**Supplementary Materials**

**Supporting Information to**

**Top 10+1 Indicators for Assessing Forest Ecosystem Conditions: A Five-Decade Fragmentation Analysis**

Table S1. Description of models’ parameters in each simulation (SIMLANDER software version 2.0.0). To select the best models, we tested different initial year maps for each prediction. For instance, to predict 2006, we tested as reference maps 1990 and 2000, to predict 2012 it was 1990, 2000, and 2006, and for predicting 2018, we tested 1990, 2000, 2006 and 2012 as initial year maps. We compared the models’ performance and quality of results, and the reference map from 1990 showed the best results only when predicting the year 2000. These results can be related to the technical characteristics of the Corine Land use land cover datasets such as the satellite data and its spatial resolution, geometric and thematic accuracy, the temporal extent, and production time. For example, comparing these characteristics for 1990 and 2000 maps, to create the oldest, it was used Landsat 5 with a spatial resolution of 50 m, against Landsat 7 with 25 m; the temporal extent was (1986-1998) and production time was set to 10 years, while for 2000 the production time was 4 years, and temporal extent was 2000 +/-1 year.

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario (year) | Initial year map | Final year map | Demand |
| 2000 | 1990 | 2000 | 1 440 |
| 2006 | 2000 | 2006 | 1 964 |
| 2012 | 2000 | 2012 | 4 261 |
| 2018 | 2000 | 2018 | 5 312 |
| 2036 | 2000 | - | 6 723 |
| 2054 | 2000 | - | 8 114 |

Table S2. Python scripts were developed to calculate 1) the summary statistics of the dataset, 2) to test different variance thresholds, and 3) to conduct the Principal Component Analysis.

