SCIENTIFIC Publishing Limited

Land Management and Utilization

https://ojs.ukscip.com/index.php/lmu

Article

Machine Learning Innovations in LULC Classification: A Comparative Study of SVM, Random Forest, and Decision Trees

Berna Çalışkan

Civil Engineering Department, Istanbul Technical University, Istanbul 34467, Türkiye; caliskanber18@itu.edu.tr

Received: 27 August 2025; **Revised:** 14 November 2025; **Accepted:** 28 December 2025; **Published:** 26 March 2026

Abstract: Classifying land use and land cover (LULC) is a fundamental process in remote sensing and geographical information systems (GIS) that is essential to many applications, including disaster assessment, urban planning, environmental monitoring, and natural resource management. Understanding the dynamics of landscapes and how they evolve over time requires accurate classification of land use and land cover groups. For this reason, straightforward classification methods like decision trees, artificial neural networks and maximum likelihood have historically been employed extensively. However, there has been an increasing interest in investigating machine learning techniques' potential to enhance the precision and effectiveness of LULC classification since their introduction. Computer vision, natural language processing, and remote sensing are just a few of the fields that have greatly benefited from the quick development of machine learning algorithms, especially deep learning approaches. Due to its capacity to automatically extract intricate patterns and features from massive datasets, machine learning-based techniques have become more and more popular in LULC classification jobs in recent years, potentially surpassing conventional approaches. This research paper aims to conduct a Land use classification by using machine learning based (ML) models (Support Vector Machine (SVM) model, Random Forest (RF) and Decision Trees (DT) models) with the use of open-sourced Python modules (Rasterio, Numpy, and Scikit-learn). The comparative analysis demonstrates that the SVM model achieved the highest performance with an Overall Accuracy (OA) of 97.30%, followed by Random Forest at 94.59%, and Decision Tree at 89.19%.

Keywords: LULC Classification; Machine Learning Algorithms; Visual Interpretation

1. Introduction

Understanding vegetation dynamics, climate change, socioeconomic issues, and landscape patterns—all of which are crucial for forecasting global weather events—requires an understanding of land use and land cover (LULC). Land cover is an essential resource that has a big impact on social, economic, and environmental well-being as well as climatic systems and sustainability. As a result, keeping an eye on LULC changes, their causes, and their effects is essential for managing natural resources, advancing sustainable development, and reducing the effects of climate change. Many social, environmental, and policy-related factors influence LULC changes in emerging nations. The need for proactive responses to the environmental issues provided by fast urbanization and land use changes in developing regions is highlighted by the need of understanding these changes for well-informed decision-making in land management and conservation initiatives [1].

Three distinct classification models—Support Vector Machine (SVM), Random Forest Machine (RFM), and Maximum Likelihood (ML)—were statistically compared by Rawat et al. (2024) [2] for the classification of Sentinel-2A and Landsat data sets. Land use and land cover (LULC) and Machine Learning techniques were investigated in the coastal Pays de Brest by Xie (2023) [3]. Hermosilla et al. (2022) [4] introduced a methodological framework that allows the creation of annual land cover classification maps from annual time-series of Landsat image composites and ancillary topographic and hydrologic data, thereby optimizing localized implementation and training data selection to improve mapping outcomes. In addition to this, Amin et al. (2024) [5] examined machine learning algorithms for classifying land cover in a complicated hilly environment. Taati et al. (2014) [6] used TM pictures from the Landsat 5 satellite to generate a land use categorization using a support vector machine (SVM) and maximum likelihood classifier (MLC) in Qazvin, Iran, as an example of employing the Support Vector Machine Model.

Precise land use and land cover maps must be created in order to properly develop urban infrastructures like pipelines, road networks, and other linear engineering constructions. There have been numerous attempts to create land use and land cover classification methods utilizing a variety of techniques, from surveying to picture interpretation employing remote sensing techniques.

The distinctive contributions of the proposed work are summarized as follows:

Methodological Innovation: Support Vector Machine (SVM) model, Random Forest (RF) and Decision Trees (DT) models are used to conduct a Land use classification.

Data Fusion and Synergy: Leveraging the synergy between multi-spectral satellite imagery to improve the detection of linear features and terrain characteristics.

Enhanced Accuracy for Linear Engineering: Development of a high-resolution mapping approach specifically optimized for the planning of urban transport networks.

Computational Efficiency: A streamlined automated processing by using Python programming language.

Robustness in Urban Complexity: Implementation of advanced algorithms to effectively distinguish between spectrally similar urban surfaces, land cover, types, etc.

1.1. Key Elements of a Machine Learning (ML) System

A strong machine learning system involves much more than a standalone algorithm, it integrates data, features, models, and optimization techniques within a solid infrastructure for training, evaluation, and deployment. These components collectively drive the system's ability to learn from data and make accurate predictions. Each part plays a vital role in ensuring the system's performance, reliability, and practical use. Therefore, understanding these elements is crucial for effectively designing, building, and refining ML models. The key elements are: Data, Features, Model (Algorithm), Training, Evaluation, Loss Function, Optimization Algorithm, Inference, Feedback Loop (Optional) and Infrastructure and Deployment [7].

a) Data

Data provides the backbone of every machine learning system and can come in structured formats like spreadsheets, semi-structured formats like XML files, or unstructured formats such as emails, audio recordings, or medical photographs. The quality and applicability of these data have a significant impact on the model's accuracy. For instance, a facial recognition model trains on image datasets, whereas a spam detection system learns from labeled emails. To properly create and evaluate the system, training and testing data are both necessary.

Finding patterns or relationships in a dataset is how machine learning algorithms operate. These patterns aid the model in comprehending the logic or "rules" behind the data, which it subsequently applies to forecast future events. In image recognition, for example, a model may be trained to distinguish between cats and dogs by examining tagged photos of each. Its ability to make precise predictions improves with the number of examples.

b) Features

The quantifiable characteristics of data that are used as inputs for machine learning models are called features. Enhancing model accuracy requires effective feature engineering, such as finding pixel patterns in medical scans for disease identification or collecting hash tags from tweets for sentiment analysis. Carefully selecting and improving these features can dramatically increase a model's performance.

c) Model (Algorithm)

The mathematical representation or method used to learn from the data and generate judgments or predictions is referred to as the model. Decision trees, support vector machines, neural networks, and ensemble approaches are examples of common algorithms. The type of issue, such as regression or classification, and the features of the dataset are taken into consideration while selecting the best model.

d) Training

Giving a machine learning model data to learn the link between input attributes and output labels is known as training. By modifying the model's parameters using optimization techniques like gradient descent or back propagation, the goal is to lower prediction errors. Over time, this procedure aids in the model's accuracy improvement.

