

ORIGINAL ARTICLE

Evaluating machine learning algorithms: The role of sample size and quality

Przemysław Kupidura ^{1*} and Michał Szkibiel ¹

¹Faculty of Geodesy and Cartography, Warsaw University of Technology, Pl. Politechniki 1, 00–661 Warsaw, Poland

*przemyslaw.kupidura@pw.edu.pl

Abstract

This study evaluates the performance of selected machine learning methods, Maximum Likelihood (MLC), Random Forest (RF), Extreme Gradient Boosting (XGB), Support Vector Machine (SVM), and Artificial Neural Networks (ANN), for land use/land cover (LULC) classification using Sentinel-2 satellite imagery. Each algorithm was tested across multiple classification scenarios that systematically varied both training sample size and training sample quality. To emulate realistic imperfections in reference data, controlled levels of label noise were introduced into the training set, and the resulting changes in classification performance were analysed. In addition, each model's susceptibility to overfitting was assessed by comparing performance on training data and independent test data. To reduce the influence of a single random draw of training and testing pixels, all sampling-based experiments were repeated 10 times using different random seed values, and the reported results were summarized using repeated-run statistics. This design enabled a comparable assessment of how classification accuracy, stability, and generalization depend on dataset size and quality. The results indicate that SVM was the most consistent and reliable method across the tested scenarios, achieving high classification accuracy over a wide range of training sample sizes and maintaining strong robustness under noisy labels. Compared with the other methods, SVM showed lower sensitivity to degraded training data and a smaller tendency to overfit, which makes it a strong baseline choice when reference data are limited or imperfect.

Keywords: overfitting, land cover classification, remote sensing, supervised machine learning, training sample size, model robustness

1 Introduction

Machine learning (ML) is becoming an increasingly common technique for acquiring spatial information from remote sensing data. This is due to the growing volume of data, which is becoming more difficult to process manually, but the primary factor is the increased effectiveness of ML itself. In recent years, many new ML algorithms have been developed that are more complex, more effective, and above all, more efficient compared to older algorithms. Their application in remote sensing allows for the creation of, among other things, land cover and land use (LULC) maps with greater accuracy.

The multitude of possible solutions – even among popular ML algorithms, there are at least several candidates, each with multiple parameter settings – is beneficial, but it also makes the selection of an optimal solution difficult. Halevy et al. (2009) argued that large datasets can allow relatively simple models to outperform more elaborate models trained on smaller datasets. At the same time, both the quantity and the quality of training data can substantially affect model performance. However, preparing better data instead of merely more data requires specialized knowledge and experience. Therefore, when producing LULC maps, it is crucial to

select an algorithm that maintains satisfactory accuracy with limited and imperfect training data. This study addresses that problem by comparing selected ML algorithms under varying training-data conditions.

This will not, of course, be the first voice in this discussion – the topic of studying the efficiency or simply the effectiveness of selected ML algorithms is well represented in the research literature.

Studies comparing the effectiveness of individual algorithms without explicitly considering the quantity or quality of training samples are numerous. Taken together, they do not support a universal conclusion that one classifier always outperforms the others. Rather, the reported performance depends on the classification task, the data type, the number of classes, and the character of the training data. Nevertheless, the comparative studies cited below more often identify SVM, RF, and XGB among the strongest performers, which justifies focusing on these methods in the present work.

The high effectiveness of XGB is indicated by Seydi et al. (2022) – in their study, it slightly outperforms deep neural networks (DNN) and SVM, and significantly outperforms Random Forest (RF) and single decision trees (DT). Similarly, Liu et al. (2021) demonstrated the superiority of XGB over RF. Bigdeli et al. (2023) also show the

high effectiveness of XGB, this time compared to artificial neural networks (ANN).

On the other hand, studies conducted by [Koppaka and Moh \(2020\)](#) show the high effectiveness of SVM – compared to RF, convolutional neural networks (CNN), RNN-LSTM, and RNN with Gated Recurrent Unit (GRU). Other evidence of high effectiveness is presented by [Ghayour et al. \(2021\)](#) – compared to ANN, minimum distance (MD), and maximum likelihood (MLC) classifiers, [Sobieraj et al. \(2022\)](#) – compared to RF, DT, and MLC, and [Shih et al. \(2018\)](#) – compared to DT, ANN, and RF, although in some cases, the superiority of DT was demonstrated.

The ambiguity regarding the choice of the optimal classifier is also visible in meta-analyses of previous publications. Some studies ([Maxwell et al., 2014a,b, 2015](#)) report the superiority of SVM over RF, whereas others ([Maxwell and Warner, 2015](#); [Burkholder et al., 2011](#)) report the opposite result. The effectiveness of RF compared to MLC is also demonstrated by [Mousavinezhad et al. \(2023\)](#). Other publications ([Zhao et al., 2024](#); [Volke and Abarca-Del-Rio, 2020](#); [Li et al., 2016](#); [Duro et al., 2012](#); [Cracknell and Reading, 2014](#)) indicate similar effectiveness of RF and SVM. In view of this ambiguity, analyses that explicitly account for changing classification conditions are particularly valuable.

Noteworthy is ([Shang et al., 2018](#)), in which the authors analyze the impact of training sample size on classification accuracy in Landsat-8 images. SVM, RF, DT, MLC, and Naive Bayes (NB) were compared. It was shown that classification accuracy increases with the size of the training sample in all cases except NB. Overall, SVM demonstrated the highest accuracy.

[Ramezan et al. \(2021\)](#) compare the effectiveness of SVM, RF, Gradient Boost (GBM), k-Nearest Neighbors (kNN), Single-Layer Perceptron ANN, and Learning Vector Quantization in LULC classification. The analysis is conducted using training data of various sizes – from 40 pixels to 10000 pixels. The best results were obtained using RF and GBM, with a slight decrease in accuracy as the training sample size decreased. High sensitivity to reduced training sample size was observed in ANN and, to a lesser extent, in SVM.

[Zheng and Jin \(2020\)](#) investigated the impact of training sample size and quality on the effectiveness of DT, RF, and XGB algorithms. A strong influence was demonstrated, indicating the lower effectiveness of DT compared to RF and XGB.

[Kupidura et al. \(2024\)](#) compared the effectiveness of XGB, RF, and SVM using a range of training samples from 100 pixels to 200,000 pixels. In the study, XGB showed the highest sensitivity to training sample size, while SVM demonstrated the greatest resilience, providing the best results with the smallest sample sizes. Additionally, [Kupidura and Niemyski \(2024\)](#) show SVM as the algorithm yielding the best results – compared to RF and MLC – when using the smallest training data sizes (less than 3000 pixels). Notably, with the largest training sample size (over 34,000 pixels), the highest accuracy was achieved using MLC. It is also worth mentioning [Fu et al. \(2023\)](#) – which also analyzes the impact of training sample size on classification accuracy, but only using RF. [Figueroa et al. \(2012\)](#) approached this issue from a different angle, analyzing the required training sample size depending on the expected classification accuracy.

[Budach et al. \(2022\)](#) demonstrate the significant impact of training data quality on accuracy – in a different context – regression, not classification. They examined the impact of training data on the effectiveness of regression trees, kNN, ANN, and RF.

Research on the relation between training sample size and machine-learning performance can be found in [Raudys and Jain \(1991\)](#) and [Halevy et al. \(2009\)](#). Although these works were not designed specifically for remote sensing, they highlight two issues that are directly relevant here: small training sets may substantially increase the variance of classifier performance, and the apparent superiority of a method may depend on the specific sample drawn. These observations support the need to evaluate algorithms under repeated sampling rather than on a single random draw.

As the above review indicates, it is difficult to definitively identify the best algorithm, as the effectiveness of individual methods largely depends on the classification conditions: the number and type of classes, as well as the quantity and quality of training data.

The aim of this article is to analyze the impact of training data on the effectiveness of popular ML algorithms: RF, XGB, SVM, MLC, and ANN, using several parameter settings specific to these algorithms. The relationship between classification effectiveness, understood here primarily as overall agreement measured with the kappa coefficient and overall accuracy, and two main characteristics of the training sample was analyzed: its size, measured by the number of training pixels, and its quality, controlled by deliberately introducing pixels representing other land-cover classes into the training data of individual classes. In addition, the susceptibility of selected algorithms to overfitting was examined by assessing classification accuracy on both independent test data and training data and then comparing the obtained results. Class-specific effects were treated as complementary information and evaluated using precision, recall, F1-score, and representative confusion matrices.

