sciendo

Reports on Geodesy and Geoinformatics, 2024, Vol. 118, pp. 53–69

DOI: 10.2478/rgg-2024-0015
Received: 13 March 2024 / Accepted: 29 July 2024
Published online: 23 September 2024

ORIGINAL ARTICLE

Comparative analysis of the performance of selected machine learning algorithms depending on the size of the training sample

Przemysław Kupidura [1*], Agnieszka Kępa [1] and Piotr Krawczyk [2]

[1]Faculty of Geodesy and Cartography, Warsaw University of Technology, Pl. Politechniki 1, 00-661, Warsaw, Poland
[2]Orbitile Ltd., Potułkały 6B/4, 02-791, Warsaw, Poland

[*]przemyslaw.kupidura@pw.edu.pl

Abstract

The article presents an analysis of the effectiveness of selected machine learning methods: Random Forest (RF), Extreme Gradient Boosting (XGB), and Support Vector Machine (SVM) in the classification of land use and cover in satellite images. Several variants of each algorithm were tested, adopting different parameters typical for each of them. Each variant was classified multiple (20) times, using training samples of different sizes: from 100 pixels to 200,000 pixels. The tests were conducted independently on 3 Sentinel-2 satellite images, identifying 5 basic land cover classes: *built-up areas*, *soil*, *forest*, *water*, and *low vegetation*. Typical metrics were used for the accuracy assessment: Cohen's kappa coefficient, overall accuracy (for whole images), as well as F-1 score, precision, and recall (for individual classes). The results obtained for different images were consistent and clearly indicated an increase in classification accuracy with the increase in the size of the training sample. They also showed that among the tested algorithms, the XGB algorithm is the most sensitive to the size of the training sample, while the least sensitive is SVM, which achieved relatively good results even when using training samples of the smallest sizes. At the same time, it was pointed out that while in the case of RF and XGB algorithms the differences between the tested variants were slight, the effectiveness of SVM was very much dependent on the gamma parameter – with too high values of this parameter, the model showed a tendency to overfit, which did not allow for satisfactory results.

Key words: efficiency, classification, machine learning, remote sensing, satellite imagery, training sample size

1 Introduction

The classification of land use/land cover (LULC) is one of the most important tasks in remote sensing. The development of machine learning (ML) algorithms, which we have witnessed for many years, has led to the widespread adoption of the (semi)automatic classification of aerial and satellite images. However, this proliferation of options also makes choosing the optimal solution problematic.

Numerous studies and publications have focused on the accuracy of classification models based on various ML algorithms. Let's briefly mention some of them, illustrating the ambiguity prevailing in this field.

We can start with the publication by Seydi et al. (2022), which compares the effectiveness of identifying flood-prone areas using methods such as deep neural networks (DNN), support vector machines (SVM), Random Forest (RF), Extreme Gradient Boosting (XGB), and individual decision trees (DT). Among these, XGB achieved the best results, followed by DNN and SVM, clearly outperforming RF and DT. The effectiveness of XGB is also demonstrated by research conducted by Bigdeli et al. (2023), which highlights its superiority compared to multi-layer perceptrons (MLP) and artificial neural networks (ANN). Similarly, Liu et al. (2021) found XGB to perform better than RF. Additionally, Ding (2024) study on precise tracking of peat soil carbon dioxide emissions indicates that XGB is highly effective, surpassing long short-term memory (LSTM) recurrent neural networks (RNN) and performing comparably to SVM. Moreover, SVM is one of the most popular machine learning methods, frequently used in remote sensing and various comparative

analyses. For instance, in the study by Koppaka and Moh (2020), SVM outperformed Random Forest (RF), convolutional neural networks (CNN), RNN-LSTM, and RSN with Gated Recurrent Unit (GRU) in crop identification. There is further evidence of SVM's superior effectiveness: Ghayour et al. (2021) demonstrated SVM's superiority over ANN, minimum distance (MD), Mahalanobis (MH), and maximum likelihood (MLC). Similarly, Sobieraj et al. (2022) compared SVM with RF and MLC, showing SVM's superior performance to RF, which, in turn, outperformed decision trees (DT), with DT performing better than MLC. Additionally, Shih et al. (2018) analysis comparing SVM, DT, ANN, and RF revealed varying superiority between SVM and DT, depending on the analyzed data. It's worth mentioning Maxwell et al. (2014a,b, 2015), which also highlight SVM's greater effectiveness compared to RF. However, studies by Maxwell and Warner (2015) and Burkholder et al. (2011) in comparative analyses show RF's advantage over SVM. The ambiguity in this meta-analysis is further deepened by a series of publications demonstrating similar accuracy for models based on SVM and RF. For example, we can mention the publication by Volke and Abarca-Del-Rio (2020), who demonstrated the superiority of SVM and RF over MLC. Li et al. (2016), and Duro et al. (2012) also found both methods to exhibit similar effectiveness, surpassing DT in this regard. In another study by Cracknell and Reading (2014), SVM and RF achieved better accuracy than ANN. Additionally, Maxwell et al. (2018) research shows varying effectiveness, with SVM outperforming RF in some cases (also compared to DT and ANN). In the literature, there are other examples of RF's clear superiority, such as when compared to SVM and DT Zhao et al. (2024) or MLC (Mousavinezhad et al., 2023).

The above overview indicates a lack of unequivocal certainty regarding the selected methods. Depending on the type of classification task, the processed data, and certainly the training data, comparing different ML methods can yield varying results. Analyses specifically focused on methodological effectiveness concerning these task aspects are relatively scarce. However, let's highlight some relevant studies. Shang et al. (2018) in their work analyzed the impact of training sample size on classification accuracy using Landsat-8 images. The methods considered included SVM, RF, DT, MLC, and Naive Bayes (NB). SVM achieved the best performance, with overall classification accuracy increasing alongside the training sample size for all algorithms, except for NB. Ramezan et al. (2021) study compared SVM, RF, Gradient Boost (GBM), k-Nearest Neighbors (kNN), Single-Layer Perceptron (SLP) ANN, and Learning Vector Quantization (LVQ) for LULC classification. They used training data ranging from very small (40 pixels) to very large (10,000 pixels). RF and GBM achieved the best results, even with a slight decrease in accuracy as the training sample size decreased. Notably, SVM and ANN, especially the latter, were sensitive to this phenomenon. Another example of the efficiency analysis of selected methods (SVM, RF, and MLC) can be found in (Kupidura and Niemyski, 2024). The study showed an increase in classification accuracy with an increase in the size of the training sample, but it also showed different dynamics of this increase, depending on the method. With complete training data, MLC showed the highest effectiveness, but its effectiveness using small training samples was definitely the poorest. SVM was characterized by the greatest immunity to a small size of training sample. Fu et al. (2023) – their analysis explored the impact of training sample size on crop identification accuracy using RF. While the literature contains numerous examples of similar analyses, this study specifically focused on crop identification. Zheng and Jin (2020) in their research compared DT, RF, XGB. The results highlighted the strong dependence of classification effectiveness on the training sample size, with DT showing weaker performance. In summary, the choice of ML method should consider the specific context, available data, and the desired trade-offs between accuracy and computational efficiency.

Budach et al. (2022), in their comprehensive analysis, also demonstrate the significant impact of training data quality on ML accuracy. In a slightly different context – regression rather than classification – they investigated the influence of training data on the effectiveness of regression trees, kNN, ANN, and RF. An interesting study by Figueroa et al. (2012) explored the prediction of the necessary sample size for achieving the expected classification accuracy. Additionally, research and discussions regarding the impact of training sample size on ML effectiveness can be found in publications such as Raudys and Jain (1991) and Halevy et al. (2009).

From the overview presented above, we can draw three key conclusions:

i. SVM, RF, and XGB are among the most popular and effective ML methods. While SVM and RF frequently appear in various studies, XGB often demonstrates high performance and is often ranked as the best choice in comparisons.
ii. The size of the training sample significantly impacts ML effectiveness. Larger training samples generally lead to better performance, but this relationship is not uniform across all methods.
iii. The nature of this impact varies for different methods, making it less straightforward to generalize. Some methods are more sensitive to changes in sample size than others.

The above conclusions are both the motivation for undertaking the research described in this article and also determine the adopted methodology of this research. The 3 most popular or most effective ML methods mentioned in the previous paragraph, SVM, RF and XGB, were analyzed. The study itself, however, concerned the effectiveness of these methods in the classic task of classifying basic LULC classes and analyzing the dynamics of this effectiveness.