Knowledge extraction from data is the aim of machine learning. It is a subject of study in the nexus of computer science, artificial intelligence, and statistics. It is also known as statistical learning or predictive analytics. In other words, machine learning (ML) is the study of mathematical models and techniques used by computer systems to progressively improve their performance on a particular task. In order to generate predictions or conclusions without being formally trained to do so, machine learning algorithms develop a mathematical model utilizing sample data, also referred to as "training data."

e) Evaluation

After the model has been trained, its performance is assessed using a separate dataset, such as a validation or test set. The evaluation measures are determined by the work at hand. For classification tasks, metrics like accuracy, precision, recall, and F1-score are used; for regression tasks, mean squared error is employed. The total percentage of accurate forecasts is known as accuracy. The proportion of optimistic predictions that turn out to be accurate is known as precision. The percentage of true positives that were accurately detected is known as recall. The F1-Score provides a balance between precision and recall by taking the harmonic mean of the two.

f) Loss Function

The loss function quantifies the discrepancy between the model's predictions and the actual outcomes. By demonstrating the accuracy of the model's performance, it plays a critical role in directing the learning process. This feedback is utilized to modify the model and increase its accuracy during training.

Loss functions are crucial to machine learning because they define the model's goal and direct its learning process. There are many different loss functions available, and selecting the right one is essential to training a precise model. Different regression or classification models can result from different loss function selections.

g) Optimization Algorithm

In order to lower the loss function and enhance performance, optimization methods modify the model's parameters. Sequential minimal optimization, grid search, and recursive partitioning are popular techniques. Accurate and efficient model learning requires effective optimization. For instance, another technique that calculates optimal hyperplanes for maximum geometric margin is Support Vector Optimization. Moreover, a structural refinement framework based on feature splitting, such as Gini impurity and information gain, is stored. Like ensemble voting, Random Forest also maintains an aggregated boundary optimization across multiple decision trees. Pruning, which performs well on overfitted nodes, and Hyperparameter Tuning, which performs well in complex and high-dimensional multi-spectral environments, respectively, can be combined to form a robust model. Instead of using a straightforward grid search as in basic parameter selection, advanced optimization uses the algorithm's structural constraints to scale the classification performance. It keeps the error rate during the iterative classification process dropping monotonically. These classical optimization routines use very little memory and are computationally efficient.

h) Inference

Inference is the process of applying the model to fresh, unknown data after training and validation in order to make predictions or judgments. This signifies the machine learning system's practical implementation. Using a trained model to generate predictions or choices based on fresh, unobserved data is known as inference in machine learning. Inference is the actual use of the model in real-world situations, whereas training entails identifying patterns in a dataset.

i) Feedback Loop (Optional)

Certain machine learning systems include a feedback loop in which the model is progressively refined based on the output or user response. This is typical of adaptive systems such as online learning platforms or recommendation engines.

The environment of the model frequently provides feedback to machine learning systems, which are subsequently fed back into the system. This can be done in a variety of ways, such as by using the model's output to train more recent iterations or by utilizing user input on the model's choices to make improvements.

Certain feedback loops can actively deteriorate the machine learning system's performance over time, even while many are beneficial and will enhance your model's performance over time. Creating useful feedback loops is a crucial part of designing machine learning systems, and it must be done carefully to guarantee the sustainability of your system.

j) Infrastructure and Deployment

To be used in the real world, machine learning models must be implemented in an appropriate computer system. This covers elements like latency, scalability, model versioning, and integration with user-facing applications or APIs. In order to bring changes into a target system environment, IT infrastructure deployment entails specifying the series of actions or steps—often referred to as a deployment plan—that must be carried out. Technology infrastructure serves an organization's business needs by supporting the enterprise architecture, facilitating communications, helping to achieve corporate goals, and more.

The basis for modeling land use has been provided by remote sensing (RS) technologies, which are frequently used to analyze LULC variations in various regions for accurate land categorization results. Additionally, it is crucial to choose a suitable method with sufficient resolution and maximum accuracy for classifying RS photos since local research requires LULC datasets with sufficient precision. Machine learning (ML) methods for LULC classification on RS images have attracted a lot of attention. Common and widely used machine learning models include radial basis function (RBF), decision tree (DT), multilayer perception (MLP), artificial neural networks (ANN), naïve Bayes (NB), maximum likelihood classifier (MLC), support vector machine (SVM), random forest (RF), and fuzzy logic. It has been noted that of the aforementioned machine learning methods, the RF and SVM models are more accurate than other classification models in various contexts [8].

1.2. Support Vector Machine (SVM) Model

SVM is a supervised machine learning method used to solve regression and classification issues. SVM classifies data by identifying the hyperplane that divides the data into the greatest number of distinct classes. To increase the model's generalizability, the hyperplane is selected to maximize the margin between the classes. Both linear and non-linear correlations between features and the target variable can be handled by SVM with the use of a technique known as a "kernel trick." Finding a hyperplane that can divide the classes is made simpler by SVM's ability to use the kernel function to convert the input data into an area with more dimensions [9].

1.3. Random Forest (RF)

The flexibility of Random Forest to handle big datasets makes it a powerful machine learning technique that is frequently employed in distant sensing applications. An ensemble classifier that integrates the predictions of several decision trees is called an RF classifier. Each instance's prediction is determined by the majority vote on the forest's trees [8].

1.4. Decision Trees (DT) Models

DTs have been used in image-based categorization because of their non-parametric character and interpretability. One of the most important steps in object-based image analysis (OBIA) for classifying land cover is the creation of decision rulesets. This stage, however, calls for class-related thresholds, which can be established by knowledge-based approaches or simple DTs. The knowledge-based approach may get complex when there are many land covers and decision factors involved [10].

1.5. Methods for Land Use and Land Cover (LULC) Modelling

Over the past ten years, there has been a lot of interest in remote sensing-based applications like LULC classification due to more advanced techniques like artificial neural networks (ANN), Support Vector Machine, Random

Forest, Decision Trees, and other models. LULC classification were done by different techniques and approaches categorized in four captions such as 'Manual Techniques', 'Numerical and Digital Techniques', 'Hybrid Techniques' and 'Other methods of classification' as shown in **Figure 1** [11]. Furthermore, some study has been conducted to identify the most accurate and suitable machine-learning classifier algorithm for LULC mapping. The accuracy levels of each machine-learning technique vary. While ANN, SVM, and RF generally offer superior accuracy when compared to the other standard classifier techniques, it has been found that SVM and RF are the best methods for the LULC classification when compared to all other machine-learning approaches. However, sensor characteristics and image data-related elements, including processing software and hardware, spatial and temporal resolution, etc., impact the accuracy of LULC classification [12].