2 Materials and methods

The research methodology aimed to answer three fundamental research questions regarding the tested ML algorithms:

1. The impact of training sample size on classification accuracy using selected ML algorithms – To this end, the accuracy of selected ML algorithms was examined using training samples with pixel counts ranging from 10 to 2500. As a result, the efficiency of the tested algorithms was analyzed.
2. Algorithm robustness to qualitative errors in training samples – For this purpose, the accuracy of selected ML algorithms was examined using training samples consisting of 2500 pixels, whose quality was deliberately disturbed by adding pixels representing other classes in reality – in sizes ranging from 10% (250 pixels) to 100% (2500 pixels) of the "clean" training sample size. This way, the robustness of the tested algorithms to errors in training samples was analyzed.
3. Susceptibility of selected algorithms to overfitting – For this purpose, classification accuracy in all variants was assessed on training data, in addition to the assessment based on independent, randomly selected test data. By comparing the results obtained from test data and training data, it was possible to infer the probable overfitting of the selected classification variants.

The general scheme of the adopted methodology is presented in Figure 1. It can be outlined as follows:

1. Selection of the test image;
2. Selection of the training and test area;
3. Selection of machine learning algorithms and their variants;
4. Creation of training sets;
5. Training the algorithms on the training set;
6. Creation of the test set;
7. Classification of the test set using the trained algorithms;
8. Evaluation of the algorithms.

2.1 Test imagery

A Sentinel-2 satellite scene from September 10, 2023, depicting the area around Szczecin at processing level 2A, was selected as the test image. The extent of the scene is shown in Figure 2a. A training area and a testing area were selected from the entire scene, as shown in Figure 2b. The training area is shown in Figure 3, while the testing area is shown in Figure 4. The areas were selected due to their similarity. Both areas include diverse urban, rural, and

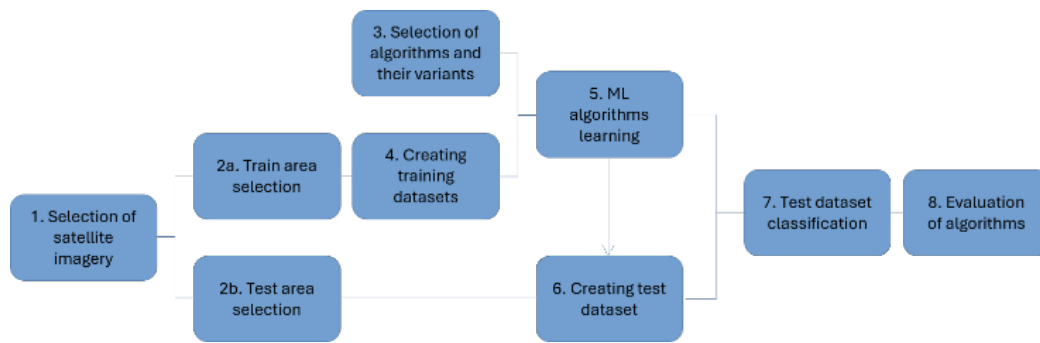


Figure 1: A methodology diagram

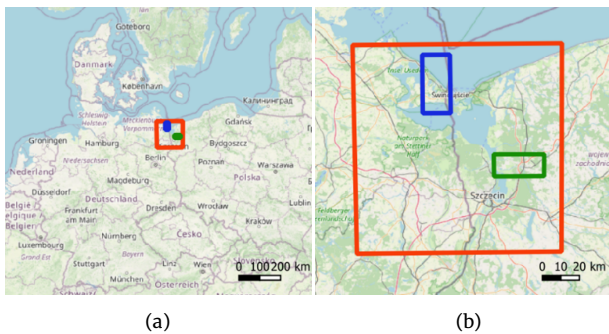


Figure 2: Location of the full Sentinel-2 scene (red) together with enlarged views indicating the training area (green) and testing area (blue) (Base map source: openstreetmap.org)

industrial buildings, large water bodies, deciduous and coniferous forests, airports, agricultural areas, and meadows.

2.2 Definition of LULC classes

- Water Bodies – areas covered by surface water, including lakes, rivers, reservoirs, and other bodies of water, regardless of their permanence or depth;
- Built-up Area – land occupied by buildings, infrastructure, and other artificial structures, including urban and industrial zones, residential areas, and transportation networks;
- Forest – land covered with trees, regardless of tree species, density, or age, including both natural and planted forests;
- Bare land – areas of bare soil with little to no vegetation, such as cultivated fields, barren land, and recently disturbed surfaces, excluding built-up areas;
- Low vegetation – land covered with short vegetation, including grasslands, meadows, pastures, and croplands with low-growing plants, whether cultivated or uncultivated.

2.3 ML algorithms and tested variants

RF, XGB, SVM, MLC, and ANN algorithms with different parameter settings were tested. The following paragraphs summarize the tested algorithms and the selected parameters. A comprehensive summary of the tested ML configurations is presented in Table 1.

Maximum Likelihood

The MLC algorithm assumes that the occurrences of a class in the feature space can be described by a normal distribution (Richards and Jia, 2006). Therefore, on the basis of the training samples, a Gaussian function parameterized by the class mean vector and covariance is fitted for each class. During classification, the discriminant value is calculated for each class and interpreted as the

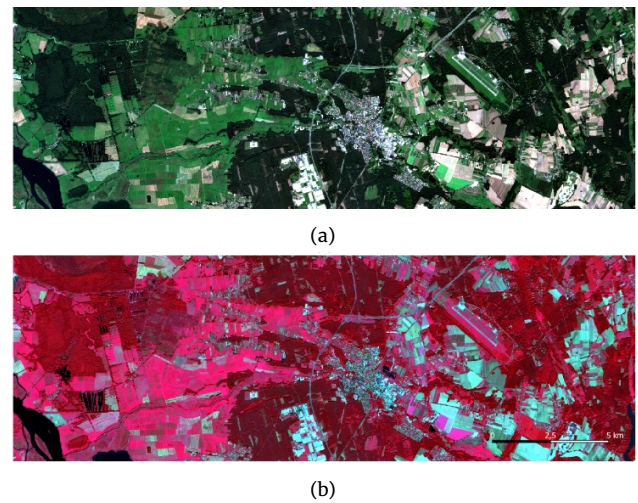


Figure 3: Training area (green) in Sentinel-2 compositions: (a) R4, G3, B2; (b) R8, G4, B3

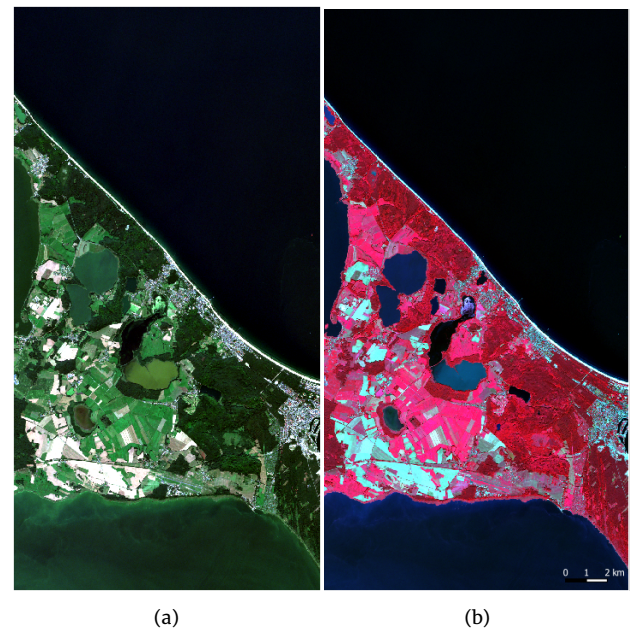


Figure 4: Testing area (blue) in Sentinel-2 compositions: (a) R4, G3, B2; (b) R8, G4, B3

Table 1: Tested hyperparameter configurations of ML algorithms

Algorithm type	Parameter				Identifier		
Maximum likelihood	none				MaxLike		
Random forest	n_estimators	150			RF_150		
		200			RF_200		
		250			RF_250		
Extreme gradient boosting	n_estimators	max_depth	9	5	XGB_9_5		
			11		XGB_11_5		
			13		XGB_13_5		
			9	6	XGB_9_6		
			11		XGB_11_6		
			13		XGB_13_6		
			9	7	XGB_9_7		
			11		XGB_11_7		
			13		XGB_13_7		
SVM	kernel	RBF	gamma	3.125e-8	SVM_3.125e-08_rbf		
				6.25e-8	SVM_6.25e-08_rbf		
				1.25e-7	SVM_12.5e-07_rbf		
		polynomial	gamma	degree	3.125e-8	SVM_3.125e-08_poly3	
					6.25e-8	3	SVM_6.25e-08_poly3
					1.25e-7		SVM_12.5e-07_poly3
					3.125e-8		4
					6.25e-8	SVM_6.25e-08_poly4	
					1.25e-7	SVM_12.5e-07_poly4	
					3.125e-8	5	SVM_3.125e-08_poly5
					6.25e-8		SVM_6.25e-08_poly5
					1.25e-7		SVM_12.5e-07_poly5
Multilayer perceptron	Number of hidden layers	Number of neurons per layer	3	5	MLP_5_3		
			4		MLP_5_4		
			5		MLP_5_5		
			3	6	MLP_6_3		
			4		MLP_6_4		
			5		MLP_6_5		
			3	7	MLP_7_3		
			4		MLP_7_4		
			5		MLP_7_5		

basis for assigning the pixel to the most probable class. The method also includes a prior probability parameter describing the relative frequency of occurrence of each class. In this study, equal prior probabilities were assumed for all classes.

