

Article

A Comparative Analysis of Dynamic Time Warping and Machine Learning Models for Crop Classification: Case Study of Limarí River Basin, Chile

Aldo A. Tapia ^{1,2,*}  and Andrew Bennett ¹ 

¹ Department of Hydrology and Atmospheric Sciences, The University of Arizona, 1133 James E. Roger Way, Tucson, AZ 85721, USA; andrbenn@arizona.edu

² Departamento de Agronomía, Facultad de Ciencias, Universidad de La Serena, Avda La Paz 1108, Ovalle 1842646, Chile

* Correspondence: aldotapia@arizona.edu

Abstract

Crop monitoring is an important aspect of agricultural management, as it provides insights into cultivated area, crop health, growth patterns, and yields potential. Mapping cultivated areas and identifying crop types was historically conducted through field surveys and manual mapping, which are time-consuming and labor-intensive. Remote sensing classification has transformed large-scale land cover mapping, including crop identification. This work aims to: (1) compare the performance of Dynamic Time Warping (DTW) and two machine learning families (artificial neural networks and decision trees) for crop classification using Sentinel-2 data; (2) assess whether reflectance data, spectral indices, or both yield better classification results; and (3) evaluate the effect of hyperparameters on model performance. Among the DTW variants evaluated, dynamic time warping without a time constraint performed the best, with an overall accuracy of 0.921 using the combination of both reflectance and spectral indices. Most machine learning methods outperformed DTW. Although the convolutional neural network reached the highest single accuracy (0.948), the transformer was selected as the best model overall (accuracy of 0.944), as it combined a comparable accuracy with the lowest sensitivity to hyperparameter variations, making it a reliable option when testing machine learning architectures applied to crop mapping. This work also provides insights for model architecture development based on an exhaustive hyperparameter search for the machine learning models.

Keywords: crop classification; dynamic time warping; machine learning



Academic Editors: Dimitrios E. Tssemelis, Pantelis E. Barouchas, Georgios Bourantas and Kleomenis Kalogeropoulos

Received: 29 May 2026

Revised: 13 July 2026

Accepted: 15 July 2026

Published: 24 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Crop monitoring is a critical component of agricultural management, providing essential information on the spatial extent of cultivated areas, crop health, phenological development, and yield estimation. Accurate crop mapping supports resource allocation, food security assessments, and sustainable land management [1]. Traditionally, these tasks relied on field surveys and manual mapping, which are labor-intensive, time-consuming, and often limited in spatial and temporal coverage [2].

Recent advances in remote sensing classification (RSC) have transformed agricultural monitoring by enabling large-scale, repeatable, and objective development of Land Use & Land Cover Change (LULCC) maps. Sentinel-2, for example, provides high-resolution multispectral imagery with frequent revisits, capturing vegetation dynamics across the

growing season [3]. These data enable automated classification algorithms to distinguish crop types by their spectral and temporal signatures, supporting more timely agricultural decision-making [4].

Crop classification represents a specialized subset of LULCC mapping, focusing on the identification and mapping of specific crop types within agricultural landscapes. Unlike broader land cover classes, crops exhibit pronounced temporal variability in their spectral response due to phenological development, management practices, crop rotation, and local climatic conditions [5]. Moreover, some crops share similar spectral and phenological characteristics, making them difficult to distinguish using traditional classification methods [6]. For example, crops such as wheat and barley may have similar spectral signatures during certain growth stages, complicating their differentiation in remote sensing imagery. The same occurs with fruit crops of the same family, such as citrus species (lemons, oranges, and mandarins), which can exhibit similar spectral responses due to their shared physiological traits and growth patterns. This variability poses unique challenges for remote sensing-based classification, as crops may be difficult to distinguish for the given reasons, and the timing of agricultural activities can vary significantly across regions and even within the same valley. Additionally, factors such as irrigation, fertilization, and harvesting schedules further influence the spectral and temporal patterns observed in satellite imagery [7].

While LULCC maps are produced by evaluating the spectral response of land cover types, crop classification maps are often produced by analyzing vegetation indices—most commonly the Normalized Difference Vegetation Index (NDVI)—and their temporal dynamics [8–10]. Vegetation indices are mathematical combinations of reflectance values from specific spectral bands (typically red and near-infrared) designed to enhance the signal related to vegetation vigor, biomass, and photosynthetic activity. For crops, these indices provide a more robust and interpretable measure of plant status than raw spectral signatures, as they reduce the influence of soil background, illumination differences, and atmospheric effects [11].

In contrast, spectral signatures represent the reflectance values across all available bands at a single point in time, which can be affected by transient atmospheric conditions, such as thin clouds or cirri. The interpretation of spectral signatures requires a comprehensive understanding of the spectral response of different crops and their phenological stages. This complexity can make spectral signatures less straightforward to use for crop classification compared to vegetation indices for pattern recognition tasks. However, whether spectral indices outperform raw reflectance for classification remains debated [12,13]. Depending on the spatial arrangement of crops, different classification approaches, from pixel-based to object-based, have been developed to capture temporal dynamics at varying scales and from traditional algorithms to machine learning applications [14,15]. Pixel-based classification assigns a class label to each pixel based on its spectral or temporal characteristics, often using statistical or rule-based approaches. Object-based classification, in contrast, groups pixels into meaningful objects (such as fields or parcels) and classifies these objects using aggregated features, which can improve accuracy in heterogeneous landscapes [16].

Traditional methods—such as maximum likelihood classification and decision trees—often rely on low-dimensional inputs and simple statistical assumptions, which may struggle to capture the temporal dynamics of crops. Dynamic time warping (DTW) has emerged as a popular method for analyzing time series data, including variants such as the time-weighted dynamic time warping (TWDWTW) method, which incorporates a temporal cost to the DTW algorithm [17–22].

Machine learning and deep learning methods, such as random forest (RF) and neural networks (NN), learn complex patterns from data and can handle high-dimensional inputs and nonlinear relationships, making them well-suited for crop classification tasks [23–25]. Feature extraction in machine learning models, especially in neural networks, involves learning representations from raw input data that are most informative for the classification task. For crop classification, this typically means identifying spectral and temporal patterns that distinguish different crop types. In fully connected neural networks, feature extraction is performed through successive layers of nonlinear transformations, while convolutional neural networks (CNNs) use convolutional filters to capture local patterns and temporal dependencies in the input time series [26]. These learned features can represent phenological cycles, sudden changes in reflectance due to management practices, or subtle differences in spectral indices between crops. The effectiveness of feature extraction depends on the architecture of the model and the quality of the input data, with deeper or more complex models generally able to capture more nuanced patterns.

However, the performance of these methods can vary depending on the characteristics of the data and the specific classification task. For instance, DTW is effective in capturing the temporal dynamics of crops, but it may struggle with high-dimensional data or when the temporal patterns are not well defined [20]. Machine learning methods can handle high-dimensional data and complex relationships, but they may require large amounts of training data and careful tuning of hyperparameters to achieve optimal performance [24]. DTW has been compared with RF, obtaining a higher overall accuracy in most cases [27–31], and neural networks have similarly been shown to outperform random forest [25,32,33].

This study compares the performance of a traditional method (dynamic time warping) with two machine learning model families (neural networks and decision trees) expressed in several architectures for crop classification using Sentinel-2 data in the Limarí River basin, Chile. The evaluation of whether reflectance data or spectral indices provide better results for crop classification was also included. Additionally, an exhaustive hyperparameter search for the different model structures was applied to evaluate the marginal contribution of the most common hyperparameters for these models. By addressing these objectives, this work advances the understanding of remote sensing classification methods for agricultural monitoring through a systematic comparison of ML models.

2. Materials and Methods

2.1. Study Area

The study area is the agricultural zone of the Limarí River basin located in the Near North (Zn) natural Region (29°00' S to 32°10' S), in the Coquimbo Region, Chile. The Limarí River basin is characterized by a snow-melt-dominated regime, with two annual peaks of streamflow. The first peak occurs in winter, due to the precipitation, and the second peak occurs in spring and summer, due to snowmelt from the Andes Mountains. The basin has an area of approximately 11,800 km² with a potential agricultural area of approximately 50,000 ha, of which 30,000 ha are cultivated in a normal period [34].

The climate of the Zn region is semi-arid with Mediterranean influence [35]. There are climatic differences between the coast and the interior of the region, mainly influenced by orography, the presence of the Humboldt Current, and the nature of wind and cloudiness [36]. The coast has a lower temperature range and lower extremes than the interior, and temperatures decrease from the interior to the Andes Mountains due to elevation.

Precipitation is also influenced by the orography, with the coastal zone receiving more than the interior due to the presence of the coastal mountain range, which blocks the passage of the humid air masses from the Pacific Ocean [37]. The precipitation intensity increases in the Andes Mountains, as a second barrier, with the presence of snow in the

winter months. The coastal areas receive a significant amount of their annual water input in the form of fog [38], leading to a lower temperature range.

These factors create a climatically complex agricultural setting. In the Limarí basin, crops experience a wide range of temperatures, precipitation, and solar radiation, making phenological characterization particularly challenging. In this valley, the climate on the coast is characterized by abundant cloud formation; the central section is a temperate zone with winter rains; and towards the Andes Mountains, there is a cold climate with snow in the winter months [39]. Differences in temperature and precipitation lead to differences in the phenology of the crops [6,40,41].

The high temperatures in the central valley towards the Andes Mountains have led to the development of fruit crops with high economic revenue, such as table grapes, avocados, and a variety of citrus species (lemons, oranges, and mandarins). The coastal area is mainly used for the cultivation of vegetables, almonds, and olives, for olive oil production [42]. The study area, and the samples collected representing the crops grown in the area, are shown in Figure 1.

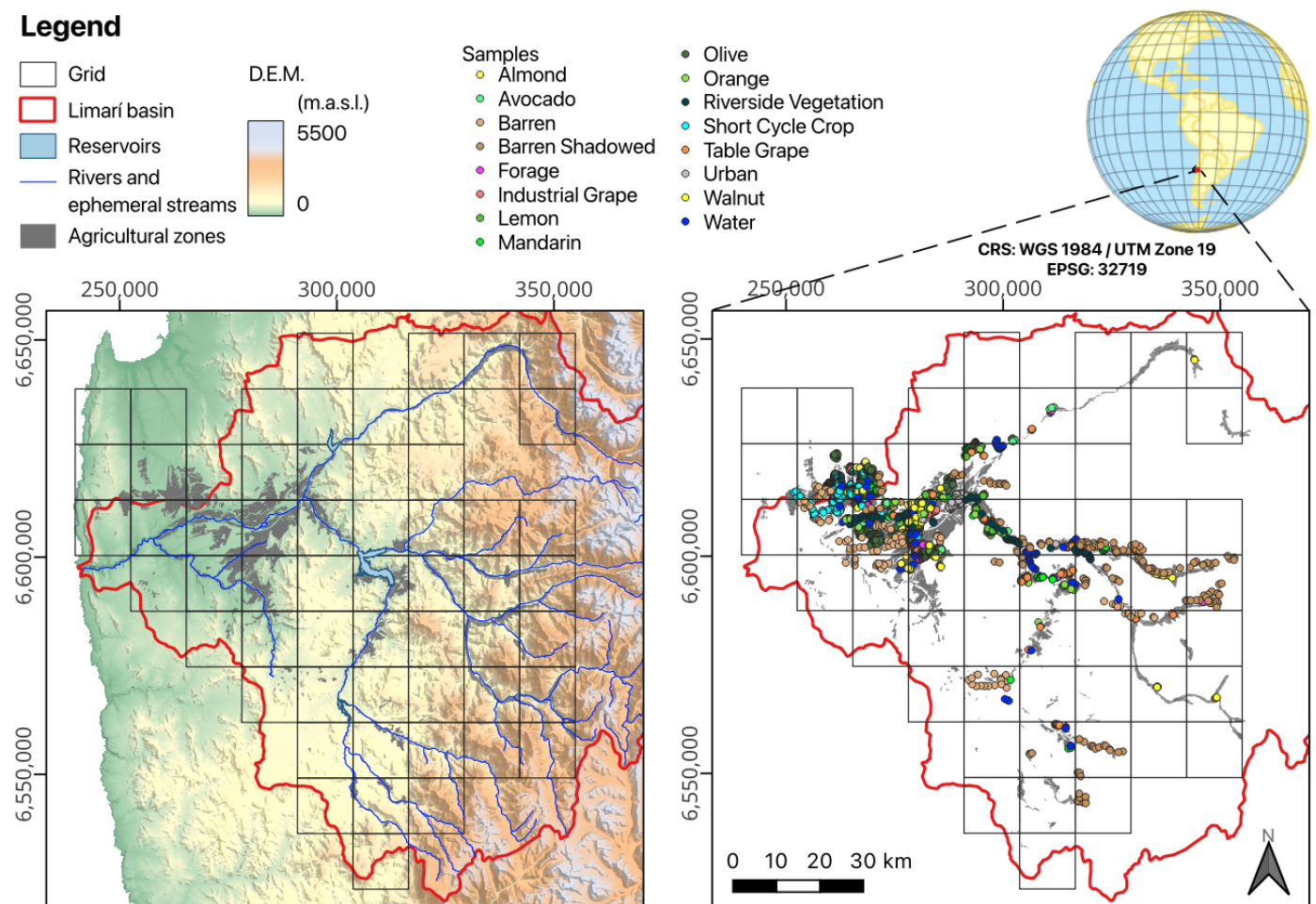


Figure 1. Study area. The (left panel) shows the agricultural zones (gray) used for filtering the grid cells (described in the next section), the final grid used, and the Digital Elevation Model showing the elevation changes across the study area. The (right panel) shows the samples collected for the different classes used in this study.

2.2. Methodology

2.2.1. Sentinel-2 Data

Sentinel-2 (S2) data were used to classify the crops in the Limarí River basin. The S2A_MSIL2A and S2B_MSIL2A products were used, which are the Level-2A products of the S2 satellite series. The S2 data was acquired from the Copernicus Open Access Hub, for the Chilean 21/22 agricultural season (from 1 May 2021, to 30 April 2022) with a two-month buffer on each end to support gap filling. The 10-m resolution bands (B2, B3, B4, and B8) were selected. Since the L2A products are already corrected for atmospheric effects, the only correction applied was the Band Reflectance to Nadir BRDF-Adjusted Reflectance (NBAR) correction, following the methodology proposed by Roy et al. [43] to adjust reflectance values to a common viewing geometry and reduce the effects of the seasonal variations.

Since the coastal zone of the Limarí River basin is affected by clouds and fog by a phenomenon called *camanchaca* [44], and the mountainous area is also affected by clouds and snow, a grid was created to split the scenes into smaller areas to remove scenes with clouds. Clouds may affect both areas, so the goal is to retain as many scenes as possible while excluding cloudy scenes. The size of the study area was also reduced by introducing a grid, which allows focusing on the agricultural areas determined by the last cadastre of native vegetation performed by the National Forestry Corporation (CONAF). The original study area has a 1280×1280 pixels, which was broken up into cells with a size of 90×90 pixels initially, with a 10-m resolution, reduced to 56 grid boxes after removing the areas without agriculture. The time series of the S2 data was split using the grid for further selection of the cells with the best quality data, evaluated using the cloud probability layer proposed by Skakun et al. [45], which improves on some shortcomings of the cloud mask provided by the S2 products [46] and has been used in similar studies [47].

To improve the cloud detection, the cloud probability layer was used as input to determine the quality of the scenes, enhancing high values. First, the probability layer was modified through an aggregation process, aggregating the 10-m resolution cloud probability layer to a 15×15 pixel resolution using the maximum value of the cloud probability layer (hence, a resolution change to 150 m). This aggregation process helps to reduce the noise in the cloud probability and detect pixels with high values. After this, a focal statistics process was applied using a 7×7 kernel to propagate the maximum value to adjacent pixels. Finally, a binary mask was created by thresholding the cloud probability values greater than 80%. If a scene had more than 10% cloud detection over the time period, it was excluded from the analysis. An example of this process is shown in Figure 2, where a synthetic example scene is used to demonstrate the cloud processing workflow, presenting the aggregation and focal statistics steps. The final computed value shows that this scene yields 20% cloud-flagged pixels and would therefore be excluded from the analysis. A manual revision was conducted to remove the scenes that presented clouds and were not detected by the cloud probability layer.

Most of the class label samples related to crops were obtained from the Fruit Cadastre of Coquimbo Region [42] and supplemented with field surveys. For sampling the crops, one sample per parcel was withdrawn within the boundaries of the parcel to minimize edge effect, mixed spectral signals, and to avoid data leakage during the experiments. The crop classes 'industrial grape', 'forage', and 'short cycle crop' are not included in the Fruit Cadastre, but were added to the study area since those are crops with economic importance in the region. Sixteen classes were defined for analysis: nine fruit tree crops (almond, avocado, industrial grape, lemon, mandarin, olive, orange, table grape, walnut), two herbaceous crop categories (forage, short cycle crop), and five other land cover types (barren, barren shadowed, riverside vegetation, urban, water). Due to limited information

and the diversity of species, short-cycle crops were grouped into a single class. In total, 2683 samples were collected, distributed over 69 dates during the 2021/2022 agricultural season. The sample count per class is shown in Table 1.

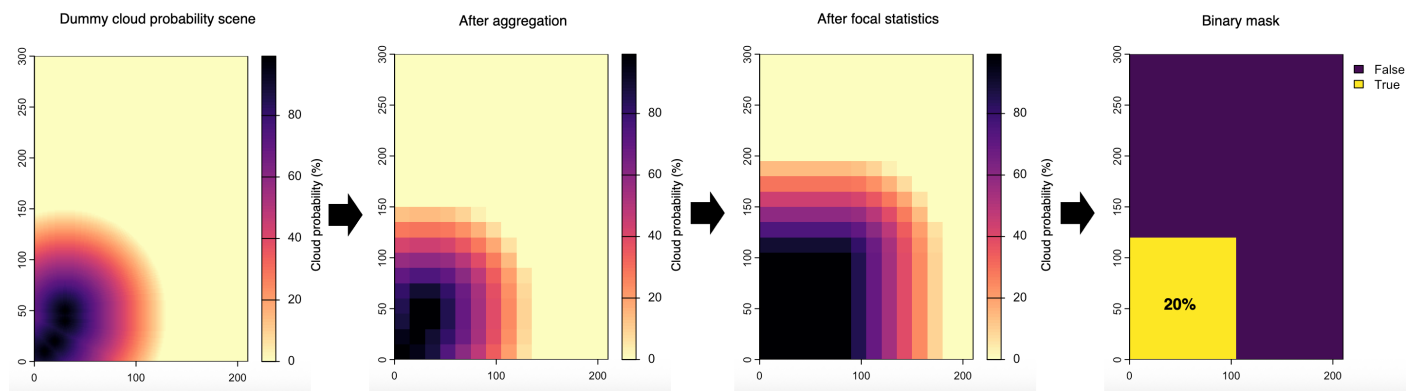


Figure 2. Cloud processing workflow demonstration with a synthetic example scene.

