} = \{(r,c): r,c \in \{1,2,3\}, (r,c) \neq (2,2)\}$ denote the eight noncentral positions. The center-versus-surround contrast Δ_b, horizontal difference G_b^h and vertical difference G_b^v are given in (2). Their signs match the right-minus-left and bottom-minus-top operations in the feature generator.

$$\begin{aligned} \Delta_b &= c_b - \tfrac{1}{8}\textstyle\sum_{(r,c)\in\mathcal{N}} X_{rcb}\,, \\ G_b^h &= \tfrac{1}{3}\textstyle\sum_{r=1}^{3}(X_{r3b} - X_{r1b})\,, \\ G_b^v &= \tfrac{1}{3}\textstyle\sum_{c=1}^{3}(X_{3cb} - X_{1cb})\,. \end{aligned} \tag{2}$$

The additional descriptors available to the richer candidates are specified in (3). Here M_b is the band median, D_b^h and D_b^v are central-row and central-column differences, V_b is the sum of absolute directional differences, and A_b is

the center-minus-mean value. In particular, V_b corresponds to the code variable named variances and is not a statistical variance.

$$\begin{aligned} M_b &= \text{median}_{r,c} X_{rcb}, \\ D_b^h &= X_{23b} - X_{21b}, \qquad D_b^v = X_{32b} - X_{12b}, \\ V_b &= \left|G_b^h\right| + |G_b^v|, \qquad A_b = c_b - \mu_b. \end{aligned} \tag{3}$$

Writing each four-band descriptor as a column vector gives the 44-value rich descriptor block in (4). The raw patch vector x has 36 entries, and $c = [c_1, c_2, c_3, c_4]^T$ contains only the four central values.

$$\psi(x) = [\mu^T, \sigma^T, R^T, \Delta^T, (G^h)^T, (G^v)^T, M^T, (D^h)^T, (D^v)^T, V^T, A^T]^T \in \mathbb{R}^{44}. \tag{4}$$

The search additionally considers rich (80 attributes), center_rich (48), center_grad (24), full_spatial (64) and raw (36). The rich block adds median values, central-row and central-column differences, the sum of absolute horizontal and vertical differences, and center-minus-mean values. The code variable named variances is the sum |dx|+|dy|, rather than a statistical variance. Table 4 records the feature choices and forest search domain.

The six eligible representations are written explicitly in (5), preserving the feature order used by the notebook. The full representation contains 60 inputs for Fusion_Untuned, while the selected model may use any eligible representation. The symbol ν denotes the feature-type choice; its winning value cannot be recovered from the stored test summaries.

$$\begin{aligned} \phi_{\text{raw}}(x) &= x \in \mathbb{R}^{36}, \\ \phi_{\text{full}}(x) &= [x^T, \mu^T, \sigma^T, R^T, \Delta^T, (G^h)^T, (G^v)^T]^T \in \mathbb{R}^{60}, \\ \phi_{\text{rich}}(x) &= [x^T, \psi(x)^T]^T \in \mathbb{R}^{80}, \\ \phi_{\text{center_rich}}(x) &= [c^T, \psi(x)^T]^T \in \mathbb{R}^{48}, \\ \phi_{\text{center_grad}}(x) &= [c^T, \mu^T, \sigma^T, \Delta^T, (G^h)^T, (G^v)^T]^T \in \mathbb{R}^{24}, \\ \phi_{\text{full_spatial}}(x) &= [x^T, \mu^T, \sigma^T, \Delta^T, (G^h)^T, (G^v)^T, M^T, V^T]^T \in \mathbb{R}^{64}. \end{aligned} \tag{5}$$

For tree u, let ω_{iu} be the effective nonnegative weight of fitting observation i, incorporating its bootstrap multiplicity and any class weight. For a node t, let $\mathcal{I}_{ut}$ contain its fitting indices and let $U_{ut} = \sum_{i \in \mathcal{I}_{ut}} \omega_{iu}$. The weighted class fraction and default Gini impurity are given in (6), for the six represented classes $\mathcal{L} = \{1,2,3,4,5,7\}$. The indicator $\mathbf{1}[P]$ is one when P is true and zero otherwise. Uniform class weights apply when class_weight is None; balanced weights follow fitting-class frequencies, with balanced_subsample using each tree's bootstrap sample.