Random Forest

The method is an extension of the decision tree algorithm. To increase the effectiveness of this method, many independently trained decision trees are used (Belgiu and Drăguț, 2016). The final result is the class most frequently indicated by individual trees. The primary parameter of the method is the number of decision trees ($n_{\text{estimators}}$), whose impact will be studied. Other parameters are inherited from the decision tree method:

- Splitting criterion – the method of calculating the quality of the split. In our research, the Gini criterion was adopted.
- Maximum depth – the limit of the tree size to the indicated value. An unlimited maximum depth was assumed.
- max_features – the maximum number of parameters among which the best split is sought. It was assumed that all parameters are examined.

Extreme Gradient Boosting

This method, like random forest, also uses an ensemble of decision trees. However, unlike the previous method, individual trees are created in a sequential process. Each subsequent tree in the sequence is trained to correct the errors of the previous one (Chen and Guestrin, 2016). Thanks to this approach, fewer trees are needed for effective classification. Additionally, XGBoost incorporates a regularized objective function and several mechanisms for controlling model complexity and reducing the risk of overfitting (Chen and Guestrin, 2016). The parameters used are:

- $n_{\text{estimators}}$ – the number of trees in the sequence,
- max_depth – the limit of the tree size to the indicated value.

Other parameters include:

- learning_rate – a value in the range $[0, 1]$ indicating the limitation of correction by the next tree in the sequence. A value of 0.3 was used.

Support Vector Machine

An algorithm whose goal is to delineate areas in the parameter hyperspace that will be unambiguously assigned to one class. Their boundaries are determined by fitting the specified function to separate the classes with the largest possible margin (Boser et al., 1992). The parameters used are:

- γ – a parameter that indicates the significance of a single point. The higher it is, the more the model fits the training data.
- kernel – the type of function being fitted,
- degree – the degree of the polynomial when using a polynomial function.

Other parameters include:

- C – a parameter controlling the margin size. A value of 1.0 was used.

Multilayer Perceptron

This is the most commonly used neural network architecture. The basic unit is neurons grouped in layers. The mandatory layers are the input layer, where each neuron represents one input parameter, and the output layer, where each neuron represents one of the output classes. Additionally, a multilayer perceptron can consist of any number of hidden layers. The greater their number, the greater the potential to fit the data. Each neuron, except for the output layer, is connected to every neuron in the next layer (fully connected). The parameters used are:

- number of hidden layers,
- Number of neurons in the hidden layers.

Table 2: Classification scenarios depending on the quantity and quality of training data

Scenario ID		Training sample size [pixels]	
Test data (independent)	Training data	Correct pixels	Incorrect pixels
test_10	train_10	10	0
test_25	train_25	25	0
test_50	train_50	50	0
test_75	train_75	75	0
test_100	train_100	100	0
test_150	train_150	150	0
test_200	train_200	200	0
test_250	train_250	250	0
test_500	train_500	500	0
test_750	train_750	750	0
test_1000	train_1000	1000	0
test_2500	train_2500	2500	0
test_2500/250	train_2500/250	2500	250
test_2500/375	train_2500/375	2500	375
test_2500/875	train_2500/875	2500	875
test_2500/1250	train_2500/1250	2500	1250
test_2500/1625	train_2500/1625	2500	1625
test_2500/2125	train_2500/2125	2500	2125
test_2500/2500	train_2500/2500	2500	2500

Other parameters include:

- Activation function – the function performed in the neuron based on the parameters passed from the previous layers. Soft-max was used.
- Loss – the function used to determine the gradient of changes in subsequent learning iterations. Categorical cross-entropy was used.

2.4 Specification of classification scenarios

Table 2 presents a list and identifiers of classification scenarios, depending on the size of the training sample and its quality.

As mentioned at the beginning of the chapter, the methodology aimed to serve three purposes:

1. Analyzing the efficiency of the algorithm depending on the size of the training sample: This purpose was served by scenarios with ID test_X (from test_10 to test_2500) – the number X indicated the number of training pixels. In these scenarios, only carefully selected training data were used, so they differed only in statistical reliability related to the number of training pixels. The classification accuracy obtained in these scenarios was verified using randomly selected independent test data.
2. Algorithm robustness to contamination of the training sample: This purpose was served by scenarios with ID test_2500/X (from test_2500/250 to test_2500/2500) – the first number (2500) indicated the number of correct training pixels, while the number /X indicated the number of pixels representing other classes in reality, introduced to contaminate the training sample. The classification accuracy obtained in these scenarios was verified using randomly selected independent test data.

Table 3: Number of test pixels assigned to individual classes

Class	Number of pixels
Water bodies	107
Built-up area	57
Forest	141
Bare land	79
Low vegetation	116
Sum	500

3. Analyzing the susceptibility of algorithms to overfitting: This purpose was served by scenarios with ID train (all variants). The main common feature is the verification of classification accuracy not using independent random test data (as in the case of test scenarios), but using training data. Comparing the results obtained in this way with the results of analogous test scenarios allowed for inferring the possibility of overfitting in a given variant – this could be indicated by a high accuracy result verified on training data and simultaneously a low accuracy result verified on test data.

2.5 Preparation of training data

Training data were prepared by manual photointerpretation of the Sentinel-2 imagery. Class-specific reference fields were first delineated within the training area shown in Figure 5. All pixels belonging to the delineated fields were then grouped by class and randomly ordered within each class. For each scenario with n correct samples, the first n pixels from the corresponding random list were used, depending on the classification scenario (Table 2).

From the remaining pixels of each class, an additional pool of 2500 pixels was selected. To generate noisy samples, these pixels were reassigned in a balanced, not random, manner to the other four classes. Thus, for a given source class, 625 pixels were assigned to each of the remaining classes. For example, 625 pixels originating from water areas were reassigned to built-up areas, forests, bare soil, and low vegetation. For a scenario of the type test_2500/X, the required number X of incorrectly labeled pixels was taken from this balanced pool and then merged with the 2500 correctly labeled pixels of the target class. Finally, the combined set was randomly shuffled before training. In this way, each noisy sample preserved equal contributions of mislabeled pixels from the four remaining classes, and each larger set contained all the data used in the smaller sets.

2.6 Preparation of testing data

To answer the first two research questions, namely the efficiency of algorithms depending on the size of the training sample and their resistance to contamination (scenarios with the prefix test), the appropriate metrics were calculated using independent test data selected from the separate testing area. Test pixels were prepared analogously to the training pixels: reference fields were delineated by photointerpretation and pixels were then sampled randomly within those fields. The number of test pixels assigned to individual classes is presented in Table 3. Because the training and testing areas were spatially separate, the evaluation was not based on neighboring pixels from the same local objects, which reduced the influence of spatial autocorrelation.

For the analysis of susceptibility/resistance to overfitting (scenarios with the prefix train), the metrics were additionally calculated on the training data. In each case, exactly the same data that were used in the training process were utilized (the method of their selection is described above). To reduce the influence of a single

random draw of training and testing pixels, all sampling-based experiments were repeated 10 times using different random seed values. The charts report mean kappa values from these repetitions, whereas Appendix reports median values and SD for all tested variants. Minimum and maximum values were additionally used in Sections 3.1–3.3 to comment on the stability of the obtained results.