For each sample and date, reflectance values (ρ) were extracted, and missing data due to cloud cover were filled using linear interpolation for the entire time series. Some class labels exhibit similarities in their spectral and phenological characteristics, which can lead to confusion during classification. For example, perennial fruit crops such as mandarin, lemon, and orange share similar evergreen phenology and spectral responses, making them difficult to distinguish using remote sensing data alone. Likewise, herbaceous classes like forage and short-cycle crops often display overlapping temporal patterns due to similar growth cycles. These similarities should be considered when interpreting classification results, as they may contribute to misclassification between certain classes. Retaining fine-grained crop classes, even when spectrally similar, deliberately creates a challenging classification scenario to test the robustness and discriminative power of each method.

In this study, three different types of data were tested for the classification: (i) reflectance data, (ii) spectral indices data, and (iii) both reflectance and spectral indices data. The spectral indices selected were the Normalized Difference Vegetation Index (NDVI), the Enhanced Vegetation Index (EVI), the Green-Red Vegetation Index (GRVI), and the Green Normalized Difference Vegetation Index (GNDVI). The equations used for the spectral indices are presented in Table 2.

The inclusion of these spectral indices follows the approach commonly used in previous crop classification studies, where vegetation indices such as NDVI, EVI, GRVI, and GNDVI serve as the primary input features for classification algorithms. This strategy is based on the premise that spectral indices can enhance the separability of crop types by emphasizing vegetation vigor and phenological differences, as demonstrated in numerous remote sensing applications for agricultural monitoring.

A total of 100 samples per class were used for the k-fold cross-validation process. These samples were split 67/33 into training and validation sets. When the combined number of available samples for a class fell below 100, bootstrapping was applied to reach that threshold, using the pre-defined train/validation splits to avoid data leakage between splits. A k-fold cross-validation process was then applied, with $k = 3$, to obtain a more robust evaluation of the performance of the methods. Samples in each of the k-fold splits were drawn so that they did not fall within the same crop field, avoiding data leakage between training and validation splits. All results are reported on the validation set for each of the three folds.

Table 1. Sample distribution for each class.

Class	Samples
almond	160
avocado	124
barren	112
barren shadowed	169
forage	94
industrial grape	152
lemon	94
mandarin	145
olive	505
orange	106
riverside vegetation	92
short cycle crop	447
table grape	204
urban	99
walnut	95
water	85

Table 2. Spectral index equations used in this study. The ρ_{NIR} , ρ_R , ρ_G and ρ_B are the reflectance values of the near infrared, red, green and blue bands, respectively. The ω is a small constant to avoid division by zero, which, in this case, was 0.000001.

Index	Equation
NDVI	$\frac{\rho_{NIR} - \rho_R}{\rho_{NIR} + \rho_R + \omega}$
EVI	$2.5 \cdot \frac{\rho_{NIR} - \rho_R}{\rho_{NIR} + 6 \cdot \rho_R - 7.5 \cdot \rho_B + 1}$
GRVI	$\frac{\rho_G - \rho_R}{\rho_G + \rho_R + \omega}$
GNDVI	$\frac{\rho_{NIR} - \rho_G}{\rho_{NIR} + \rho_G + \omega}$

2.2.2. Classification Algorithms

DTW was selected as a baseline for time series classification and compared against two families of ML methods. The first family is decision tree-based methods, which encompasses algorithms such as random forest (RF) and extreme gradient boosting (XGBoost). The second family is neural networks (NN), which includes architectures such as multi-layer perceptron (MLP), convolutional neural networks (CNN), recurrent neural networks (RNN), and transformers. These methods were selected to compare a traditional time series classification method (DTW) with the more complex methods widely used in recent years for crop classification. This work aims to explore the hyperparameter spaces of the machine learning methods to offer insights into the most critical architectural choices that affect the performance of the models. Although this research does not aim to interpret the role of the different features in the classification, the hyperparameter tuning process can provide insights into the importance of certain features and how they contribute to the classification performance for further research.

The overall setup of each method is as follows: for the DTW method, the performance was evaluated with and without the temporal cost, using a logistic function to incorporate the temporal cost. For the machine learning methods, different architectures and hyperparameters were evaluated to find the best configuration for each method. The performance of each method was evaluated using standard classification metrics such as accuracy, precision, recall, and F1-score.

The DTW method has been widely implemented for crop classification tasks [16–20,29,30,48,49]. DTW is a distance-based method that measures the similarity

between two time series by finding the optimal alignment that minimizes their distance. It is particularly suited to crop classification because it accommodates phenological variability arising from irrigation, fertilization, and local climatic differences.

The algorithm consists of two main steps: (i) the calculation of the distance matrix, which measures the distance between each pair of time series, and (ii) the classification step, where the class label is assigned to a new time series based on the nearest neighbor in the distance matrix. The DTW method can be implemented using different distance metrics, such as Euclidean distance or Manhattan distance, and different warping constraints, such as Sakoe-Chiba band or Itakura parallelogram. The mathematical formulation of the DTW method is as follows:

$$DTW(X, Y) = \min_W \sum_{k=1}^K d(w_k), \quad (1)$$

where X and Y are two time series, w is defined as the distance between the k -th element of X and Y , and W corresponds to the warping path, from which the minimum path for each sample is extracted.

A variation of this method is the time-weighted dynamic time warping (TWDTW) method, which incorporates a temporal cost to the DTW algorithm, allowing for a more flexible alignment of time series that may have different lengths or temporal patterns [17]. The temporal cost (ω) is defined as a function of the time difference between the two time series, and it can be used to penalize alignments that are too far apart in time. For this work, DTW was evaluated with and without the temporal cost, using a logistic function as follows:

$$\omega_{i,j} = \frac{1}{1 + e^{\alpha(g(t_i, t_j) - \beta)}}, \quad (2)$$

where $g(t_i, t_j)$ is the temporal distance between the two time series, α is a parameter that controls the steepness of the logistic function, and β is a parameter that controls the location of the function. Based on previous research, the values of α and β were set to the following combinations: 0.1, 50; 0.1, 100; 0.05, 35 [17,18] for α and β , respectively.

For the machine learning methods, a grid search was conducted to obtain the marginal contribution of each hyperparameter to the performance of the model and to find the best configuration for each method. The approach and hyperparameters evaluated for each architecture are summarized in Table 3, while the details of each hyperparameters and values tested are described in Appendix C. The general description of each neural network architecture is the following:

Table 3. Model architectures and the respective hyperparameters tested.

Model	Hyperparameters
FFNN1	<i>l1, l2, l3, lr, and dropout</i>
FFNN2	<i>l1, l2, l3, lr, and dropout</i>
CNN1	<i>conv1_channels, conv2_channels, l1, l2, l3, lr, and dropout</i>
CNN2	<i>conv1_channels, conv2_channels, conv3_channels, l1, l2, lr, and dropout</i>
LSTM	<i>projection_size, lstm_hidden_state, lstm_num_layers, dense1, dense2, lr, and dropout</i>
GRU	<i>projection_size, gru_hidden_state, gru_num_layers, dense1, dense2, lr, and dropout</i>
TRF	<i>d_model, num_heads, num_layers, d_ff, lr, and dropout</i>
RF	<i>n_estimators, max_features, criterion, min_impurity_decrease, ccp_alpha</i>
XGB	<i>n_estimators, max_depth, min_child_weight, learning_rate, subsample, colsample_bytree, reg_lambda, reg_alpha</i>

- Feed-forward neural networks case 1 (FFNN1): consists of a simple feed-forward neural network of 3 layers, with dropout between layers, and a final softmax layer for classification.
- Feed-forward neural networks case 2 (FFNN2): this experiment is a copy of the previous one with one extra pre-processing step, where a Principal Component Analysis (PCA) is applied to the input data, reducing the dimensionality of the data to 15 components to test if this pre-processing step can improve the performance of the model by reducing the noise and redundancy in the data, especially considering the expected confusions between classes with similar spectral and phenological characteristics.
- Convolutional neural networks case 1 (CNN1): consists of a simple convolutional neural network with 2 convolutional layers, followed by a feed-forward neural network with 3 layers, and a final softmax layer for classification. Dropout was included after the batch normalization layer for each convolutional block.
- Convolutional neural networks case 2 (CNN2): similar to CNN1, but with three convolutional layers and two feed-forward layers, with dropout after each feed-forward layer.
- Recurrent neural networks case LSTM (LSTM): consists of a bi-directional LSTM layer, followed by a feed-forward neural network with 2 layers. This implementation includes a linear projection layer before the LSTM layer to expand the dimensionality of the input data, and dropout before the final layer.
- Recurrent neural networks case GRU (GRU): similar to the LSTM, but with a GRU layer instead of an LSTM layer.
- Transformer (TRF): consists of a transformer encoder layer, followed by a feed-forward neural network, similar to the architecture in [50], with a linear projection layer and a sinusoidal positional encoding as pre-processing steps.

For the decision tree-based methods, the description of the models are:

- Random forest (RF): consists of an ensemble of decision trees, where each tree is trained on a random subset of the data and features.
- Extreme gradient boosting (XGBoost): consists of an ensemble of decision trees, where each tree is trained to correct the errors of the previous trees. Since this architecture is not as common to the previous one for RS classification, the hyperparameter search was extended to learn more about this model.

The selection of the hyperparameters for each model is based on the architecture—especially for the neural network family—and model complexity—especially for decision tree-based methods. Since extreme gradient boosting is the model that has been reported less in the literature for RS crop classification, the hyperparameter search was extended to learn more about this model applied to this specific task. Hence, the number of hyperparameter combinations for XGBoost is higher than for the other models. With 8 different hyperparameters and at least 3 values for each, the total number of hyperparameter combinations for XGBoost is 20,736 per fold of the k-fold cross-validation and per dataset.

2.2.3. Evaluation Metrics

As mentioned before, the performance of each method was evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score, using a grid search approach to evaluate the marginal contribution of each hyperparameter to the performance of the model, and to find the best configuration for each method.

To evaluate the performance of the classification methods, the Overall Accuracy (OA) was calculated for each class, which is defined as the ratio of correctly classified samples to the total number of samples for that class [51]:

$$OA = \frac{\sum_{i=1}^k n_{ii}}{n}, \quad (3)$$

The OA ranges from 0 to 1, where 1 indicates perfect classification, and 0 indicates no correct classification. This metric was computed for the top-1 to top-5 candidate classes for the top model of each experiment to understand the performance of the models in terms of the most likely classes. This was done by selecting the top-5 classes with the highest probability for each evaluated sample, and then computing the OA coefficient for k -classes (more details in the Appendix B).

The Precision, Recall, and F1-score were also calculated for each class, which are defined as follows:

$$Precision = \frac{TP}{TP + FP}, \quad (4)$$

$$Recall = \frac{TP}{TP + FN}, \quad (5)$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. The F1-score is the harmonic mean of Precision and Recall, defined as:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}, \quad (6)$$

These metrics provide a more detailed evaluation of the performance of the classification methods, especially in cases where there is an imbalance in the number of samples for each class, which is common in crop classification tasks. Although for this case, the performance of the models is evaluated using a balanced set of samples for each class, these metrics can provide insights into the performance of the models for each class and help to identify which classes are more difficult to classify.

Overall accuracy is a summary metric; it does not capture differences in how two models distribute predictions across classes. To address this, the non-parametric Stuart-Maxwell test was applied [52] with Holm-Bonferroni correction, which assesses marginal homogeneity in a contingency table formed from two models' outputs with three or more categories—a multiclass extension of McNemar's test. This is a paired test that compares how the two models distribute their predictions across classes, not their overall classification accuracy; a significant result therefore indicates that the two models assign samples to classes differently, and should not be interpreted as one model being more accurate than the other.

2.2.4. Marginal Contribution of Hyperparameters

To evaluate the marginal contribution of each hyperparameter, the fANOVA framework proposed by Hutter et al. [53] was used, which fits a random forest surrogate to predict model performance across the hyperparameter space. The resulting variance is then decomposed into the contribution of each hyperparameter and its interactions. The surrogate model learns a predictive $\hat{f}$, given a set of f algorithms with hyperparameter settings θ , a subset of the configuration space Θ . After training the surrogate model, it can be used to predict the performance of the ML model ($\hat{y}(\theta)$) for a given combination of hyperparameters. Then, the surrogate performance is decomposed additively as:

$$\hat{y}(\theta) = \sum_{U \subseteq N} \hat{f}_U(\theta_U), \quad (7)$$

where U is a subset of the hyperparameters, and θ_U is the corresponding hyperparameters for the subset U . Then, variance of $\hat{y}$ across Θ is defined as:

$$\mathbb{V} = \frac{1}{||\Theta||} \int (\hat{y}(\theta) - \hat{f}_{\mathcal{O}})^2 d\theta, \quad (8)$$

where $\hat{f}_{\mathcal{O}}$ is the average performance of the model across all the hyperparameters. Functional ANOVA method decomposes the variance considering all subsets of variables:

$$\mathbb{V} = \sum_{U \subseteq N} \mathbb{V}_U, \quad (9)$$

where $\mathbb{V}_U$ is defined as:

$$\mathbb{V}_U = \frac{1}{||\Theta_U||} \int \hat{f}_U(\theta_U)^2 d\theta_U, \quad (10)$$

where Θ_U is the subset of the configuration space for the subset U . Finally, the importance of the main effect of the hyperparameter subset U is given by:

$$\mathbb{F}_U = \frac{\mathbb{V}_U}{\mathbb{V}}. \quad (11)$$

This method allows identifying which hyperparameters have the most significant impact on the model performance, and also identify the models that are less sensitive to changes in the hyperparameters, which can be useful to evaluate the robustness of the models under this scenario. Since the surrogate model is key to obtain the marginal contribution of the hyperparameters, each surrogate model was optimized with a grid search approach using 81 combinations of hyperparameters for each experiment using 80% of the data for training and 20% for testing, and then the surrogate model was trained using all the data from the grid search to obtain a more robust estimation of the marginal contribution of each hyperparameter to the model performance. Finally, the fANOVA method was applied to decompose the variance of the performance into the contribution of each hyperparameter, following the method proposed by Hutter et al. [53].

2.2.5. Final Classification of the Study Area

After the evaluation of all the models (including the baseline), a final classification of the study area was performed using the best overall model, based on the robustness and performance of the models evaluated. The models from the k-fold analysis were used for the evaluation. To blend the models, a number of methods for combining their predictions were tested. The approaches are the following:

Product of experts: this approach consists of multiplying the class probabilities of the three models for each class, as follows:

$$p_{\text{ens}}(c) \propto \prod_{m=1}^3 p_m(c) \quad (12)$$

where $p_{\text{ens}}(c)$ is the ensemble probability for class c , and $p_m(c)$ is the probability of class c predicted by model m . This approach gives more weight to the classes that are consistently predicted with high probability across the three models.

Averaging: this approach consists of averaging the class probabilities of the three models for each class, as follows:

$$p_{\text{ens}}(c) \propto \frac{1}{3} \sum_{m=1}^3 p_m(c) \quad (13)$$

This approach gives equal weight to the predictions of the three models.

Maximum rule: this approach consists of assigning the probability of the class based on the maximum value across the three models, as follows:

$$p_{\text{ens}}(c) \propto \max_{m=1}^3 p_m(c) \quad (14)$$

This approach gives more weight to the model that predicts the highest probability for a given class, giving more weight to the most confident predictions from the individual models.

Finally, the ensemble scores $p_{\text{ens}}(c)$ were normalized to sum to one to obtain the final class probabilities for each pixel, and the class with the highest probability was assigned as the final class label for each pixel in the study area. The output normalization is as follows:

$$p'_{\text{ens}}(c) = \frac{p_{\text{ens}}(c)}{\sum_{i=1}^n p_{\text{ens}}(i)} \quad (15)$$

where $p'_{\text{ens}}(c)$ is the normalized ensemble probability for class c , and n is the total number of classes. This normalization step ensures that the final probabilities for each class sum to 1, which is necessary for the final classification step, where the class with the highest probability is assigned as the final class label for each pixel in the study area:

$$\hat{c} = \arg \max_c p'_{\text{ens}}(c) \quad (16)$$

3. Results

A total of 384 scenes across 96 dates were acquired from Sentinel-2 tiles T19JBG, T19JBF, T19JCG, and T19JCF. After creating date-based mosaics, cropping to grid cells, and applying cloud probability filtering, each cell retained between 24 to 73 valid scenes (Figure 3). The coastal area retained fewer valid scenes owing to persistent *camanchaca* fog, which is common in northern Chile. In the mountainous area, cloud cover is higher than in the valley, but lower than in the coastal area. Missing data were filled through linear interpolation, and samples were extracted from all cells and dates for the 69 dates of the 21/22 agricultural season for model training and testing, homogenizing the input for the different models.

3.1. Experiment Results

3.1.1. Dynamic Time Warping Algorithm

For the DTW, 4 different algorithm sets were tested for the 3 different datasets (reflectance, spectral indices, and reflectance + spectral indices). The algorithms were the following: (i) DTW without temporal cost, (ii) DTW with temporal cost using a logistic function with $\alpha = 0.1$ and $\beta = 50$, (iii) DTW with temporal cost using a logistic function with $\alpha = 0.1$ and $\beta = 100$, and (iv) DTW with temporal cost using a logistic function with $\alpha = 0.05$ and $\beta = 35$.

The results of the DTW and TWDTW algorithms for the different runs are presented in Table 4. The highest performance was achieved by the DTW algorithm using both reflectance and spectral indices data, with an accuracy of 0.921, followed by the TWDTW algorithm with $\alpha = 0.1$ and $\beta = 50$ using both datasets, with an accuracy of 0.920. The lowest performance was achieved by the TWDTW algorithm with $\alpha = 0.05$ and $\beta = 35$ using indices data, with an accuracy of 0.905. In general, the DTW algorithms achieved higher performance than the TWDTW algorithms, with the exception of the TWDTW algorithm with $\alpha = 0.1$ and $\beta = 50$ using both datasets, which achieved a similar performance to the DTW algorithm using both datasets. The performance of the algorithms was generally

higher when using both reflectance and spectral indices data, compared to using only one of the datasets. That said, all DTW-based methods show accuracy of roughly 91% with a small spread based on the dataset and algorithmic choices.