$$\begin{aligned} p_{ukt} &= \frac{\sum_{i \in \mathcal{I}_{ut}} \omega_{iu} \mathbf{1}[y_i = k]}{U_{ut}}, \qquad k \in \mathcal{L}, \\ G_u(t) &= 1 - \sum_{k \in \mathcal{L}} p_{ukt}^2 . \end{aligned} \tag{6}$$

If a candidate split creates children t_L and t_R, its weighted impurity decrease is (7), where U_{u0} is the total effective fitting weight at the tree root. The tree greedily selects admissible splits with the largest decrease among the considered features, subject to the configured depth and minimum-sample constraints. The max_features setting controls random feature subsampling, and bootstrap controls whether observations are sampled with replacement for each tree.

$$D_u(t) = \frac{U_{ut}}{U_{u0}} \left[G_u(t) - \frac{U_{utL}}{U_{ut}} G_u(t_L) - \frac{U_{utR}}{U_{ut}} G_u(t_R) \right]. \tag{7}$$

For a forest with B trees, let $t_u\left(\phi_\nu(x)\right)$ be the leaf reached by the transformed observation in tree u. Equation (8) averages the leaf class fractions and predicts the class with the highest mean probability estimate. This is the implemented forest decision rule; the resulting scores are not treated as externally calibrated probabilities.

$$\begin{aligned} \hat{p}_k(x) &= \frac{1}{B} \sum_{u=1}^{B} p_{u,k,t_u(\phi_\nu(x))} , \\ \hat{y}(x) &= \arg\max_{k \in \mathcal{L}} \hat{p}_k(x). \end{aligned} \tag{8}$$

Table 4. Configured feature and Random Forest search space

Design element	Operational definition
Feature representation	full (60); rich (80); center_rich (48); center_grad (24); full_spatial (64); raw (36)
n_estimators	Categorical: 100, 180 or 280 trees
max_features	Continuous: 0.16–0.88
Minimum samples	min_samples_leaf: integer 1–4; min_samples_split: integer 2–9
max_depth	Categorical: None, 14, 22 or 32
Bootstrap / class weight	bootstrap: True or False; class_weight: None or balanced_subsample, with balanced used when bootstrap=False
Objective and budget	Mean macro-F1 over 3 stratified folds; 36 requested trials, 12 startup trials, seed 42

Optuna is configured to maximize mean three-fold training macro-F1 through TPESampler(seed=42, n_startup_trials=12) with n_trials=36. This budget permits adaptive proposals after the startup phase if the requested trials complete; the trial history is not embedded, so completed trial counts and the winning CV score are not reconstructed. If bootstrap=False and balanced_subsample is suggested, the code changes the effective class_weight to balanced. Deterministic descriptors are computed row by row, and no scaling, imputation, oversampling, calibration or geographic blocking is executed.

For any validation fold, the macro-F1 criterion is the unweighted six-class mean in (9). Here TP_k, FP_k and FN_k are the true-positive, false-positive and false-negative counts for class k within that fold. The class weights used to fit a candidate forest do not change this unweighted macro-average evaluation rule.

$$\mathrm{MacroF1} = \frac{1}{6}\sum_{k\in\mathcal{L}} \frac{2\mathrm{TP}_k}{2\mathrm{TP}_k+\mathrm{FP}_k+\mathrm{FN}_k}. \tag{9}$$

Let $\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3$ be the three fixed stratified folds of the official training records. For feature type ν and forest settings θ, $\hat{y}_{\mathcal{I}_f}^{(-f;\nu,\theta)}$ denotes predictions on fold f from a forest fitted only on the other two folds. The objective in (10) is the arithmetic mean of the three fold-specific macro-F1 values, rather than a score computed from the official test set.

$$J(\nu,\theta) = \frac{1}{3}\sum_{f=1}^{3} \mathrm{MacroF1}\left(y_{\mathcal{I}_f}, \hat{y}_{\mathcal{I}_f}^{(-f;\nu,\theta)}\right). \tag{10}$$

Let $\mathcal{H}_{\mathrm{done}}$ contain the candidate configurations for which the search completes evaluation. Equation (11) formalizes selection among those evaluated candidates. The configured budget is 36 trials with 12 random startup trials; the archive does not establish how many completed or identify the winning pair. Once selected, the feature type and effective forest parameters are fixed before refitting on all official training records.