2.7 Accuracy assessment and verification data selection

To determine the accuracy of individual classification variants, the following accuracy measures, commonly used in the analysis of classification accuracy in remote sensing, were used:

- For individual land cover classes: F1-score (Hand et al., 2021), precision, and recall (Powers, 2007);
- Overall for the entire classification: kappa coefficient (Cohen, 1960) and overall accuracy (or overall success rate) (Labatut and Cherifi, 2012) to determine the effectiveness of precision and recall (Hand et al., 2021);

in the form of:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (1)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

where: TP – True Positive, FP – False Positive, and FN – False Negative.

For F1 score (Hand et al., 2021):

$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (3)$$

For the kappa coefficient (Sim and Wright, 2005):

$$\text{kappa} = \frac{P_o - P_c}{1 - P_c}, \quad (4)$$

where: P_o – observed agreement, P_c – chance agreement.

For overall accuracy:

$$\text{Overall accuracy} = \frac{C_c}{C_o}, \quad (5)$$

where: C_c – number of correctly classified pixels, C_o – total number of classified pixels.

Because the study compares a large number of classifier variants and scenarios, the kappa coefficient is treated as the principal summary measure in the results section, while class-specific metrics are used as complementary indicators and are illustrated through representative confusion matrices.

3 Results

This section presents the results and their interpretation. Its structure follows the three main research questions introduced in Section 2. The principal results are shown in charts in the main text, whereas the complete numerical results are provided in Appendix. Confusion matrices for representative MLC scenarios are included in Appendix. Unless stated otherwise, the kappa values reported below are mean values from 10 random repetitions performed for each scenario. All charts use the same vertical scale to facilitate comparison; therefore, some very low values are not visible in detail in the plots, but they remain available in the appendix tables.

To complement these mean-based plots, Appendix reports the median kappa values and their SD for all variants and scenarios. Minimum and maximum values are used below as additional indicators of the spread of the repeated results.

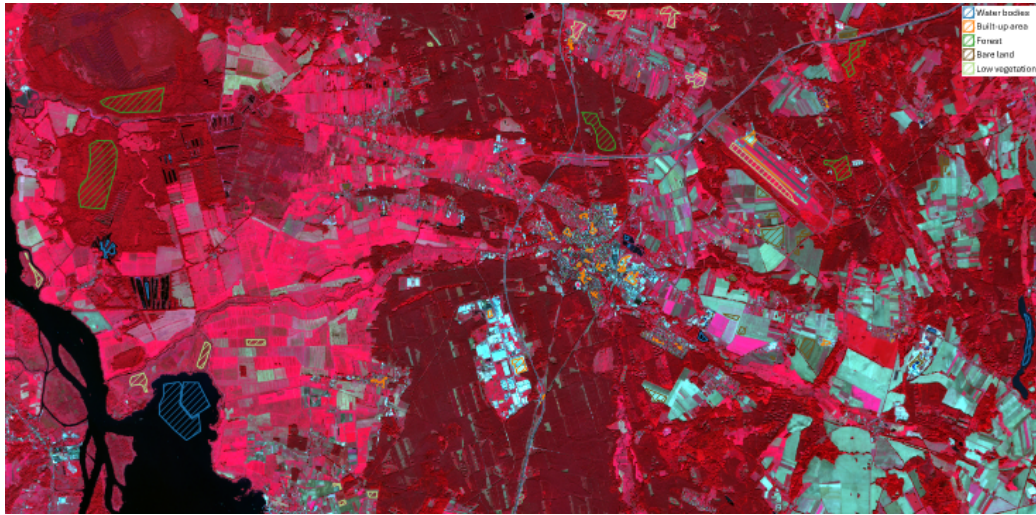


Figure 5: Selected training fields for individual classes

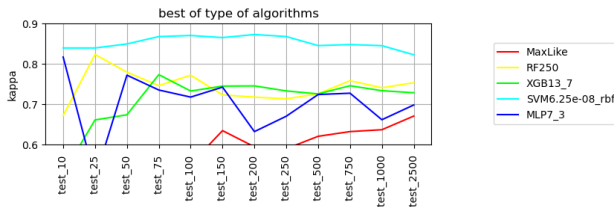


Figure 6: Mean kappa values obtained for the representative variants of individual types of ML algorithms – test_X scenarios (10 repetitions)

3.1 Efficiency analysis (test_X scenarios)

First, the results of the representative variants among the five tested algorithm families were analyzed (Figure 6). For each family, the representative variant was defined as the configuration that achieved the highest kappa coefficient in at least one test_X scenario when the mean value over 10 repetitions was considered. Accordingly, the variants shown in Figure 6 are MaxLike, RF_250, XGB_13_7, SVM_6.25e-08_rbf, and MLP_7_3.

The averaged results again indicate that SVM is the strongest performer in this comparison. Across all test_X scenarios, the selected SVM variant remains at a high level of kappa, approximately 0.82–0.87, and therefore combines very good accuracy with low sensitivity to training sample size. RF and XGB yield lower mean kappa values, usually in the range of about 0.67–0.82 and 0.53–0.77, respectively. RF reaches its maximum already for relatively small training sets and then remains stable, whereas XGB improves from very small samples and reaches its best mean result around test_75. The MLP family shows the highest variability. The selected MLP_7_3 variant achieves very high mean accuracy for some scenarios, including test_10 and test_50, but the results fluctuate strongly between scenarios, which suggests greater sensitivity to the exact composition of the training data. MLC remains clearly below the best SVM and tree-based variants and improves more gradually with increasing sample size, reaching its highest mean kappa only for the largest training scenarios.

The dispersion measures confirm that this ranking is not caused by averaging alone. For the representative SVM_6.25e-08_rbf variant in the test_X series, SD remains low, typically between 0.65% and 1.68%, and the min–max range stays narrow at about 3.29–5.13 percentage points even for the smallest clean test scenarios. By contrast, the representative MLP_7_3 variant shows much greater

instability for small samples, with SD reaching 17.43% for test_10 and 12.95% for test_50, and with a min–max range of 48.15 percentage points for test_10. This indicates that some favorable average or median results obtained for MLP may still be associated with limited repeatability across random draws.

Figure 7 presents the results obtained for all tested variants, broken down by individual algorithm families.

The best algorithm in this comparison, SVM (Figure 7e), again shows differences depending on the adopted parameters. The strongest effect is associated with the kernel type. The RBF-based variants provide the highest mean kappa values and remain consistently high across all training-sample sizes. Polynomial variants are usually weaker, although some of them still provide competitive results. Differences related to gamma are smaller than the differences between kernels, but lower gamma values generally lead to better overall performance. In contrast to SVM, both RF and XGB show relatively small between-variant differences, and the relative ordering of their variants changes only slightly across scenarios. MLP remains the most variable family, which is consistent with the stronger dependence of neural-network performance on architecture and on the composition of the sampled training data.

In the case of MLP, the greatest variation in classification accuracy can be observed depending on the network's characteristics – its architecture. The impact of the number of hidden layers is particularly noticeable. With a greater number of hidden layers, a significantly larger training sample was required to achieve satisfactory accuracy, and this effect was even more pronounced in networks with a larger number of neurons per layer. Therefore, it can be inferred that a more complex network structure has greater requirements for the training sample. Interestingly, in the case of networks with the smallest number of hidden layers (3), better results were obtained with a larger number of neurons per layer, especially for scenarios with the smallest training sample sizes. For scenarios with the largest number of training pixels (500 and more), all variants achieved relatively good, comparable results – the noticeable differences between them are no longer systematic and can be attributed to the slight variability of models built based on neural networks.

Both decision tree-based algorithms (RF and XGB) show a very high similarity between variants within each algorithm. Neither the number of trees (in both cases) nor the maximum tree depth (XGB) significantly affect classification accuracy – some differences can be observed, but they are small and, more importantly, do not show a systematic and clear pattern.