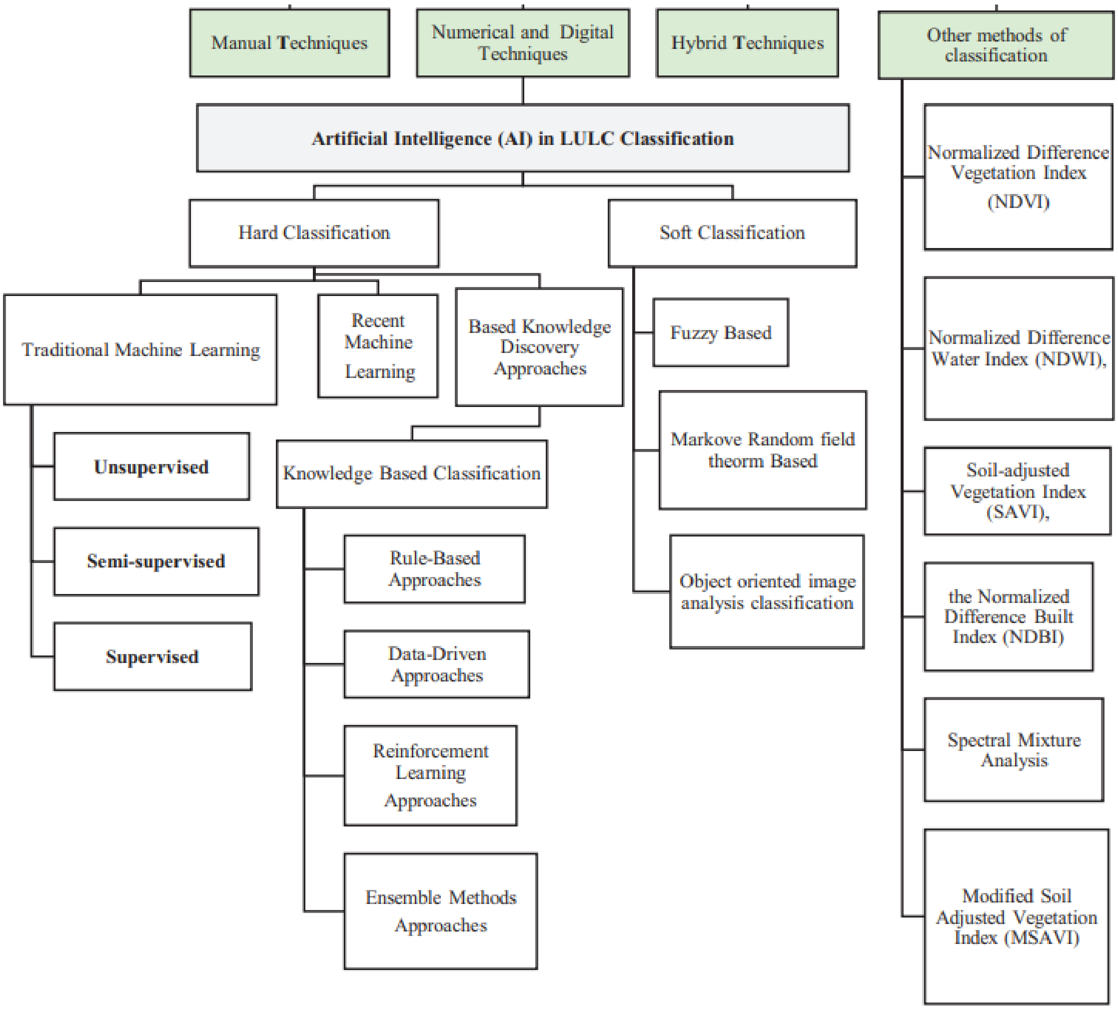

Figure 1. Categorization of methods for LULC classification and other similar tasks.

A geographic information system platform (such as ArcGIS Pro) or Google Earth Engine can be used to analyze various LULC categories utilizing machine learning techniques and remote sensing datasets. The geographic information system can be used to have a better understanding of the temporal and spatial analysis related to LULC classification. We can better understand the LULC trend of change with the use of these analyses. Google Earth Engine is a platform that speeds up and streamlines the processing of satellite images by fusing remote sensing data—that is, satellite imagery from many sources—with high-performance computer services. Google Earth Engine provides satellite images from a variety of sources, including Landsat 8, Sentinel 2, MODIS, and many more publicly accessible datasets. Python is used by Google Earth Engine to modify code, and JavaScript is used to create client libraries [13].

2. Materials and Methods

Validation and ML classifier development are the two main steps that make up the model classifier creation phase. Model selection, or selecting the best machine learning algorithm or strategy for the given problem, is the first step in the development process. The next step is model training, which utilizes the dataset to train the machine learning model. This entails employing an iterative optimization method to determine the ideal values of the model parameters and/or hyperparameters. Lastly, model evaluation involves evaluating the trained model's performance on a validation dataset to make sure it is effectively generalizing to new and untested data. The training dataset in the suggested model is subjected to a variety of machine learning (ML) classification algorithms, including SVM, random forests (RFs), and decision trees (DT). The raster mosaic image's composition and a vector layer containing pre-defined samples are the two primary inputs that the ML classifier model uses. Taking into account the results, the machine learning classifier's parameters are set and assigned values. The LULC Raster Vector (LULCRV) classification model was developed by Mahmoud et al. (2023) [14] based on machine learning (ML). It has four phases including pre-processing, labeling, model building, model classification, and prediction. The Sankey diagram illustrates the hierarchical transition from data acquisition (a) and feature extraction (b) to comparative model training (c) and final geospatial exporting (d), highlighting the taxonomic distinction between traditional and ensemble machine learning approaches (**Figure 2**).

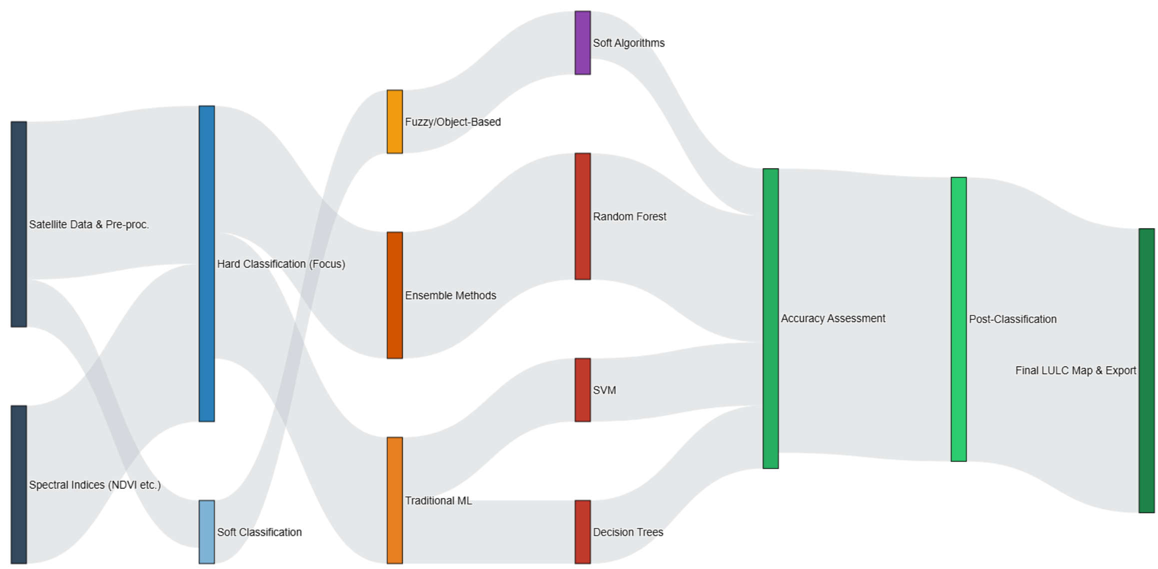

Figure 2. Sankey Diagram for Integrated Methodological Workflow for LULC Comparative Analysis (Illustrated by author).