$$(\nu^*, \theta^*) = \underset{(\nu,\theta)\in\mathcal{H}_{\mathrm{done}}}{\arg\max}\ J(\nu,\theta). \tag{11}$$

Implementation and reproducibility. The final evaluation reads the selected feature type and forest parameters from rf_best.json. That file and rf_trials.csv are not embedded in the supplied notebook, and the optimization cell has no saved search output. Consequently, the exact selected input, parameter vector and CV score are unresolved. The learning curves use 1,164, 2,161 and 3,326 fitting records and a fixed 1,109-record validation subset from the official training set, with selected settings held fixed. This subset is reused within data already involved in configuration selection. The recorded workflow uses seed 42 and n_jobs=2; software versions, hardware, repeated-seed results, uncertainty intervals and full raw-data provenance are not archived.

3. Results and Discussion

3.1. Overall Classification Performance

The stored evaluation compares three Random Forest configurations on the same 2,000 official test observations after fitting on the 4,435 official training records. Table 5 reports accuracy, macro-F1, weighted F1 and balanced accuracy. The selected development objective is macro-F1, which is distinct from optimizing the number of correct test predictions.

Table 5. Recorded official-test performance of the three Random Forest configurations

Model	Accuracy (%)	Macro F1 (%)	Weighted F1 (%)	Balanced acc. (%)
Baseline	90.85	89.37	90.66	88.91
Fusion_Untuned	91.40	89.77	91.12	89.19
Optimized	91.20	89.94	91.10	89.64

Table 5 shows that Optimized has the highest macro-F1, 89.94%, compared with 89.37% for Baseline and 89.77% for Fusion_Untuned. Calculated from the unrounded confusion-matrix metrics, the gains are 0.568374 and 0.171234 percentage points, rounded to 0.57 and 0.17 points. Accuracy increases from 90.85% for Baseline to 91.20% for Optimized, but Fusion_Untuned reaches 91.40%. The three models correctly classify 1,817, 1,828 and 1,824 test observations, respectively. Thus, Optimized improves the selected macro-average endpoint while making four fewer correct predictions than the untuned fusion model.

Balanced accuracy is also highest for Optimized at 89.64%, whereas weighted F1 is slightly higher for Fusion_Untuned, 91.12% versus 91.10%. These descriptive differences do not establish statistical significance, optimizer superiority or an independent contribution from every fused feature. The winning representation and parameter vector are unresolved in the archived records, so the third arm is reported using its stored Optimized label.

3.2. Class-Level Errors and Confusion Matrices

Table 6 compares recall and F1 for each represented class. These values are reconstructed from the stored confusion-matrix counts and agree with the rounded classification reports. Class 4 remains the weakest class, making its behavior important to the macro-F1 interpretation.

Table 6. Per-class recall and F1 for the three recorded Random Forest models

Class	Support	Base R	Base F1	Fusion R	Fusion F1	Opt. R	Opt. F1
1	461	0.9913	0.9870	0.9935	0.9849	0.9913	0.9870
2	224	0.9688	0.9753	0.9732	0.9732	0.9643	0.9730
3	397	0.9471	0.9148	0.9547	0.9199	0.9345	0.9115
4	211	0.6351	0.6943	0.6161	0.6989	0.6919	0.7300
5	237	0.8987	0.9083	0.8987	0.9122	0.9030	0.9030
7	470	0.8936	0.8824	0.9149	0.8968	0.8936	0.8917

Class-4 recall is 0.6351 for Baseline, 0.6161 for Fusion_Untuned and 0.6919 for Optimized; its F1 increases to 0.7300 in the selected model. This improvement coexists with lower class-2 and class-3 recall relative to Baseline. Class-5 F1 is lower for Optimized than for either fixed alternative, while its class-7 recall equals Baseline and falls below Fusion_Untuned. The results therefore describe a redistribution of errors rather than a uniform class-wise improvement.