2 Materials and Methods

The aim of our research was to analyze the dynamics of the effectiveness of selected ML methods used with great success in remote sensing. This goal dictated the methodology of the analysis. It was not an analysis of the accuracy of individual classes, or the dependence of classification accuracy on selected data, etc.

Below we present the characteristics of the imagery on which the described research was conducted, the methods tested, and the methodology of the research itself.

2.1 Methodology

The general scheme of the methodology is presented in Figure 1.

The methodology written in points is as follows:

i. Selection of test areas.
ii. Classification of land cover based on visual interpretation to prepare training and test data.
iii. Drawing training samples of different sizes.
iv. Performing classification using cross-validation with selected ML methods.
v. Accuracy analysis and interpretation of results .

Details regarding each stage of the study are presented in the following subsections.

2.2 Test imagery

The analysis used fragments of Sentinel-2 satellite scenes from the area of Warsaw (Poland) and its surroundings. To ensure diversity and statistical credibility of the data, photos from 3 different dates were selected for analysis, and within the scenes, 3 areas of 4 km x 4 km were chosen. They differ in the characteristics of land cover. Individual areas from the same dates were treated in the classification and test process as one whole. This means that both the training data, the model, and the test data were common to all of them. The selected areas are presented in Figure 2, while the

Table 1. Details of test imagery

Image symbol	Date of aquisition	Imagery type	Spectral bands	GSD	Number of pixels
A	19.04.2020	Sentinel-2 L2A	2, 3, 4, 8 (blue, green, red, near infrared)	10 m	480 000
B	09.05.2020				
C	22.08.2020				

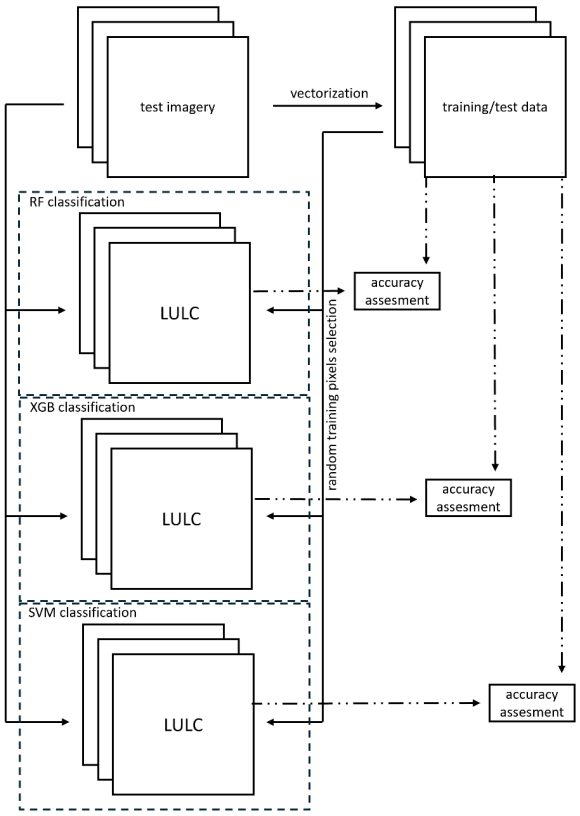

Figure 1. General scheme of the methodology for analyzing the effectiveness of ML algorithms

details of the photos are presented in Table 1. Figure 3 presents 3 fragments of the satellite scene that make up a single test image.

2.3 Definition of the analyzed land cover classes

In order to ensure full credibility of the obtained results, the entire test areas were vectorized. Five land cover classes were distinguished:

i. *built-up area* – building complexes or other architectural objects e.g. roads;
ii. *soil* – crop fields not covered by vegetation, river patches, sand mines;
iii. *water* – water bodies e.g. ponds, lakes, reservoirs or watercourses;
iv. *forest* – forests, bushy areas;
v. *low vegetation* – meadows, cultivated fields covered with vegetation.

These classes were identified through visual interpretation, independently on each of the 9 images (3 test areas and 3 dates). An example result of such vectorization for one of the areas based on a single photo is presented in Figure 4.

2.4 Selection of training samples

In order to ensure the appropriate precision of the analysis, the classification was performed in 20 variants of the sample size – from 100 to 200,000 for all classes in total. The number of samples for individual classes varied, as this was proportional to the overall share of a given class in the image. The drawing of training pixels took place randomly and independently for each of the dates of the images being taken, which is why their number for individual classes may also vary, mainly due to the changing proportional share of individual classes in the entire analyzed area, but also due to the random nature of the selection of training pixels. Details regarding individual variants: numbers of training pixels – general and for individual classes for individual variants and dates are presented in Tables 2–4.

2.5 Brief presentation of the tested ML algorithms

In the article, we present the results of the comparison of three popular ML algorithms:

- Random Forests (RF),
- Extreme Gradient Boosting – XGBoost (XGB),
- Support Vector Machine (SVM).

Random Forests

RF (Ho, 1995, 1998; Breiman, 2001) is an ML method that has been gaining popularity in remote sensing image processing for some time due to its high effectiveness, resistance to overfitting, and relatively high resistance to the quality of training data (Belgiu and Drăguţ, 2016). It is an ensemble learning method – it involves the use of a large number of individual decision trees. Single decision trees are effective in extracting classes of a multimodal nature, while at the same time suffering from a tendency to overfit. The composition of a random forest – many decision trees built on randomly limited data – helps to significantly reduce the tendency to overfit, while maintaining the advantages of individual trees. This makes it applicable in the classification of various types of remote sensing data. The key factor influencing the accuracy and computational efficiency of the model is the number of decision trees: increasing the number of trees can make the model less prone to overfitting (Liaw et al., 2002). On the other hand, it increases the computational complexity of the model, which can slow down the training and classification process. The second important parameter is the number of features in each split in a single decision tree. Many studies use the value of this parameter equal to the square root of the number of variables (Duro et al., 2012).

XGBoost

XGB (Chen and Guestrin, 2016) is also a method of ensemble learning, also based on decision trees. The difference is that while in the case of RF the trees are generated independently, in XGB they are generated sequentially, and each new tree corrects the errors of the previous one. The correction is controlled by the gradient of the loss function. The final result is the sum of the predictions of all trees, with models that handle errors better having a greater impact on the final result. The process includes a regularization mechanism that penalizes for overcomplicating models or assigning too-large weights to individual features, which avoids overfitting models

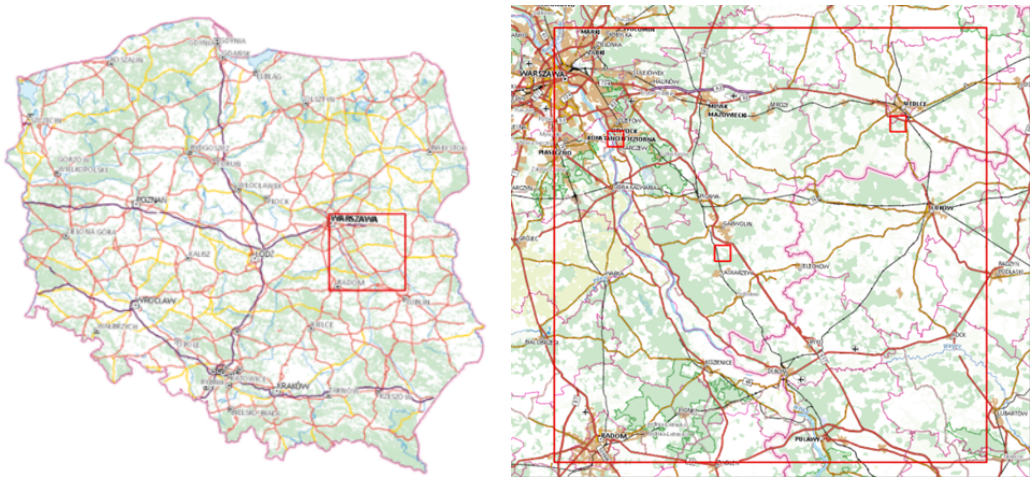

Figure 2. Location of test areas

Figure 3. Example of test images, color composition RGB 843

Table 2. Training sample size for image A (taken on 19.04.2020): total (in pixels and in percentage, relative to the size of the test area) and for individual classes (in pixels)

overall [px]	**overall relative** [%]	**built-up area** [px]	**soil** [px]	**forest** [px]	**water** [px]	**low vegetation** [px]
100	0.02%	16	35	25	5	19
200	0.04%	36	69	41	9	45
300	0.06%	45	127	56	17	55
500	0.10%	78	219	91	26	86
700	0.15%	116	303	124	32	125
1000	0.21%	160	441	171	46	182
1500	0.31%	252	650	255	70	273
2000	0.42%	346	875	327	94	358
3000	0.63%	531	1281	494	141	553
5000	1.04%	892	2139	793	241	935
7000	1.46%	1258	2999	1106	348	1289
10000	2.08%	1792	4275	1585	490	1858
15000	3.13%	2689	6415	2392	730	2774
25000	5.21%	4411	10728	4006	1204	4651
30000	6.25%	5279	12923	4814	1414	5570
50000	10.42%	8749	21552	8026	2336	9337
75000	15.63%	13212	32329	11878	3465	14116
100000	20.83%	17632	43121	15909	4604	18734
150000	31.25%	26671	64637	23764	6778	28150
200000	41.67%	35635	86108	31641	9020	37596

