Comparison of Random Forest and Support Vector Machine Classifiers for Regional Land Cover Mapping Using Coarse Resolution FY-3C Images

Tesfaye Adugna 1,2,3,*; Wenbo Xu 1,2 and Jinlong Fan 4

1 School of Resources and Environment, University of Electronic Science and Technology of China, Chengdu 611731, China; xuwenbo@uestc.edu.cn
2 Yangtze Delta Region Institute, University of Electronic Science and Technology of China, Huzhou 313001, China
3 College of Applied Sciences, Addis Ababa Science and Technology University (AASTU), Addis Ababa P.O. Box 16417, Ethiopia
4 National Satellite Meteorological Center, China Meteorological Administration, Beijing 100081, China; fanjl@cma.gov.cn
* Correspondence: 201814010010@std.uestc.edu.cn or tesfishmom@gmail.com; Tel.: +86-13258256072

Abstract: The type of algorithm employed to classify remote sensing imageries plays a great role in affecting the accuracy. In recent decades, machine learning (ML) has received great attention due to its robustness in remote sensing image classification. In this regard, random forest (RF) and support vector machine (SVM) are two of the most widely used ML algorithms to generate land cover (LC) maps from satellite imageries. Although several comparisons have been conducted between these two algorithms, the findings are contradicting. Moreover, the comparisons were made on local-scale LC map generation either from high or medium resolution images using various software, but not Python. In this paper, we compared the performance of these two algorithms for large area LC mapping of parts of Africa using coarse resolution imageries in the Python platform by employing the Scikit-Learn (sklearn) library. We employed a big dataset, 297 metrics, comprised of systematically selected 9-month composite FengYun-3C (FY-3C) satellite images with 1 km resolution. Several experiments were performed using a range of values to determine the best values for the two most important parameters of each classifier, the number of trees and the number of variables, for RF, and penalty value and gamma for SVM, and to obtain the best model of each algorithm. Our results showed that RF outperformed SVM yielding 0.86 (OA) and 0.83 (k), which are 1–2% and 3% higher than the best SVM model, respectively. In addition, RF performed better in mixed class classification; however, it performed almost the same when classifying relatively pure classes with distinct spectral variation, i.e., consisting of less mixed pixels. Furthermore, RF is more efficient in handling large input datasets where the SVM fails. Hence, RF is a more robust ML algorithm especially for heterogeneous large area mapping using coarse resolution images. Finally, default parameter values in the sklearn library work well for satellite image classification with minor/or no adjustment for these algorithms.

Keywords: large area; land cover mapping; machine learning (ML); Python; Scikit-Learn (sklearn); random forest (RF); support vector machine (SVM); parameter tuning; FengYun-3C; Africa

1. Introduction

Land cover (LC) information provides some of the most indispensable data in various sectors including environmental, ecological and climate change studies, and resource management and monitoring [1–3]. One of the best ways of recording and conveying land cover information is by using land cover maps. LC map production requires considering numerous issues that determine the property of the map such as purpose, thematic content,
scale, type of input data, and algorithms employed. It can be derived at different scales and broadly divided into three, i.e., either based on the areal extent it covers: local scale (covers a small area 100–10^3 km^2), regional scales (10^4–10^6 km^2), and continental to global scales (>10^6 km^2) [4] or according to its spatial resolution: coarse (≥1 km), moderate (1 km–100 m), and fine (<10 m) resolution [5].

It has been more than four decades since the first land cover map production from remote sensing data was realized [6]. Within these years, quite dramatic improvements have been made on the image classification methods. In the early days, traditional image classification such as supervised parametric and unsupervised techniques were the most widely employed techniques to produce the LC maps with different resolutions (e.g., Friedl, et al. [7]; Mayaux, et al. [8]; Tchuenté, et al. [9]; Loveland and Belward [10], Hansen, et al. [11]; Arino, et al. [12]). However, the significance of these techniques has started to decline due to the notable limitations they possess. The former assumes Gaussian normal data distribution, which is rarely the case in remote sensing data [13,14], and the latter requires limited expert involvement, i.e., the algorithm clusters pixels with similar spectral characteristics into a single class based on some predefined criteria [15,16].

Therefore, in order to address the drawbacks associated with traditional image classification methods and to obtain a better classification result, advanced classification techniques have been developed and implemented in recent years [13,17,18]. In this regard, machine learning (ML) algorithms are among the foremost and robust classifiers that have gained great acceptance (e.g., Mountrakis, Im, and Ogole [13]; Ghimire, Rogan, Galiano, Panday, and Neeti [14]; Waske, van der Linden, Benediktsson, Rabe, Hostert, and Sensing [17]; Foody and Mathur [19]; Maxwell, et al. [20]; Belgiu and Drăgu¸t [21]; Ghosh, et al. [22]; Rodriguez-Galiano, et al. [23]; Pal and Mather [24]; and Pal [25]).

Machine learning algorithms are non-parametric supervised methods that make no assumptions about the statistical distribution of input datasets in contrary to the traditional supervised algorithms [13,14,16]. There exist several types of ML algorithms; where, some have been rigorously tested, including in practical (non-research) environments, and hence they are referred to as matured, for instance, support vector machines (SVM), single decision trees (DTs), Random Forests (RF), boosted DTs, artificial neural networks (ANN), and k-nearest neighbors (k-NN). On the contrary, some classifiers are relatively less thoroughly studied and considered as immature, e.g., the extreme learning machines kernel-based extreme learning machines (KELMs), and deep convolution neural networks (CNN) [18]. Concerning their performance, studies have demonstrated that ML techniques yield more accurate results than conventional parametric classifiers, particularly for complicated, multi-modal data consisting of several features (e.g., Ghimire, Rogan, Galiano, Panday, and Neeti [14]; Waske, van der Linden, Benediktsson, Rabe, Hostert, and Sensing [17]; Huang, et al. [26]; Pal and Mather [24]; Gualtieri and Chhetri [27]; Khatami, et al. [28]; Friedl and Brodley [29]; and Otukei and Blaschke [30]).