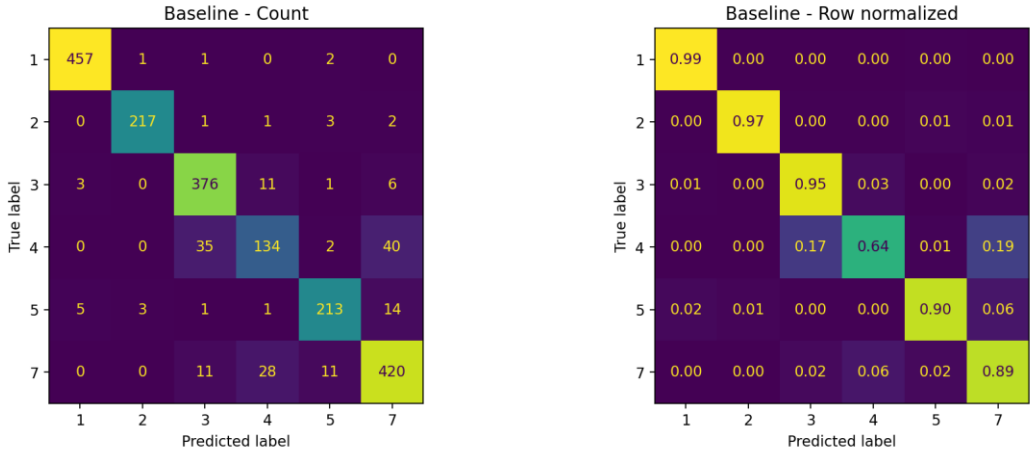


Figure 3. Baseline confusion matrix with counts and row-normalized values

Figure 3 shows that Baseline correctly predicts 134 of 211 class-4 observations. Its main class-4 errors are assignments to class 7 (40 cases) and class 3 (35 cases). Class 7 has 420 correct predictions among 470 observations.

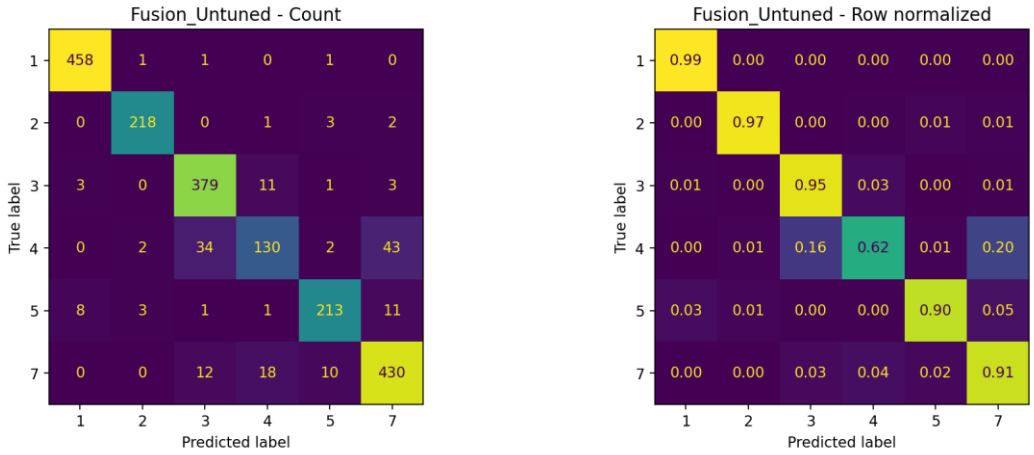


Figure 4. Fusion_Untuned confusion matrix with counts and row-normalized values

Figure 4 shows that untuned fusion increases the total number of correct predictions to 1,828. However, correct class-4 predictions fall from 134 to 130, while correct class-7 predictions rise from 420 to 430. The higher overall accuracy therefore does not imply better recall for the weakest class.