2.1. Data Acquisition and Preprocessing

Getting the data is the first step in classifying land use. Aerial photos, satellite images, and ground surveys can all be used for this. Following acquisition, the data must be preprocessed to eliminate any noise or artifacts that can impede the classification procedure. This may entail adjusting for variations in illumination and eliminating clouds or other meteorological factors. We used an example image of Sentinel-2 over Tokyo, Japan, with bands (Blue, Green, Red, and NIR), with minimal cloud cover. The coordinates for the study area is "140.1876614° E 36.0692671° N" (**Figure 3**). We used map tile services to illustrate the study map (tiles = 'https://server.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/\{z\}/\{y\}/\{x\}'). Satellite Images can be downloaded from USGS Earth Explorer and Earth Engine Data Catalog.

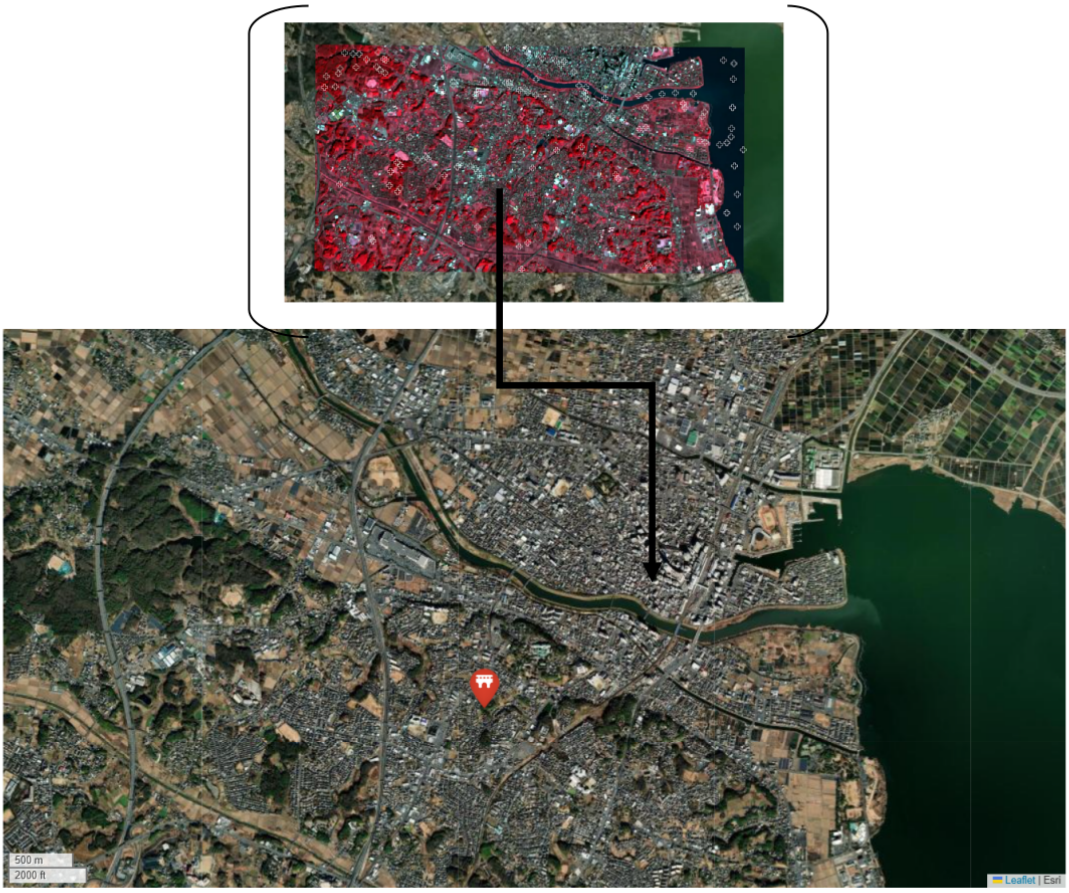

Figure 3. Study Area [15] (Illustrated by author).

2.2. Feature Extraction

We imported the required Python libraries and determined the input locations for our image and labels, and a list of input image band names. Input files can be listed as raster_loc (materials/rasters/s2image.tif), points_loc (materials/shapefiles/samples.shp) and temp_point_loc (materials/temp/temp_y_points.shp). Land cover names (lulc_name) are 'Water', 'Dense Veg', 'Veg', 'Impervious'.

Due to the complexity of urban and suburban landscapes and the limitations of remotely sensed data in terms of spectral and spatial resolutions, extracting impermeable surfaces remains a difficulty. Because urban environments are so diverse, the extractions based on picture classification are frequently understated. Other types of land cover, like trees, grasses, and soils, may be combined with the impervious surfaces. One of the main causes of incorrect land-use/cover classification is the challenge of choosing training samples for impervious surfaces. Shadows from tall buildings or big tree crowns can pose a serious challenge to efficient extraction when using high spatial resolution data, like IKONOS, for impervious surface mapping [16].

The primary method for acquiring VT/LC information is now remote sensing data because of its affordability, long-term repeatability, and extensive coverage. In arid and semi-arid locations, for example, there is less green cover and a scant vegetation canopy, which makes remote sensing observations extremely challenging. As a result, it is questionable to characterize the plant cover in these settings using conventional remote sensing techniques. Therefore, it becomes crucial to evaluate more effective VT/LC categorization methods in these domains. This realization has prompted important advances in the processing of remote sensing data, which have resulted in the development of efficient image classification and mapping systems [17].

We displayed the bands of available satellite images as well as composites of natural and false colors. The resulting bands are shown in **Figures 4** and **5** shows the visualization of individual bands as well as image composites for the input image.

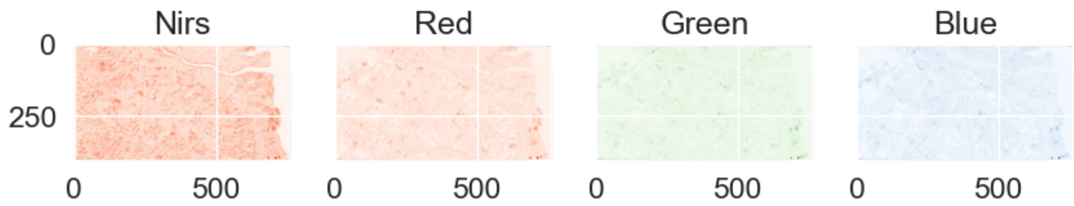

Figure 4. Individual Bands.