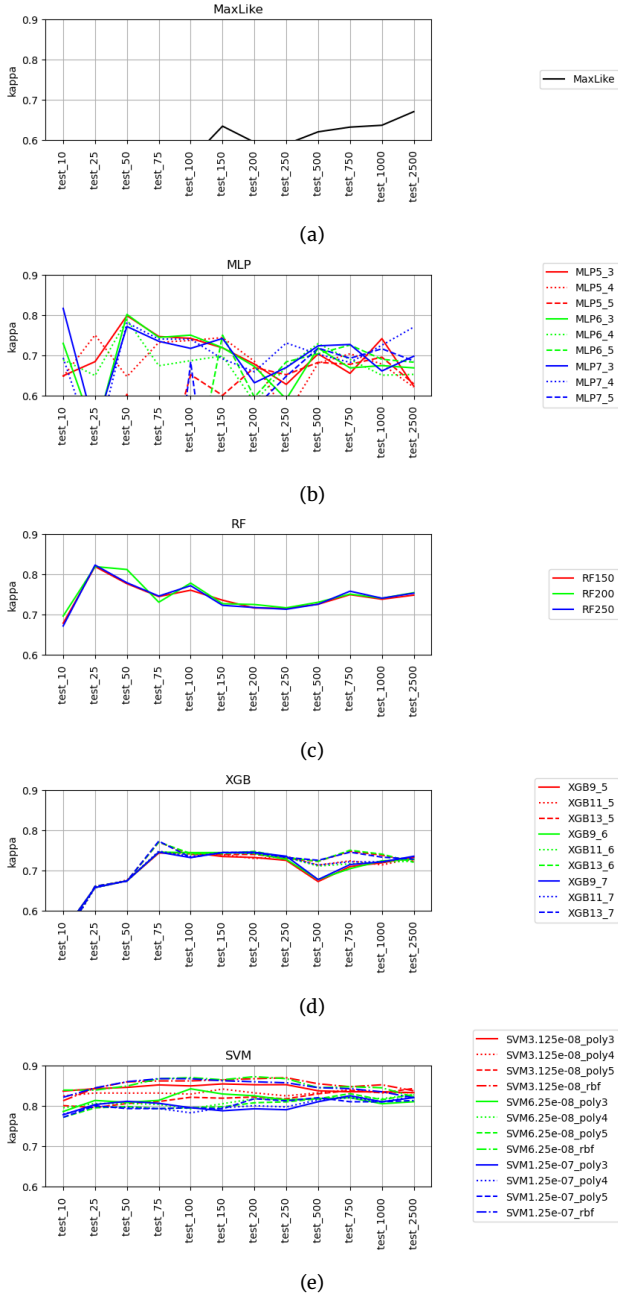


Figure 7: Mean kappa values obtained for all tested variants broken down by individual algorithms – test_X scenarios (10 repetitions): (a) Maximum Likelihood (MaxLike); (b) Multilayer Perceptron (MLP); (c) Random Forest (RF); (d) Extreme Gradient Boosting (XGB); (e) Support Vector Machine (SVM)

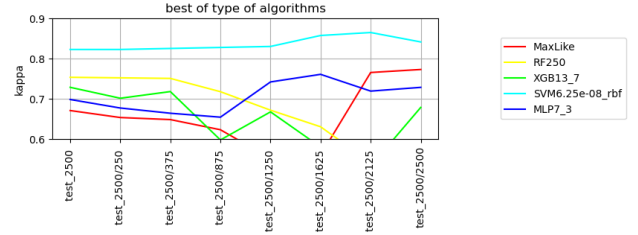


Figure 8: Mean kappa values obtained for the representative variants of individual types of ML algorithms – test_2500/X scenarios (10 repetitions)

3.2 Analysis of resistance to noise (test_2500/X scenarios)

First, the representative variants identified in Section 3.1 were analyzed for the test_2500/X scenarios (Figure 8). Because the values shown are averages over 10 repetitions, the curves reflect the general tendency rather than the result of a single random sample composition.

The selected SVM configuration, SVM_6.25e-08_rbf, remains the most robust method under label noise. Its mean kappa does not decrease with increasing contamination and even slightly increases in the most contaminated scenarios, reaching about 0.84 for test_2500/2500. This result indicates that the superiority of SVM in the previous version was not caused solely by one particular random draw of training pixels. At the same time, this robustness still does not apply equally to all SVM configurations, as several polynomial-kernel variants remain clearly weaker in Table 8 (Appendix). A possible explanation is that, in the adopted balanced-noise design, the mislabeled pixels were dispersed across the other classes and remained diluted by a still large pool of correctly labeled samples. For the RBF SVM, such contamination may behave more like weak boundary perturbation than like a systematic class shift; in some repetitions it may even slightly smooth the decision boundary and produce a marginal improvement on the independent test data.

The SD values support the same interpretation. In the noisy test_2500/X scenarios, the representative SVM variant keeps SD below 1.02% for all contamination levels and its min-max range does not exceed 3.37 percentage points. In contrast, RF_250 and XGB_13_7 show progressively broader dispersion as contamination increases, reaching SD values of 6.01% and 7.13%, respectively, in the most contaminated scenario. The representative MLP_7_3 variant is even less stable, with SD rising to 10.77% and the min-max range to 27.44 percentage points at the 65% noise level. Therefore, SVM is not only the most accurate on average, but also the most repeatable under noisy labeling.

RF and XGB are clearly more sensitive to contamination. For RF, the mean kappa decreases nearly monotonically from about 0.75 to about 0.52 as noise increases. XGB also shows an overall decrease, although with stronger local fluctuations. The selected MLP variant does not exhibit a monotonic loss of accuracy; instead, its mean kappa changes irregularly and even increases for the more contaminated scenarios. This again suggests that MLP behaviour is less stable and more dependent on the particular composition of the training sample than in the case of SVM. Figure 9 presents the results obtained for all tested variants, broken down by individual algorithm families.

An interesting phenomenon can be observed in the case of MLC: scenarios with the highest share of erroneous training samples show the highest kappa values. Analyzing the confusion matrices obtained for this algorithm in various classification scenarios can help explain this paradox. Analyzing especially these key matrices for the MLC algorithm – the test_2500, test_2500/1250 (for which the lowest kappa value is obtained), and test_2500/2500 (for which

the highest kappa value is obtained) scenarios, an interesting dynamic can be observed (Tables 4–6). In the test_2500 scenario, the pairs of classes: Built-up/Bare land and Forest/Low veg have the greatest impact on reducing accuracy. In the test_2500/1250 scenario, we observe a significant decrease in the classification accuracy of these 4 classes, while in subsequent scenarios (with a greater share of noise) – an increase – to still relatively low levels, but this translates into an increase in the overall classification accuracy. However, this should be interpreted solely as a random effect – the analysis of the entire accuracy dynamics depending on the presence of noise – specifically: its instability – allows us to conclude that MLC is an algorithm significantly susceptible to noise in test data.

The averaged results therefore confirm that the non-monotonic behaviour of MLC under increasing noise is not merely a single-run artifact, although the method still remains clearly less robust than the best SVM configuration.

3.3 Overfitting susceptibility analysis (train scenarios)

First, the representative variants identified in Section 3.1 were analyzed for the train scenarios (Figure 10). As in the previous sections, the curves show mean kappa values over 10 repetitions.

In these scenarios, kappa values are calculated on the training data rather than on independent test data. High values therefore do not automatically mean good generalization, especially in the train_2500/X scenarios, where noisy training samples were also used for evaluation. RF remains the clearest example of overfitting: its mean kappa equals 1.00 in all scenarios, including the noisy ones. XGB also reaches values very close to 1.00 for the clean scenarios and then declines as noise increases. By contrast, SVM shows lower training-data kappa for the smallest clean samples and therefore a noticeably smaller gap between training and independent test performance. This supports the interpretation that SVM generalizes better than RF and XGB under the analyzed conditions. MLP occupies an intermediate position: for most clean scenarios it attains very high training accuracy, but its average values are less stable than those of RF and XGB, especially for the smallest samples. MLC again behaves differently, with no meaningful result for train_10 and then high but gradually decreasing values for larger clean scenarios.

The additional descriptive statistics are also informative here. For RF, SD equals 0.00% in all train scenarios, which means that the algorithm fits the training data in exactly the same way in every repetition. XGB behaves similarly, with only negligible SD values. This numerical stability should not be interpreted as better generalization; rather, together with the perfect or near-perfect training kappa, it confirms a systematic tendency to overfit. By contrast, SVM and especially MLP remain more variable in the smallest clean training scenarios. For example, for train_10 the SD equals 5.10% for SVM_6.25e-08_rbf and 24.17% for MLP_7_3, while the min–max range for MLP_7_3 reaches 52.50 percentage points.

The other algorithms show similar dynamics: high values in “clean” scenarios and a proportional decrease as noise increases. The similarity is particularly evident in scenarios with noisy training/test data; three algorithms: XGB, MLP, and SVM show very similar dynamics of kappa value decline. It is worth noting that in this case, the decline in kappa value is a desirable phenomenon, as high kappa values (when validation is performed on training data) indicate the model’s adaptation to noise in the training data, hence overfitting. The three algorithms: XGB, SVM, and MLP show almost identical declines, with MLC values decreasing slightly more, though still proportionally.