Table 3. Training sample size for image B (taken on 09.05.2020): total (in pixels and in percentage, relative to the size of the test area) and for individual classes (in pixels)

overall [px]	**overall relative** [%]	**built-up area** [px]	**soil** [px]	**forest** [px]	**water** [px]	**low vegetation** [px]
100	0.02%	14	29	21	4	32
200	0.04%	33	59	44	9	55
300	0.06%	44	96	66	15	79
500	0.10%	75	148	114	24	139
700	0.15%	99	231	151	30	189
1000	0.21%	151	331	213	34	271
1500	0.31%	228	496	308	46	422
2000	0.42%	302	670	407	65	556
3000	0.63%	471	974	600	99	856
5000	1.04%	779	1643	963	164	1451
7000	1.46%	1105	2303	1347	245	2000
10000	2.08%	1621	3310	1919	313	2837
15000	3.13%	2405	4943	2852	473	4327
25000	5.21%	4018	8253	4753	810	7166
30000	6.25%	4812	9936	5685	982	8585
50000	10.42%	8017	16562	9435	1615	14371
75000	15.63%	11990	24819	14230	2510	21451
100000	20.83%	15928	32999	18919	3441	28713
150000	31.25%	23906	49576	28435	4048	43035
200000	41.67%	31902	66286	37836	6670	57306

Table 4. Training sample size for image C (taken on 22.08.2020): total (in pixels and in percentage, relative to the size of the test area) and for individual classes (in pixels)

overall [px]	**overall relative** [%]	**built-up area** [px]	**soil** [px]	**forest** [px]	**water** [px]	**low vegetation** [px]
100	0.02%	9	24	33	2	32
200	0.04%	23	45	70	5	57
300	0.06%	38	65	102	12	83
500	0.10%	72	96	157	28	147
700	0.15%	97	129	222	39	213
1000	0.21%	140	194	309	53	304
1500	0.31%	215	296	475	77	437
2000	0.42%	287	395	630	103	585
3000	0.63%	437	594	930	147	892
5000	1.04%	741	972	1561	227	1499
7000	1.46%	1003	1392	2212	312	2081
10000	2.08%	1410	1982	3139	477	2992
15000	3.13%	2149	2959	4744	695	4453
25000	5.21%	3618	4947	7912	1186	7337
30000	6.25%	4373	5922	9412	1434	8859
50000	10.42%	7327	9922	15623	2382	14746
75000	15.63%	10995	14888	23512	3612	21993
100000	20.83%	14672	19850	31367	4820	29291
150000	31.25%	22019	29617	47031	7194	44139
200000	41.67%	29226	39520	62674	9649	58931

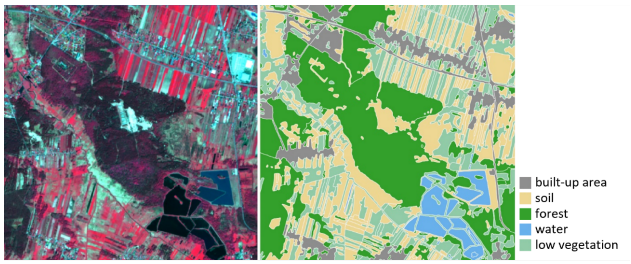

Figure 4. Example of vectorization result of a fragment of one of the test images: Image B, acquisition date 09.05.2020.

Table 5. Tested classification variants

ML algorithm	parameter	tested parameters	variant description
RF	The number of decision trees	50 100	RF50 RF100
XGB	The number of node levels in a single decision tree	3 5 7 9	XGBoost3 XGBoost5 XGBoost7 XGBoost9
SVM	Gamma – parameter defining the influence reach of a single training example	0.00003 0.00001 0.000003	SVM03 SVM01 SVM003

(more effectively than in the case of RF: Allwright (2023); Kumar (2023)). It is assessed that XGB is characterized by greater efficiency for large data sets than RF (Kapoor and Perrone, 2021).

There are many types of options and parameters that affect the operation of the XGBoost algorithm. One of them is the booster type. We distinguish Gradient Boosting Tree (GBT), Gradient Boosting Linear, and Dart. The GBT booster (this type of algorithm was used in the studies presented below) supports such parameters as number of trees, learning rate, and maximum tree depth (maximum number of node levels in a single tree).

SVM

SVM (Boser et al., 1992; Cortes and Vapnik, 1995) is an algorithm based on the search for hyperplanes in the feature space that optimally divide data into different classes. Originally designed to distinguish classes with linear separation, the use of kernel functions and kernel tricks allows it to also effectively solve problems related to non-linear separation (Schölkopf, 2002). It is a method considered especially effective for data sets with a large number of features/dimensions (Nalepa and Kawulok, 2018).

In the case of the SVM method, we also deal with a large number of options and parameters. The most important include the regularization parameter C, which helps control the size of the margin, the type of kernel, the gamma parameter, which determines the influence of a single training element (a higher gamma value leads to a better fit of the model to the training data), the maximum number of iterations, and the tolerance (difference between loss or result between successive iterations) for the stopping criterion.

Tested classification variants

For individual algorithms, some options and parameters were assumed to be constant (these were default values or those specified in the literature analysis as optimal for the conducted experiment). At the same time, to assess a wider spectrum of possibilities of these algorithms, variable parameters were chosen, which were used to test different variants of the process. The list of these variable parameters and their values are presented in Table 5.

The list of options and parameters for individual algorithms, which remained unchanged in various variants, is as follows:

- for RF:
  - number of features in each split in a single decision tree: square root of the number of variables,
- for XGBoost:
  - booster type: Gradient Boosting Tree;
  - number of decision trees: 100;
  - learning rate: 0.1;
- for SVM:
  - regularization parameter C: 1.0;
  - kernel type: radial basis function;
  - maximum number of iterations: 1000;
  - tolerance: 1e-04.

It should be emphasized that the conducted experiment was not aimed at analyzing the impact of the parameters of individual methods, but at a general analysis of the effectiveness of selected algorithms. Therefore, it largely relied on default options and parameters.

2.6 Accuracy assessment

In order to determine the accuracy of individual classification variants, the following accuracy metrics, 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);
- generally for the entire classification: kappa coefficient (Cohen, 1960) and overall accuracy (or overall success rate) (Labatut and Cherifi, 2012) to determine the effectiveness o for precision and recall (Hand et al., 2021):

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{1}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{2}$$

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

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

$$\text{F1} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{3}$$

For Kappa index (Sim and Wright, 2005):

$$\text{kappa} = \frac{P_o - P_c}{1 - P_c}, \tag{4}$$

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

For overall accuracy:

$$OA = \frac{C_c}{C_o}, \tag{5}$$

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

The entire test images were used for the accuracy analysis. The number of test pixels was 480,000. Individual classes were identified in the photointerpretation process.

3 Results and discussion

Below, we present the results of the conducted research and their analysis. First, we provide an analysis performed for individual classes (primarily based on F1-score values), and finally, an overall classification analysis (mainly relying on the kappa coefficient). Due to the extensive number of conducted analyses and the resulting comprehensive tables containing analysis outcomes for specific classes, they have been included in the Appendix.