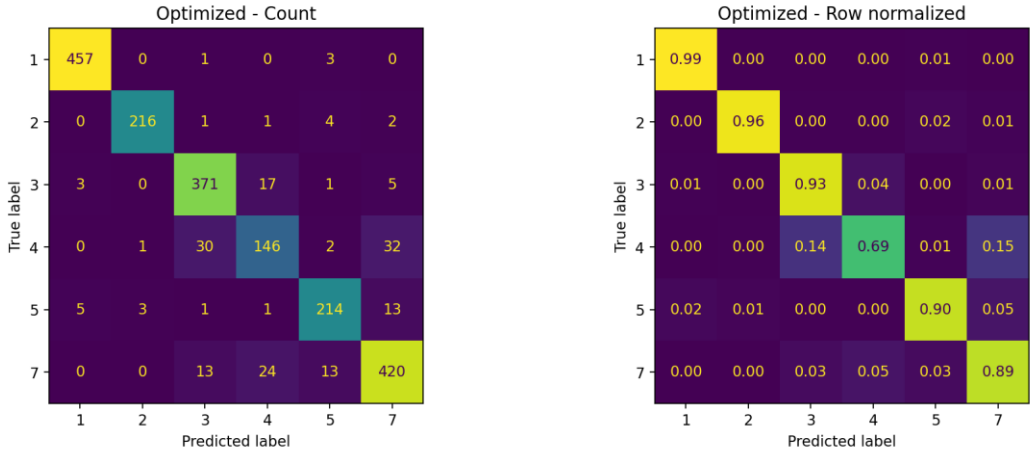


Figure 5. Optimized confusion matrix with counts and row-normalized values

Figure 5 shows 146 correct class-4 predictions, twelve more than Baseline and sixteen more than Fusion_Untuned. Class-4 assignments to class 7 fall to 32. Conversely, class-3 correct predictions decrease to 371, compared with 376 and 379 in the fixed models, and class-7 correct predictions return to 420. These class-specific changes account for the different rankings by accuracy and macro-F1.

3.3. Internal Learning Curves

Figure 6 contains the recorded training and validation curves for Baseline and Optimized. The fitting subsets contain 1,164, 2,161 and 3,326 observations, with the same 1,109-record internal validation set. The selected settings are fixed during these refits, and Fusion_Untuned is not included in this curve. The validation curves rise as fitting size increases. Optimized is slightly below Baseline at the smallest recorded size and above it at the two larger sizes. Near-perfect training scores coexist with lower validation performance. These are resubstitution and reused-development diagnostics, not fresh test results or repeated-seed estimates. Because model selection already uses the official training records, this curve is not an independent confirmation of the selected configuration.

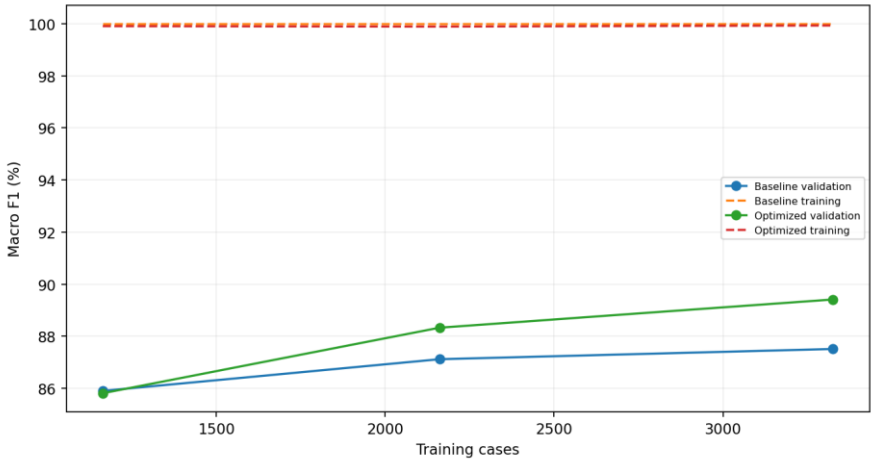


Figure 6. Recorded learning curves within the official training partition

3.4. Discussion and Limitations

The recorded evidence supports a modest macro-F1 improvement for Optimized relative to the fixed alternatives. It also supports an accuracy improvement from adding the full descriptor block without tuning. These contrasts do not identify a separate benefit of statistics, gradients or TPE. No statistics-only, gradients-only, separately tuned raw-only or equal-budget representation study is reported, and raw is eligible in the joint search. Consequently, the title does not establish which input representation was actually selected.

The missing winning feature type, final parameter vector, CV score and trial history prevent exact reproduction of the selected model from the archived outputs alone. The 36-trial, 12-startup budget is a code specification rather than a verified completed search log. Other limitations are one seed, stratified rather than geographic validation, missing environment details, prior historical-test inspection and absent uncertainty intervals. Edge-cover performance, feature-importance stability, calibration and inference cost were not measured. An independent geographic test and complete search records are needed before making confirmatory, attribution or deployment claims.