|  |  |
| --- | --- |
|  | Script to run in Windows command line - python |
| 1. Summary statistics
 | import pandas as pd# Load the dataset from a CSV fileinput\_csv\_path = 'data.csv' # Replace with your file pathdf = pd.read\_csv(input\_csv\_path)# Step 1: Calculate summary statisticssummary\_stats = df.describe(include='all') # Includes basic stats for both numerical and categorical data# Step 2: Additional metricssummary\_stats.loc['median'] = df.median() # Adding the median to the statisticssummary\_stats.loc['skew'] = df.skew() # Adding skewness (asymmetry of the data)summary\_stats.loc['kurtosis'] = df.kurt() # Adding kurtosis (tailedness of the data)summary\_stats.loc['range'] = df.max() - df.min() # Adding the range (max - min)summary\_stats.loc['variance'] = df.var() # Adding variance# Step 3: Save the summary statistics to a CSV fileoutput\_csv\_path = 'summary\_statistics.csv'summary\_stats.to\_csv(output\_csv\_path)print(f"Summary statistics saved to {output\_csv\_path}")# Print the summary statistics for reviewprint(summary\_stats) |
| 1. Test variance threshold
 | import numpy as npimport pandas as pdfrom sklearn.feature\_selection import VarianceThreshold# Load the dataset from a CSV fileinput\_csv\_path = 'data.csv' # Replace this with your input file pathdf = pd.read\_csv(input\_csv\_path)# Convert all data to numeric (skip errors by using 'coerce' if any non-numeric values exist)df = df.apply(pd.to\_numeric, errors='coerce')# Extract the feature names (header) and the actual data (without the header)feature\_names = df.columnsoriginal\_data = df.values# Define the thresholds you want to testthresholds = [0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] for threshold in thresholds: print(f"\nApplying VarianceThreshold with threshold = {threshold}...") var\_threshold = VarianceThreshold(threshold=threshold)  try: # Fit and transform the data with the current threshold data\_after\_threshold = var\_threshold.fit\_transform(original\_data)  # Get the indices of the selected features selected\_features = var\_threshold.get\_support(indices=True)  if data\_after\_threshold.shape[1] > 0: # Ensure there are selected features # Get the names of the selected features selected\_feature\_names = feature\_names[selected\_features] # Create a DataFrame with the selected features reduced\_df = pd.DataFrame(data\_after\_threshold, columns=selected\_feature\_names) # Save the reduced dataset after applying VarianceThreshold output\_csv\_path\_reduced = f'reduced\_data\_threshold\_{threshold}.csv' reduced\_df.to\_csv(output\_csv\_path\_reduced, index=False)  print(f"Dataset with reduced features saved to {output\_csv\_path\_reduced}") print(f"Number of features after applying threshold {threshold}: {data\_after\_threshold.shape[1]}") else: print(f"No features selected at threshold {threshold}.") except Exception as e: print(f"Error applying VarianceThreshold with threshold {threshold}: {e}") |
| 1. Principal component analysis
 | import numpy as npimport pandas as pdimport matplotlib.pyplot as pltfrom sklearn.decomposition import PCAfrom sklearn.linear\_model import Ridgefrom sklearn.preprocessing import StandardScaler# Load the dataset from a CSV fileinput\_csv\_path = 'data.csv' # Replace with your file pathdata = pd.read\_csv(input\_csv\_path)# Prepare the features (X)X = data.valuesfeature\_names = data.columns# Step 1: Standardize the data (important for PCA and Ridge)scaler = StandardScaler()X\_scaled = scaler.fit\_transform(X)# Step 2: Perform PCA with the desired number of componentsn\_components = 10pca = PCA(n\_components=n\_components)X\_pca = pca.fit\_transform(X\_scaled)# Explained variance of each componentexplained\_variance\_ratio = pca.explained\_variance\_ratio\_# Step 3: Use Ridge regression to rank features' importance in predicting PC1ridge = Ridge(alpha=1.0)ridge.fit(X\_scaled, X\_pca[:, 0]) # Fit Ridge regression on scaled features with PC1 as the target# Save feature importances (Ridge coefficients) to CSVfeature\_importances = np.abs(ridge.coef\_)importance\_df = pd.DataFrame({'Feature': feature\_names, 'Importance': feature\_importances})importance\_df = importance\_df.sort\_values(by='Importance', ascending=False)importance\_df.to\_csv('feature\_importancesN.csv', index=False)print("Feature importances saved to 'feature\_importances1.csv'.")# Step 4: Feature contributions (loadings) for the first three principal componentsloadings = pd.DataFrame(pca.components\_[:3].T, columns=['PC1', 'PC2', 'PC3'], index=feature\_names)loadings.to\_csv('loadings\_PC1\_PC2\_PC3N.csv', index=True)print("Feature contributions (loadings) for the first three components saved to 'loadings\_PC1\_PC2\_PC3.csv'.")# Step 5: Projection of variables (correlations between initial variables and PCs)# PC1 vs PC2plt.figure(figsize=(10, 8))for i in range(len(feature\_names)): plt.arrow(0, 0, pca.components\_[0, i], pca.components\_[1, i], color='r', alpha=0.7, head\_width=0.02, head\_length=0.03) plt.text(pca.components\_[0, i] \* 1.25, pca.components\_[1, i] \* 1.25, feature\_names[i], color='g', ha='center', va='center', fontsize=20)plt.xlim([-1, 1])plt.ylim([-1, 1])plt.xlabel('PC1', fontsize=20)plt.ylabel('PC2', fontsize=20)plt.title('Projection of Variables on PC1 and PC2', fontsize=20)plt.grid(True)plt.tight\_layout()plt.savefig('projection\_variables\_PC1\_PC2\_highres.png', dpi=1300)plt.close()# PC1 vs PC3plt.figure(figsize=(10, 8))for i in range(len(feature\_names)): plt.arrow(0, 0, pca.components\_[0, i], pca.components\_[2, i], color='r', alpha=0.7, head\_width=0.02, head\_length=0.03) plt.text(pca.components\_[0, i] \* 1.25, pca.components\_[2, i] \* 1.25, feature\_names[i], color='g', ha='center', va='center', fontsize=20)plt.xlim([-1, 1])plt.ylim([-1, 1])plt.xlabel('PC1', fontsize=20)plt.ylabel('PC3', fontsize=20)plt.title('Projection of Variables on PC1 and PC3', fontsize=20)plt.grid(True)plt.tight\_layout()plt.savefig('projection\_variables\_PC1\_PC3\_highres.png', dpi=1300)plt.close()# PC2 vs PC3plt.figure(figsize=(10, 8))for i in range(len(feature\_names)): plt.arrow(0, 0, pca.components\_[1, i], pca.components\_[2, i], color='r', alpha=0.7, head\_width=0.02, head\_length=0.03) plt.text(pca.components\_[1, i] \* 1.25, pca.components\_[2, i] \* 1.25, feature\_names[i], color='g', ha='center', va='center', fontsize=20)plt.xlim([-1, 1])plt.ylim([-1, 1])plt.xlabel('PC2', fontsize=20)plt.ylabel('PC3', fontsize=20)plt.title('Projection of Variables on PC2 and PC3', fontsize=20)plt.grid(True)plt.tight\_layout()plt.savefig('projection\_variables\_PC2\_PC3\_highres.png', dpi=1300)plt.close()# # Step 6: Plot Variance of Principal Components# plt.figure(figsize=(8, 6))# plt.bar(range(1, n\_components + 1), explained\_variance\_ratio, alpha=0.7)# plt.xlabel('Principal Components', fontsize=14)# plt.ylabel('Variance Ratio', fontsize=14)# plt.title('Variance of Principal Components', fontsize=16)# plt.tight\_layout()# plt.savefig('variance\_pcs1\_highres.png', dpi=300)# plt.close()# # Print Explained Variance for first three components# print(f"Explained Variance (first 3 components): {explained\_variance\_ratio[:3]}") |