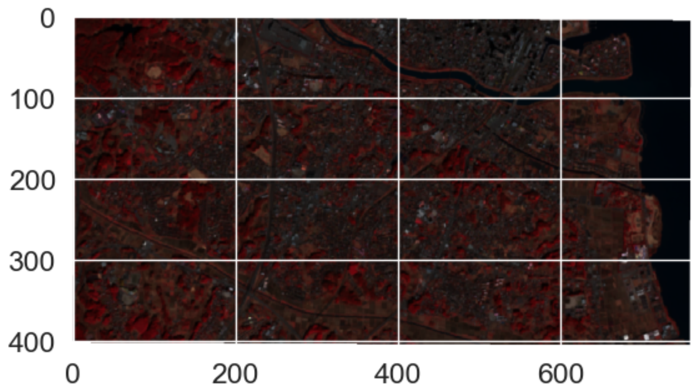

Figure 5. Visualization of individual bands along with image false color composites for the input image.

Additionally, the shapefile is designated as a geopandas dataframe, and the input data is defined for reading the satellite picture as a rasterio array. A unique ID number was assigned and saved as a new temporary file for feature extraction. Next, we made a blank Pandas series dataframe and used a Python for loop and Fiona to iterate through each label. Each label location's corresponding band value (as well as additional attributes like NDVI) is kept for every iteration. The data were transformed into a list format following iteration.

After sampling the features, we separated the data into X (independent variable—features) and Y (dependent variable—labels). The Scikit-learn train_test_split method was then used to divide the data into a 70/30 ratio, with 70% going to training and 30% to testing. Values from the sampled data are displayed in **Table 1**.

Table 1. Sampled Data Values.

Index	Band1	Band2	Band3	Band4	ID	Labels
0	0.0268	0.0404	0.0228	0.01	0	1
1	0.0284	0.043	0.0248	0.0111	1	1
2	0.0272	0.0402	0.0234	0.0107	2	1
3	0.0258	0.0408	0.0231	0.0099	3	1
4	0.0248	0.0436	0.0228	0.0107	4	1
...	...	...	...	...	...	...
116	0.1066	0.108	0.1208	0.1368	116	4
117	0.1022	0.1092	0.1192	0.1432	117	4
118	0.3632	0.3724	0.3644	0.3596	118	4
119	0.0849	0.0797	0.0773	0.0918	119	4
120	0.0818	0.0902	0.0934	0.113	120	4

Note: The dataset consists of 4 input image bands named 'band1', 'band2', 'band3', and 'band4', comprising a total of 121 rows and 6 columns. Following a 70/30 train/test split ratio, the data dimensions are partitioned as follows: X_train shape: (84, 4), X_test shape: (37, 4), y_train shape: (84,), and y_test shape: (37,).

2.3. Model Training and Accuracy Assessment

Following the preparation of the training and testing datasets, machine learning models can be readily executed using their default configurations. We began our analysis by employing a Support Vector Machine (SVM) classifier configured with a Radial Basis Function (RBF) kernel. Upon initialization, the SVM model was trained on the training features (x_train) and corresponding labels (y_train) using the standard fitting procedure.

We evaluate the sample prediction with x test data and the model accuracy for that specific sample prediction using Scikit-learn accuracy_score consecutively. (Accuracy SVM: 97.297) Scikit-learn, sometimes abbreviated as "sklearn," is a free and open-source machine learning package for Python that offers easy-to-use and effective tools for modeling and data analysis. Many data scientists and machine learning practitioners now use scikit-learn, a package built on top of NumPy, SciPy, and matplotlib [18]. For example, SVM achieved 100% accuracy by properly classifying every pixel. Subsequently, we acquired the confusion matrix and visualize it using the Seaborn "sns.heatmap" function to better examine the model's performance. The resulting confusion matrices can be seen in **Figures 6–8**. Confusion matrix visualization can be done by adding a label and percentage to a confusion matrix plotted using a Seaborn heatmap [19].

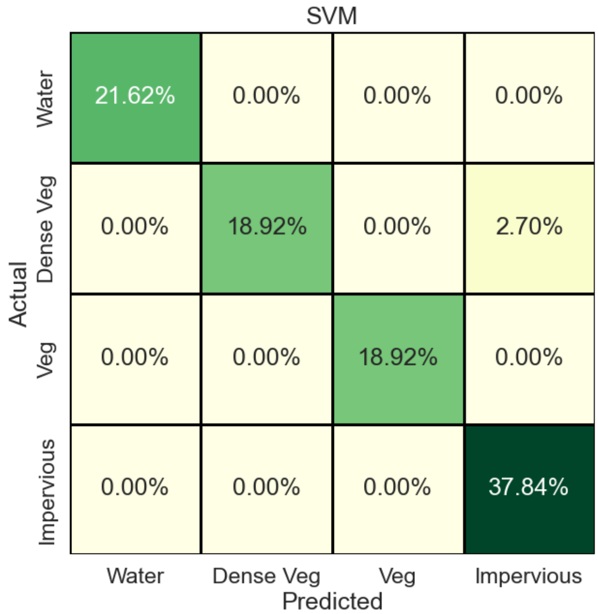

Figure 6. Confusion matrix for the Support Vector Machine (SVM) model.

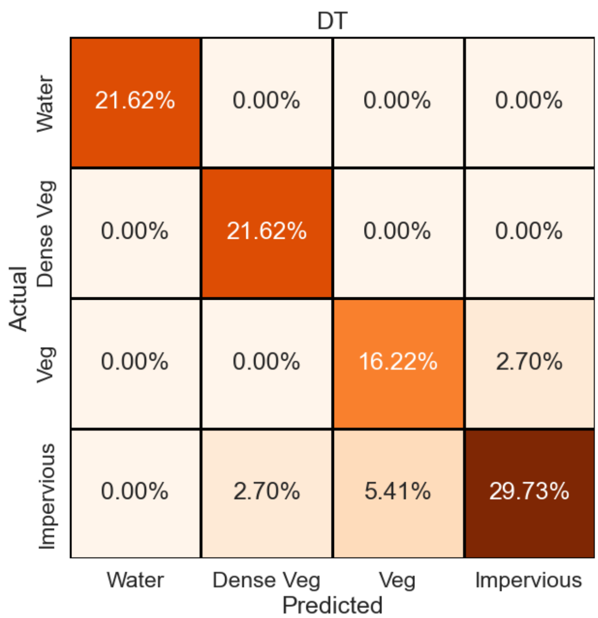

Figure 7. Confusion matrix for the Decision Trees (DT) model.

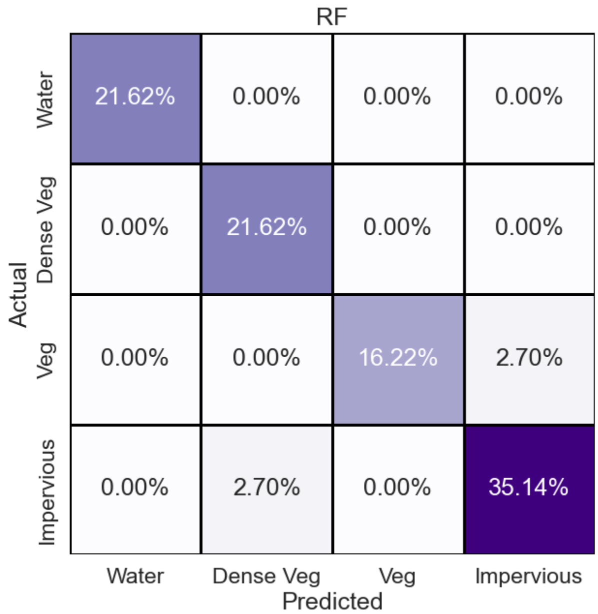

Figure 8. Confusion matrix for the Random Forest (RF) model.