4. Conclusion

This study compared raw, untuned-fusion and selected Random Forest configurations on the official Statlog Landsat split. The full fusion representation adds local statistics and gradients to the original 36 attributes, yielding 60 inputs for the fixed fusion model. Stored test macro-F1 is 89.37%, 89.77% and 89.94% for Baseline, Fusion_Untuned and Optimized. Optimized gains 0.57 percentage points over Baseline and improves class-4 recall from 63.51% to 69.19%. Fusion_Untuned nevertheless retains the highest accuracy, 91.40%, versus 91.20% for Optimized. The comparison therefore supports a small aggregate macro-F1 gain with class-wise trade-offs. The selected feature type and parameter vector remain unrecorded in the archived notebook, and individual descriptor or optimizer contributions are not isolated. Further work should preserve complete winning-model and trial records, compare components under equal search budgets, repeat seeds and evaluate independently collected geographic scenes.

References

[1] Y. Wang, Y. Sun, X. Cao, Y. Wang, W. Zhang, and X. Cheng, “A review of regional and Global scale Land Use/Land Cover (LULC) mapping products generated from satellite remote sensing,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 206, pp. 311–334, 2023, doi: 10.1016/j.isprsjprs.2023.11.014.

[2] S. Zhao, K. Tu, S. Ye, H. Tang, Y. Hu, and C. Xie, “Land Use and Land Cover Classification Meets Deep Learning: A Review,” *Sensors*, vol. 23, no. 21, p. 8966, 2023, doi: 10.3390/s23218966.

[3] D. Moraes, M. L. Campagnolo, and M. Caetano, “Training data in satellite image classification for land cover mapping: a review,” *Eur. J. Remote Sens.*, vol. 57, no. 1, p. 2341414, 2024, doi: 10.1080/22797254.2024.2341414.

[4] I. Kotaridis and M. Lazaridou, “CNNs in land cover mapping with remote sensing imagery: a review and meta-analysis,” *Int. J. Remote Sens.*, vol. 44, no. 19, pp. 5896–5935, 2023, doi: 10.1080/01431161.2023.2255354.

[5] P. Dou, H. Shen, C. Huang, Z. Li, Y. Mao, and X. Li, "Large-scale land use/land cover extraction from Landsat imagery using feature relationships matrix based deep-shallow learning," *International Journal of Applied Earth Observation and Geoinformation*, vol. 129, p. 103866, 2024, doi: 10.1016/j.jag.2024.103866.

[6] Y. Maulani and K. Surendro, "Detailed land use classification model based on vegetation indices and texture features," *Remote Sens. Appl.*, vol. 40, p. 101786, 2025, doi: 10.1016/j.rsase.2025.101786.

[7] V. Sangeetha and L. Agilandeeswari, "Artificial intelligence enabled spectral-spatial feature extraction techniques for land use and land cover classification using hyperspectral images – An inclusive review," *The Egyptian Journal of Remote Sensing and Space Sciences*, vol. 28, no. 3, pp. 455–467, 2025, doi: 10.1016/j.ejrs.2025.06.004.

[8] Y. Zhang, J. Liu, and W. Shen, "A Review of Ensemble Learning Algorithms Used in Remote Sensing Applications," *Applied Sciences*, vol. 12, no. 17, p. 8654, 2022, doi: 10.3390/app12178654.

[9] A. D. Purwanto, K. Wikantika, A. Deliar, and S. Darmawan, "Decision Tree and Random Forest Classification Algorithms for Mangrove Forest Mapping in Sembilang National Park, Indonesia," *Remote Sens. (Basel).*, vol. 15, no. 1, p. 16, 2023, doi: 10.3390/rs15010016.

[10] E. Mensah, X. Liu, Z. Zhang, and G. Campbell, "A big data ensemble machine learning framework for robust land use/land cover classification with landsat time-series," *Remote Sens. Appl.*, vol. 42, p. 102095, 2026, doi: 10.1016/j.rsase.2026.102095.

[11] K. Lasko, F. D. O'Neill, and E. Sava, "Automated Mapping of Land Cover Type within International Heterogenous Landscapes Using Sentinel-2 Imagery with Ancillary Geospatial Data," *Sensors*, vol. 24, no. 5, p. 1587, 2024, doi: 10.3390/s24051587.