Before proceeding with the presentation of results and analysis, we want to emphasize that the purpose of this research was to analyze the performance of individual methods, rather than comparing the accuracies obtained for different classes based on image acquisition dates. These differences may arise from various factors, such as the varying characteristics of individual classes on different dates, their consistency, or similarities to other classes. However, we want to focus on analyzing the differences in results obtained by specific ML algorithms.

3.1 Built-up area

The results obtained for the *built-up area* class are presented in Figure 5 and Tables 9–11 in the Appendix.

Firstly, a clear trend is visible in virtually every analyzed scenario: as the size of the training sample increases, classification accuracy improves, although the dynamics of this trend can vary. When using the largest training samples, the best results were achieved with XGBoost7, which was slightly better (with an F1 score difference at the level of thousandths) than XGBoost9 and XGBoost5. Applying XGBoost3 consistently yields noticeably weaker results compared to scenarios with more tree levels, likely indicating that XGBoost3 is too inflexible. However, these differences are still small, at the level of thousandths or hundredths. Additionally, there is a consistent (and equally small) advantage of XGBoost models over Random Forest (RF). When comparing the two RF models, RF100 performs better due to its larger number of decision trees, although the difference is only at the level of thousandths.

With a smaller training sample size, the RF variants perform slightly better than XGBoost (though there are occasional exceptions). One of the SVM variants – with the smallest *gamma* parameter value: SVM003 – is worth commenting on. For the least numerous training sample, it consistently gives much better results than the other models (from a few hundredths to even over 0.1). This advantage consistently decreases as the size of the training sample increases. In the case of the largest training samples, the accuracy obtained using SVM003 is clearly worse than the best results obtained (by 0.02–0.03).

The accuracies obtained using SVM with greater *gamma* parameter values are noticeably lower. The values obtained for SVM03 deviate from the results of other methods from about 0.1 to even 0.2 – depending, on the size of the training sample, among other factors. The SVM01 variant gives slightly better results, but still clearly lags behind the results obtained by other models. This indicates the significant importance of overfitting that occurs in SVM models when using too large a *gamma* value (Wainer and Fonseca, 2021). This is also confirmed by the ambiguous dependence of classification accuracy on the size of the training sample. While with the use of other variants it increases quite consistently with the training sample, in these two cases (especially SVM03) we are dealing with large fluctuations of this dependence. This seems to indicate a large dependence on the quality – or "purity" – of the training sample and problems with generalizing the issue, which would indicate overfitting. It should be noted that the built-up area class is demanding to extract due to its large diversity and potential similarity of features to other land cover classes.

3.2 Soil

The results obtained for the *soil* class are presented in Figure 6 and in Tables 12-14 in the Appendix.

The results show similar tendencies to the case of the *built-up area* class. A clear increase in accuracy with the size of the training sample is visible, although the fluctuations of this dynamic are less noticeable (than with the built-up area) in the case of the SVM03 and SVM01 variants, which still perform the worst among the analyzed, though in this case the difference is not that great. This may be due to the lesser diversity of pixel values of this class, compared to the *built-up area* class, and therefore less exposure to overfitting.

The highest accuracy with the use of the largest training sample was again obtained using the XGBoost algorithm, in the following order: XGBoost7, XGBoost9, XGBoost5, XGBoost3, with the differences being, again, very small (at the level of thousandths of F1). Slightly worse (about one hundredth of F1) was obtained using RF (RF100 again imperceptibly better than RF50).

Again, in the case of the smallest training sample, the best results were obtained using the SVM003 variant, although not so unambiguously: in 2 cases (Images A and B) a clear advantage of this variant is noticeable, but in the third case (Image C) it gives slightly worse results than other best variants (in this case it is the XGBoost5 variant, and the difference between them is 0.012 F1).

3.3 Forest

The results obtained for the *forest* class are presented in Figure 7 and in Tables 15-17 in the Appendix.

In the case of this class, essentially one significant change is noticeable, compared to the two classes analyzed above. This concerns the efficiency of SVM variants: here, such a large drop in accuracy when using larger *gamma* parameter values is not noticeable. This may be due to a certain distinctiveness of pixel values representing forests, as well as their relative homogeneity. Such features of the analyzed class could prevent the effect of overfitting and, as a result, a decrease in accuracy. Still generally better results (among SVM variants) are obtained using variants with a smaller *gamma* value, but these differences are much smaller. Generally, the efficiency of all 3 SVM variants is relatively higher: with the largest number of training samples, the accuracy obtained using the SVM003 variant and, to a lesser extent, SVM01, are comparable with other best methods (again these are XGBoost7, XGBoost5 and XGBoost9 variants – with F1 value differences at the level of 0.001). However, the previously observed advantage of SVM003 over other algorithms when using a small training sample also becomes the share of SVM01 here. And it is generally larger than in the case of other classes.

3.4 Water

The results obtained for the *water* class are presented in Figure 8 and in Tables 18-20 in the Appendix.

Here we can observe very strong fluctuations in the dynamics of the effectiveness of SVM variants: SVM03 and, to a lesser extent, SVM01, i.e., with larger *gamma* parameter values. It is worth noting that in the case of this class, individual samples were very few due to the small share of water surfaces in the test area. While the other variants produced stably effective models, these 2 mentioned variants probably overfitted. Interestingly, this does not apply to one of the test images: Image C. The analysis of images allows us to assume that this is a matter of greater homogeneity of pixels of this class in the image from this date. After all, the results obtained for this image very much resemble the results obtained for the forest class. Besides, one can notice a high accuracy of classification, the highest with the most numerous training samples, with differences – excluding SVM03 and SVM01 – both between variants and training data of different sizes. This is probably due to the fact that this is a class spectrally significantly different from the other analyzed classes.

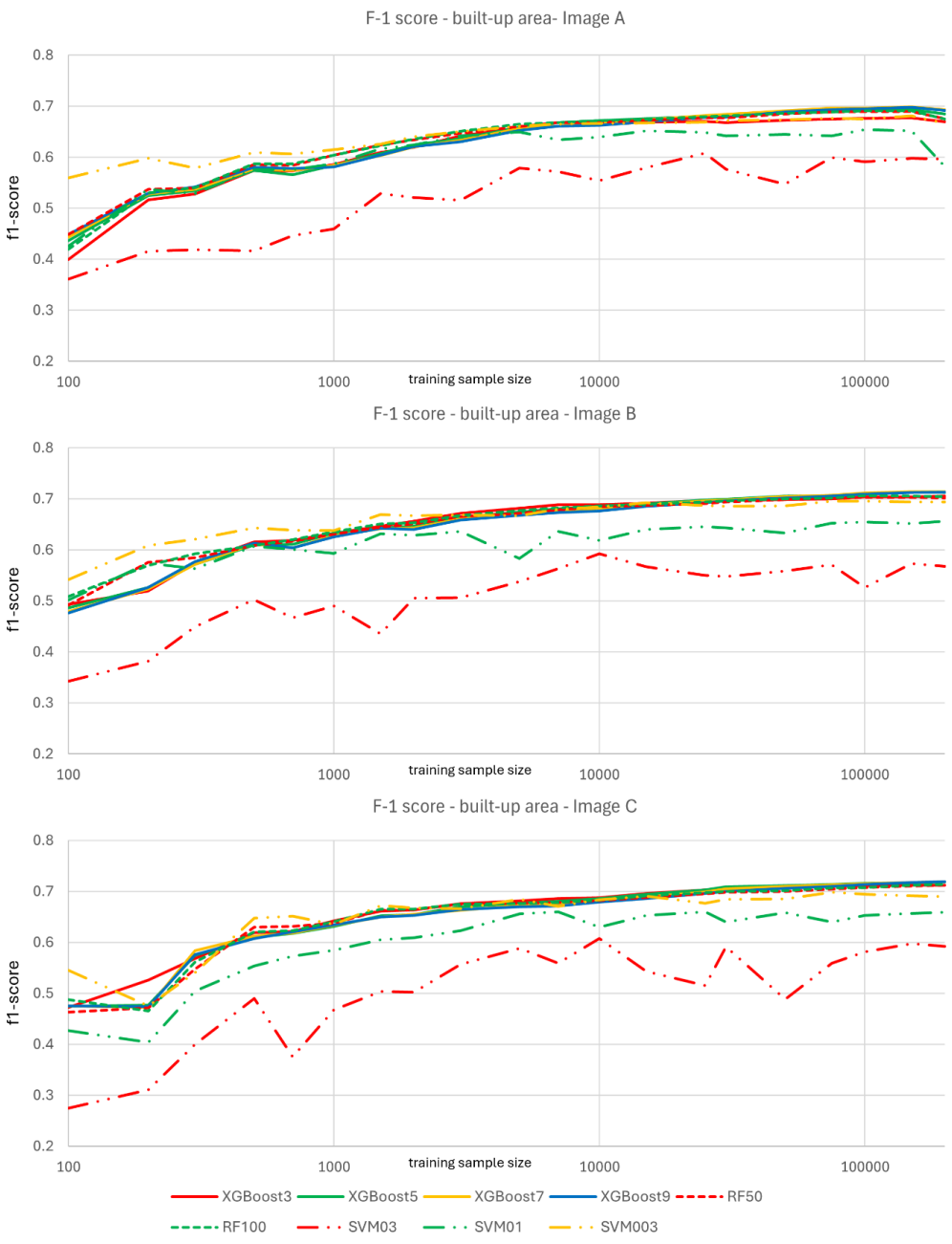

Figure 5. Diagrams of F-1 score values for the built-up area class, in relation to the size of the training sample (the scale of the training sample size is logarithmic)