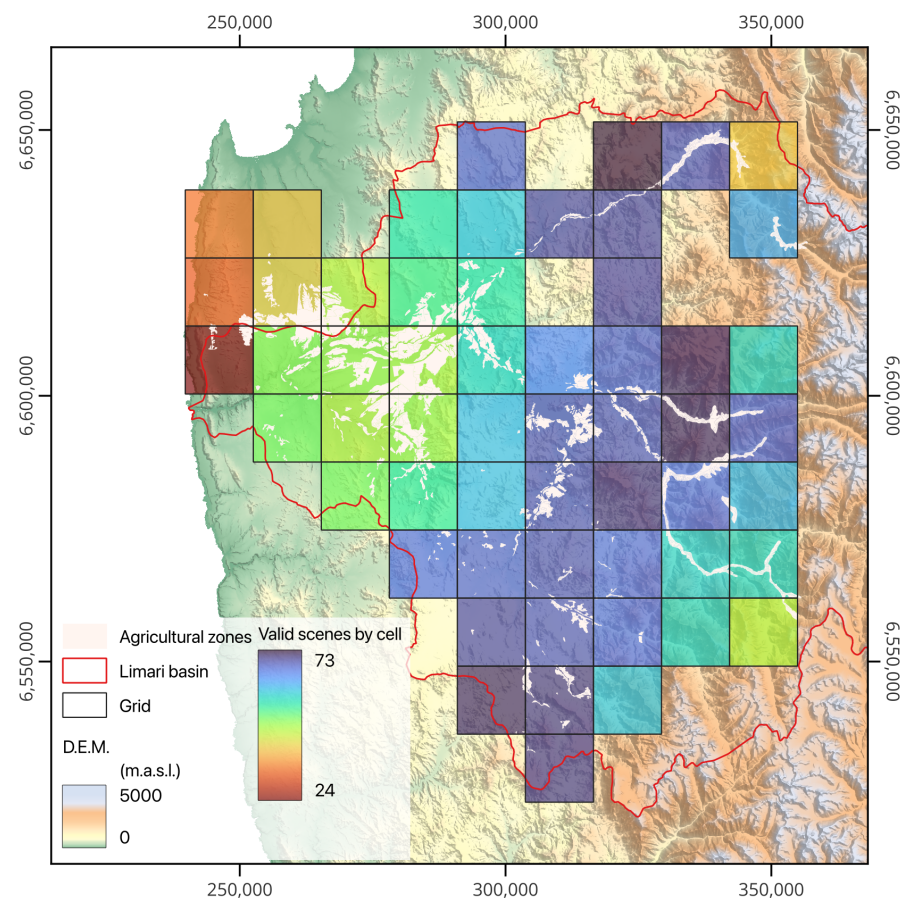


Figure 3. Valid scenes by cell. The color palette indicates the number of valid scenes for each cell, with red and orange colors indicating fewer valid scenes, while blue indicates cells with the highest number of valid scenes.

After evaluating the performance of the different algorithm combinations, the Stuart-Maxwell test was applied to evaluate the significance of the class distribution difference between the algorithm combinations evaluated. The results are added as footnote in Table 4. The only significant difference is between the TWDTW algorithm with $\alpha = 0.1$ and $\beta = 100$ using reflectance data and spectral indices data, and the TWDTW algorithm with $\alpha = 0.1$ and $\beta = 100$ using spectral data and both datasets. For the remaining algorithm combinations, the difference between the outputs' distributions is not significant.

For the DTW and most of the TWDTW algorithms, the performance was generally higher when using both reflectance and spectral indices data, compared to using only one of the datasets. The $d(w_k)$ tends to be lower for spectral indices, since the spectral indices are designed to capture specific characteristics of the vegetation, such as the amount of chlorophyll or the water content, which can be more stable and less affected by noise compared to the raw reflectance data. However, the combination of both datasets can provide complementary information that can improve the performance of the algorithms, as observed in the results. This is reflected in the highest performance by dataset: reflectance (0.912), spectral indices (0.914), and both (0.921).

Table 4. DTW and TWDTW performance for the different runs. The performance metrics informed are the Accuracy, Precision, Recall and F1-score for the top-1 candidate class for the testing set (mean across folds). The values of α and β are only informed for the TWDTW algorithms, since for the DTW algorithms these parameters are not used. In bold are highlighted the highest values for each metric and in underline the second highest values.

Name	Data Set	α	β	Accuracy	Precision	Recall	F1
DTW_1	Reflectance	-	-	0.911	0.918	0.911	0.911
DTW_2	Indices	-	-	0.914	0.920	0.915	0.914
DTW_3	Both	-	-	0.921	<u>0.925</u>	0.921	0.921
$TWDTW_{a1}$	Reflectance	0.1	50	0.911	0.919	0.910	0.912
$TWDTW_{a2}$	Indices	0.1	50	0.912	0.919	0.912	0.912
$TWDTW_{a3}$	Both	0.1	50	<u>0.920</u>	0.926	<u>0.920</u>	<u>0.920</u>
$TWDTW_{b1}$	Reflectance	0.1	100	0.912	0.919	0.913	0.913
$TWDTW_{b2}$	Indices	0.1	100	0.908	0.916	0.908	0.908
$TWDTW_{b3}$	Both	0.1	100	0.916	0.922	0.916	0.916
$TWDTW_{c1}$	Reflectance	0.05	35	0.905	0.915	0.905	0.907
$TWDTW_{c2}$	Indices	0.05	35	0.903	0.912	0.903	0.904
$TWDTW_{c3}$	Both	0.05	35	0.913	0.921	0.913	0.914

Statistical significance: a Stuart–Maxwell test with Holm–Bonferroni correction was applied to all pairs of runs. Only $TWDTW_{b1}$ differed significantly ($p < 0.05$) from $TWDTW_{b2}$ and from $TWDTW_{b3}$ in the distribution of their predictions across classes; all other pairs of runs showed no significant difference. The test assesses marginal homogeneity of the predicted class distributions, not differences in classification accuracy.

The confusion matrix of the run with the highest performance is presented in Figure 4. The highest confusion is observed on the riverside vegetation class, misclassified as olive, lemon, and avocado. Considering all those classes represent perennial vegetation with similar phenological patterns, this confusion is expected. Barren and urban are also often confused with each other, which is also expected since both classes have a similar spectral signature and phenological pattern, with low vegetation cover and high reflectance in the visible and near-infrared bands.

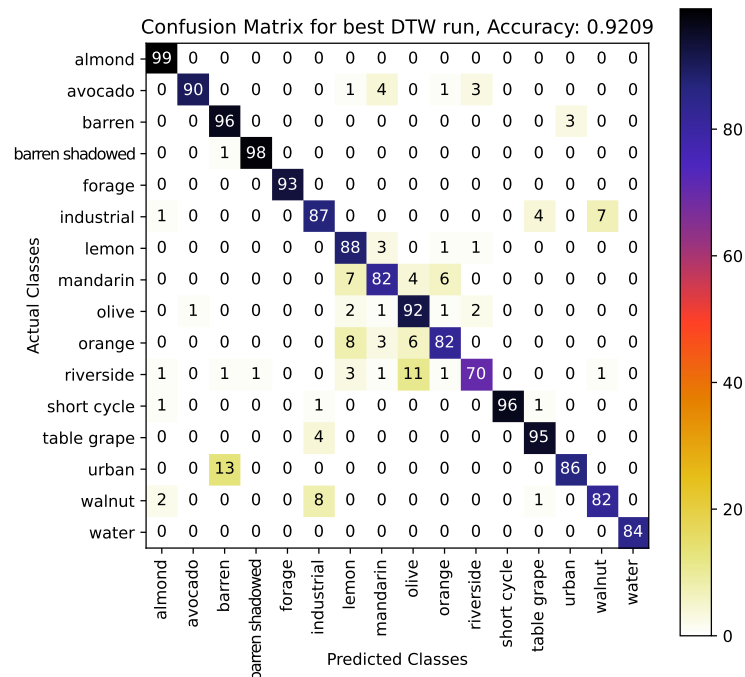


Figure 4. Confusion matrix of the DTW algorithm, mixed dataset

3.1.2. Machine Learning Methods

For each machine learning architecture, hyperparameter tuning was performed on the training set using 3-fold cross-validation under the three input configurations (reflectance, spectral indices, and their combination). The size of each architecture's hyperparameter grid and the resulting number of tuning runs are reported in Appendix C. The performance of the models for the different experiments is shown in Figure 5.

Most of the machine learning models achieved the highest performance using only reflectance data, with the lowest performance using the combination of reflectance and spectral indices data. In Table 5 the performance of the best model for each experiment by dataset is presented, where the values in bold indicate the highest performance for each dataset. The CNN2 model achieved the highest performance for reflectance data, followed by TRF, while the GRU model achieved the highest performance for both the spectral indices and the combination of reflectance and spectral indices datasets. The FFNN2 model achieved the lowest performance for all datasets, which indicates that the PCA pre-processing step did not improve the performance of the model, and in fact reduced it compared to the FFNN1 model.

Table 6 shows the statistical test (Stuart-Maxwell) results between the 3 different combinations of model pairs. As the table states, even when the accuracy can differ with a magnitude of +0.02 or more, if the class distributions are equivalent among the models, the test detects no significant difference in their predicted class distributions, i.e., the RF case. On the other hand, two models' output with a similar accuracy (i.e., CNN1 for the datasets Spectral indices and Both, with an OA of 0.914 and 0.913, respectively) can have a significant difference in the class distributions. This is reflected in the GRU model, which achieved the highest performance for both the spectral indices and the combination of reflectance and spectral indices datasets, but the predicted class distributions are not significantly different. It should be stressed that these results concern the distribution of predictions across classes, not the classification accuracy. The TRF model, which achieved the second-highest performance for the reflectance dataset, has a significant difference compared with the same model using the other two datasets.

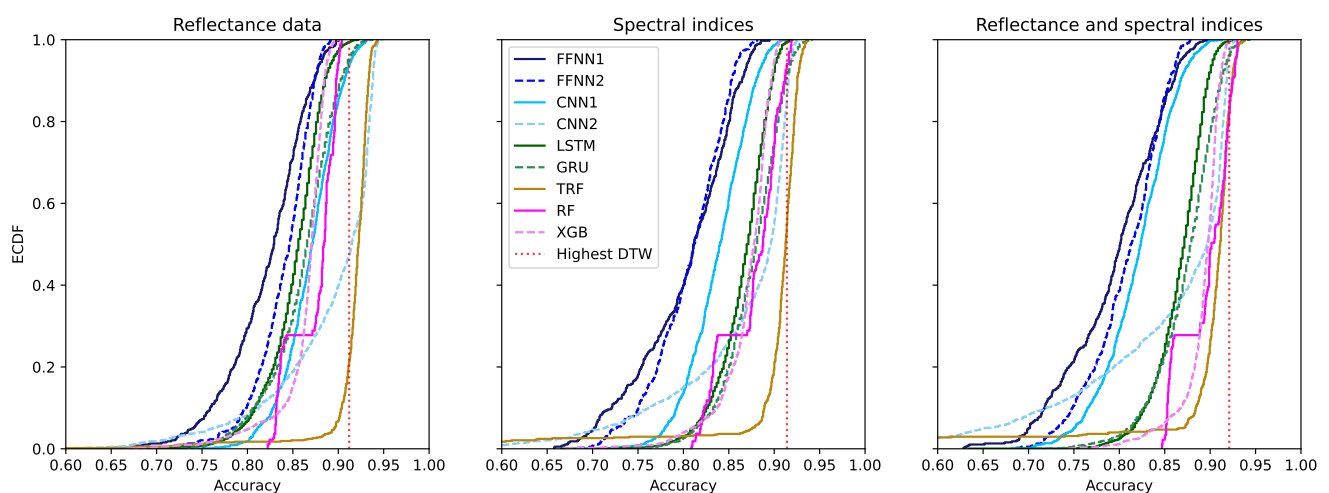


Figure 5. Empirical Cumulative Distribution Function (ECDF) of the accuracy of the machine learning models for the different experiments (mean across folds). The dotted vertical red line indicates the maximum performance of the DTW algorithm for the different experiments, as a baseline for comparison.

Table 5. Accuracy of the best model for each experiment by dataset, taken as the mean across folds. The values in bold indicate the highest performance for each dataset, and the values in underline indicate the second highest performance.

Architecture	Reflectance	Spectral Indices	Both
FFNN1	0.901	0.895	0.899
FFNN2	0.892	0.881	0.878
CNN1	0.933	0.914	0.913
CNN2	0.948	0.929	0.930
LSTM	0.922	0.920	0.924
GRU	0.938	0.942	0.944
TRF	<u>0.944</u>	<u>0.937</u>	<u>0.938</u>
RF	0.904	0.919	0.930
XGB	0.901	0.910	0.923

Table 6. Pairwise statistical comparison of the input datasets for each architecture (Stuart–Maxwell test with Holm–Bonferroni correction). ✓ indicates a significant difference ($p < 0.05$) in the class distribution of the predictions between the two datasets, and × indicates no significant difference.

Architecture	Reflectance vs. Indices	Reflectance vs. Both	Indices vs. Both
FFNN1	×	×	×
FFNN2	✓	×	✓
CNN1	×	✓	✓
CNN2	×	×	✓
LSTM	✓	✓	×
GRU	×	×	✓
TRF	✓	✓	✓
RF	×	×	×
XGB	×	×	×

“Both” denotes the combined reflectance and spectral indices dataset.

The models from the architecture classes FFNN1, FFNN2, RF, and XGB performed worse than DTW runs. RF and XGB were superior only when the training included both reflectance and spectral indices data. Since both are decision-tree-based models, they benefit from the combination of both data since these models can select a specific feature for each split, which can be either a reflectance or a spectral index. Unlike CNN, RNN, and Transformer architectures, FFNN models lack mechanisms to weight or attend to the most discriminative input channels, which may explain their lower performance when combining both data types. The models from the architectures CNN1, CNN2, LSTM, GRU, and TRF performed better than DTW runs for all datasets, with the exception of the CNN1 model for the combination of both datasets. These later models can learn more complex patterns via pattern recognition (CNN) and temporal dependencies (LSTM, GRU, TRF), leading to better performance compared to the DTW algorithm.

The hyperparameter combination for the CNN2 model with the highest accuracy had a hyperparameter selection of: 128 filters in convolutional layer 1, 256 filters in convolutional layer 2, 256 filters in convolutional layer 3, 128 neurons in dense layer 1, 128 neurons in dense layer 2, 0.3 of dropout, and a learning rate of 0.0001. The best TRF model has the following hyperparameters: projection size of 256, 8 attention heads, 1 encoding layer, 128 neurons on the feed-forward layer in the encoder, 0.3 of dropout, and a learning rate of 0.001.

Figure 6 shows the confusion matrix for the best CNN2 and TRF models. Both models perform well across most classes. The main sources of confusion are between barren and urban, and among the citrus classes (lemon, mandarin, orange), which share similar phenology and canopy structure.

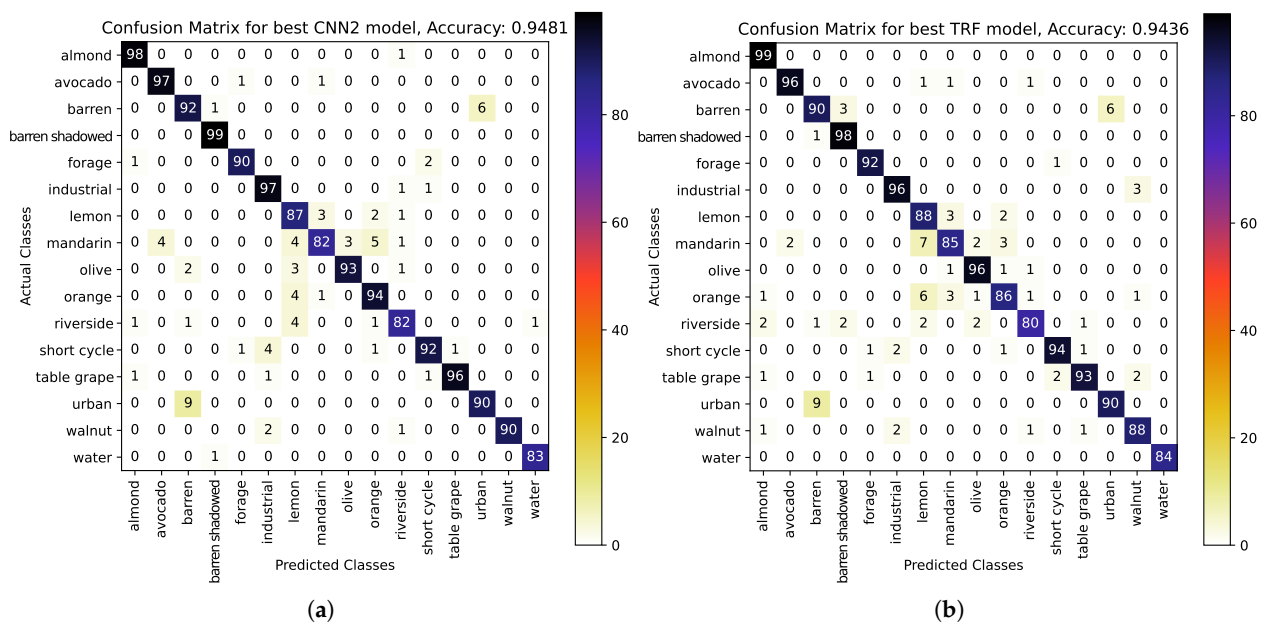


Figure 6. Confusion matrix of the best CNN2 (a) and TRF (b) models

3.1.3. Hyperparameters Marginal Contribution

A total of 9 surrogate random forest models were trained to compute the variance of each of the forest's trees' predictions using Functional ANOVA (fANOVA) to decompose the variance of the performance into the contribution of each hyperparameter and their interactions. Since each surrogate model depends on the performance of the original model, all the surrogate models differ in their structure and performance. While most of the surrogate models achieved a high performance, with a high R^2 and low RMSE, the model with the lowest performance was the surrogate model for CNN1.

This model achieved an R^2 of 0.759 and a RMSE of 0.015, although for the models with highest performance, the R^2 was higher than 0.9 and the RMSE was lower than 0.01, which indicates that the surrogate models can predict the performance of the original models with a high accuracy, which allows decomposing the variance of the performance into the contribution of each hyperparameter with a high confidence. The performance of the surrogate models for each experiment is shown in Table 7.

Table 7. Performance of the surrogate models for the fANOVA analysis. The performance metrics informed are the R^2 and RMSE for the testing set.