The values obtained for individual algorithms in the train_10 – train_2500 scenarios, which are free of erroneous training data, look slightly different. XGB, like RF, achieves a kappa value of 1 even for the scenario with the smallest number of training/test data and maintains it for all scenarios in this range (with larger samples,

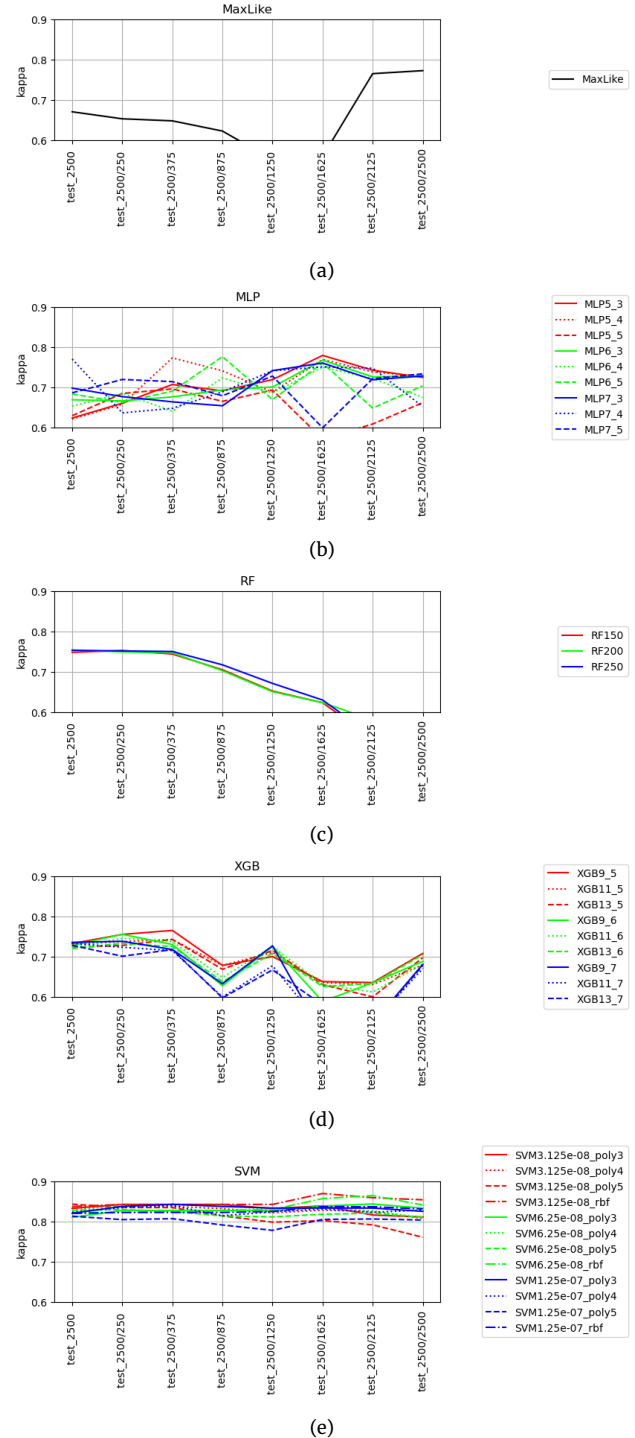


Figure 9: Mean kappa values obtained for all tested variants broken down by individual algorithms – test_2500/X scenarios (10 repetitions): (a) Maximum Likelihood (MaxLike); (b) Multilayer Perceptron (MLP); (c) Random Forest (RF); (d) Extreme Gradient Boosting (XGB); (e) Support Vector Machine (SVM).

Table 4: Confusion matrix for the MLC, test_2500 scenario

Classified\Truth	Water	Built-up	Forest	Bare land	Low veg	Sum	Precision
Water	81	0	0	0	0	81	1.0000
Built-up	26	55	7	31	14	133	0.4135
Forest	0	0	94	0	4	98	0.9592
Bare land	0	0	0	44	0	44	1.0000
Low vegetation	0	2	40	4	98	144	0.6806
Sum	107	57	141	79	116	500	
Recall	0.7570	0.9649	0.6667	0.5570	0.8448	Accuracy	0.7440
F1 Score	0.8617	0.5789	0.7866	0.7154	0.7538	Kappa	0.6796

Table 5: Confusion matrix for the MLC, test_2500/1250 scenario

Classified\Truth	Water	Built-up	Forest	Bare land	Low veg	Sum	Precision
Water	67	6	0	0	0	73	0.9178
Built-up	5	48	3	28	12	96	0.5000
Forest	5	0	64	0	4	73	0.8767
Bare land	29	0	0	44	0	73	0.6027
Low vegetation	1	3	74	7	100	185	0.5405
Sum	107	57	141	79	116	500	
Recall	0.6262	0.8421	0.4539	0.5570	0.8621	Accuracy	0.6460
F1 Score	0.7444	0.6275	0.5981	0.5789	0.6645	Kappa	0.5557

Table 6: Confusion matrix for the MLC, test_2500/2500 scenario

Classified\Truth	Water	Built-up	Forest	Bare land	Low veg	Sum	Precision
Water	101	6	21	0	0	128	0.7891
Built-up	3	27	0	6	2	38	0.7105
Forest	1	14	111	2	14	142	0.7817
Bare land	0	4	0	55	0	59	0.9322
Low vegetation	2	6	9	16	100	133	0.7519
Sum	107	57	141	79	116	500	
Recall	0.9439	0.4737	0.7872	0.6962	0.8621	Accuracy	0.7880
F1 Score	0.8596	0.5684	0.7845	0.7971	0.8032	Kappa	0.7268

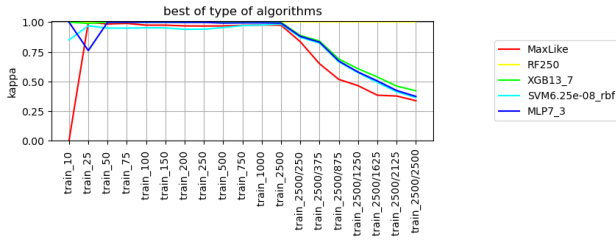


Figure 10: Mean kappa values obtained for the representative variants of individual types of ML algorithms – train scenarios (10 repetitions)

kappa decreases slightly, by less than 0.01). Relatively low values in scenarios with a small number of training data are obtained by SVM, which – again – may indicate resistance to overfitting and the ability to generalize the phenomenon; the difference between kappa values obtained in scenarios with validation based on training data and independent test data is relatively small – the smallest among all analyzed algorithms.

MLC presents an interesting dynamic: for the train_10 scenario, with the smallest number of training data, the kappa value is 0 (the same as in the test_10 scenario), likely due to the small number of training data, insufficient to build reliable probability distributions for individual classes. However, in the next scenario (in terms of training data), we obtain a kappa value of 1, which then decreases slightly with the increase in training data. If we compare this with the results obtained for the test/X scenarios, we conclude (successive increase in accuracy with the increase in the training sample size) that this is likely due to the increased ability to generalize the analyzed phenomenon. The results for all analyzed algorithms are shown in Figure 11.

4 Discussion

The results confirm a positive impact of training sample size on classification effectiveness, which is consistent with the findings of Shang et al. (2018); Ramezan et al. (2021); Zheng and Jin (2020); Kupidura et al. (2024); Kupidura and Niemyski (2024); Fu et al. (2023). They also confirm that SVM is the most reliable method under the conditions tested, especially when the training sample is limited or noisy. The repeated experiments are important here, because they reduce the risk that the observed ranking is driven by a single favorable or unfavorable random selection of pixels.

The additional stability measures refine this interpretation. On the independent test data, SVM combines high central values with low SD and narrow min-max ranges, which indicates robust generalization. RF and XGB are also relatively stable for many scenarios, but their dispersion increases under strong label noise. The widest spreads are observed for several MLP configurations, especially for small training samples, which shows that some neural-network results are strongly dependent on the exact random composition of the reference data. At the same time, the zero SD observed for RF on the training data reinforces the conclusion that its excellent train performance is partly a manifestation of consistent overfitting rather than robust predictive behavior.