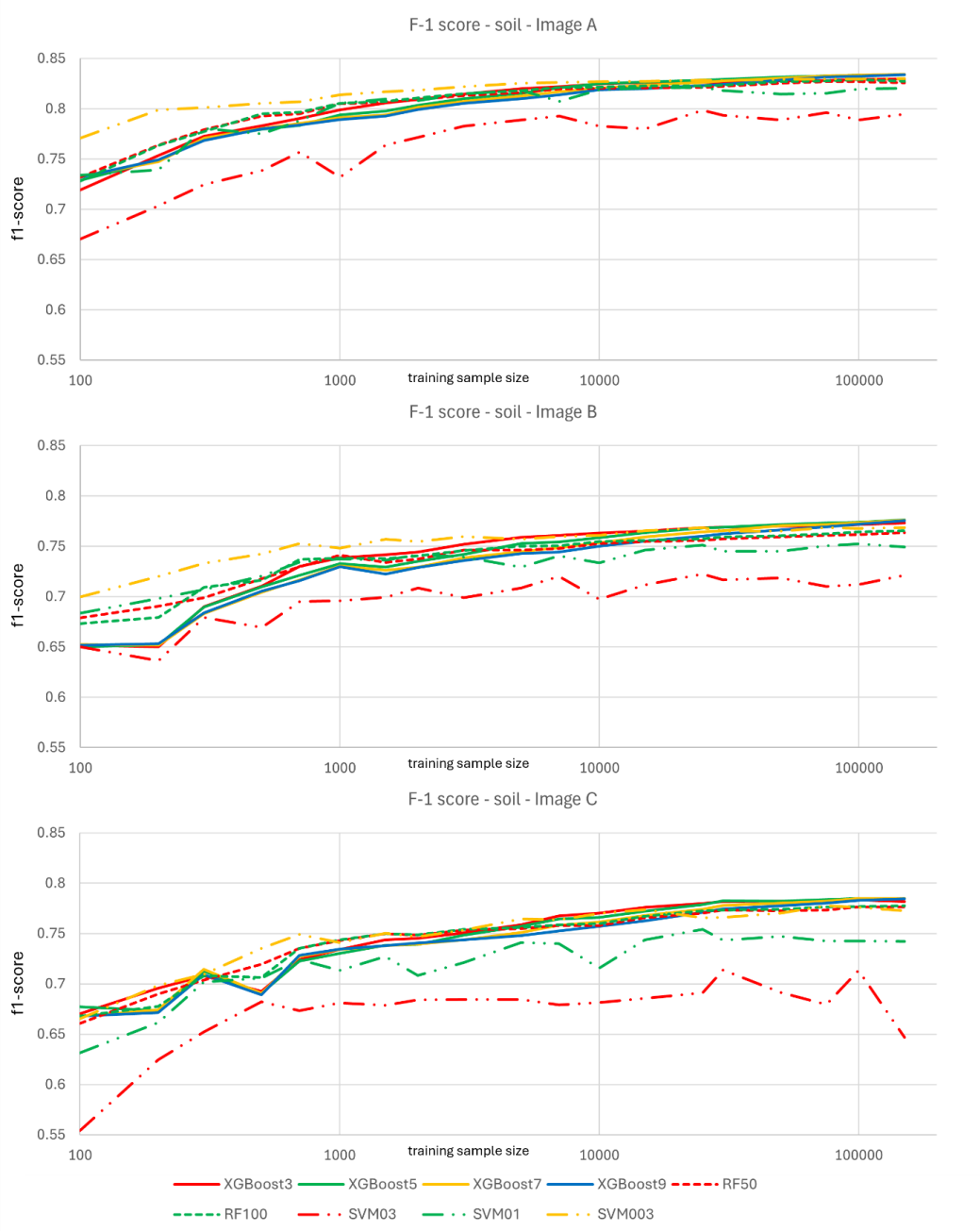

Figure 6. Diagrams of F-1 score values for the soil class, in relation to the size of the training sample (the scale of the training sample size is logarithmic)

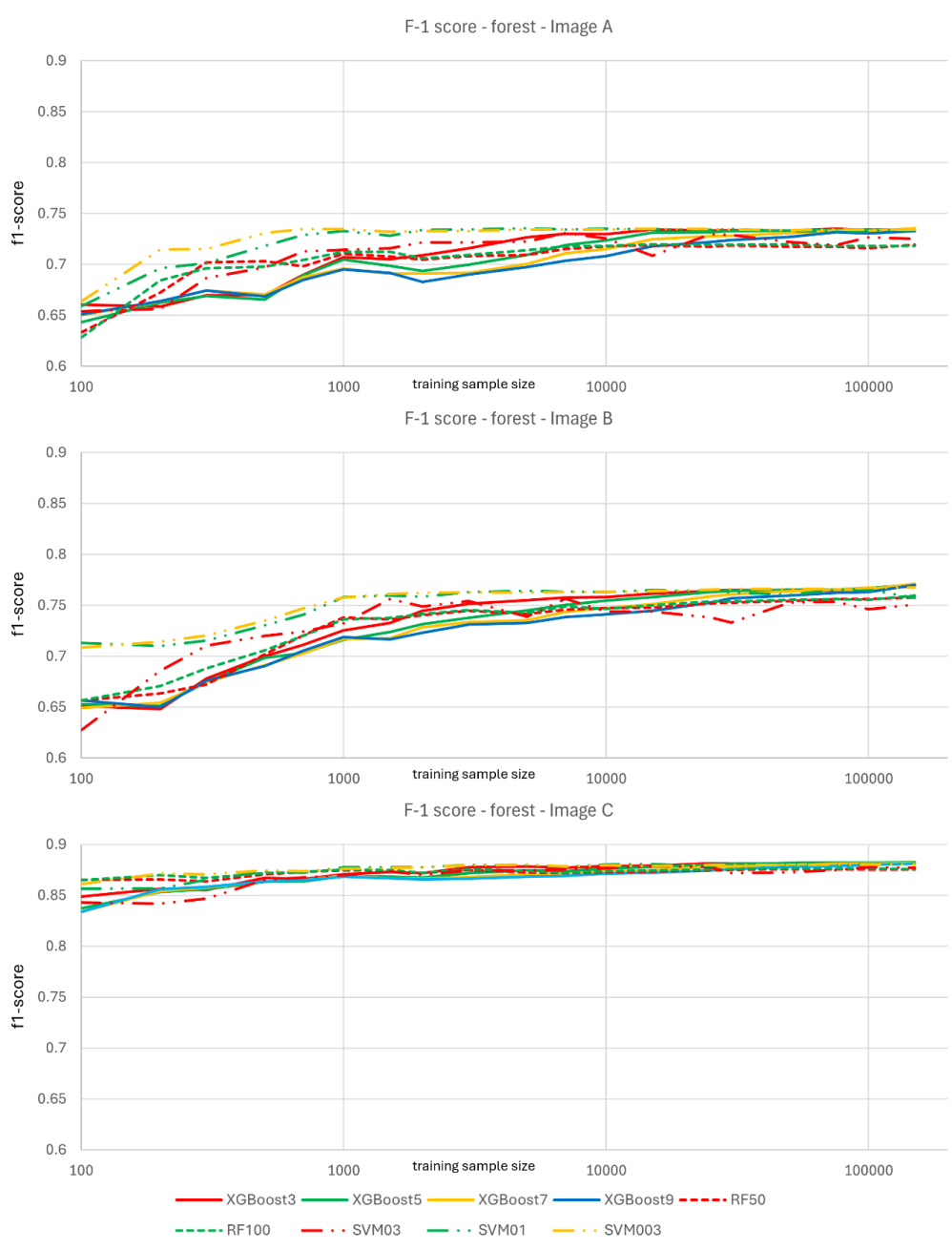

Figure 7. Diagrams of F-1 score values for the forest class, in relation to the size of the training sample (the scale of the training sample size is logarithmic)