Metric	FFNN1	FFNN2	CNN1	CNN2	LSTM	GRU	TRF	RF	XGB
R^2	0.835	0.864	0.759	0.931	0.882	0.932	0.966	0.998	0.999
RMSE	0.017	0.013	0.015	0.017	0.007	0.007	0.006	0.001	0.001

Since the number of hyperparameters for each model is different, the number of hyperparameters and interactions that contribute to the performance of the model also differ. For instance, the CNN2 model has 7 hyperparameters, which means that there are 7 main effects and 21 two-way interactions, while the RF model has 5 hyperparameters, which means that there are 5 main effects and 10 two-way interactions. Because the number of hyperparameters varies across models, direct comparison of explained variance is not straightforward. Instead, the relative contribution of each hyperparameter is compared. However, the relative contribution of each hyperparameter to the performance of the model can be compared, which can provide insights into which hyperparameters are more im-

portant for the performance of the model. Models whose performance depends weakly on hyperparameter settings are inherently more robust; conversely, high sensitivity indicates that a poor hyperparameter configuration could substantially degrade performance. The contribution of each hyperparameter is shown in Figure 7.

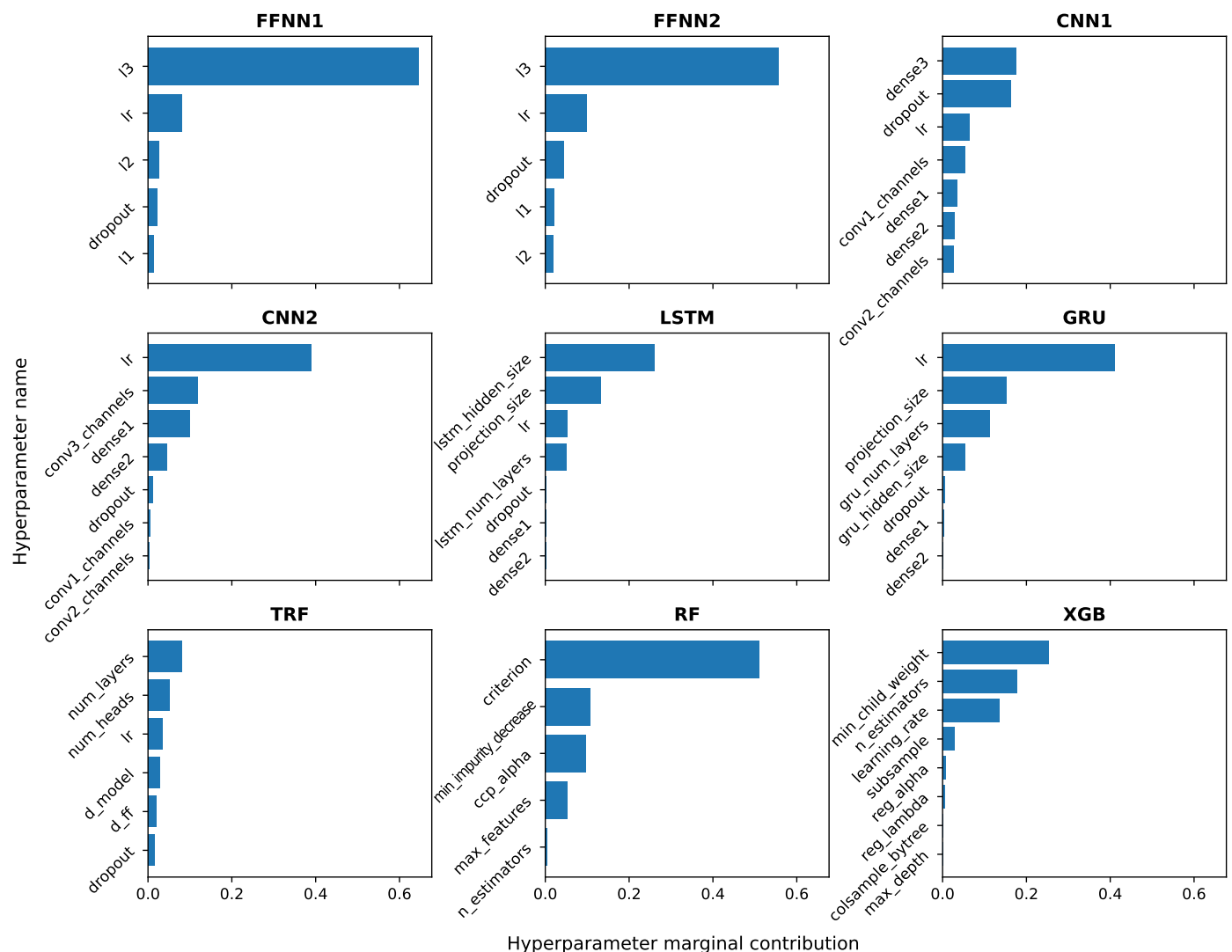


Figure 7. Marginal contribution of the hyperparameters for the different machine learning models obtained by fANOVA analysis. The acronyms for the hyperparameters are related with the previous description of the hyperparameters for each model.

The architectures FFNN1, FFNN2, and RF have hyperparameters with a high marginal contribution. In Figure 5, the bimodal behavior of RF is linked with the criterion hyperparameter. Initializing the model with gini as criterion (which is the default value for RandomForestClassifier in Python (version 3.10)'s Scikit-learn library (version 1.3.2)) has lower performance compared to using entropy as the criterion value.

To analyze the relationship between the performance and the robustness of the model, the distribution of the accuracy and the marginal contribution of the hyperparameters for each model is shown in Figure 8. The highest and lowest 5% of the accuracy distribution were dropped to avoid outliers, and the highest hyperparameter marginal contribution was used as the main value for the y-axis, while the second and third most important hyperparameters are used to create the error bars for the y-axis based on their total contribution from Figure 7. Focusing only on the two highest performing models (TRF and CNN2),

the CNN2 model reached a marginally higher peak accuracy (0.948 versus 0.944 for TRF). However, the TRF model has a lower contribution of the hyperparameters compared to the CNN2 model, which indicates that the performance of the TRF model is less sensitive to changes in the hyperparameters and that its accuracy distribution is narrower than that of CNN2. This is a decisive advantage for an operational setting, where extensive per-region hyperparameter tuning is costly: TRF attains near-peak accuracy with substantially less tuning effort. For this reason, and not on the basis of peak accuracy alone, the TRF model was selected as the best overall model for generating the final classification map for the full study area.

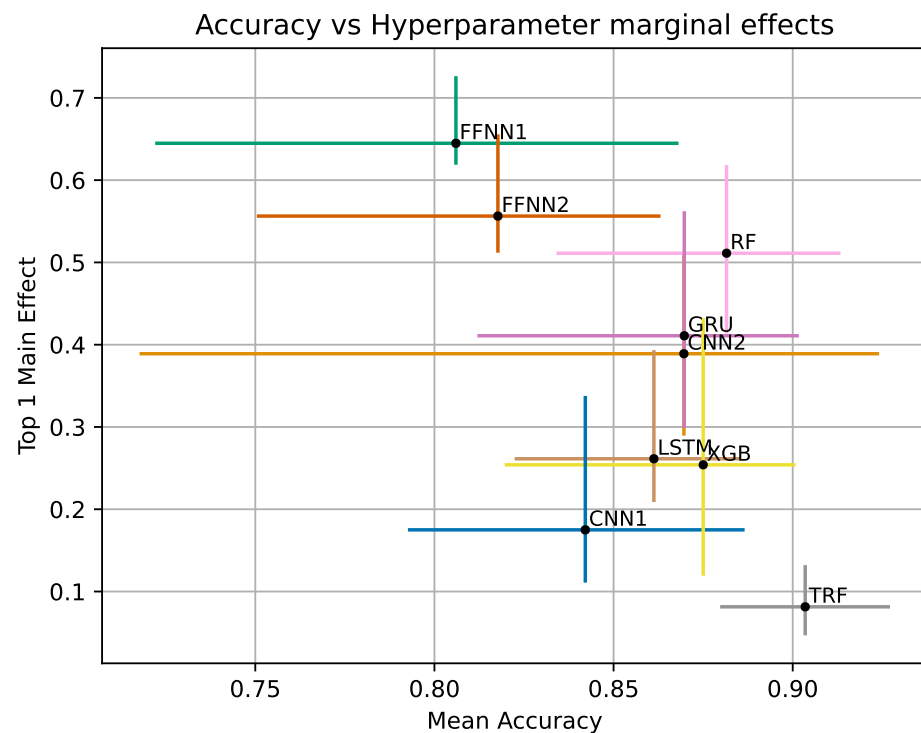


Figure 8. Summary of the distribution of the accuracy and the marginal contribution of the hyperparameters. The x-axis corresponds to the accuracy, while the y-axis represents the marginal contribution. The point that represents each model is located in the coordinate (x: mean accuracy, y: top-1 hyperparameter marginal contribution). The error bar of the x-axis represents the quantile 0.05 (left) and quantile 0.95 (right) of the accuracy distribution, while the error bar of the y-axis represents the top-2 hyperparameter marginal contribution (top) and the top-3 hyperparameter marginal contribution (bottom).

FFNN1 and FFNN2 exhibit a single dominant hyperparameter, whereas XGB distributes importance more evenly across several hyperparameters. This can be identified with Figure 7 as noted before, but this Figure gives a holistic view of the model performance and robustness, summarizing the most important hyperparameters for each model, and the distribution of the performance of the models to ease the comparison between models. Based on this analysis, plus the high performance obtained, the best model overall is the TRF model, exhibiting a high performance and robustness to changes in the hyperparameters.

Since the TRF architecture is the best model overall, the metrics (Precision, Recall, and F1-score) for the predicted class for the test set (mean across folds) for the best TRF model configuration are shown in Table 8. The TRF model achieved a high performance for all classes, with a mean accuracy of 0.944, a mean precision of 0.944, a mean recall of 0.945, and a mean F1-score of 0.944. The similarity between accuracy, precision, recall,

and F1-score indicates that the model errors are evenly distributed across classes and that no systematic bias toward specific classes was observed.

The highest performance was achieved for the water class, with a precision, recall, and F1-score of 1.000, while the lowest performance was achieved for the lemon class, with a precision of 0.946, a recall of 0.846, and an F1-score of 0.893.

The confusion identified in the confusion matrix for the TRF model (Figure 6) is reflected in precision and recall for the classes mandarin, lemon, and orange. Precision is low for mandarin and orange, while recall is low for lemon. The model is more likely to misclassify lemon samples as mandarin or orange. Similarly, the same behavior was observed for the barren and urban classes.

Table 8. Performance metrics for the best TRF model. The performance metrics informed are the Precision, Recall, and F1-score for the top-1 candidate class for the testing set (mean across folds). The values in bold indicate the highest performance for each metric, and the values in underline indicate the second highest performance.

Class	Precision	Recall	F1
almond	1.000	0.952	0.975
avocado	0.970	<u>0.980</u>	0.975
barren	0.909	<u>0.891</u>	0.900
barren shadowed	<u>0.990</u>	0.951	0.970
forage	<u>0.989</u>	0.979	<u>0.984</u>
industrial	0.970	0.960	<u>0.965</u>
lemon	0.946	0.846	0.893
mandarin	0.859	0.914	0.885
olive	0.970	0.950	0.960
orange	0.869	0.925	0.896
riverside	0.889	0.952	0.920
short cycle	0.949	0.969	0.959
table grape	0.939	0.969	0.954
urban	0.909	0.938	0.923
walnut	0.946	0.936	0.941
water	1.000	1.000	1.000
Mean	0.944	0.945	0.944

3.1.4. Final Classification Map

The three approaches to combine the predictions of the three models (Product of experts, Averaging, and Maximum rule) were tested to generate the final classification map for the study area using the best model configuration for each fold. The results of the final classification map obtained by the best model (TRF) and the three approaches to combine the predictions of the three models are shown in Figure 9.

A pixel that corresponds to the almond class was selected to show the effect over $p'_{\text{ens}}(c)$ of the different approaches. The Product of experts approach gives more weight to the classes that are consistently predicted with high probability across the three models, which can be observed in the top row of the figure, where the probability for the almond class is higher compared to the other approaches, and at the same time, the classes with lower original probabilities (like short cycle and table grape) have a lower probability compared to the other approaches, with values < 0.001 .

The Averaging approach gives equal weight to the predictions of the three models, which can be observed in the top row of the figure, where the probability for the almond class is lower compared to the Product of experts approach, but still higher than the Maximum rule approach.

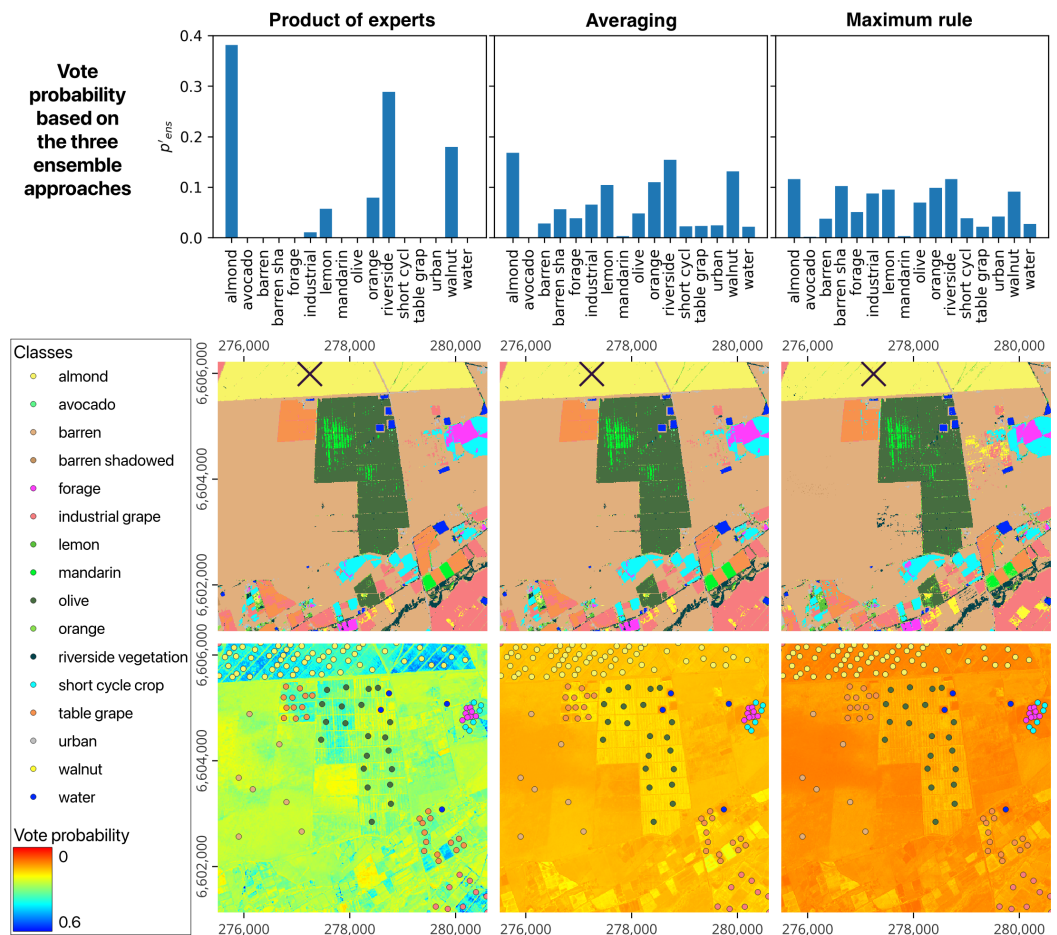


Figure 9. Comparison between the final classification map obtained by the best model (TRF) and the three approaches to combine the predictions of the three models (Product of experts, Averaging, and Maximum rule). The bar plot at the top row corresponds to $(p'_{ens}(c))$ for each class, which is the normalized ensemble probability for class c obtained by the different approaches. The map in the middle row corresponds to the classes after applying each approach ($\hat{c}$). The map in the bottom row corresponds to the $\max_c p'_{ens}(c)$, the samples collected are shown as point data. The X symbol in the middle row shows the selected pixel of the normalized ensemble probability shown in the top row.

The Maximum rule approach gives more weight to the model that predicts the highest probability for a given class, which can be observed in the top row of the figure, where the probability for the almond class is lower compared to both the Product of Experts and Averaging approaches. The potential issue with this approach emerges when a class is incorrectly assigned with a high probability by one of the models, which can lead to confusion in that pixel.

In terms of the final classification map (middle row), all three approaches correctly classify the selected pixel as almond, but there are some differences in the surrounding pixels, where the Product of experts and Averaging approaches show a more consistent classification with fewer pixels wrongly labeled (salt and pepper effect close to the crop limits) compared to the Maximum rule approach.

The effect of the different approaches is relatively marginal (at least for the Product of Experts and Averaging approaches) for the final classification map, but $p'_{ens}(c)$ varies drastically between the different approaches. The approach of Product of experts can be more robust to misclassifications with high probability, since it gives more weight to the classes that are consistently predicted with high probability across the three models. This is the approach used to generate the final classification map for the study area, since it can help to improve the performance of the final classification by reducing the impact of

outliers or misclassifications from individual models, while still giving some weight to the most confident predictions.

The final classification map for the study area is shown in Figure 10. There are small boxes to highlight some important agricultural areas to see in detail how the classification map identifies the crop borders. In some of the boxes (like boxes A and C), with a confusion between the green classes (like lemon/mandarin/olive/orange), which is what was found in the confusion matrix for the best TRF model, while in other boxes (like boxes B and D), the classification map shows a good performance with a clear identification of the crop borders. Box E and F show small populated areas with urban class identified surrounded by agricultural areas, which is a common pattern in the study area, where the rural population is associated with agricultural activities. In the left part of the study, some areas classified as water can be identified, which corresponds to the Andes Mountain range, where water bodies are not as common as the classification map shows. The reflectance of those areas, with steep slopes and high elevation, can be affected by shadows (especially during the winter season), hence the model can misclassify those areas as water.