For several models, we followed the same procedures. See this list for a list of Scikit-learn's available classification models. In addition to Scikit-learn, you can utilize other models like XGBoost and LightGBM. We run Random Forest (RF) and Decision Trees (DT) models for additional testing; the confusion matrix for these models is shown below. We implemented the model tests in Python3 programming language. Python codes are taken from GitHub which is a cloud-based platform [20].

To evaluate the effectiveness of the implemented machine learning algorithms (SVM, RF, and DT) in Land Use and Land Cover (LULC) classification, a comprehensive comparative analysis was conducted, the computational cost (runtime/complexity) of each classifier was assessed. **Table 2** summarizes these quantitative findings, providing a clear overview of the trade-offs between classification precision and algorithmic complexity.

Table 2. Comparative Performance Metrics and Computational Complexity Analysis.

Algorithm	Overall Accuracy (OA %)	Precision (Avg)	Recall (Avg)	F1-Score (Avg)	Computational Cost (Runtime)
Support Vector Machine (SVM)	97.30%	0.97	0.97	0.97	Moderate
Random Forest (RF)	94.59%	0.95	0.95	0.95	High (Ensemble)
Decision Tree (DT)	89.19%	0.91	0.89	0.89	Low

2.4. Predicting and Exporting Data

Predicting complete data using our trained model and exporting the numerically predicted results in native Geotiff format is the most important step [21]. Even though there are a lot of solutions (such GDAL or using GeoPandas/Geocube), they are far less effective when it comes to expanding this process to broader geographic areas (I am now conducting analysis on a national scale, and those techniques are bottlenecks). Thus, the following is a better method to use rasterio. Firstly, the original input image introduced to the program and the image should then be reshaped in the same manner as the training input data. When other characteristics were added like elevation or NDVI, combination should be done with the input image before proceeding.

Subsequent to this stage, we predicted the entire set of altered data using the pre-existing model and the predict command. We reshaped it to the input (original image height, width) shape after making the prediction. Finally, we exported the predicted/reshaped results using the raster export program [22]. After exporting, it can be seen in Geographic Information System (GIS) software (ArcGIS Pro or QGIS) or in Python. The output will appear when colors, names, map key, etc. were specified.

3. Results

We evaluate the sample prediction with x test data and the model accuracy for that specific sample prediction using Scikit-learn accuracy_score. In our case, SVM correctly classified, an accuracy of 97% is performed with Accuracy SVM: 97.297297297297. Random Forest (RF) model get results with Accuracy RF: 94.5945945945946 and Decision Trees (DT) get results with Accuracy DT: 91.8918918918919. Support Vector Machine (SVM) model performed well in comparison with the other models (See **Figure 9**). The Impervious class of LULC exhibits the highest level of confusion.

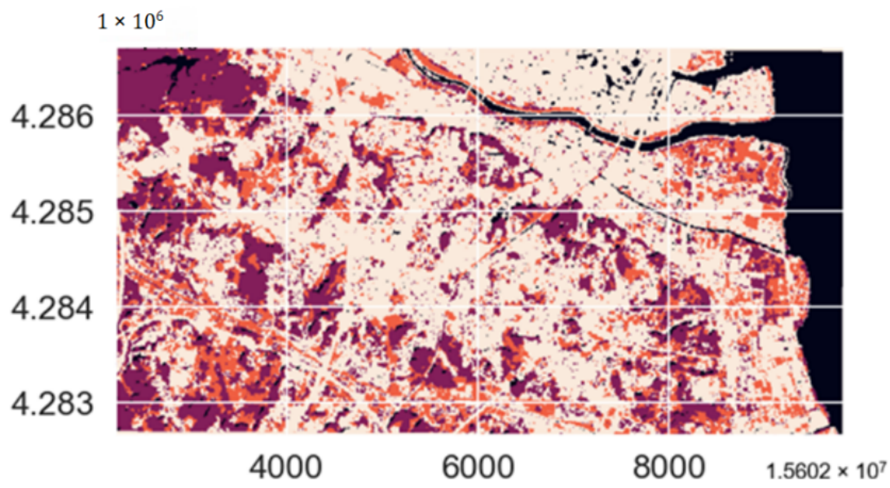

Figure 9. Predicted ML-based LULC results for Support Vector Machine (SVM) model.

Colormaps help map pixel values to colors when showing raster data, whereas colorbars provide a scaling reference for interpreting the pixel values. We displayed the first band of the DEM using a custom colormap called "terrain" and included a colorbar, as shown in **Figure 10**, to help visualize the elevation values in this case.

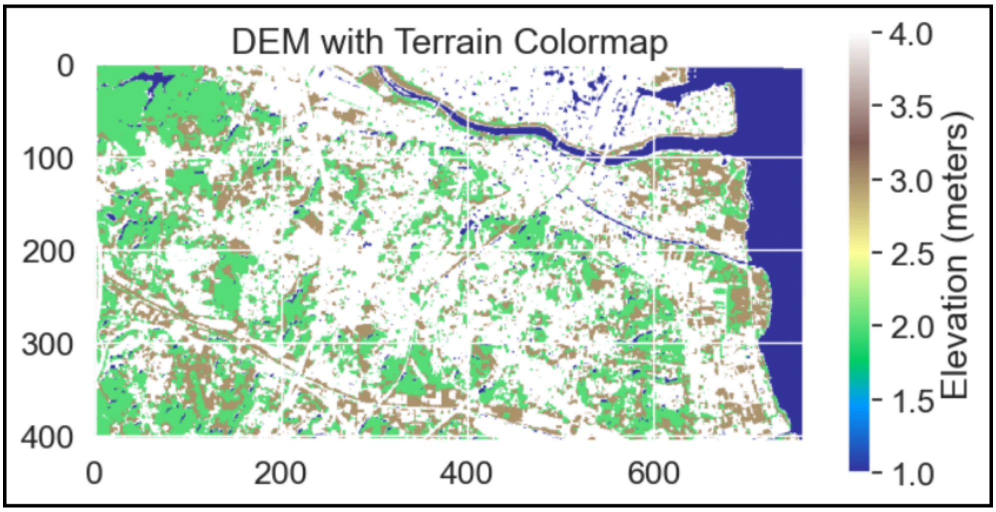

Figure 10. Terrain Colormap.