Table S3. The 27 LM characterizing forest configuration is grouped by type. From the aggregation type, there were three metrics (division, mesh, and split). From the area and edge type, there were five metrics (area, gyrate, pland - % of landscape of class, lpi - largest patch index, and ca - total class area). Metrics characterizing the core area (cai - core area index, dcore - disjunct core areas, core - core area, cpland - % of core area in a landscape, and tca – total core area), diversity (sidi - Simpson’s diversity index, msidi - modified Simpson’s diversity index, siei - Simpson’s evenness index, and msiei - modified Simpson’s evenness index), the mean contiguity index (contig\_mn) from the shape metric type, and the mutinf (mutual Information) metric grouped in the complexity type. At the landscape level, 10 metrics showed negative variation above the threshold of 50% (dcore\_mn, mutinf, sidi, siei, gyrate\_mn, contig\_mn, division, msiei, msidi,cai\_mn). At the class level, 16 metrics were identified (split, contig\_mn, gyrate\_mn, dcore\_sd, gyrate\_sd, ca, pland, cpland, tca, cai\_mn, area\_mn, core\_mn, core\_sd, area\_sd, lpi, mesh), and one metric at patch level (cai).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type | Level | Abbrev. | Name | Unit | References |
| Aggregation metric | Landscape | division | Landscape division index | proportion | (Jaeger, 2000)(McGarigal et al., 2012) |
| Aggregation metric | Class | mesh | Effective Mesh Size | ha | (Jaeger, 2000)(McGarigal et al., 2012) |
| Aggregation metric | Class | split | Splitting index | none | (Jaeger, 2000)(McGarigal et al., 2012) |
| Area and edge metric | Class | area\_mn, area\_sd | Mean of patch area, Standard deviation of patch area | ha | (McGarigal K et al., 2012) |
| Area and edge metric | Landscape; Class | gyrate\_mn; gyrate\_sd | Mean radius of gyration; Standard deviation radius of gyration | m | (Keitt et al., 1997)(McGarigal K et al., 2012) |
| Area and Edge metric | Class | pland | Percentage of landscape of class | percentage | (McGarigal K et al., 2012) |
| Area and Edge metric | Class | lpi | Largest patch index | percentage | (McGarigal K et al., 2012) |
| Area and edge metric | Class | ca | Total class area | ha | (McGarigal et al., 2012) |
| Core area metric | Landscape; Class; | cai\_mn; cai | Mean of core area index; Core area index | m | (Keitt et al., 1997)(McGarigal K et al., 2012) |
| Core area metric | Landscape; Class | dcore\_mn | Mean number of disjunct core areas | none | (McGarigal K et al., 2012) |
| Core area metric | Class | cpland | Core area percentage of landscape | percentage | (McGarigal K et al., 2012) |
| Core area metric | Class | core\_mn, core\_sd, tca | Mean of core area, Standard deviation of patch core area, Total core area | ha | (McGarigal et al., 2012) |
| Shape metric | Landscape; Class | contig\_mn | Mean of Contiguity index | none | (Lagro, 1991)(McGarigal et al., 2012) |
| Diversity metric | Landscape | sidi, siei, msidi, msiei | Simpson’s diversity index, Simpson’s evenness index, Modified Simpson’s diversity index, Modified Simpson’s evenness index | none | (Simpson, 1949) (May, 1975) (Romme, 1982) (McGarigal K et al., 2012) |
| Complexity | Landscape | mutinf | Mutual information | none | (Nowosad and Stepinski, 2019b) |