This study also showed the highest overall effectiveness of SVM across the analyzed scenarios. Comparing this result with previous studies is not straightforward. For small training sets, most of the cited studies also indicate SVM as the strongest method, whereas Ramezan et al. (2021) reported better performance for RF and GBM. For large training samples, the situation is more diverse: in some studies other algorithms perform better, such as XGB or, under specific conditions, MLC. One likely reason is that the present study considered five broad LULC classes, which resulted in more heterogeneous spectral distributions. This setting is favorable for flexible

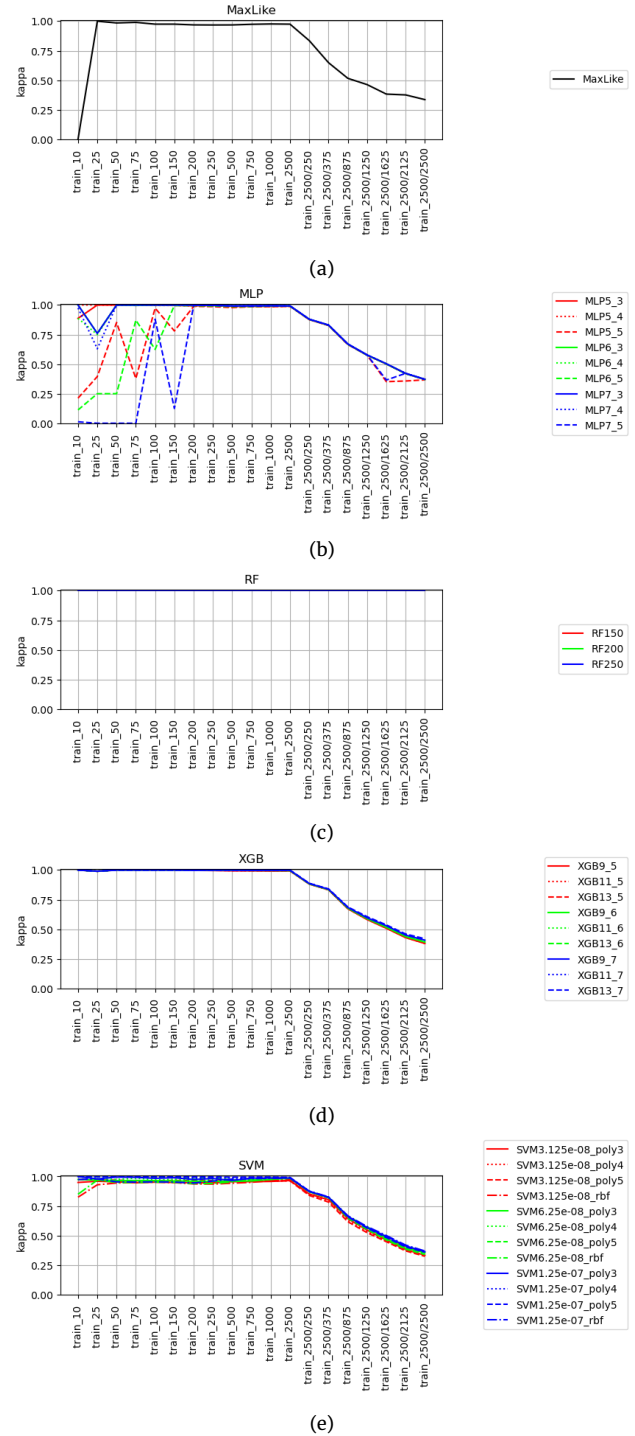


Figure 11: Mean kappa values obtained for all tested variants broken down by individual algorithms – train scenarios (10 repetitions): (a) Maximum Likelihood (MaxLike); (b) Multilayer Perceptron (MLP); (c) Random Forest (RF); (d) Extreme Gradient Boosting (XGB); (e) Support Vector Machine (SVM)

margin-based methods such as SVM, whereas it is less favorable for MLC, whose assumptions are closer to normally distributed subclasses. From a practical perspective, the present results suggest that SVM is particularly attractive when reference data are limited or imperfect and when robustness is more important than exploiting highly specialized training distributions. A plausible interpretation is that the broad classes used here are internally heterogeneous but still sufficiently distinct at the class level, which makes a flexible non-linear maximum-margin classifier particularly effective. In a more thematically detailed or structurally complex landscape, especially one divided into many subclasses with subtler transitions, the relative advantage of SVM could be smaller.

Regarding the tested SVM parameters, the obtained results are consistent with previous studies. The RBF kernel yielded the best results (Mansor et al., 2024), and lower gamma values produced better performance, which is also consistent with the findings reported by Kupidura et al. (2024). This pattern can be interpreted as evidence that lower gamma values improved generalization, whereas higher values increased sensitivity to the training data and therefore the risk of overfitting.

Another source of differences between studies is the design of label noise. In the present experiment, incorrect pixels were added in a controlled and balanced manner, which isolated the effect of noise from class imbalance. In practical applications, mislabeled samples are often spatially clustered and class-dependent; therefore, the relative robustness reported here should be interpreted for the experimental setting adopted in this study.

The present study also has limitations. It focused on two spatially separate areas, five broad LULC classes, and accuracy-oriented evaluation. Additional study regions, a larger number of classes or subclasses, visual comparison of classification maps, and a systematic class-specific analysis for all scenarios would allow the practical meaning of the observed differences to be assessed more fully. Computational cost and implementation complexity were also outside the scope of this study, although these aspects may influence method choice in large-scale applications. In other words, the robustness estimates reported here concern balanced and spatially dispersed label contamination; real training data may contain localized, class-specific, and spatially clustered annotation errors that could affect the tested algorithms differently.

5 Conclusions

The conducted research demonstrated that, under the analyzed experimental design, SVM is the most effective of the tested algorithms for classifying the Sentinel-2 images. It achieved high mean kappa values for both small and large training samples and showed the greatest robustness to noisy training data, especially for the RBF kernel. It also exhibited a smaller discrepancy between training and test performance than RF and XGB, which indicates better generalization. These conclusions are supported not only by a single run but by averaged results from 10 repetitions carried out with different random seeds.

Among the other tested algorithms, RF and XGB also showed high resistance to small training sample sizes (the latter slightly less so) – however, the classification accuracies obtained using these algorithms were lower than those of SVM. Both algorithms also showed less resistance to noisy training samples – classification accuracy decreased proportionally to the increase in the degree of noise in the sample.

Some MLP variants – those with less complex architectures – also showed little impact from the training sample size. For more complex variants, the training sample size had a significant impact – a relatively large number of training pixels was necessary to achieve satisfactory accuracy. Regardless of the architecture, all variants showed very good resistance to noise in the training sample – although the classification accuracies obtained were significantly

lower than those of SVM.

MLC showed a strong dependence on the training sample size and low resistance to noise in the training sample. This does not mean that SVM should automatically be treated as the best choice in every LULC task. Rather, the present results indicate that it is a strong baseline when the number of reference samples is limited, their quality is uncertain, or resistance to overfitting is a priority. Other methods may still be preferable when the landscape is more complex, when a larger number of subclasses must be separated, when class boundaries are subtler or more irregular, or when practical constraints such as computation time are decisive. In the present study, the combination of five broad LULC classes, relatively clear class-level contrasts, and the need to remain robust to limited and noisy reference data favored the RBF SVM.

Appendix

Appendix presents complementary descriptive statistics for the repeated experiments. The tables report median kappa values and SD for all tested variants and scenarios over 10 repetitions performed with different random seed values. Minimum and maximum values were used in Sections 3.1–3.3 to discuss the spread of the repeated results. For completeness, separate appendix tables with minimum and maximum kappa values across the 10 repetitions are also provided.

Tables 7–21 are attached in a file available at:
<https://rgg.edu.pl/SuppFile/225583/1/>.

References

- Belgiu, M. and Drăguț, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114:24–31, doi:10.1016/j.isprsjprs.2016.01.011.
- Bigdeli, A., Maghsoudi, A., and Ghezelbash, R. (2023). A comparative study of the XGBoost ensemble learning and multilayer perceptron in mineral prospectivity modeling: a case study of the Torud–Chahshirin belt, NE Iran. *Earth Science Informatics*, 17(1):483–499, doi:10.1007/s12145-023-01184-4.
- Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT'92, page 144–152. ACM, doi:10.1145/130385.130401.
- Budach, L., Feuerpfeil, M., Ihde, N., Nathansen, A., Noack, N., Patzlaß, H., Naumann, F., and Harmouch, H. (2022). The effects of data quality on machine learning performance. *arXiv preprint arXiv:2207.14529*.
- Burkholder, A., Warner, T. A., Culp, M., and Landenberger, R. (2011). Seasonal trends in separability of leaf reflectance spectra for *Ailanthus altissima* and four other tree species. *Photogrammetric Engineering and Remote Sensing*, 77(8):793–804, doi:10.14358/PERS.77.8.793.
- Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794. ACM, doi:10.1145/2939672.2939785.
- Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, doi:10.1177/001316446002000104.
- Cracknell, M. J. and Reading, A. M. (2014). Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information. *Computers and Geosciences*, 63:22–33, doi:10.1016/j.cageo.2013.10.008.
- Duro, D. C., Franklin, S. E., and Dubé, M. G. (2012). A comparison