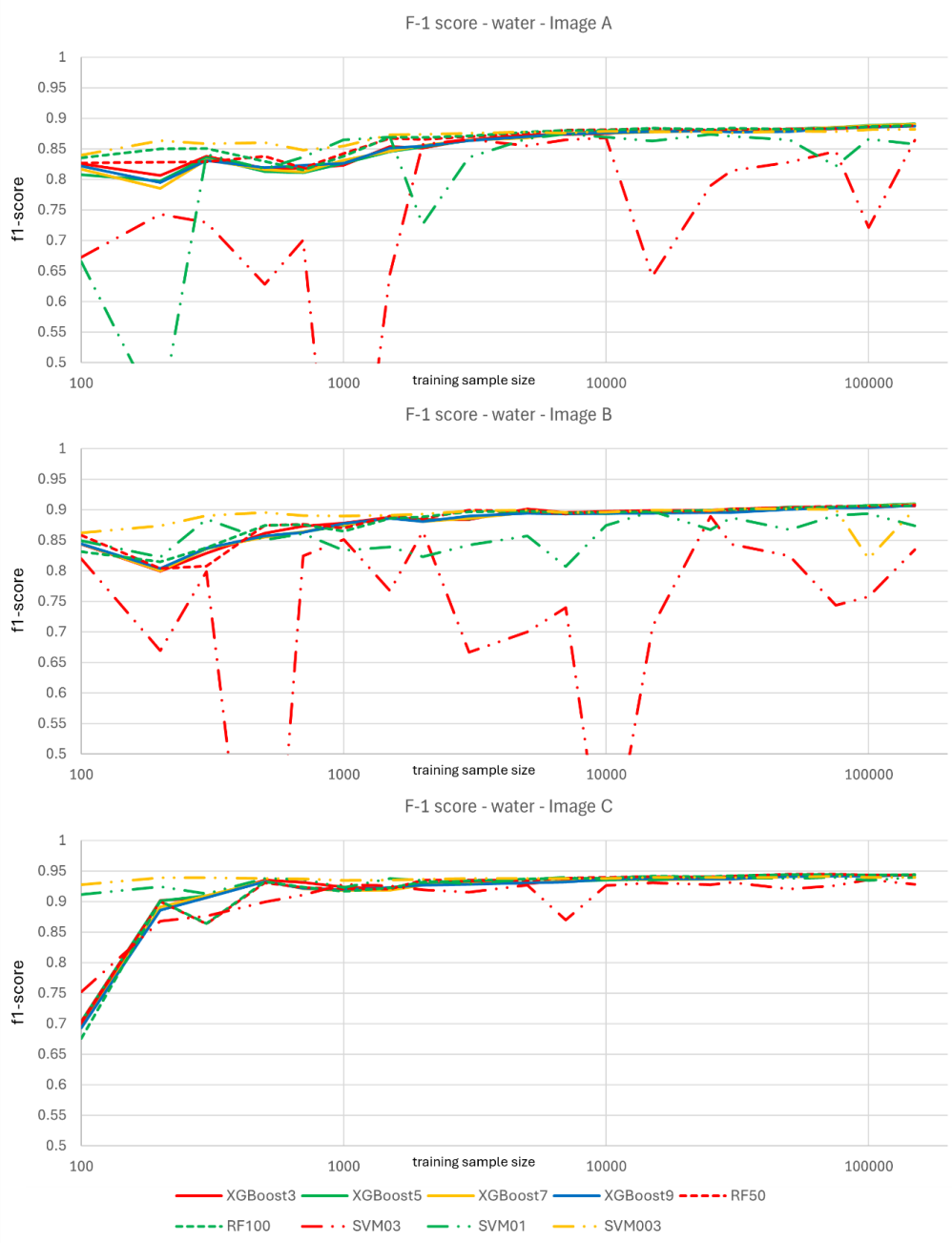

Figure 8. Diagrams of F-1 score values for the water class, in relation to the size of the training sample (the scale of the training sample size is logarithmic)

3.5 Low vegetation

The results obtained for the *low vegetation* class are presented in Figure 9 and in Tables 21-23 in the Appendix.

The results obtained for this class show significant similarities to the results for the other classes. With the largest training samples, the models based on XGBoost are characterized by the best efficiency, although this time the best results were achieved using XGBoost5, with the differences between individual variants being again very small. However, with smaller samples, their accuracy clearly drops, significantly below the accuracy not only of SVM003, which traditionally already shows the highest efficiency, understood as high effectiveness with few training samples, but also below the accuracy of both RF variants.

3.6 Overall classification

The results obtained for the entire images are presented in Figure 10 and in Tables 24-26 in the Appendix.

The values determining the accuracy of the entire classification are, of course, the result of the accuracy of identifying individual classes. Thus, of all the scenarios analyzed, the variants of the XGBoost algorithm were characterized by the highest efficiency when using the training sample with the highest number of pixels. Consistently, the best results were obtained for the XGBoost7 variant, followed by XGBoost5, XGBoost9 – with the differences in kappa values obtained for the best results being about 0.001. The XGBoost3 variant turned out to be slightly weaker in this respect, but the difference in kappa values was only a few thousandths. At the same time, however, the XGBoost algorithms proved to be the most sensitive to changes in the size of the training sample. In the case of scenarios with the least numerous training sample, it was the XGBoost variants that showed the lowest efficiency, excluding the SVM03 variant, which turned out to be the worst of all analyzed in every respect. The variant that coped best with a small training sample was another SVM variant: SVM003. It consistently and significantly outperformed other tested variants in this respect (in some cases close to 0.1 kappa difference). At the same time, it gave very good accuracy with large training samples, worse than the best variants only by about 0.01 kappa, but better than those obtained thanks to RF.

Tables 6–8 contain a statistical summary of kappa values for all scenarios (for all types of training samples) for individual variants. They indicate that in the overall comparison, the SVM003 variant performs the best. Both the mean and median values are unambiguously (and often significantly) the best for this variant. Of course, the highest kappa values were obtained for selected XGBoost scenarios, but at the same time, the lowest values were obtained for other scenarios of XGBoost variants (excluding SVM03). On the other hand, the SVM003 variant has the highest minimum kappa value among the analyzed scenarios – consistently for all test images. Combined with the smallest standard deviation value, this allows us to state that this is the most stable variant, resistant to imperfections in training data. Considering the differences between the results obtained by the 3 SVM variants, we might conclude that this is an algorithm very sensitive to the value of the *gamma* parameter. The differences between the results of variants of other algorithms were relatively small.

It's worth noting that the SVM003 variant is better than the others with training samples of a total pixel count of about 10,000 (and smaller). In the case of samples with a count of 15,000 and 25,000 pixels, it shows similar accuracy to the XGBoost variants. Only in the case of samples of 30,000 (so, very large) and larger, can we observe the superiority of selected XGBoost variants, increasing with the training sample.