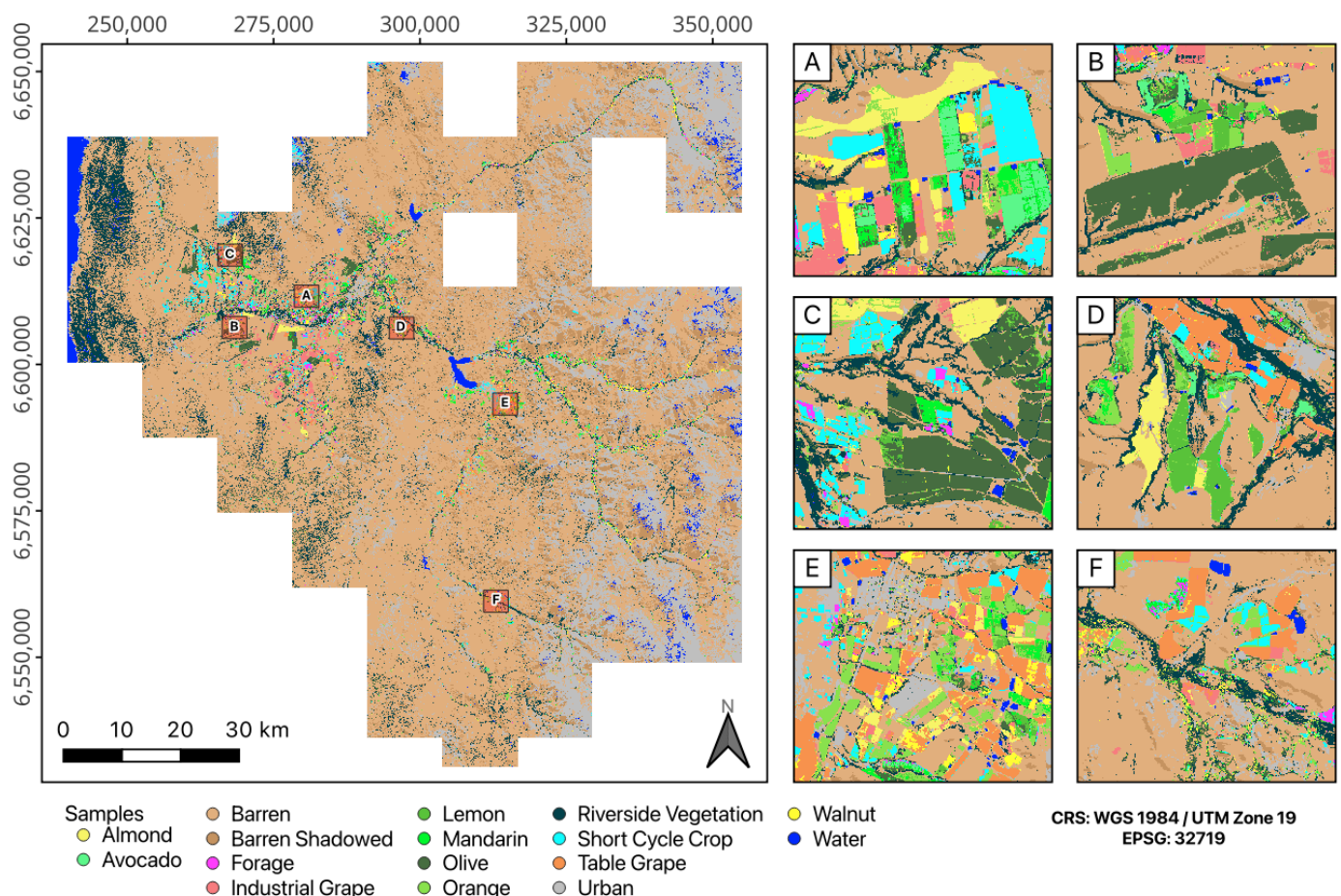


Figure 10. Final classification map for the study area obtained using the Product of Experts approach.

The result of the application of the ensemble was compared with the Fruit Cadastre of Coquimbo Region [42], extracting the class for each polygon of the cadastre, and then comparing with the classification map. This comparison should not be interpreted as a fully independent validation, since the Fruit Cadastre was also the main source of the crop labels used to train the models. Rather, its purpose is to assess whether remote sensing products can reproduce the level of detail contained in an operational cadastre, particularly for crops with similar phenology and spectral characteristics, such as the citrus classes. The Fruit Cadastre includes a larger number of classes related to fruit production, like

cherry trees and pomegranate, which were not included in this study due to a lack of representation over the study area (118.12 and 20.87 ha, respectively). Also, this cadastre does not include some of the classes of this study, like Industrial grape, which is used for base wine fabrication, which is later distilled to produce pisco (a national spirit). Hence, the comparison between the classification map and the Fruit Cadastre was done only for the 8 fruit crops included in both datasets (almond, avocado, lemon, mandarin, olive, orange, table grape and walnut), the classes short cycle crops and forage are also out of the Fruit Cadastre, since it is focused only in fruit trees, and the remainder classes are not considered agricultural areas.

In Table 9, the results of the classification map versus the Fruit Cadastre of Coquimbo Region [42] are presented. The results show that the classification results are able to identify the main classes present in the study area. The almond class was identified with an agreement of 98.62% over 1145.1 ha. In contrast, avocado showed a lower agreement of 49.15% across 2119.43 ha, with most mis-classifications attributed to olive (13.88%) and mandarin (8.81%), likely due to their similar perennial phenology (see Appendix A). Lemon achieved 75.29% agreement over 531.58 ha. For mandarin, the agreement was 34.41% over 4001.08 ha, with notable confusion with barren (15.18%) and orange (29.69%); according to CIREN, Centro de Información de Recursos Naturales [42], 788.45 ha of mandarin were newly planted between 2020 and 2021, making it the region's most expanded crop in that period. Orange reached 80.46% agreement over 296.52 ha, with confusion mainly with barren (10.13%) and lemon (3.58%). Olive was classified with 95.11% agreement over 3095.84 ha, with minor confusion with barren (4.59%). Table grape achieved 66.18% agreement over 4945.09 ha, with most confusion involving short cycle crops (8.64%), barren (7.49%), and industrial grape (6.12%). Finally, walnut was classified with 80.29% agreement over 999.67 ha, with confusion mainly with barren (12.75%) and urban (2.89%).

Table 9. Comparison of the classification map with the Fruit Cadastre of Coquimbo Region. Columns correspond to the fruit crops reported by [42] while rows represent the class results of the application of the best model over the study area. The values are in hectares (ha). The columns represent the classes included in the Fruit Cadastre, while the rows represent the classes included in the classification map. The values in bold correspond to the diagonal of the table, which represents the area of agreement between the classification map and the Fruit Cadastre for each class.

Class	Almond	Avocado	Lemon	Mandarin	Olive	Orange	Table Grape	Walnut
almond	1129.34	2.98	17.45	4.4	2.03	0.15	139.35	7.42
avocado	0	1041.59	0	24.7	0	1.2	0	0.14
barren	4.86	99.78	12.32	607.39	142.14	30.05	372.51	127.47
barren shadowed	0	1.99	0	1.65	0	0	0	0.85
forage	0	20.87	0	31.86	0	0	2.1	0.62
industrial	2.2	67.72	0	10.46	0	0	302.47	12.41
lemon	0.05	145.42	400.23	349.87	0	10.61	3.46	0.55
mandarin	0	186.82	6.15	1376.65	0	9.87	11.02	0
olive	0	294.27	34.67	194.08	2944.35	2.69	3.09	0.05
orange	0	157.13	45.81	1187.83	2.14	238.59	191.16	5.9
riverside vegetation	0	60.14	4.19	21.57	5.18	0	3.75	2.07
short cycle	8.65	9.81	1.15	75.63	0	0	427.23	6.09
table grape	0	1.54	0	3.79	0	0	3272.54	4.49
urban	0	25.71	9.61	67.14	0	3.3	80.84	28.94
walnut	0	3.66	0	44.06	0	0.06	133	802.67
water	0	0	0	0	0	0	2.57	0
Total	1145.1	2119.43	531.58	4001.08	3095.84	296.52	4945.09	999.67

The same scenario of new sowing of crops like mandarin applies broadly to the other crops. Due to the biomass accumulation through the years, the spectral signature of new plantations is more related to the barren class since the canopy coverage fraction is lower compared to older plantations, which can lead to a misclassification of young plants as barren.

4. Discussion

4.1. Comparison Between DTW and ML Models

Most of the ML architectures beat the DTW benchmark, with the exception of FFNN1 and FFNN2, with a max accuracy of 0.901 and 0.892, respectively. The baseline method achieved a max accuracy of 0.922, while the best ML model achieved an accuracy of 0.948. Most of the ML methods performed better only using reflectance data, while DTW, using the mix of reflectance and spectral indices, reached the highest performance, not only for the best run, but for every algorithm configuration.

Numerically, GRU, RF, and XGB were the only architectures whose best run was obtained on the combined reflectance and spectral indices dataset. However, the Stuart-Maxwell test (Table 6) shows that, for these three architectures, the predicted class distributions on the combined dataset are not significantly different from those obtained with reflectance only; hence, the models distribute their predictions across classes in a statistically indistinguishable way. This is not a test of whether the small accuracy gain itself is significant. For DTW, since it is based on the distance measurement between two time series, the use of spectral indices will obtain low distances for samples with similar phenology, and the use of reflectance data helps to finally distinguish between classes with similar phenology, such as mandarin, lemon, and orange. For the case of FFNN architecture, since all the input data have the same weight, the use of spectral indices, whose signature can be similar among some classes, leads to a decrease in the performance of the model. For the case of recurrent neural networks, the use of spectral indices can help to capture the temporal dynamics of the classes, which can lead to an improvement in the performance of the model. For the case of RF and XGB, since they are based on decision trees, the lowest performance was observed when only reflectance data was used, while the best performance was obtained on the combined reflectance and spectral indices dataset; this is consistent with the ability of tree-based models to select the most discriminative feature at each split, which benefits from a richer feature pool.

CNN and Transformer architectures process input differently: CNNs capture local patterns through convolution, whereas Transformers can capture long-range dependencies via self-attention. The use of spectral indices can help to define patterns in a class, although some confusion could arise based on similar phenology, and this is shown as the models achieved a better performance using only reflectance data.

The confusion matrices of the best two models (CNN2 and TRF) are similar to each other but differ from the one obtained with DTW, suggesting that combining ML and DTW predictions through ensembling could leverage their complementary error patterns to improve the final classification map. This was not tested in this work, but it remains an interesting avenue for future research.

The models presented here have been used previously for crop classification [54–58]. Several studies have reported that DTW (or TWDTW) outperforms random forest in vegetation and crop classification tasks [27–31,59], while both methods generally perform better than other classical machine learning approaches such as support vector machines and decision trees [60]. In contrast, Rafif et al. [49] compared DTW, RF, Extra Trees, and XGBoost for crop intensity mapping and found that RF achieved higher performance than DTW—a result consistent with the findings of the present work.

In this work, the performance of RF is superior to DTW, matching the findings of Rafif et al. [49]. It is also demonstrated that the performance of RF is highly sensitive to the hyperparameter selection, especially to the criterion selection. Based on the hyperparameter search and the experiments conducted, RF performed better than DTW for the datasets Spectral Indices and the mixed dataset, showing a lower performance using only reflectance data. The use of spectral indices can help to capture the temporal dynamics of the classes, which can lead to an improvement in the performance of the model, and, as stated before, the decision trees can select the most important features and the temporal window to make the classification, which can lead to a better performance than DTW.

Similarly, there is no consensus in the literature about the performance of DTW compared with neural networks for remote sensing image classification. Finnegan et al. [61] reported similar performance between DTW and a CNN-LSTM model, and proposed splitting the time series into four segments to reduce the DTW computation cost. However, their NN architecture differs substantially from the ones used here, which makes a direct comparison difficult.

The approach of 1D convolutional layers for pixel-based classification has been used in previous works for crop classification [32,62–65]. The use of 2D convolutional layers for crop classification is also reported, although this approach is related to object-based classification approaches. Kiranyaz et al. [66] compared both structures, where 1D CNN had 10k parameters, while 2D CNN had more than 1M parameters, and the 1D CNN achieved a better performance than the 2D CNN, showing that for pixel-based classification, the use of 1D convolutional layers is well suited.

Further evidence comparing DTW with neural networks for crop classification is scarce, although DTW has been used as a preprocessing step whose output feeds an NN model [56,67,68]. Other works report that bidirectional-LSTM and transformer-based architectures consistently outperform classical baselines such as RF, SVM, KNN and feedforward NNs [13,69–71]. Zhong et al. [72] similarly found that deep-learning architectures (LSTM, 1D CNN, FFNN) outperformed XGBoost, RF and SVM, in line with the present results, where 1D CNN and GRU are among the best models. The strong performance of the transformer in this domain has been reported by Rußwurm and Körner [73], who attribute it to the ability of self-attention to select or discard features—a mechanism also available to LSTMs through recurrence.

Despite their different mechanisms (self-attention in TRF, local convolutions in CNN, and recurrence in RNN) the three families of models converged to similar performance, suggesting that the dominant requirement for this task is the ability to weight discriminative time steps rather than any specific inductive bias.

4.2. Confusion Between Classes

A central challenge of this classification task is discriminating between classes with similar phenology, tree structure, and reflectance response, such as mandarin, lemon, and orange. The confusion matrices in Figure 6 reflect this: the highest confusion is observed among perennial classes (mandarin, olive, orange, and riverside vegetation), which all keep a high vegetation cover throughout the year since they do not have a dormant season. Confusion was also observed between barren and urban classes, both with similar reflectance responses in this arid region, where the spectral signal only changes when the rainy season wets the soil enough to germinate wild plants.

The TRF model achieved perfect precision, recall, and F1 for water, as expected for a class with a very distinctive spectral response, and also for almond, despite NDVI similarities with industrial grape and walnut (see Appendix A). On the other end, lemon had the lowest recall, with most confusion concentrated on mandarin and orange—a result

expected for citrus trees that share phenology, tree structure, and reflectance response. The only classes with an F1 score below 0.9 were Mandarin and Lemon, which had the most expanded crops in the region, typically planted on hillslopes with low canopy cover. In the early stages of the growing cycle their reflectance response is similar to the barren class, which explains the observed confusion.

The shadowed barren class helped to differentiate barren mountain areas from water bodies. Both have very low reflectance during winter—water through absorption, and the shadowed areas because the satellite overpass occurs when these slopes are in the mountains' shadow. Including this class as a separate label reduced misclassification of mountainous shadows as water in the final map.

One important output to highlight is the ability to classify an area with a variable number of valid scenes for each grid cell, where the coastal and mountainous areas have a lower number of valid scenes due to the presence of clouds. The models were able to learn the key features that differentiate the classes, even with a variable number of valid scenes, which is important for the operational application of the model, since in real scenarios, there will be a variable number of valid scenes for each grid cell. There is no artifact in the classification map related to the number of valid scenes, and there is no evidence of cell boundaries in the classification map. This is an advantage when applying the model to a larger area.

The use of spectral indices may introduce confusion in the neural network models, since indices like NDVI tend to be similar across classes with similar phenology (mandarin, lemon, orange), reducing rather than enhancing class separability. Accordingly, the performance of TRF, CNN, and LSTM decreases when indices are added on top of reflectance. In contrast, GRU and decision tree-based models (RF, XGB) benefit from spectral indices, since their internal feature selection mechanisms can pick the most informative inputs at each step. The Stuart-Maxwell test (Table 6) confirms that the choice of input dataset has a significant effect on the class distribution of the predictions for the neural networks, while for decision tree-based models, the differences between using spectral indices alone and the combined dataset are not significant. The selection of the input dataset is therefore a model-dependent design decision, and adding more information is not always beneficial.

The TRF model is a particularly clear example: its best performance was obtained on reflectance data, but it also achieved 0.937 and 0.938 on spectral indices and the combined dataset—both higher than most of the models analyzed with the exception of GRU on any dataset. The different class distributions across these inputs suggest that ensembling the same architecture with different inputs, or different architectures with shared inputs, could further improve the final classification map. This was outside the scope of this work, but it remains a valuable direction for future research.

For DTW, the best run combined reflectance and spectral indices: the indices shorten the distances between samples with similar phenology, while reflectance helps to differentiate them. Combining DTW and ML predictions through ensembling could leverage their complementary error patterns, although DTW's computational cost would need to be addressed before operational use (see Appendix E for reference).

The top-k analysis (see Appendix B) helps to distinguish between confusions arising from genuinely similar classes and confusions arising from a model's inability to learn the discriminative features. The bump in accuracy from top-1 to top-2, followed by diminishing returns at top-3 to top-5, indicates that the residual confusion is concentrated on a few class pairs, and that the remaining gap to 100% accuracy is at least partly due to mislabeling in samples collected through visual interpretation of high-resolution imagery. FFNN1, FFNN2, and CNN1 do not catch up even at top-5: the FFNN family weighs all inputs equally and is hurt by adding redundant inputs, and CNN1 lacks the third convolutional layer that lets

CNN2 generalize the discriminative features. This contrast highlights the importance of architectural choices when developing deep learning models for crop classification.

FFNN1, FFNN2, and RF were the models most sensitive to hyperparameter selection. In the FFNN family, the key hyperparameter was the number of neurons in the last hidden layer; in RF, the criterion (entropy outperformed Gini impurity) drives the bimodal accuracy distribution observed in Figure 5. The transformer, on the contrary, exhibited the lowest hyperparameter marginal contribution, indicating that it is more robust to hyperparameter selection. The combination of high accuracy and low hyperparameter sensitivity makes the transformer a strong candidate for operational deployment, since it reduces the tuning effort that would be required when transferring the model to new regions or sensors. It should be noted, however, that this potential transferability was not tested here and remains a hypothesis to be confirmed through independent validation in other regions and with other sensors.

When applied to the entire study area, the best model produces a classification map whose spatial patterns are broadly consistent with the Fruit Cadastre of Coquimbo Region [42], with fruit crops concentrated in the central valley, herbaceous crops in the coastal zone, and barren and urban classes predominant in the arid mountainous area. This comparison is not an independent validation, since the cadastre also provided most of the training labels; it should instead be read as an assessment of how well the remote sensing classification reproduces the detail of the cadastre. The agreement is class-dependent: it is high for crops with distinctive phenology (e.g., almond and olive), but substantially lower for mandarin, avocado, and table grape, which share evergreen or soil-dominated signals with neighboring classes and, in the case of mandarin, were affected by recent planting between the cadastre date and the analysis period. Since the training set did not include barren samples from the mountainous area, exposed rock in this region is sometimes misclassified as urban concrete. The remaining confusion sources mirror those identified in the test set: perennial crops with similar evergreen phenology (mandarin, olive, orange, riverside vegetation), young or recently established crops with soil-dominated reflectance signals, and herbaceous classes (forage, short cycle crops) with overlapping phenological cycles. These limitations highlight the challenges of crop classification in heterogeneous agricultural landscapes when using medium-resolution satellite imagery and when crop development stages vary within the same season.