The morphology, genesis, chronology, and dynamic processes of the topography are typically the foundation of landform mapping. A DEM uses elevation data to simulate the surface of the Earth. DEM data can be used to calculate a number of topography parameters, including aspect, slope, and curvature, from which landform information can be inferred. As a result, DEM data are useful for analyzing and classifying topography, and topographic morphology is typically the basis for landform mapping using DEM. The classification of topography using DEM data has been the subject of numerous studies, most of which employ pixel-based and object-based methods. By allocating each pixel to one or more landform classes based on DEM threshold values and topography variables, pixel-based techniques automatically cluster pixels [23].

The elevation data from the DEM is represented as the first band of the raster, which we read using src.read(1) (elev_band = src.read(1)). The 'terrain' colormap is used by the plt.imshow() function to display the raster (plt.imshow (elev_band, cmap = "terrain"). In order to provide a reference for interpreting the elevation values, we also add a colorbar using plt.colorbar(). The colorbar is scaled to fit the Figure using the shrink = 0.5 option (plt.colorbar(label = "Elevation (meters)", shrink = 0.5). The plot displays the DEM together with a color map that uses the 'terrain' color scheme to emphasize various altitudes. We titled the map by plt.title("DEM with Terrain Colormap"). The distribution of the pixel values, which represent elevation in meters, is easily understood thanks to the colorbar [24].

We used rasterio export raster command to export the predicted/reshaped results. All Geotiff formatted files are saved as lulc_SVM.tif, lulc_RF.tif and lulc_DT.tif. **Figure 11** shows the predicted ML-based LULC results according to lulc_names ['Water', 'Dense Veg', 'Veg', 'Impervious']. Python or Geographic Information System (GIS) software (ArcGIS Pro or QGIS) can be used to access the exported file.

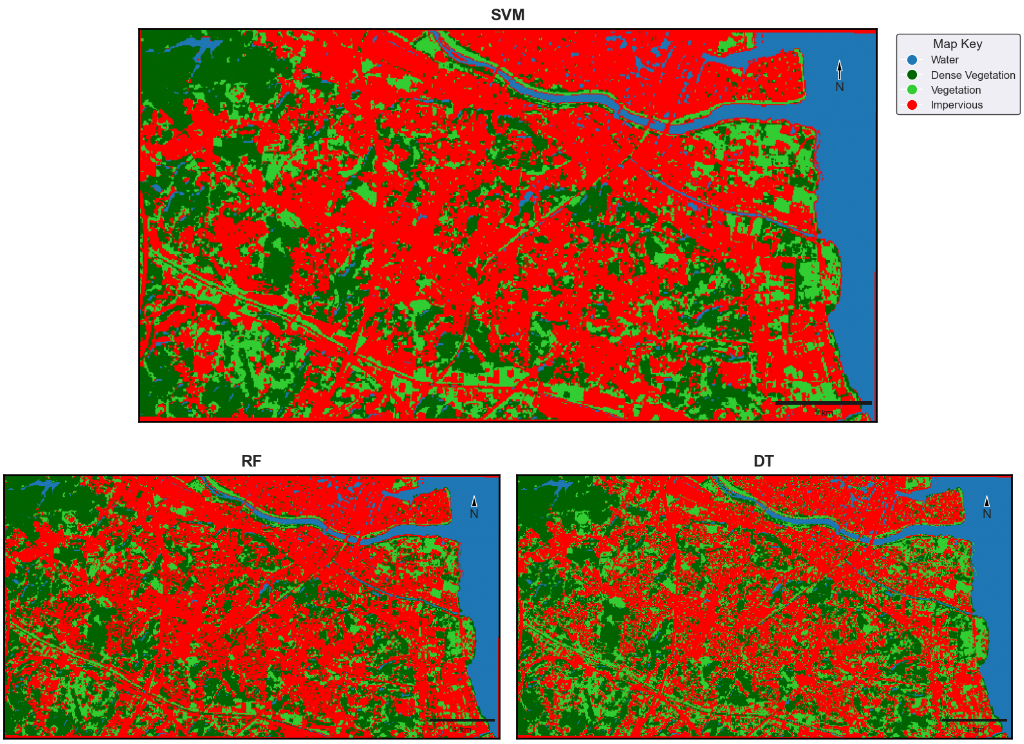

Figure 11. Predicted ML-based LULC results for SVM, RF, and DT.

4. Conclusions

In conclusion, the empirical findings of this study successfully validate the robust capabilities of Support Vector Machines (SVM), Random Forest (RF), and Decision Trees in thematic land mapping. The superior performance observed in our comparative framework corroborates the benchmarks established by Chachondhia et al. (2021) [25], confirming that advanced machine learning architectures effectively manage complex multi-band spectral responses. Ultimately, this comparative execution provides a reliable and scalable approach for high-resolution LULC classification in heterogeneous environments.

We checked the performance of the models by performing the confusion matrix and visualize it using the Seaborn "sns.heatmap" function in Python. Firstly, confusion matrix were prepared for the Support Vector Machine (SVM) model. For further testing, we run Random Forest (RF) and Decision Trees (DT) models. SWM gets an accuracy of 97%, RF gets an accuracy of 95%, whereas DT gets 92% accuracy.

Predicting complete data using our trained model and exporting numerically predicted results in native Geotiff format is the most important step. The predicted/reshaped results were exported using the raster export program. This method can be scaled over broader geographic areas more effectively with this file format. Python or Geographic Information System (GIS) software (ArcGIS Pro or QGIS) can be used to visualize the output file. Overlaying your LULC data over the input image will allow you to cross-check the results and see if the classes match the correct labels and places. As a consequence, we explored how to classify Machine Learning-based LULC using Python programming language. The procedure entails gathering and preprocessing data, extracting features, training the classification model, and classifying fresh data. Python is an effective tool for classifying land use since it contains multiple libraries that may be used for each of these processes. We may better comprehend and manage our natural resources by using land use classification with the appropriate tools and methods.

Classifying various landscapes according to their distinct features—such as flora, soil type, terrain, and land cover—is known as land use and land cover (LULC) classification. In this study, land cover names are ['Water', 'Dense Vegetation', 'Vegetation', 'Impervious']. Understanding vegetation dynamics, climate change, socioeconomic issues, and landscape patterns—all of which are crucial for forecasting global weather events—requires an understanding of land use and land cover (LULC).

The parameterized LCCS approach to categorization aims for a logical and functional hierarchical arrangement of the parameters, thereby accommodating different levels of information. It begins with broad-level classes that allow further systematic subdivision into more detailed subclasses. At every level, the designated classes are mutually exclusive. It is forbidden to apply criteria from one level of the classification to another. There is a correlation between the number of parameters utilized and the level of depth in a class definition. In other words, the class becomes more comprehensive as more parameters are supplied. The class border is determined by the presence of one or more different sorts of parameters, or by a different number of parameters. The collection of factors used to define this land-cover class is highlighted rather than the derived class name, which is the conventional approach. Geographic accuracy affects how parameters are used and how they are arranged in a hierarchy. The configuration of parameters will guarantee a high level of geographic accuracy at the highest levels of classification, or the most aggregated levels. Since land cover deals with a wide variety of classes, the features are unique to the main land-cover categories of LCCS (version 2.0) [26], which are grouped under the distinction of mostly vegetated and primarily non-vegetated territory.