- of pixel-based and object-based image analysis with selected machine learning algorithms for the classification of agricultural landscapes using SPOT-5 HRG imagery. *Remote sensing of environment*, 118:259–272, doi:10.1016/j.rse.2011.11.020.
- Figueroa, R. L., Zeng-Treitler, Q., Kandula, S., and Ngo, L. H. (2012). Predicting sample size required for classification performance. *BMC Medical Informatics and Decision Making*, 12(1), doi:10.1186/1472-6947-12-8.
- Fu, Y., Shen, R., Song, C., Dong, J., Han, W., Ye, T., and Yuan, W. (2023). Exploring the effects of training samples on the accuracy of crop mapping with machine learning algorithm. *Science of Remote Sensing*, 7:100081, doi:10.1016/j.srs.2023.100081.
- Ghayour, L., Neshat, A., Paryani, S., Shahabi, H., Shirzadi, A., Chen, W., Al-Ansari, N., Geertsema, M., Pourmehdi Amiri, M., Gholamnia, M., Dou, J., and Ahmad, A. (2021). Performance Evaluation of Sentinel-2 and Landsat 8 OLI Data for Land Cover/Use Classification Using a Comparison between Machine Learning Algorithms. *Remote Sensing*, 13(7):1349, doi:10.3390/rs13071349.
- Halevy, A., Norvig, P., and Pereira, F. (2009). The Unreasonable Effectiveness of Data. *IEEE Intelligent Systems*, 24(2):8–12, doi:10.1109/mis.2009.36.
- Hand, D. J., Christen, P., and Kirielle, N. (2021). F*: an interpretable transformation of the F-measure. *Machine Learning*, 110(3):451–456, doi:10.1007/s10994-021-05964-1.
- Koppaka, R. and Moh, T.-S. (2020). Machine Learning in Indian Crop Classification of Temporal Multi-Spectral Satellite Image. In *2020 14th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, page 1–8. IEEE, doi:10.1109/imcom48794.2020.9001718.
- Kupidura, P., Kepa, A., and Krawczyk, P. (2024). Comparative analysis of the performance of selected machine learning algorithms depending on the size of the training sample. *Reports on Geodesy and Geoinformatics*, 118(1), doi:10.2478/rgg-2024-0015.
- Kupidura, P. and Niemyski, S. (2024). Analysis of the effectiveness of selected machine learning algorithms in the classification of satellite image content depending on the size of the training sample. *Teledetekcja Środowiska*, 64.
- Labatut, V. and Cherifi, H. (2012). Accuracy measures for the comparison of classifiers. *arXiv preprint arXiv:1207.3790*.
- Li, X., Chen, W., Cheng, X., and Wang, L. (2016). A Comparison of Machine Learning Algorithms for Mapping of Complex Surface-Mined and Agricultural Landscapes Using ZiYuan-3 Stereo Satellite Imagery. *Remote Sensing*, 8(6):514, doi:10.3390/rs8060514.
- Liu, J., Zuo, Y., Wang, N., Yuan, F., Zhu, X., Zhang, L., Zhang, J., Sun, Y., Guo, Z., Guo, Y., Song, X., Song, C., and Xu, X. (2021). Comparative Analysis of Two Machine Learning Algorithms in Predicting Site-Level Net Ecosystem Exchange in Major Biomes. *Remote Sensing*, 13(12):2242, doi:10.3390/rs13122242.
- Mansor, N. S., Awang, H., Malami, S. T. S., Zolkafli, A., Taiye, M. A., and Maulana, H. (2024). *Support Vector Machine for Satellite Images Classification Using Radial Basis Function Kernel Method*, page 301–312. Springer Nature Singapore, doi:10.1007/978-981-99-9589-9_23.
- Maxwell, A., Strager, M., Warner, T., Zégre, N., and Yuill, C. (2014a). Comparison of NAIP orthophotography and RapidEye satellite imagery for mapping of mining and mine reclamation. *GIScience and Remote Sensing*, 51(3):301–320, doi:10.1080/15481603.2014.912874.
- Maxwell, A., Warner, T., Strager, M., Conley, J., and Sharp, A. (2015). Assessing machine-learning algorithms and image- and lidar-derived variables for GEOBIA classification of mining and mine reclamation. *International Journal of Remote Sensing*, 36(4):954–978, doi:10.1080/01431161.2014.1001086.
- Maxwell, A. E. and Warner, T. A. (2015). Differentiating mine-reclaimed grasslands from spectrally similar land cover using terrain variables and object-based machine learning classification. *International Journal of Remote Sensing*, 36(17):4384–4410, doi:10.1080/01431161.2015.1083632.
- Maxwell, A. E., Warner, T. A., Strager, M. P., and Pal, M. (2014b). Combining RapidEye Satellite Imagery and Lidar for Mapping of Mining and Mine Reclamation. *Photogrammetric Engineering and Remote Sensing*, 80(2):179–189, doi:10.14358/pers.80.2.179-189.
- Mousavinezhad, M., Feizi, A., and Aalipour, M. (2023). Performance Evaluation of Machine Learning Algorithms in Change Detection and Change Prediction of a Watershed's Land Use and Land Cover. *International Journal of Environmental Research*, 17(2), doi:10.1007/s41742-023-00518-w.
- Powers, D. (2007). Evaluation: From precision, recall and F-factor to ROC, informedness, markedness and correlation. Technical report, Techn. Rep. SIE-07-001, Flinders University, Adelaide, Australia.
- Ramezan, C. A., Warner, T. A., Maxwell, A. E., and Price, B. S. (2021). Effects of Training Set Size on Supervised Machine-Learning Land-Cover Classification of Large-Area High-Resolution Remotely Sensed Data. *Remote Sensing*, 13(3):368, doi:10.3390/rs13030368.
- Raudys, S. and Jain, A. (1991). Small sample size effects in statistical pattern recognition: recommendations for practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):252–264, doi:10.1109/34.75512.
- Richards, J. A. and Jia, X. (2006). *Remote Sensing Digital Image Analysis: An Introduction*. Springer Berlin Heidelberg, doi:10.1007/3-540-29711-1.
- Seydi, S. T., Kanani-Sadat, Y., Hasanlou, M., Sahraei, R., Chanussot, J., and Amani, M. (2022). Comparison of Machine Learning Algorithms for Flood Susceptibility Mapping. *Remote Sensing*, 15(1):192, doi:10.3390/rs15010192.
- Shang, M., Wang, S.-X., Zhou, Y., and Du, C. (2018). Effects of Training Samples and Classifiers on Classification of Landsat-8 Imagery. *Journal of the Indian Society of Remote Sensing*, 46(9):1333–1340, doi:10.1007/s12524-018-0777-z.
- Shih, H.-c., Stow, D. A., and Tsai, Y. H. (2018). Guidance on and comparison of machine learning classifiers for Landsat-based land cover and land use mapping. *International Journal of Remote Sensing*, 40(4):1248–1274, doi:10.1080/01431161.2018.1524179.
- Sim, J. and Wright, C. C. (2005). The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements. *Physical Therapy*, 85(3):257–268, doi:10.1093/ptj/85.3.257.
- Sobieraj, J., Fernández, M., and Metelski, D. (2022). A Comparison of Different Machine Learning Algorithms in the Classification of Impervious Surfaces: Case Study of the Housing Estate Fort Bema in Warsaw (Poland). *Buildings*, 12(12):2115, doi:10.3390/buildings12122115.
- Volke, M. I. and Abarca-Del-Rio, R. (2020). Comparison of machine learning classification algorithms for land cover change in a coastal area affected by the 2010 Earthquake and Tsunami in Chile. *Natural Hazards and Earth System Sciences Discussions*, 2020:1–14, doi:10.5194/nhess-2020-41.
- Zhao, Z., Islam, F., Waseem, L. A., Tariq, A., Nawaz, M., Islam, I. U., Bibi, T., Rehman, N. U., Ahmad, W., Aslam, R. W., Raza, D., and Hatamleh, W. A. (2024). Comparison of Three Machine Learning Algorithms Using Google Earth Engine for Land Use Land Cover Classification. *Rangeland Ecology and Management*, 92:129–137, doi:10.1016/j.rama.2023.10.007.
- Zheng, W. and Jin, M. (2020). The Effects of Class Imbalance and Training Data Size on Classifier Learning: An Empirical Study. *SN Computer Science*, 1(2), doi:10.1007/s42979-020-0074-0.