4.3. Computing Time

Although the DTW algorithm was implemented with parallel processing, this method takes significantly longer than any other method applied—details about the computation time are found in Appendix E. Even more, the time spent on the ML model for each run was at max between 5–10 min, for the recurrent neural networks (LSTM and GRU), while the other models were faster due to the size of the input data, which allowed evaluating all hyperparameter combinations within a reasonable time frame. A single cell of 1280×1280 pixels would require several days to classify with DTW, rendering operational deployment infeasible.

Other studies have also reported high computation times for the DTW method, and there have been proposed methods to reduce the computation time, such as the use of parallel processing [74] or the use of GPU [17,75,76]. Another approach is to use a reduced set of samples, such as the k-means clustering method [77] or with fewer samples for running the algorithm, like Belgiu et al. [20] and Cheng and Wang [27], using 50 samples for training and testing, for 7 and 8 classes, respectively. This approach can significantly reduce the computation time by computing the DTW distance between samples and re-

moving the ones with a high similarity, but it reduces model fidelity as it lumps similar signals together.

Another approach is to use a pattern response by class instead of multiple samples [76], which significantly reduces the computation time, but it is not suitable for all cases, since it requires a representative sample from the class. Since this work aims to differentiate between classes with similar phenology, such as mandarin, lemon, and orange, the use of a single pattern response would likely lead to misclassification. On the other hand, a class like forage, whose dynamic will depend on harvesting practices, does not respond to a single pattern response (See Appendix A for the NDVI time series of each class).

To reduce the matrix computation time, the Sakoe-Chiba band [78] can be used, which restricts the search space. The same applies to Kim et al. [79] and Keogh and Ratanamahatana [80] bands for setting upper and lower limits for the DTW distance [81]. These strategies limit the distance calculation to specific cells, avoiding the need to compute the distance for all pairs of samples, which can significantly reduce the computation time 5 to 20 times for a narrow band. Once again, the forage class could be a problem for this approach, due to its dynamic response, which can lead to a wide range of reflectance responses. Even adopting this strategy, the DTW method would still take significantly longer than the machine learning methods at deployment time.

Figure A3 confirms that the number of samples is the dominant factor in DTW computation time. Reducing the number of training samples per class—for example, via k-means selection—is therefore the most effective way to lower this cost, while keeping a higher number of samples for classes with irregular phenological patterns, such as forage.

5. Conclusions

The performance of different machine learning and dynamic time warping algorithms was evaluated for crop classification using Sentinel-2 data in the Limarí basin, Chile. The analysis also evaluated whether reflectance data, spectral indices, or the combination of both provide better classification results, in contrast to previous works that focus on using spectral indices alone. The results demonstrate that machine learning models can outperform dynamic time warping, either using a neural network or a decision tree-based model. Architectures containing convolutional layers or transformer architectures performed the best, with an accuracy of 0.948 and 0.944, respectively, while the best dynamic time warping algorithm had an accuracy of 0.921.

The best performing DTW was when both reflectance and spectral indices were used, indicating that this algorithm benefits from the combination of both datasets. For machine learning models, the best performance was observed when using only reflectance data, which could indicate that the use of only spectral data is not enough to learn the key features that differentiate the classes. This finding challenges the common practice of relying solely on spectral indices and suggests that raw reflectance carries discriminative information not fully captured by spectral indices. This could be due to the similarity between different crops, which have similar phenological patterns, and the use of spectral indices may add noise to the model instead of adding separability between the classes. For the decision tree-based models, the best performance was observed when using both reflectance and spectral indices data, which could indicate that these models can learn the key features that differentiate the classes from both datasets. The Limarí basin covers three different climatic zones, from the coastal area affected by the *camanchaca*, to the temperate central valley, and the cold Andean section with winter snow. Since the same crop classes are observed under different temperature, precipitation, and radiation conditions within the study area, the main findings of this work, in particular the value of reflectance data for tree-based models

and the limited contribution of spectral indices when classes share similar phenology, are expected to be applicable to other semi-arid Mediterranean regions with similar crop types. For regions with short-cycle herbaceous crops with stronger phenological contrasts, the contribution of spectral indices may be different and should be evaluated in future work.

One key aspect to consider when choosing the model for this task is the computational cost. The dynamic time warping algorithm is computationally expensive, and an operational application over a large-scale field is unfeasible (see Appendix E for reference). Although there are techniques to reduce the computation time, such as the use of a temporal window or clustering the samples, the computation time is still significant, and cannot be compared with the inference time of the machine learning models.

The results of fANOVA analysis showed that the different architectures have different sensitivity to the hyperparameter selection. For the feed-forward neural networks, the number of neurons in the last layer before the output layer had the highest marginal contribution to the performance of the model, while for the convolutional neural networks, the number of filters in the convolutional layers had the highest marginal contribution. For the decision tree-based models, the criterion hyperparameter had the highest marginal contribution to the performance of the random forest model. For the transformer-based model, the number of attention heads and the number of transformer encoder layers had the highest marginal contribution to the performance of the model, although this model exhibited the lowest sensitivity to the hyperparameter selection, indicating that this model is more robust to the hyperparameter selection. The latter indicates that the structure of the model is more important than the hyperparameter selection for this model, and there is a good chance to obtain an initial high performance with a good structure, and then fine-tune the hyperparameters to improve the performance, suggesting that a well-chosen architecture can yield strong baseline performance, which can then be refined through hyperparameter tuning.

As expected, confusion was observed in classes with similar characteristics, such as barren and urban, or between perennial crops, such as mandarin, lemon, and orange. The top-k analysis showed that the confusion can be solved by including the top-2 candidate classes. Compared with the Fruit Cadastre, there was confusion between crop labeled polygons from the cadastre with barren class, and this is related to the age of the crops, where new or recently established crops are more likely to be classified as barren.

Finally, from the three ensemble approaches evaluated, the Product of Experts approach showed the best separation of the final label compared to the other two approaches, indicating that this approach is more robust to the outliers or misclassifications from the individual models.

Overall, all evaluated models performed well, but they differ in accuracy, computational cost, and sensitivity to hyperparameter selection, and these differences directly inform model selection for operational crop classification. For practitioners aiming at large-scale deployment, dynamic time warping is not a viable choice given its computational cost, even when implemented with parallel processing or band restrictions. Among the machine learning models, the transformer is the most attractive option, since it combines high accuracy with low sensitivity to the hyperparameter selection, which is expected to reduce the tuning effort when transferring the model to new regions or sensors. This expectation is supported, within the limits of this study, by the fact that the same crop classes were classified across the three climatic zones of the Limarí basin (coastal, central valley, and Andean), but it has not been validated outside the study area and should be confirmed with independent data before any operational use beyond this region. CNN-based architectures are a competitive alternative when peak accuracy is the priority and

computational resources are available for hyperparameter tuning, while tree-based models such as random forest and XGBoost remain useful when interpretability or fast inference on tabular features is needed, and benefit the most from combining reflectance and spectral indices. For researchers, the main contribution of this work is the evidence that raw reflectance, rather than spectral indices, is the more informative input for most machine learning architectures in landscapes dominated by phenologically similar perennial crops, which suggests that the common practice of relying on spectral indices alone should be revisited in future crop classification studies.

Author Contributions: Conceptualization, A.A.T.; methodology, A.A.T. and A.B.; software, A.A.T.; validation, A.A.T. and A.B.; formal analysis, A.A.T.; investigation, A.A.T.; resources, A.A.T. and A.B.; data curation, A.A.T.; writing—original draft preparation, A.A.T. and A.B.; writing—review and editing, A.A.T. and A.B.; visualization, A.A.T.; supervision, A.B.; project administration, A.A.T. and A.B.; funding acquisition, A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by startup provided to Andrew Bennett by the University of Arizona College of Sciences and the Department of Hydrology and Atmospheric Sciences. The APC was funded by the University of Arizona College of Sciences and the Department of Hydrology and Atmospheric Sciences.

Data Availability Statement: The data processing and classification code is available at <https://github.com/aldotapia/S2-ClassifierComparison> (accessed on 14 July 2026). We also provide the scene list used in this work, which can be found at the same repository. Due to the size of the input data, we do not provide the raw scenes or pre-processed data, but the user can download the scenes from the Copernicus Browser <https://browser.dataspace.copernicus.eu> (accessed on 14 July 2026). A sample to reproduce the results and the 3 TRF models with the best hyperparameters are included in the repository. The class probabilities predicted by the aforementioned models and the final classification map are stored in Zenodo: <https://doi.org/10.5281/zenodo.19868872> (accessed on 14 July 2026).

Acknowledgments: We thank Carlos Villalobos for providing field-collected and secondary-source samples. His contribution was essential to this work. We would like to express our gratitude to the anonymous reviewers for the insightful suggestions, which improved the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. NDVI Time-Series of Classes

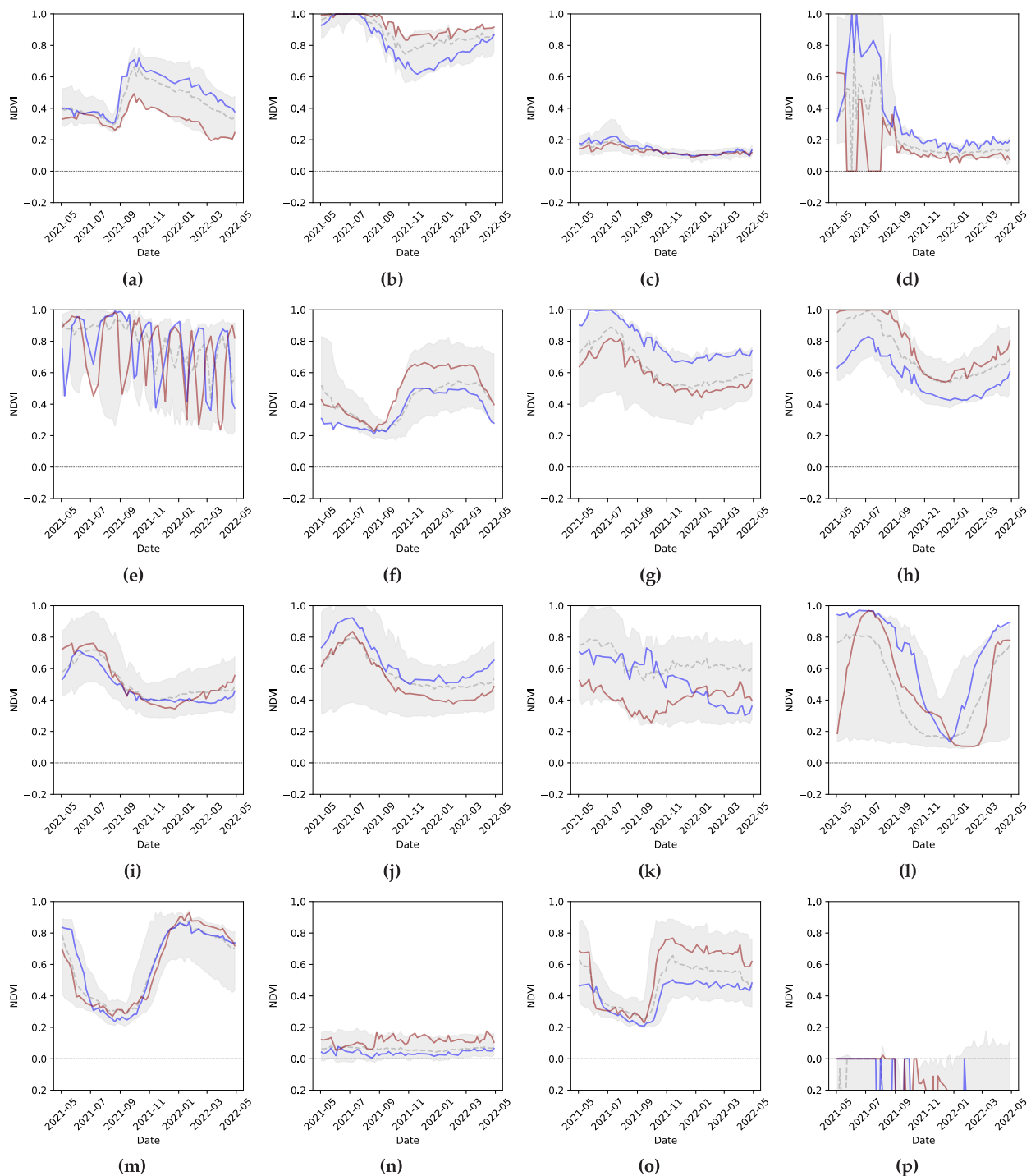


Figure A1. Sample of the NDVI time series for each class. The x-axis represents the date, and the y-axis represents the NDVI value. The light gray area represents the 5th and 95th percentiles of the NDVI values for each class, and the gray line represents the median of the NDVI values for each class. The red and blue lines and blue are two examples of the NDVI time series for each class. The classes are: (a) almond, (b) avocado, (c) barren, (d) barren shadowed, (e) forage, (f) industrial grape, (g) lemon, (h) mandarin, (i) olive, (j) orange, (k) riverside vegetation, (l) short cycle crop, and (m) table grape, (n) urban, (o) walnut, and (p) water.

Appendix B. Top- k Analysis

The top- k analysis is a technique to evaluate the performance of a model by considering the top- k most probable predictions for each sample. This analysis is useful to understand the performance of the model in the presence of confusion between classes with similar characteristics. With this analysis, the performance of the model can be observed in the presence of confusion between classes with similar characteristics and the performance of the model in the presence of confusion between classes with different characteristics.

To compute the top- k analysis, the raw output of each of the models is processed, sorting the probabilities of the predictions and cutting the list at the top- k most probable predictions. Then, the accuracy of the model can be computed by considering the true class among the top- k predictions.

The top- k results for the DTW and the best model of each ML experiment are presented in Figure A2.

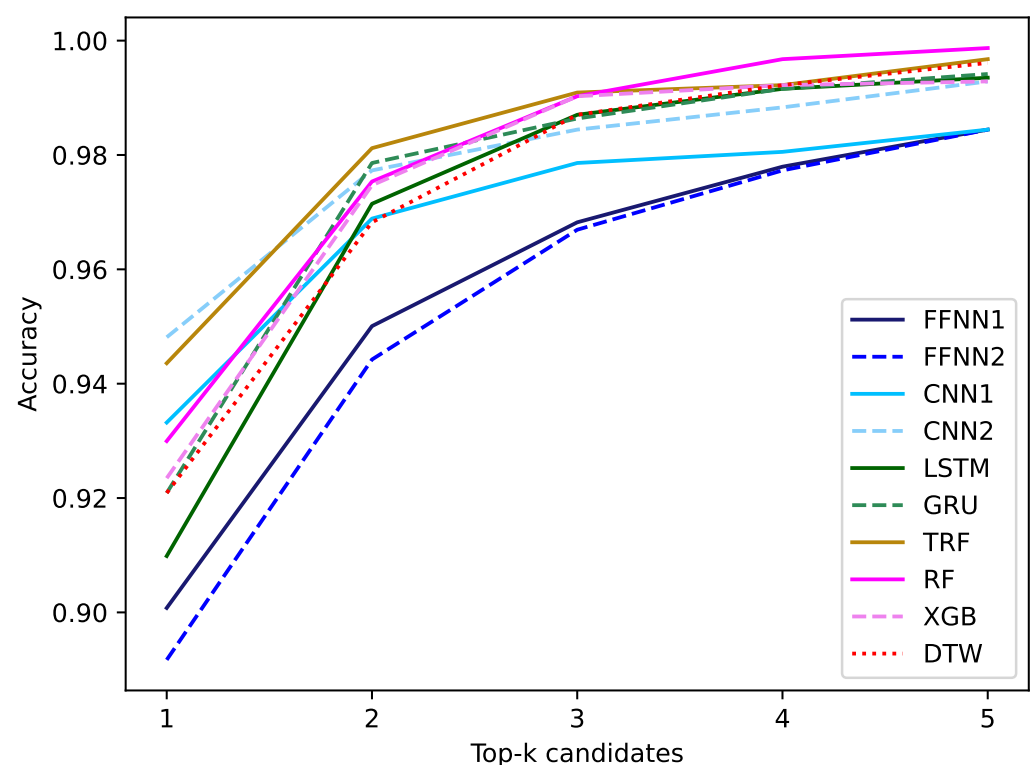


Figure A2. Top-1 to top-5 performance comparison for DTW and the best model of each ML experiment.

The results show that the performance of the model improves as more predictions are considered, but the improvement is not significant beyond the top-2 predictions. This is expected since the majority of the confusion comes from classes with similar characteristics, and the model is likely to include the true class in the top-2 predictions (improving the performance from 0.922 to 0.970).

The model that benefits the most from top-1 to top-2 is the best LSTM model, with an increase of 0.062 in accuracy, although all models show similar increases from their baselines. Including more classes in the top- k analysis shows diminishing returns beyond the top-2, and may limit the applicability of such predictions. All models show almost perfect performance at top-5, which is expected since there are only 16 classes in total, and the model is likely to include the true class in the top-5 predictions.

Appendix C. Hyperparameter Combinations per Architecture

Appendix C.1. Hyperparameter Description

A description and grouping of the hyperparameters used in this research are the following:

- Shared across all neural network architectures:
 - *lr* (learning rate, Adam optimizer): step size controlling how much model weights are updated at each optimization step.
 - *dropout* (dropout rate): fraction of units randomly set to zero during training, used as a regularizer to mitigate overfitting.
- Dense (fully-connected) layer width—*l1*, *l2*, *l3* (FFNN1, FFNN2, CNN1, CNN2) and *dense1*, *dense2* (LSTM, GRU): number of neurons in each fully-connected layer of the model's dense head. Larger values increase representational capacity at the cost of more parameters.
- Convolutional filter count—*conv1_channels*, *conv2_channels*, *conv3_channels* (CNN1, CNN2): number of convolutional filters (output channels) learned at each convolutional layer, controlling the number of distinct feature maps extracted from the input.
- Input projection dimension—*projection_size* (LSTM, GRU) and *d_model* (Transformer): dimensionality to which input features are linearly projected before being fed to the sequence model. For LSTM/GRU, a value of 0 disables the projection; for the Transformer, it defines the embedding size used throughout the encoder.
- Recurrent hidden size—*lstm_hidden_size*, *gru_hidden_size*: number of units in each recurrent cell, controlling the dimensionality of the hidden state propagated through time.
- Recurrent depth—*lstm_num_layers*, *gru_num_layers*: number of stacked recurrent layers, allowing hierarchical temporal feature extraction.
- Transformer-specific:
 - *num_heads*: number of parallel attention heads in the multi-head self-attention mechanism, each attending to different representation subspaces.
 - *num_layers*: number of stacked encoder blocks in the Transformer.
 - *d_ff*: dimensionality of the position-wise feed-forward sublayer inside each encoder block.
- Shared across tree-based ensembles (RF, XGBoost):
 - *n_estimators*: number of decision trees in the ensemble.
- Random forest specific:
 - *max_features*: number of features considered when searching for the best split at each node (sqrt and log2 use $\sqrt{p}$ and $\log_2 p$ of the p available features; None uses all of them).
 - *criterion*: function used to measure the quality of a split (Gini impurity or Shannon entropy/information gain).
 - *min_impurity_decrease*: minimum impurity reduction required for a node to be split, used as a pre-pruning criterion.
 - *ccp_alpha*: complexity parameter for cost-complexity (post-)pruning; larger values produce smaller, more regularized trees.
- XGBoost specific:
 - *max_depth*: maximum depth of each individual tree, bounding model complexity per boosting round.
 - *min_child_weight*: minimum sum of instance Hessian weights required in a child node; higher values yield more conservative splits.