Table 6. Image A; statistical summary of Cohen's kappa coefficient values for all scenarios (dependent on the size of the training sample) for each variant

variant	kappa				
	med.	avg.	st. dev.	max.	min.
XGBoost3	0.681	0.655	0.049	0.695	0.521
XGBoost5	0.675	0.655	0.049	0.702	0.538
XGBoost7	0.668	0.653	0.047	0.704	0.544
XGBoost9	0.663	0.651	0.045	0.702	0.545
RF50	0.673	0.657	0.038	0.689	0.536
RF100	0.677	0.658	0.041	0.691	0.525
SVM03	0.608	0.583	0.061	0.639	0.433
SVM01	0.665	0.649	0.039	0.681	0.542
SVM003	0.687	0.676	0.021	0.695	0.609

Table 7. Image B; statistical summary of Cohen's kappa coefficient values for all scenarios (dependent on the size of the training sample) for each variant

variant	kappa				
	med.	avg.	st. dev.	max.	min.
XGBoost3	0.651	0.626	0.050	0.668	0.503
XGBoost5	0.642	0.622	0.051	0.673	0.502
XGBoost7	0.632	0.618	0.051	0.674	0.497
XGBoost9	0.627	0.616	0.049	0.672	0.499
RF50	0.634	0.622	0.036	0.657	0.533
RF100	0.638	0.624	0.037	0.660	0.526
SVM03	0.575	0.562	0.039	0.599	0.465
SVM01	0.628	0.619	0.025	0.645	0.551
SVM003	0.652	0.644	0.023	0.664	0.575

Table 8. Image C; statistical summary of Cohen's kappa coefficient values for all scenarios (dependent on the size of the training sample) for each variant

variant	kappa				
	med.	avg.	st. dev.	max.	min.
XGBoost3	0.737	0.722	0.037	0.753	0.616
XGBoost5	0.732	0.718	0.039	0.756	0.610
XGBoost7	0.726	0.716	0.039	0.756	0.601
XGBoost9	0.721	0.714	0.038	0.755	0.601
RF50	0.729	0.719	0.031	0.744	0.624
RF100	0.730	0.720	0.031	0.746	0.628
SVM03	0.680	0.664	0.043	0.703	0.518
SVM01	0.721	0.706	0.031	0.732	0.625
SVM003	0.739	0.728	0.023	0.748	0.666

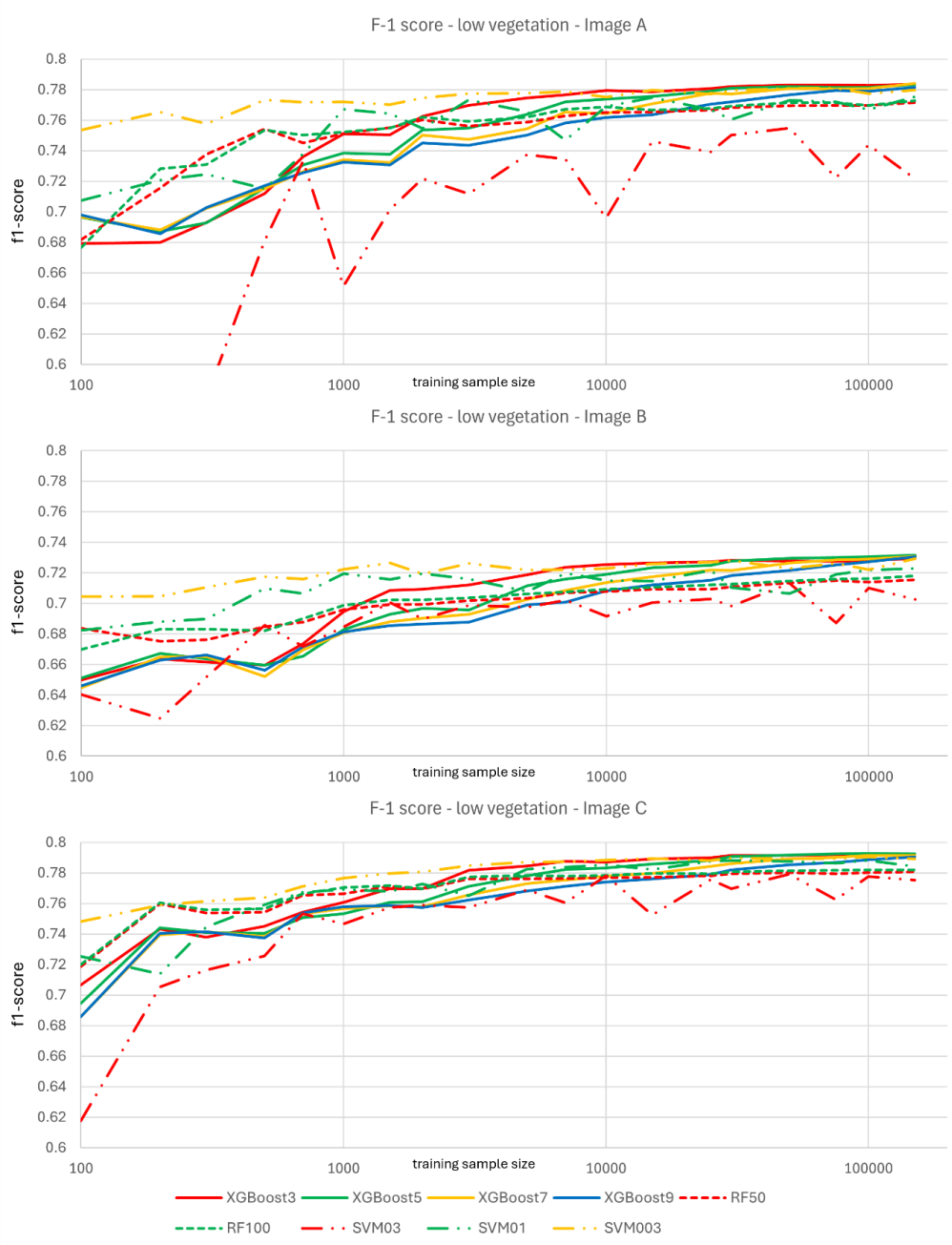

Figure 9. Diagrams of F-1 score values for the low vegetation class, in relation to the size of the training sample (the scale of the training sample size is logarithmic)

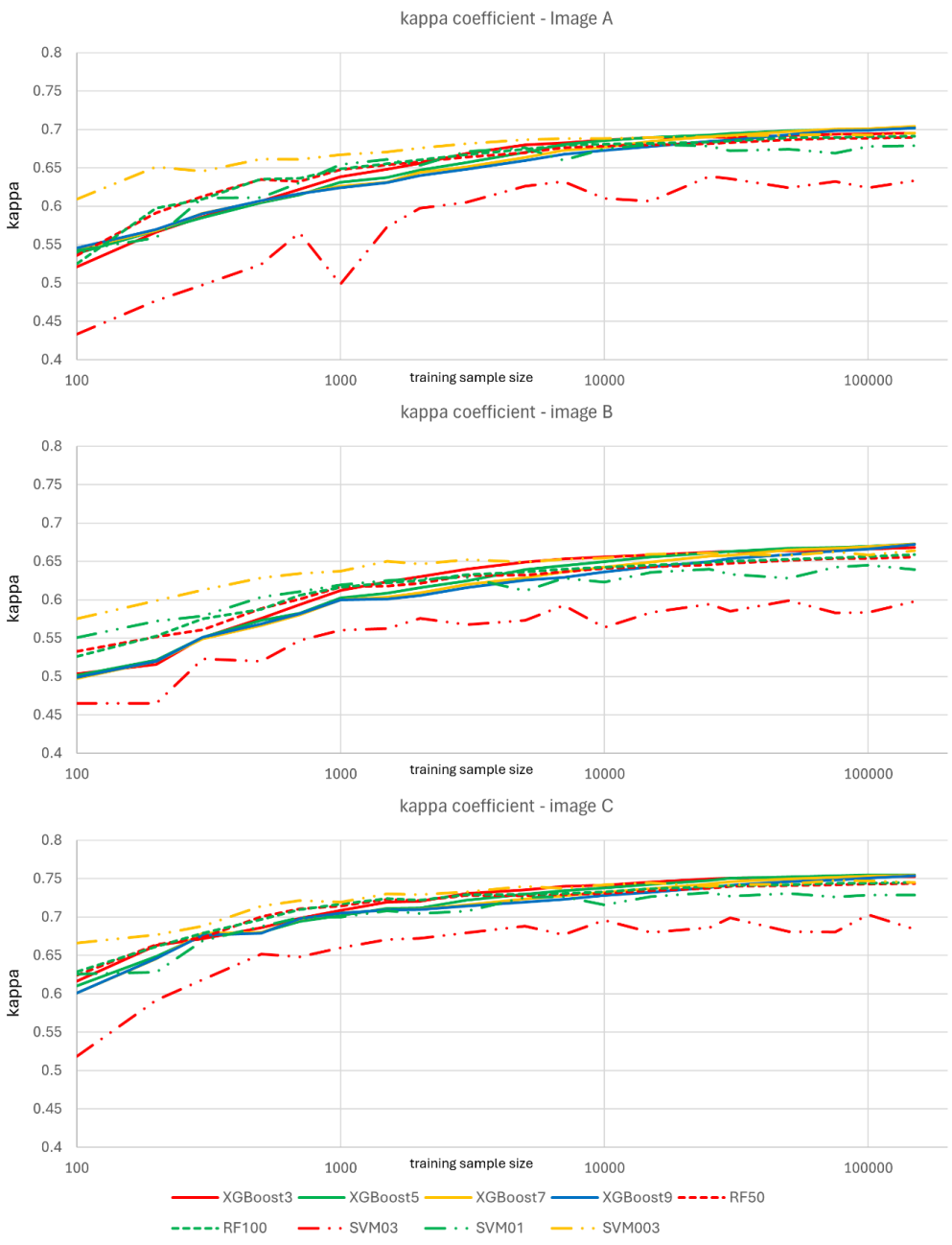

Figure 10. Diagrams of Cohen's kappa coefficient values in relation to the size of the training sample (the scale of the training sample size is logarithmic)