[12] S. Wang, W. Pang, and S. Jin, "Land Use/Land Cover Classification and Its Variability Along the Jiangsu Coast Based on Pearson-SHAP-RFE Feature Selection from Landsat Imagery," *Sensors*, vol. 26, no. 18, p. 5915, 2026, doi: 10.3390/s26185915.

[13] P. Koukaras and C. Tjortjis, "Data Preprocessing and Feature Engineering for Data Mining: Techniques, Tools, and Best Practices," *AI*, vol. 6, no. 10, p. 257, 2025, doi: 10.3390/ai6100257.

[14] M. R. Islam, A. A. Lima, S. C. Das, M. F. Mridha, A. R. Prodeep, and Y. Watanobe, "A Comprehensive Survey on the Process, Methods, Evaluation, and Challenges of Feature Selection," *IEEE Access*, vol. 10, pp. 99595–99632, 2022, doi: 10.1109/access.2022.3205618.

[15] B. Bischl *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *WIREs Data Mining and Knowledge Discovery*, vol. 13, no. 2, p. e1484, 2023, doi: 10.1002/widm.1484.

[16] M. Baratchi *et al.*, "Automated machine learning: past, present and future," *Artif. Intell. Rev.*, vol. 57, no. 5, p. 122, 2024, doi: 10.1007/s10462-024-10726-1.

[17] F. Alonso-Sarría, C. Valdivieso-Ros, and F. Gomariz-Castillo, "Analysis of the hyperparameter optimisation of four machine learning satellite imagery classification methods," *Comput. Geosci.*, vol. 28, no. 3, pp. 551–571, 2024, doi: 10.1007/s10596-024-10285-y.

[18] H. İ. Gündüz, O. Orhan, S. N. G. Hamal, and S. Ekercin, "An Explainable and Optimized Machine Learning Framework for Multisensor Remote Sensing-Based Land Use and Land Cover Classification," *Transactions in GIS*, vol. 30, no. 3, p. e70293, 2026, doi: 10.1111/tgis.70293.

[19] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, p. 100804, 2023, doi: 10.1016/j.patter.2023.100804.

[20] M. A. Lones, "Avoiding common machine learning pitfalls," *Patterns*, vol. 5, no. 10, p. 101046, 2024, doi: 10.1016/j.patter.2024.101046.

[21] A. Apicella, F. Isgrò, and R. Prevete, "Don't push the button! Exploring data leakage risks in machine learning and transfer learning," *Artif. Intell. Rev.*, vol. 58, no. 11, p. 339, 2025, doi: 10.1007/s10462-025-11326-3.

[22] S. Kapoor *et al.*, "REFORMS: Consensus-based Recommendations for Machine-learning-based Science," *Sci. Adv.*, vol. 10, no. 18, p. eadk3452, 2024, doi: 10.1126/sciadv.adk3452.

[23] S. Friedrich and T. Friede, "On the role of benchmarking data sets and simulations in method comparison studies," *Biometrical Journal*, vol. 66, no. 1, p. 2200212, 2024, doi: 10.1002/bimj.202200212.

[24] J. L. J. Pereira, K. Smith-Miles, M. A. Muñoz, and A. C. Lorena, "Optimal selection of benchmarking datasets for unbiased machine learning algorithm evaluation," *Data Min. Knowl. Discov.*, vol. 38, no. 2, pp. 461–500, 2024, doi: 10.1007/s10618-023-00957-1.

[25] S. Bates, T. Hastie, and R. Tibshirani, "Cross-Validation: What Does It Estimate and How Well Does It Do It?," *J. Am. Stat. Assoc.*, vol. 119, no. 546, pp. 1434–1445, 2024, doi: 10.1080/01621459.2023.2197686.

[26] Y. Wang, M. Khodadadzadeh, and R. Zurita-Milla, "Spatial+: A new cross-validation method to evaluate geospatial machine learning models," *International Journal of Applied Earth Observation and Geoinformation*, vol. 121, p. 103364, 2023, doi: 10.1016/j.jag.2023.103364.

[27] S. de Bruin, D. J. Brus, G. B. M. Heuvelink, T. van Ebbenhorst Tengbergen, and A. M. J.-C. Wadoux, "Dealing with clustered samples for assessing map accuracy by cross-validation," *Ecol. Inform.*, vol. 69, p. 101665, 2022, doi: 10.1016/j.ecoinf.2022.101665.