- *learning_rate*: shrinkage applied to the contribution of each new tree, trading training speed for generalization.
- *subsample*: fraction of training samples randomly drawn (without replacement) to fit each tree, providing stochastic regularization.
- *colsample_bytree*: fraction of input features randomly sampled to build each tree, decorrelating the ensemble.
- *reg_lambda*: L_2 regularization coefficient applied to leaf weights.
- *reg_alpha*: L_1 regularization coefficient applied to leaf weights, encouraging sparsity.

Appendix C.2. Model Hyperparameter Configuration

For the neural network architectures, the models and hyperparameters evaluated are as follows:

- Feed-forward neural networks case 1 (FFNN1):
 - Number of neurons in layer 1 ($l1$): 64, 128, 256, 512
 - Number of neurons in layer 2 ($l2$): 32, 64, 128, 256
 - Number of neurons in layer 3 ($l3$): 16, 32, 64, 128
 - Learning rate (Adam optimizer) (lr): 0.001, 0.0001
 - Dropout rate (*dropout*): 0.0, 0.15, 0.3
- Feed-forward neural networks case 2 (FFNN2):
 - Number of neurons in layer 1 ($l1$): 64, 128, 256, 512
 - Number of neurons in layer 2 ($l2$): 32, 64, 128, 256
 - Number of neurons in layer 3 ($l3$): 16, 32, 64, 128
 - Learning rate (Adam optimizer) (lr): 0.001, 0.0001
 - Dropout rate (*dropout*): 0.0, 0.15, 0.3
- Convolutional neural networks case 1 (CNN1):
 - Number of filters in convolutional layer 1 (*conv1_channels*): 32, 64, 128
 - Number of filters in convolutional layer 2 (*conv2_channels*): 16, 32, 64
 - Number of neurons in layer 1 ($l1$): 128, 256, 512
 - Number of neurons in layer 2 ($l2$): 64, 128, 256
 - Number of neurons in layer 3 ($l3$): 32, 64, 128
 - Learning rate (Adam optimizer) (lr): 0.001, 0.0001
 - Dropout rate (*dropout*): 0.0, 0.15, 0.3
- Convolutional neural networks case 2 (CNN2):
 - Number of filters in convolutional layer 1 (*conv1_channels*): 32, 64, 128
 - Number of filters in convolutional layer 2 (*conv2_channels*): 64, 128, 256
 - Number of filters in convolutional layer 3 (*conv3_channels*): 128, 256, 512
 - Number of neurons in layer 1 ($l1$): 128, 256, 512
 - Number of neurons in layer 2 ($l2$): 128, 256, 512
 - Learning rate (Adam optimizer) (lr): 0.001, 0.0001
 - Dropout rate (*dropout*): 0.0, 0.15, 0.3
- Recurrent neural networks case LSTM (LSTM):
 - Projection size (*projection_size*): 0, 64, 128 (0 means no projection layer)
 - Number of units in the LSTM layer (*lstm_hidden_size*): 64, 128, 256
 - Number of LSTM layers (*lstm_num_layers*): 1, 2, 3
 - Number of neurons in layer 1 (*dense1*): 128, 256
 - Number of neurons in layer 2 (*dense2*): 128, 256
 - Learning rate (Adam optimizer) (lr): 0.001, 0.0001
 - Dropout rate (*dropout*): 0.0, 0.15, 0.3

- Recurrent neural networks case GRU (GRU):
 - Projection size (*projection_size*): 0, 64, 128 (0 means no projection layer)
 - Number of units in the GRU layer (*gru_hidden_size*): 64, 128, 256
 - Number of GRU layers (*gru_num_layers*): 1, 2, 3
 - Number of neurons in layer 1 (*dense1*): 128, 256
 - Number of neurons in layer 2 (*dense2*): 128, 256
 - Learning rate (Adam optimizer) (*lr*): 0.001, 0.0001
 - Dropout rate (*dropout*): 0.0, 0.15, 0.3
- Transformer (TRF):
 - Projection size (*d_model*): 128, 256
 - Number of heads in the multi-head attention (*num_heads*): 2, 4, 8, 16
 - Number of transformer encoder layers (*num_layers*): 1, 2, 4, 8
 - Number of neurons on feed-forward layer in the encoder (*d_ff*): 128, 256, 512
 - Learning rate (Adam optimizer) (*lr*): 0.001, 0.0001
 - Dropout rate (*dropout*): 0.0, 0.15, 0.3

For the decision tree-based methods, the models and hyperparameters evaluated are as follows:

- Random forest (RF):
 - Number of trees (*n_estimators*): 50, 100, 200, 300, 400
 - Maximum features (*max_features*): sqrt, log2, None
 - Criterion (*criterion*): gini, entropy
 - Minimum impurity decrease (*min_impurity_decrease*): 0.0, 0.01, 0.001
 - Ccp alpha (*ccp_alpha*): 0.0, 0.01, 0.001
- Extreme gradient boosting (XGBoost):
 - Number of trees (*n_estimators*): 100, 500, 1000, 2000
 - Maximum depth (*max_depth*): 3, 5, 9, 12
 - Minimum child weight (*min_child_weight*): 1, 5, 10
 - Learning rate (*learning_rate*): 0.01, 0.1, 0.2
 - Subsample (*subsample*): 0.5, 0.7, 0.9, 1.0
 - Column sample by tree (*colsample_bytree*): 0.5, 0.7, 0.9, 1.0
 - L2 regularization term on weights (*reg_lambda*): 0.1, 1.0, 10.0
 - L1 regularization term on weights (*reg_alpha*): 0.0, 0.1, 1.0

Table A1 reports the number of hyperparameter combinations evaluated for each machine learning architecture, together with the resulting number of training runs once the 3-fold cross-validation and the three input configurations (reflectance, spectral indices, and their combination) are taken into account. Each run corresponds to a single fit used to score a particular hyperparameter setting, not to an independent final model. The large total is mostly explained by XGB, whose grid is substantially denser than the other architectures because it includes a wider range of tree-based hyperparameters (number of estimators, learning rate, maximum depth, subsample, and column sampling rates).

Table A1. Number of hyperparameter combinations and total tuning runs per architecture. Total runs = combinations $\times$ 3 folds $\times$ 3 input configurations.

Architecture	Hyperparameter Combinations	Total Tuning Runs
FFNN1	384	3456
FFNN2	384	3456
CNN1	1458	13,122
CNN2	1458	13,122

Table A1. Cont.

Architecture	Hyperparameter Combinations	Total Tuning Runs
LSTM	648	5832
GRU	648	5832
TRF	576	5184
RF	270	2430
XGB	20,736	186,624
Total	26,562	239,058

Appendix D. Stuart-Maxwell Test Computation

The Stuart-Maxwell test is a non-parametric test that assesses the marginal homogeneity of contingency tables between two models' output with three or more categories (an extension of McNemar's test, for multiclass classification). Given a $k \times k$ contingency table N , where n_{ij} denotes the number of samples assigned to class i by the first model and to class j by the second, the row and column marginals are $n_{i\cdot} = \sum_j n_{ij}$ and $n_{\cdot j} = \sum_i n_{ij}$, respectively. The marginal difference vector is defined as $\mathbf{d} = (d_1, d_2, \dots, d_{k-1})$, where $d_i = n_{i\cdot} - n_{\cdot i}$ is the difference between the number of times each model predicts class i . One class is excluded because the differences sum to zero. The estimated variance and covariance of these differences are:

$$\hat{\text{Var}}(d_i) = n_{i\cdot} + n_{\cdot i} - 2n_{ii}, \quad (\text{A1})$$

$$\hat{\text{Cov}}(d_i, d_j) = -(n_{ij} + n_{ji}), \quad i \neq j. \quad (\text{A2})$$

These elements form the covariance matrix S :

$$S = \begin{pmatrix} \hat{\text{Var}}(d_1) & \hat{\text{Cov}}(d_1, d_2) & \cdots & \hat{\text{Cov}}(d_1, d_{k-1}) \\ \hat{\text{Cov}}(d_2, d_1) & \hat{\text{Var}}(d_2) & \cdots & \hat{\text{Cov}}(d_2, d_{k-1}) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\text{Cov}}(d_{k-1}, d_1) & \hat{\text{Cov}}(d_{k-1}, d_2) & \cdots & \hat{\text{Var}}(d_{k-1}) \end{pmatrix}. \quad (\text{A3})$$

The test statistic is then:

$$\chi^2 = \mathbf{d}^\top S^{-1} \mathbf{d}, \quad (\text{A4})$$

which follows a chi-squared distribution with $k - 1$ degrees of freedom under the null hypothesis. The null hypothesis (H_0) states that the marginal distributions are equal, i.e., both models assign the same overall frequency to each class. The alternative hypothesis (H_1) states that at least one class is predicted at a different rate by the two models.

Since the evaluation includes more than 2 models, the Holm-Bonferroni correction was applied to adjust the significance level for multiple comparisons. The corrected p -value (sorted from lowest to highest) is computed as:

$$P_k \leq \frac{\alpha}{m - k + 1}, \quad (\text{A5})$$

where P_k is the p -value for the k -th comparison, α is the significance level, and m is the number of comparisons. The corrected p -value is then used to determine the significance of the difference between the models.

Appendix E. DTW Computation Time

The DTW/TWDTW algorithm is computationally expensive, especially when the number of samples and the length of the time series are large. On average, the computation

time to run a DTW experiment is 475.10 min, with a minimum of 185.85 and a maximum of 986.42 min. This computation increases significantly when classifying the whole study area, considering that for each run, the algorithm classified 528 samples versus 1072 references. Each cell has approximately 1.6 million pixels, so the total number of samples to classify for the whole study area is approximately 90 million, which would require a significant amount of time to classify the whole area using the DTW method. In contrast, the machine learning methods are significantly faster.

The high computation time for DTW is primarily due to its pairwise comparison approach, where each test sample is compared to every training sample, and the cost path is evaluated for each combination. This process is repeated for each fold in cross-validation, further increasing the total time. Machine learning methods such as NN and RF, on the other hand, require an initial training phase, but once trained, their inference time is much lower. NN models take longer to train than RF due to the iterative optimization and larger number of parameters, but both are significantly faster than DTW for both training and evaluation. Each test sample was compared with 1072 training samples for 69 dates, and for either 4 or 8 channels.

To better understand the factors that impact the DTW computation time, a small experiment was conducted to measure the time required to compute the distance matrix for different numbers of samples (from 2^0 to 2^{10}), dates (from 10 to 90 with a step of 20), and channels (1, 4, and 8). The results of this experiment are presented in Figure A3.

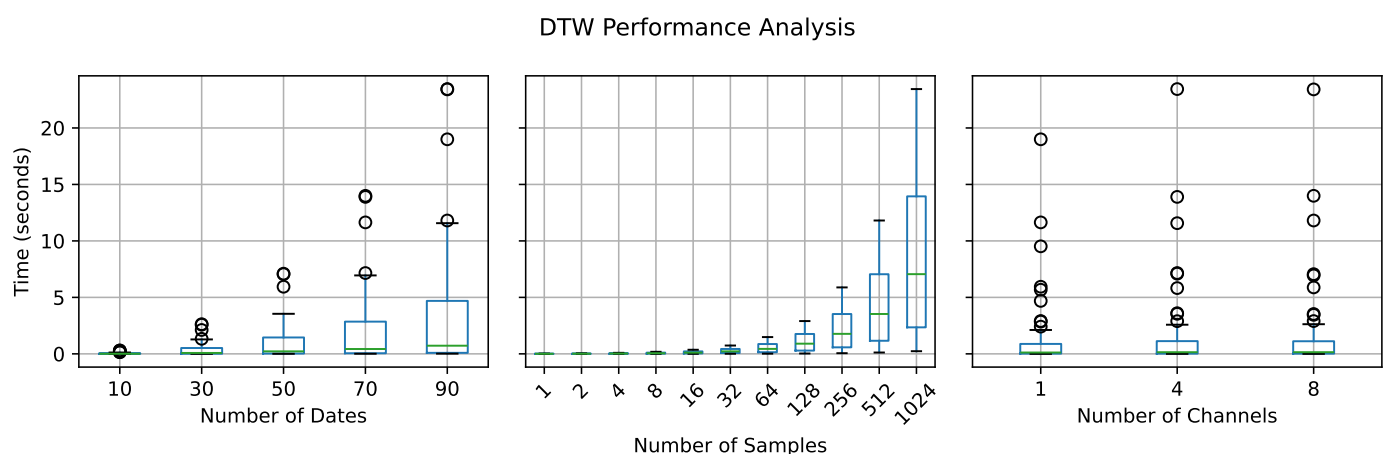


Figure A3. DTW computation time by number of samples, dates and channels.

The DTW computation time increases with the number of samples and dates, with a higher impact of the number of samples. There is no trend in the computation time with the number of channels. For 70 dates, 1024 samples, and 8 channels, the DTW computation time was 13.99 s. Extrapolating to 16 classes, 33 samples per class, and 3 folds yields an estimated total of 370 min—consistent with the average observed in this study.

Based on Figure A3, the most sensitive factor for the DTW computation time is the number of samples, hence reducing the number of samples per class or using k-means to select samples can eventually reduce the computation time, considering using multiple samples for forage for the given reasons.

References

1. Jain, M.; Mondal, P.; DeFries, R.S.; Small, C.; Galford, G.L. Mapping Cropping Intensity of Smallholder Farms: A Comparison of Methods Using Multiple Sensors. *Remote Sens. Environ.* **2013**, *134*, 210–223. [CrossRef]
2. Congalton, R.; Green, K. *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2019.

3. Immitzer, M.; Vuolo, F.; Atzberger, C. First Experience with Sentinel-2 Data for Crop and Tree Species Classifications in Central Europe. *Remote Sens.* **2016**, *8*, 166. [\[CrossRef\]](#)
4. Sitokonstantinou, V.; Papoutsis, I.; Kontoes, C.; Lafarga Arnal, A.; Armesto Andrés, A.P.; Garraza Zurbano, J.A. Scalable Parcel-Based Crop Identification Scheme Using Sentinel-2 Data Time-Series for the Monitoring of the Common Agricultural Policy. *Remote Sens.* **2018**, *10*, 911. [\[CrossRef\]](#)
5. De Beurs, K.M.; Henebry, G.M. Spatio-Temporal Statistical Methods for Modelling Land Surface Phenology. In *Phenological Research: Methods for Environmental and Climate Change Analysis*; Hudson, I.L., Keatley, M.R., Eds.; Springer: Dordrecht, The Netherlands, 2009; pp. 177–208. [\[CrossRef\]](#)
6. García-Gutiérrez, V.; Meza, F. Modeling Phenology Combining Data Assimilation Techniques and Bioclimatic Indices in a Cabernet Sauvignon Vineyard (*Vitis vinifera* L.) Central Chile. *Remote Sens.* **2023**, *15*, 3537. [\[CrossRef\]](#)
7. Heupel, K.; Spengler, D.; Itzerott, S. A Progressive Crop-Type Classification Using Multitemporal Remote Sensing Data and Phenological Information. *PFG- Photogramm. Remote Sens. Geoinf. Sci.* **2018**, *86*, 53–69. [\[CrossRef\]](#)
8. Simonneaux, V.; Duchemin, B.; Helson, D.; Er-Raki, S.; Olioso, A.; Chehbouni, A.G. The Use of High-resolution Image Time Series for Crop Classification and Evapotranspiration Estimate over an Irrigated Area in Central Morocco. *Int. J. Remote Sens.* **2008**, *29*, 95–116. [\[CrossRef\]](#)
9. Piedadlobo, L.; Hernández-López, D.; Ballesteros, R.; Chakhar, A.; Del Pozo, S.; González-Aguilera, D.; Moreno, M.A. Scalable Pixel-Based Crop Classification Combining Sentinel-2 and Landsat-8 Data Time Series: Case Study of the Duero River Basin. *Agric. Syst.* **2019**, *171*, 36–50. [\[CrossRef\]](#)
10. Kobayashi, N.; Tani, H.; Wang, X.; Sonobe, R. Crop Classification Using Spectral Indices Derived from Sentinel-2A Imagery. *J. Inf. Telecommun.* **2020**, *4*, 67–90. [\[CrossRef\]](#)
11. Lebourgeois, V.; Dupuy, S.; Vintrou, É.; Ameline, M.; Butler, S.; Bégué, A. A Combined Random Forest and OBIA Classification Scheme for Mapping Smallholder Agriculture at Different Nomenclature Levels Using Multisource Data (Simulated Sentinel-2 Time Series, VHRS and DEM). *Remote Sens.* **2017**, *9*, 259. [\[CrossRef\]](#)
12. Hu, Q.; Wu, W.-B.; Song, Q.; Lu, M.; Chen, D.; Yu, Q.-Y.; Tang, H.-J. How do temporal and spectral features matter in crop classification in Heilongjiang Province, China? *J. Integr. Agric.* **2017**, *16*, 324–336. [\[CrossRef\]](#)
13. Lozano-Tello, A.; Siesto, G.; Fernández-Sellers, M.; Caballero-Mancera, A. Evaluation of the Use of the 12 Bands vs. NDVI from Sentinel-2 Images for Crop Identification. *Sensors* **2023**, *23*, 7132. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Li, M.; Zang, S.; Zhang, B.; Li, S.; Wu, C. A Review of Remote Sensing Image Classification Techniques: The Role of Spatio-contextual Information. *Eur. J. Remote Sens.* **2014**, *47*, 389–411. [\[CrossRef\]](#)
15. Teixeira, I.; Morais, R.; Sousa, J.J.; Cunha, A. Deep Learning Models for the Classification of Crops in Aerial Imagery: A Review. *Agriculture* **2023**, *13*, 965. [\[CrossRef\]](#)
16. Csillik, O.; Belgiu, M.; Asner, G.P.; Kelly, M. Object-Based Time-Constrained Dynamic Time Warping Classification of Crops Using Sentinel-2. *Remote Sens.* **2019**, *11*, 1257. [\[CrossRef\]](#)
17. Maus, V.; Câmara, G.; Cartaxo, R.; Sanchez, A.; Ramos, F.M.; de Queiroz, G.R. A Time-Weighted Dynamic Time Warping Method for Land-Use and Land-Cover Mapping. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 3729–3739. [\[CrossRef\]](#)
18. Belgiu, M.; Csillik, O. Sentinel-2 Cropland Mapping Using Pixel-Based and Object-Based Time-Weighted Dynamic Time Warping Analysis. *Remote Sens. Environ.* **2018**, *204*, 509–523. [\[CrossRef\]](#)
19. Maus, V.; Câmara, G.; Appel, M.; Pebesma, E. dtwSat: Time-Weighted Dynamic Time Warping for Satellite Image Time Series Analysis in R. *J. Stat. Softw.* **2019**, *88*, 1–31. [\[CrossRef\]](#)
20. Belgiu, M.; Zhou, Y.; Marshall, M.; Stein, A. Dynamic Time Warping for Crops Mapping. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, XLIII-B3-2020, 947–951. [\[CrossRef\]](#)
21. Peng, Q.; Shen, R.; Dong, J.; Han, W.; Huang, J.; Ye, T.; Zhao, W.; Yuan, W. A New Method for Classifying Maize by Combining the Phenological Information of Multiple Satellite-Based Spectral Bands. *Front. Environ. Sci.* **2023**, *10*, 1089007. [\[CrossRef\]](#)
22. Liu, M.; He, W.; Zhang, H. WPS: A Whole Phenology-Based Spectral Feature Selection Method for Mapping Winter Crop from Time-Series Images. *ISPRS J. Photogramm. Remote Sens.* **2024**, *210*, 141–159. [\[CrossRef\]](#)
23. Belgiu, M.; Drăguț, L. Random Forest in Remote Sensing: A Review of Applications and Future Directions. *ISPRS J. Photogramm. Remote Sens.* **2016**, *114*, 24–31. [\[CrossRef\]](#)
24. Maxwell, A.E.; Warner, T.A.; Fang, F. Implementation of Machine-Learning Classification in Remote Sensing: An Applied Review. *Int. J. Remote Sens.* **2018**, *39*, 2784–2817. [\[CrossRef\]](#)
25. Li, Q.; Tian, J.; Tian, Q. Deep Learning Application for Crop Classification via Multi-Temporal Remote Sensing Images. *Agriculture* **2023**, *13*, 906. [\[CrossRef\]](#)
26. Song, J.; Gao, S.; Zhu, Y.; Ma, C. A Survey of Remote Sensing Image Classification Based on CNNs. *Big Earth Data* **2019**, *3*, 232–254. [\[CrossRef\]](#)
27. Cheng, K.; Wang, J. Forest-Type Classification Using Time-Weighted Dynamic Time Warping Analysis in Mountain Areas: A Case Study in Southern China. *Forests* **2019**, *10*, 1040. [\[CrossRef\]](#)