Fig. S1. Projection of features in the principal components: PC1 and PC2 and, P1 and PC3. The selected 27 LM were analysed through PCA by 10 components. The explained variance percentages for PC1, PC2 and PC3 are respectively 62.5%, 18.4% and 16.2%. The top 10 loadings for PC1, are the features that contribute the most to this principal component and capture the largest variance in the dataset. These 10 most important features were selected as LM-based indicators of forest condition (Table S4).

Table S4. Assessment of forest ecosystem conditions for Mainland Portugal. The top 10 LM-based indicators were calculated for observed and predicted maps. Legend: obs – observed, pred – predicted. Future projections refer to the calculations that were predicted for 2036 and 2054.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Metric | 2000 | 2006 | 2012 | 2018 | 2036 | 2054 | Diagram | Future projections |
| mesh obs | 34 658.3 | 13 465.4 | 13 069.8 | 4 501.7 |   |   |  |  |
| mesh pred | 34 334.8 | 22 585.4 | 26 281.8 | 9 781.6 | 4 514.1 | 1 852.1 |  | \* |
| core\_mn obs | 312.1 | 211.8 | 200.1 | 159.6 |   |   |  |  |
| core\_mn pred | 262.2 | 243.2 | 353.2 | 253.8 | 279 | 208 |  | \* |
| gyrate\_mn obs | 587.9 | 540.2 | 544.6 | 514.6 |   |   |  |  |
| gyrate\_mn pred | 425.8 | 421 | 493 | 476.2 | 516.4 | 484 |  | \* |
| split obs | 257.5 | 662.8 | 682.8 | 1 982.4 |   |   |  |  |
| split pred | 259.9 | 395.1 | 339.6 | 912.3 | 1 977 | 4 818.5 |  | \* |
| cpland obs | 18.9 | 15 | 14 | 11.9 |   |   |  |  |
| cpland pred | 19.5 | 17.3 | 18 | 13.2 | 11.8 | 8.9 |  | \* |
| lpi obs | 5.7 | 3 | 3.2 | 1.4 |   |   |  |  |
| lpi pred | 5.6 | 4.6 | 5.0 | 3.0 | 1.5 | 0.9 |  | \* |
| siei obs | 0.8 | 0.7 | 0.7 | 0.6 |   |   |  |  |
| siei pred | 0.8 | 0.7 | 0.7 | 0.6 | 0.5 | 0.4 |  | \* |
| contig\_mn obs | 0.7 | 0.7 | 0.7 | 0.7 |   |   |  |  |
| contig\_mn pred | 0.6 | 0.6 | 0.7 | 0.7 | 0.7 | 0.7 |  | \* |
| mutinf obs | 0.5 | 0.5 | 0.5 | 0.4 |   |   |  |  |
| mutinf pred | 0.5 | 0.5 | 0.6 | 0.5 | 0.5 | 0.4 |  | \* |
| sidi obs | 0.4 | 0.3 | 0.3 | 0.3 |   |   |  |  |
| sidi pred | 0.4 | 0.4 | 0.4 | 0.3 | 0.3 | 0.2 |  | \* |