[28] A. Stock, "Choosing blocks for spatial cross-validation: lessons from a marine remote sensing case study," *Frontiers in Remote Sensing*, vol. 6, p. 1531097, 2025, doi: 10.3389/frsen.2025.1531097.

[29] D. Abriha, P. K. Srivastava, and S. Szabó, "Smaller is better? Unduly nice accuracy assessments in roof detection using remote sensing data with machine learning and k-fold cross-validation," *Heliyon*, vol. 9, no. 3, p. e14045, 2023, doi: 10.1016/j.heliyon.2023.e14045.

[30] A. Comber and N. Tsutsumida, "Geographically weighted accuracy for hard and soft land cover classifications: 5 approaches with coded illustrations," *Int. J. Remote Sens.*, vol. 44, no. 19, pp. 6233–6257, 2023, doi: 10.1080/01431161.2023.2264503.

[31] S. Farhadpour, T. A. Warner, and A. E. Maxwell, "Selecting and Interpreting Multiclass Loss and Accuracy Assessment Metrics for Classifications with Class Imbalance: Guidance and Best Practices," *Remote Sens. (Basel).*, vol. 16, no. 3, p. 533, 2024, doi: 10.3390/rs16030533.

[32] J. Opitz, "A Closer Look at Classification Evaluation Metrics and a Critical Reflection of Common Evaluation Practice," *Trans. Assoc. Comput. Linguist.*, vol. 12, pp. 820–836, 2024, doi: 10.1162/tacl_a_00675.

[33] A. Reinke *et al.*, "Understanding metric-related pitfalls in image analysis validation," *Nat. Methods*, vol. 21, no. 2, pp. 182–194, 2024, doi: 10.1038/s41592-023-02150-0.

[34] L. Maier-Hein *et al.*, "Metrics reloaded: recommendations for image analysis validation," *Nat. Methods*, vol. 21, no. 2, pp. 195–212, 2024, doi: 10.1038/s41592-023-02151-z.

[35] T. Viering and M. Loog, "The Shape of Learning Curves: A Review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7799–7819, 2023, doi: 10.1109/tpami.2022.3220744.

[36] T. Silva Filho, H. Song, M. Perello-Nieto, R. Santos-Rodriguez, M. Kull, and P. Flach, "Classifier calibration: a survey on how to assess and improve predicted class probabilities," *Mach. Learn.*, vol. 112, no. 9, pp. 3211–3260, 2023, doi: 10.1007/s10994-023-06336-7.

[37] D. Valle, R. Leite, R. Izbicki, C. Silva, and L. Haneda, "Local uncertainty maps for land-use/land-cover classification without remote sensing and modeling work using a class-conditional conformal approach," *International Journal of Applied Earth Observation and Geoinformation*, vol. 135, p. 104288, 2024, doi: 10.1016/j.jag.2024.104288.

[38] A. Paleyes, R.-G. Urma, and N. D. Lawrence, "Challenges in Deploying Machine Learning: A Survey of Case Studies," *ACM Comput. Surv.*, vol. 55, no. 6, pp. 1–29, 2023, doi: 10.1145/3533378.

[39] H. Andalib, H. Ebadi, and R. Naanjam, "An intelligent dual-stage fusion framework of optical and radar data for land cover classification," *Remote Sens. Appl.*, vol. 42, p. 101987, 2026, doi: 10.1016/j.rsase.2026.101987.

[40] R. Gomes Mantovani *et al.*, "Better trees: an empirical study on hyperparameter tuning of classification decision tree induction algorithms," *Data Min. Knowl. Discov.*, vol. 38, no. 3, pp. 1364–1416, 2024, doi: 10.1007/s10618-024-01002-5.

[41] A. Candelieri, A. Ponti, and F. Archetti, "Fair and green hyperparameter optimization via multi-objective and multiple information source Bayesian optimization," *Mach. Learn.*, vol. 113, no. 5, pp. 2701–2731, 2024, doi: 10.1007/s10994-024-06515-0.

[42] A. Morales-Hernández, I. Van Nieuwenhuyse, and S. Rojas Gonzalez, "A survey on multi-objective hyperparameter optimization algorithms for machine learning," *Artif. Intell. Rev.*, vol. 56, no. 8, pp. 8043–8093, 2023, doi: 10.1007/s10462-022-10359-2.