28. Saha, A.; Sastry, S.; Dave, V.A.; Ghosh, R. Evaluation of Tree Species Classification Methods Using Multi-Temporal Satellite Images. In *Proceedings of the 2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS)*; IEEE: Piscataway, NJ, USA, 2020; pp. 40–43. [\[CrossRef\]](#)
29. Wang, X.; Hou, M.; Shi, S.; Hu, Z.; Yin, C.; Xu, L. Winter Wheat Extraction Using Time-Series Sentinel-2 Data Based on Enhanced TWDTW in Henan Province, China. *Sustainability* **2023**, *15*, 1490. [\[CrossRef\]](#)
30. Xiao, X.; Jiang, L.; Liu, Y.; Ren, G. Limited-Samples-Based Crop Classification Using a Time-Weighted Dynamic Time Warping Method, Sentinel-1 Imagery, and Google Earth Engine. *Remote Sens.* **2023**, *15*, 1112. [\[CrossRef\]](#)
31. Yu, Y.; Zhang, S.; Bai, Y.; Sun, Y.; Ge, B. Mapping the Planting Area of Winter Wheat at 10-m Resolution Using Sentinel-2 Data and Multimodel Fusion Method. *IEEE Access* **2024**, *12*, 149115–149129. [\[CrossRef\]](#)
32. Zhang, H.K.; Roy, D.P.; Luo, D. Demonstration of Large Area Land Cover Classification with a One Dimensional Convolutional Neural Network Applied to Single Pixel Temporal Metric Percentiles. *Remote Sens. Environ.* **2023**, *295*, 113653. [\[CrossRef\]](#)
33. Bai, M.; Peng, P.; Zhang, S.; Wang, X.; Wang, X.; Wang, J.; Pellikka, P. Mountain Forest Type Classification Based on One-Dimensional Convolutional Neural Network. *Forests* **2023**, *14*, 1823. [\[CrossRef\]](#)
34. Vicuña, S.; McPhee, J.; Garreaud, R.D. Agriculture Vulnerability to Climate Change in a Snowmelt-Driven Basin in Semiarid Chile. *J. Water Resour. Plan. Manag.* **2012**, *138*, 431–441. [\[CrossRef\]](#)
35. Montecinos, S.; Gutiérrez, J.R.; López-Cortés, F.; López, D. Climatic Characteristics of the Semi-Arid Coquimbo Region in Chile. *J. Arid Environ.* **2016**, *126*, 7–11. [\[CrossRef\]](#)
36. Castro, L.; Gironás, J. Precipitation, Temperature and Evaporation. In *Water Resources of Chile*; Fernández, B., Gironás, J., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 31–60. [\[CrossRef\]](#)
37. Falvey, M.; Garreaud, R. Wintertime Precipitation Episodes in Central Chile: Associated Meteorological Conditions and Orographic Influences. *J. Hydrometeorol.* **2007**, *8*, 171–193. [\[CrossRef\]](#)
38. Espinoza, V.; Lobos-Roco, F.; del Río, C. Synoptic Control of the Spatiotemporal Variability of Fog and Low Clouds under ENSO Phenomena along the Chilean Coast (17°–36° S). *Atmos. Res.* **2024**, *308*, 107533. [\[CrossRef\]](#)
39. Verbist, K.M.J.; Maureira-Cortés, H.; Rojas, P.; Vicuña, S. A Stress Test for Climate Change Impacts on Water Security: A CRIDA Case Study. *Clim. Risk Manag.* **2020**, *28*, 100222. [\[CrossRef\]](#)
40. Del Pozo, A.; Ovalle, C.; Aronson, J.; Avendaño, J. Ecotypic Differentiation in *Medicago Polymorpha* L. along an Environmental Gradient in Central Chile. I. Phenology, Biomass Production and Reproductive Patterns. *Plant Ecol.* **2002**, *159*, 119–130. [\[CrossRef\]](#)
41. Lara, C.; Saldías, G.S.; Cazelles, B.; Rivadeneira, M.M.; Muñoz, R.; Galán, A.; Paredes, A.L.; Fierro, P.; Broitman, B.R. Climatic Regulation of Vegetation Phenology in Protected Areas along Western South America. *Remote Sens.* **2021**, *13*, 2590. [\[CrossRef\]](#)
42. CIREN, Centro de Información de Recursos Naturales. *Catastro Frutícola. Región de Coquimbo. Actualización 2021*; Informe técnico; Centro de Información de Recursos Naturales: Santiago, Chile, 2021.
43. Roy, D.P.; Li, Z.; Zhang, H.K. Adjustment of Sentinel-2 Multi-Spectral Instrument (MSI) Red-Edge Band Reflectance to Nadir BRDF Adjusted Reflectance (NBAR) and Quantification of Red-Edge Band BRDF Effects. *Remote Sens.* **2017**, *9*, 1325. [\[CrossRef\]](#)
44. Strub, P.T.; James, C.; Montecino, V.; Rutllant, J.A.; Blanco, J.L. Ocean Circulation along the Southern Chile Transition Region (38°–46°S): Mean, Seasonal and Interannual Variability, with a Focus on 2014–2016. *Prog. Oceanogr.* **2019**, *172*, 159–198. [\[CrossRef\]](#) [\[PubMed\]](#)
45. Skakun, S.; Wevers, J.; Brockmann, C.; Doxani, G.; Aleksandrov, M.; Batič, M.; Frantz, D.; Gascon, F.; Gómez-Chova, L.; Hagolle, O.; et al. Cloud Mask Intercomparison eXercise (CMIX): An Evaluation of Cloud Masking Algorithms for Landsat 8 and Sentinel-2. *Remote Sens. Environ.* **2022**, *274*, 112990. [\[CrossRef\]](#)
46. Tarrio, K.; Tang, X.; Masek, J.G.; Claverie, M.; Ju, J.; Qiu, S.; Zhu, Z.; Woodcock, C.E. Comparison of Cloud Detection Algorithms for Sentinel-2 Imagery. *Sci. Remote Sens.* **2020**, *2*, 100010. [\[CrossRef\]](#)
47. Shen, R.; Pan, B.; Peng, Q.; Dong, J.; Chen, X.; Zhang, X.; Ye, T.; Huang, J.; Yuan, W. High-Resolution Distribution Maps of Single-Season Rice in China from 2017 to 2022. *Earth Syst. Sci. Data* **2023**, *15*, 3203–3222. [\[CrossRef\]](#)
48. Kate, R.J. Using Dynamic Time Warping Distances as Features for Improved Time Series Classification. *Data Min. Knowl. Discov.* **2015**, *30*, 283–312. [\[CrossRef\]](#)
49. Rafif, R.; Kusuma, S.S.; Saringatin, S.; Nanda, G.I.; Wicaksono, P.; Arjasakusuma, S. Crop Intensity Mapping Using Dynamic Time Warping and Machine Learning from Multi-Temporal PlanetScope Data. *Land* **2021**, *10*, 1384. [\[CrossRef\]](#)
50. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
51. Congalton, R. A Review of Assessing the Accuracy of Classifications of Remotely Sensed Data. *Remote Sens. Environ.* **1991**, *37*, 35–46. [\[CrossRef\]](#)
52. Maxwell, A.E. Comparing the Classification of Subjects by Two Independent Judges. *Br. J. Psychiatry* **1970**, *116*, 651–655. [\[CrossRef\]](#) [\[PubMed\]](#)
53. Hutter, F.; Hoos, H.; Leyton-Brown, K. An Efficient Approach for Assessing Hyperparameter Importance. In *Proceedings of the 31st International Conference on Machine Learning*; PMLR: New York, NY, USA, 2014; pp. 754–762.

54. Ghazaryan, G.; Dubovyk, O.; Löw, F.; Lavreniuk, M.; Kolotii, A.; Schellberg, J.; Kussul, N. A Rule-Based Approach for Crop Identification Using Multi-Temporal and Multi-Sensor Phenological Metrics. *Eur. J. Remote Sens.* **2018**, *51*, 511–524. [\[CrossRef\]](#)
55. Afrasiabi, M.; khotanlou, H.; Mansoorizadeh, M. DTW-CNN: Time Series-Based Human Interaction Prediction in Videos Using CNN-extracted Features. *Vis. Comput.* **2019**, *36*, 1127–1139. [\[CrossRef\]](#)
56. Iwana, B.K.; Frinken, V.; Uchida, S. DTW-NN: A Novel Neural Network for Time Series Recognition Using Dynamic Alignment between Inputs and Weights. *Knowl.-Based Syst.* **2020**, *188*, 104971. [\[CrossRef\]](#)
57. Boulila, W.; Ghandorh, H.; Khan, M.A.; Ahmed, F.; Ahmad, J. A Novel CNN-LSTM-based Approach to Predict Urban Expansion. *Ecol. Inform.* **2021**, *64*, 101325. [\[CrossRef\]](#)
58. Trelinski, J.; Kwolek, B. CNN-based and DTW Features for Human Activity Recognition on Depth Maps. *Neural Comput. Appl.* **2021**, *33*, 14551–14563. [\[CrossRef\]](#)
59. Fang, S.; Liu, Y.; Li, H.; Liu, X.; Hu, Y.; Xu, A. Crop Type Identification Using Spatio-Temporal Fusion of Multi-Source Remote Sensing Data Based on Time-Weighted Dynamic Time Warping. *J. Appl. Remote Sens.* **2024**, *18*, 044501. [\[CrossRef\]](#)
60. Khosravi, I. Towards Sustainable Agriculture in Iran Using a Machine Learning-Driven Crop Mapping Framework. *Eur. J. Remote Sens.* **2025**, *58*, 2490787. [\[CrossRef\]](#)
61. Finnegan, R.; Metcalfe, J.; Sharifzadeh, S.; Caraffini, F.; Xie, X.; Hornero, A.; Synes, N.W. Temporal Optimisation of Satellite Image-Based Crop Mapping: A Comparison of Deep Time Series and Semi-Supervised Time Warping Strategies. *IET Comput. Vis.* **2025**, *19*, e70014. [\[CrossRef\]](#)
62. Sidike, P.; Asari, V.K.; Sagan, V. Progressively Expanded Neural Network (PEN Net) for Hyperspectral Image Classification: A New Neural Network Paradigm for Remote Sensing Image Analysis. *ISPRS J. Photogramm. Remote Sens.* **2018**, *146*, 161–181. [\[CrossRef\]](#)
63. Ma, N.; Sun, L.; Zhou, C.; He, Y. Cloud Detection Algorithm for Multi-Satellite Remote Sensing Imagery Based on a Spectral Library and 1D Convolutional Neural Network. *Remote Sens.* **2021**, *13*, 3319. [\[CrossRef\]](#)
64. Cheng, X.; Sun, Y.; Zhang, W.; Wang, Y.; Cao, X.; Wang, Y. Application of Deep Learning in Multitemporal Remote Sensing Image Classification. *Remote Sens.* **2023**, *15*, 3859. [\[CrossRef\]](#)
65. Ivanda, A.; Šerić, L.; Žagar, D.; Oštir, K. An Application of 1D Convolution and Deep Learning to Remote Sensing Modelling of Secchi Depth in the Northern Adriatic Sea. *Big Earth Data* **2023**, *8*, 82–114. [\[CrossRef\]](#)
66. Kiranyaz, S.; Avci, O.; Abdeljaber, O.; Ince, T.; Gabbouj, M.; Inman, D.J. 1D Convolutional Neural Networks and Applications: A Survey. *Mech. Syst. Signal Process.* **2021**, *151*, 107398. [\[CrossRef\]](#)
67. Kenji Iwana, B.; Uchida, S. Time Series Classification Using Local Distance-Based Features in Multi-Modal Fusion Networks. *Pattern Recognit.* **2020**, *97*, 107024. [\[CrossRef\]](#)
68. Seydi, S.T.; Hasanlou, M. A New Structure for Binary and Multiple Hyperspectral Change Detection Based on Spectral Unmixing and Convolutional Neural Network. *Measurement* **2021**, *186*, 110137. [\[CrossRef\]](#)
69. Filho, H.C.d.C.; Júnior, O.A.d.C.; de Carvalho, O.L.F.; de Bem, P.P.; de Moura, R.d.S.; de Albuquerque, A.O.; Silva, C.R.; Ferreira, P.H.G.; Guimarães, R.F.; Gomes, R.A.T. Rice Crop Detection Using LSTM, Bi-LSTM, and Machine Learning Models from Sentinel-1 Time Series. *Remote Sens.* **2020**, *12*, 2655. [\[CrossRef\]](#)
70. Wang, H.; Zhao, X.; Zhang, X.; Wu, D.; Du, X. Long Time Series Land Cover Classification in China from 1982 to 2015 Based on Bi-LSTM Deep Learning. *Remote Sens.* **2019**, *11*, 1639. [\[CrossRef\]](#)
71. Mohanrajan, S.N.; Loganathan, A. Novel Vision Transformer-Based Bi-LSTM Model for LU/LC Prediction—Javadi Hills, India. *Appl. Sci.* **2022**, *12*, 6387. [\[CrossRef\]](#)
72. Zhong, L.; Hu, L.; Zhou, H. Deep Learning Based Multi-Temporal Crop Classification. *Remote Sens. Environ.* **2019**, *221*, 430–443. [\[CrossRef\]](#)
73. Rußwurm, M.; Körner, M. Self-Attention for Raw Optical Satellite Time Series Classification. *ISPRS J. Photogramm. Remote Sens.* **2020**, *169*, 421–435. [\[CrossRef\]](#)
74. Escudero-Arnanz, O.; Marques, A.G.; Soguero-Ruiz, C.; Mora-Jiménez, I.; Robles, G. dtwParallel: A Python Package to Efficiently Compute Dynamic Time Warping between Time Series. *SoftwareX* **2023**, *22*, 101364. [\[CrossRef\]](#)
75. de Oliveira, S.S.T.; do S. Rodrigues, V.J.; Ferreira, L.G.; Martins, W.S. P-TWDTW: Parallel Processing of Time Series Remote Sensing Images Using Manycore Architectures. In *Proceedings of the 2018 Symposium on High Performance Computing Systems (WSCAD)*; IEEE: Piscataway, NJ, USA, 2018; pp. 252–258. [\[CrossRef\]](#)
76. Singh, R.K.; Rizvi, J.; Behera, M.D.; Biradar, C. Automated Crop Type Mapping Using Time-Weighted Dynamic Time Warping-A Basis to Derive Inputs for Enhanced Food and Nutritional Security. *Curr. Res. Environ. Sustain.* **2021**, *3*, 100032. [\[CrossRef\]](#)
77. Petitjean, F.; Inglada, J.; Gancarski, P. Satellite Image Time Series Analysis Under Time Warping. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 3081–3095. [\[CrossRef\]](#)
78. Sakoe, H.; Chiba, S. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Trans. Acoust. Speech Signal Process.* **1978**, *26*, 43–49. [\[CrossRef\]](#)

79. Kim, S.W.; Park, S.; Chu, W.W. An Index-Based Approach for Similarity Search Supporting Time Warping in Large Sequence Databases. In *Proceedings of the 17th International Conference on Data Engineering*; IEEE: Piscataway, NJ, USA, 2001; pp. 607–614.
80. Keogh, E.; Ratanamahatana, C.A. Exact Indexing of Dynamic Time Warping. *Knowl. Inf. Syst.* **2005**, *7*, 358–386. [[CrossRef](#)]
81. Zhang, Z.; Tang, P.; Hu, C.; Liu, Z.; Zhang, W.; Tang, L. Seeded Classification of Satellite Image Time Series with Lower-Bounded Dynamic Time Warping. *Remote Sens.* **2022**, *14*, 2778. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.