4 Summary and Conclusions

The conducted analyses allow for the presentation of several conclusions regarding the effectiveness of the 3 analyzed ML algorithms: RF, XGB, and SVM.

Firstly, in all cases, it was clear that as the size of the training sample increased, so did the effectiveness of the classification. However, the nature of this relationship was different for each algorithm. The size of the training sample was most significant in the case of the XGB algorithm variants. For the largest training samples (from 30,000 to 200,000 pixels), the XGB algorithm demonstrated the highest effectiveness. However, this effectiveness significantly decreased as the training sample size decreased. Although this rule generally applies to all analyzed methods, the decrease was greatest in the case of XGB. As a result, for smaller samples (below 10,000 pixels), XGB variants showed lower effectiveness than the best SVM variant, and for the smallest samples (from 100 to 1,000 pixels), they showed the lowest effectiveness among the tested variants (excluding 2 weaker SVM variants, more on that below). On the other hand, the size of the training sample was least significant in the case of the SVM method, although this mainly applies to one of the analyzed variants – with the smallest *gamma* parameter value, therefore the least prone to overfitting. The decrease in the effectiveness of this algorithm with the reduction of the training sample size was the smallest among all analyzed variants. Although the effectiveness was slightly lower than the best XGB variants for the largest training samples, their effectiveness equaled for samples below 30,000 pixels, and below 10,000 pixels, the SVM algorithm gave unequivocally and very significantly the best classification results. Generally, it can be stated that it was this algorithm that allowed for the best results across all analyzed scenarios. The RF algorithm performs moderately well. Although the decrease in accuracy with the reduction of the training sample is not as significant as in the case of XGB, in none of the analyzed scenarios does RF give the best results: for the largest training samples its effectiveness is slightly lower than XGB (comparable to SVM), and for smaller ones it is clearly lower than SVM (although slightly better than XGB).

The obtained results confirm the dependency observed in other publications (Shang et al., 2018; Ramezan et al., 2021; Fu et al., 2023; Kupidura and Niemyski, 2024): a larger training sample size results in greater classification accuracy. Confirmation is also found in the relatively high effectiveness (compared to other algorithms) of SVM with a small number of training data (Kupidura and Niemyski, 2024). The above results also correspond with the results of other studies, confirming the high effectiveness of SVM (Koppaka and Moh, 2020; Ghayour et al., 2021; Sobieraj et al., 2022; Maxwell et al., 2014a,b, 2015; Volke and Abarca-Del-Rio, 2020). The obtained results also seem to confirm previous studies confirming the high effectiveness of XGB, especially compared to RF (Liu et al., 2021; Seydi et al., 2022). To a large extent, it was also possible to confirm the general ambiguity resulting from the comprehensive analysis of the literature on the subject: different algorithms may have different effectiveness, depending on the conditions (in this case: the size of the training sample). On the other hand, in the obtained results, no confirmation was found of earlier studies indicating a greater effectiveness of RF compared to, for example, SVM (Mousavinezhad et al., 2023; Zhao et al., 2024).

The individual algorithms were tested in various variants – with different parameter values – specific to each method. The analysis results show that in the case of XGB and RF algorithms, the impact of these parameters is relatively small. For the XGB algorithm, the best results were obtained for 7 node levels – in the case of a larger number of levels, a decrease in model effectiveness was observed (very small, but clearly on all 3 test images), which could indicate increasing overfitting. On the other hand, a smaller number of node levels could result in too little model flexibility. However, as we mentioned, the differences between individual variants are very small. In the case of RF, better results were obtained thanks to the model using a larger number of decision trees, but here too the difference between the effectiveness of both variants is small. The value of the *gamma* parameter, on the other hand, was of great importance. One of the SVM variants, with the smallest *gamma* value, showed excellent effectiveness against all tested variants, while the remaining 2, with larger *gamma* values, were already much less effective, and the variant with the largest *gamma* value clearly lagged behind all tested algorithms, which can be explained by model overfitting.

In light of the above, it is difficult to unequivocally indicate the best algorithm. The recommendation may depend on the size or quality of the training sample. The highest accuracy among the analyzed scenarios was achieved thanks to the XGB algorithm. However, the condition was a very large, high-quality training sample. In the case of a smaller number of pixels, the SVM algorithm proved to be the best solution, which allowed for the construction of an effective model despite worse training data, but provided that the *gamma* parameter value was properly tuned.

When analyzing the results, conclusions, and recommendations presented above, it should be remembered that they were based on a specific number of features and classes. The performance of the selected ML algorithms may vary depending on the nature of the task.

Appendix

Tables 9–26 are attached in a file available at:
https://rgg.edu.pl/SuppFile/191766/1/.

References

Allwright, S. (2023). XGBoost vs Random Forest, which is better? Technical report.

Belgiu, M. and Drăguţ, 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*, COLT92. ACM, doi:10.1145/130385.130401.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32, doi:10.1023/a:1010933404324.

Budach, L., Feuerpfeil, M., Ihde, N., Nathansen, A., Noack, N., Patzlaff, H., Naumann, F., and Harmouch, H. (2022). The effects of data quality on machine learning performance. doi:10.48550/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 & 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. 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.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297, doi:10.1007/bf00994018.

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.

Ding, H. (2024). Establishing a soil carbon flux monitoring system based on support vector machine and XGBoost. *Soft Computing*, 28(5):4551–4574, doi:10.1007/s00500-024-09641-y.

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.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition, 14-16 August 1995, Montreal, QC, Canada*, volume 1, pages 278–282. IEEE, doi:10.1109/ICDAR.1995.598994.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, doi:10.1109/34.709601.

Kapoor, S. and Perrone, V. (2021). A simple and fast baseline for tuning large XGBoost models. *arXiv preprint arXiv:2111.06924*, doi:10.48550/arXiv.2111.06924.

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)*. IEEE, doi:10.1109/imcom48794.2020.9001718.

Kumar, A. (2023). Random Forest vs XGBoost: Which one to use? Examples. Technical report.

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*, 64:24–38.

Labatut, V. and Cherifi, H. (2012). Accuracy measures for the comparison of classifiers. *arXiv preprint arXiv:1207.3790*, doi:10.48550/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.

Liaw, A., Wiener, M., et al. (2002). Classification and regression by Random Forest. *R news*, 2(3):18–22.

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.

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., and Fang, F. (2018). Implementation of machine-learning classification in remote sensing: An applied review. *International Journal of Remote Sensing*, 39(9):2784–2817, doi:10.1080/01431161.2018.1433343.

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.

Nalepa, J. and Kawulok, M. (2018). Selecting training sets for support vector machines: A review. *Artificial Intelligence Review*, 52(2):857–900, doi:10.1007/s10462-017-9611-1.

Powers, D. M. W. (2007). Evaluation: From precision, recall and F-factor to ROC, informedness, markedness & correlation. Technical report SIE-07-001. Technical report, 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.

Schölkopf, B. (2002). *Learning with kernels*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass.

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 [preprint]*, doi:10.5194/nhess-2020-41.

Wainer, J. and Fonseca, P. (2021). How to tune the RBF SVM hyperparameters? An empirical evaluation of 18 search algorithms. *Artificial Intelligence Review*, 54(6):4771–4797, doi:10.1007/s10462-021-10011-5.

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.