Land cover is an essential resource that has a big impact on social, economic, and environmental well-being as well as climatic systems and sustainability. Therefore, fostering sustainable development, managing natural resources, and reducing climate change all depend on keeping an eye on LULC changes, as well as their causes and effects. For future developments, Predictive accuracy and applicability could be further improved by adopting deep learning models, growing datasets, and integrating real-time environmental and policy monitoring. Higher-temporal-resolution imagery, uncertainty analysis, and datasets like socioeconomic status, settlement growth, agroforestry, reforestation facilities can be incorporated into future studies.

Funding

This work received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Informed consent was obtained from all subjects involved in the study.

Data Availability Statement

The data and Python codes that support the findings of this study have been deposited in Mendeley Data and can be accessed under the following citation: Çalışkan, Berna (2026), "LULC Classification Python Codes", Mendeley Data, V1, doi: 10.17632/ysc784fk5s.1.

Conflicts of Interest

The author declares no conflict of interest.

AI Use Statement

No generative AI tools were used for content generation or data analysis. AI was employed solely for language editing and proofreading support.

References

1. Bogale, T.; Degefa, S.; Dalle, G.; et al. Machine learning-based analysis of land use and land cover trends in southeastern Ethiopia using Google Earth Engine. *Discov. Sustain.* **2025**, *6*, 878. [CrossRef]
2. Rawat, K.S.; Kumar, S.; Garg, N. Statistical Comparison of Simple and Machine Learning Based Land Use and Land Cover Classification Algorithms: A Case Study. *J. Water Manag. Model.* **2024**. [CrossRef]
3. Xie, G. Machine Learning Methods and Land Use/Land Cover (LULC) in the Coastal Pays de Brest. PhD Thesis, Université de Bretagne Occidentale–Brest, Brest, France, 2023.
4. Hermosilla, T.; Wulder, M.A.; White, J.C.; et al. Land cover classification in an era of big and open data: Optimizing localized implementation and training data selection to improve mapping outcomes. *Remote Sens. Environ.* **2022**, *268*, 112780. [CrossRef]
5. Amin, G.; Imtiaz, I.; Haroon, E.; et al. Assessment of Machine Learning Algorithms for Land Cover Classification in a Complex Mountainous Landscape. *J. Geovis. Spat. Anal.* **2024**, *8*, 34. [CrossRef]
6. Taati, A.; Sarmadian, F.; Mousavi, A.; et al. Land Use Classification using Support Vector Machine and Maximum Likelihood Algorithms by Landsat 5 TM Images. *Walailak J. Sci. Technol.* **2014**, *12*, 681–687. [CrossRef]
7. Singh, A.R.; Sharma, R. Prevailing Trends and Innovations in Machine Learning Techniques and Applications. *Int. J. Sci. Technol.* **2025**, *16*, 1–10. Available online: https://www.ijsat.org/papers/2025/2/6316.pdf
8. Ahmad, H.; Abdallah, M.; Jose, F.; et al. Evaluation and mapping of predicted future land use changes using hybrid models in a coastal area. *Ecol. Inform.* **2023**, *78*, 102324. [CrossRef]
9. Dapke, P.; Syed, A.; Nagare, S.; et al. A Comparative Analysis of Machine Learning Techniques for LULC Classification Using Landsat-8 Satellite Imagery. *Int. J. Eng. Geosci.* **2025**, *10*, 1–10. [CrossRef]
10. Phiri, D.; Simwanda, M.; Nyirenda, V.; et al. Decision Tree Algorithms for Developing Rulesets for Object-Based Land Cover Classification. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 329. [CrossRef]
11. Alshari, E.A.; Gawali, B.W. Development of classification system for LULC using remote sensing and GIS. *Glob. Transit. Proc.* **2021**, *2*, 8–17. [CrossRef]
12. Talukdar, S.; Singha, P.; Mahato, S.; et al. Land-Use Land-Cover Classification by Machine Learning Classifiers for Satellite Observations—A Review. *Remote Sens.* **2020**, *12*, 1135. [CrossRef]
13. Basheer, S.; Wang, X.; Farooque, A.A.; et al. Comparison of Land Use Land Cover Classifiers Using Different Satellite Imagery and Machine Learning Techniques. *Remote Sens.* **2022**, *14*, 4978. [CrossRef]
14. Mahmoud, R.; Hassanin, M.; Al Feel, H.; et al. Machine Learning-Based Land Use and Land Cover Mapping Using Multi-Spectral Satellite Imagery: A Case Study in Egypt. *Sustainability* **2023**, *15*, 9467. [CrossRef]
15. Waleed, M. Mastering Machine Learning Based Land Use Classification with Python: A Comprehensive Guide! Available online: https://waleedgeo.medium.com/lulc-py-78cb954673d (accessed on 26 May 2025).
16. Lu, D.; Weng, Q. Use of impervious surface in urban land-use classification. *Remote Sens. Environ.* **2006**, *102*, 146–160. [CrossRef]
17. Deval, K.; Joshi, P.K. Vegetation type and land cover mapping in a semi-arid heterogeneous forested wetland of India: Comparing image classification algorithms. *Environ. Dev. Sustain.* **2022**, *24*, 3947–3966. [CrossRef]
18. Codecademy Team. Scikit-Learn Tutorial: Python Machine Learning Model Building. Available online: https://www.codecademy.com/article/scikit-learn-tutorial (accessed on 26 May 2025).
19. Dennis, T. Confusion Matrix Visualization. Available online: https://medium.com/@dtuk81/confusion-matrix-visualization-fc31e3f30fea (accessed on 26 May 2025).
20. Waleedgeo/lulc_py. Available online: https://github.com/waleedgeo/lulc_py/tree/main/materials/results (accessed on 26 May 2025).
21. Bhat, R. Extracting Metadata from TIFF Images Using Python. Available online: https://ra-bhat2002.medium.com/extracting-metadata-from-tiff-images-using-python-c1f3b38ab5ce (accessed on 26 May 2025).
22. Rasterio. Available online: https://github.com/rasterio/rasterio (accessed on 26 May 2025).
23. Yang, J.; Xu, J.; Lv, Y.; et al. Deep learning-based automated terrain classification using high-resolution DEM

data. *Int. J. Appl. Earth Obs. Geoinf.* **2023**, *118*, 103249. [CrossRef]

24. Rasterio. Available online: https://geog-510.gishub.org/book/geospatial/rasterio.html (accessed on 26 May 2025).
25. Chachondhia, P.; Shakya, A.; Kumar, G. Performance evaluation of machine learning algorithms using optical and microwave data for LULC classification. *Remote Sens. Appl. Soc. Environ.* **2021**, *23*, 100599. [CrossRef]
26. Jansen, L.J.M.; Badea, A.; Milenov, P.; et al. The Use of the Land-Cover Classification System in Eastern European Countries: Experiences, Lessons Learnt and the Way Forward. In *Land Use and Land Cover Mapping in Europe*; Springer: Dordrecht, The Netherlands, 2014; pp. 297–314.