Although several comparison works have been conducted between the two most widely applied and effective methods of machine learning algorithms, RF and SVM, which are also known for finding the global minimum [13], using various remote sensing datasets for different purposes, the conclusions drawn are inconsistent and contradicting. For example, Adam, et al. [31]; Ghosh, Fassnacht, Joshi, and Koch [22]; Dalponte, et al. [32]; and Pal [25] concluded that SVM and RF produce similar classification accuracy implying both are equally reliable; whereas, Khatami, Mountrakis, and Stehman [28]; Raczko and Zagarowski [33]; Li, et al. [34]; Thanh Noi and Kappas [35]; Zhang and Xie [36]; Maxwell, Warner, Strager, Conley, and Sharp [20]; Maxwell, et al. [37]; and Ghosh and Joshi [38] reported that SVM outperformed RF. In contrast to both findings, studies by Abdel-Rahman, et al. [39], Shang and Chisholm [40], and Lawrence and Moran [41] indicted that RF is superior to SVM. Moreover, these studies were carried out on a small area, local scale entirely employing either high or medium resolution imageries as their principal input datasets. To the best of our knowledge, comparisons of these algorithms for regional/large area mapping using several inputs of coarse resolution images have not been conducted so far.
In this study, therefore, we aimed to compare the performance of these ML algorithms to generate a large area land cover by manipulating big input dataset of coarse resolution images obtained from the FengYun-3C (FY-3C) satellite.

Our work systematically evaluated the performance of these powerful algorithms on regional mapping of the parts of Africa using FY-3C composite imageries that have 1 km spatial resolution and collected over several months, i.e., consisting of several variables (bands). To perform the comparison, we selected the best models of each classifier that was determined by testing several models that were created by varying the values of the two most influential parameters of each classifier, i.e., number of trees (Ntree) and the number of variables (Mtry) of RF and the penalty function/cost value (C) and gamma (γ) of SVM. Ranges of parameter values were tested including the default values given in the sklearn python platform to find the best/optimum values.

Furthermore, employing Scikit-Learn (sklearn) and other libraries of the Python platform for this work, which is becoming the most popular tool in the remote sensing community, allows us to evaluate the effectiveness of the various default parameters’ values set in the software.

2. Study Area

The study area, which is shaded in gray and is the same region considered in our previous work Adugna, et al. [42], covers about one-third of the total area of the continent of Africa (Figure 1). It is situated between northing: 11°58′31.72″S, 33°0′18.56″N, and easting: 19°4′35.02″E, 51°24′37.32″E. It includes about 18 nations, partially or fully, which are located in the central, eastern, and northeastern parts of the continent. Heterogeneous land cover types and drastic climatic variations characterize the region. Three climatic zones characterize the regions: the arid, Sahelian, tropical, and equatorial; where the majority of the northern part is arid and includes the Sahara, the world’s largest desert, whereas central Africa is known for its tropical rain forest cover.

Figure 1. Study area, shaded in gray, and its location in the continent of Africa.
3. Materials and Methods

To perform the various experiments and achieve our objective we used some of the techniques and materials that were employed in our earlier work, see Adugna, Xu, and Fan [42]. For instance, in this research, we considered the same study area, input datasets, and reference data as the previous work, as the main aim of this work is to evaluate the performance of the two ML classifiers for large area/regional land cover mapping using big datasets (time series data) of coarse resolution imageries. Figure 2 below elucidates the technical route implemented for this study.

Figure 2. Technical route.

3.1. Classification Scheme

We broadly categorized the land cover types of the study area into 8 major classes (Table 1) according to the land cover classification system (LCCS) developed by United Nations (UN) Food and Agriculture Organization (FAO).

Table 1. Land cover classes and their definition, modified after Smets, et al. [43].

| Land Cover Class     | Class Definition Based on UN LCCS                                                                                     |
|---------------------|-----------------------------------------------------------------------------------------------------------------------|
| 1. Bare/Sparse vegetation | Barren land or surface covered by plants that should not exceed 10% in every season of the year.                      |
| 2. Built-up         | Area of land covered with buildings and other man-made features.                                                       |
| 3. Cropland         | Intermittently cultivated land that is harvested and then left fallow (e.g., single and multiple cropping systems). Perennial woody crops are classified as either forest or shrub according to the criteria. |
Table 1. Cont.

| Land Cover Class | Class Definition Based on UN LCCS |
|------------------|----------------------------------|
| 4. Forest        | Woody plants cover more than 15% of the land and grow to a height of more than 5 m. Exceptions: even if its height is less than 5 m but larger than 3 m, a woody plant with a characteristic physiognomic trait of a tree can be classified as a tree. |
| 5. Herbaceous wetland | A persistent mixture of water and herbaceous or woody vegetation covers the land. The plants can exist in salt, brackish, or freshwater. |
| 6. Herbaceous vegetation | Plants with no persistent branches or shoots above the surface and no apparent solid structure. Up to 10% of the area may be comprised of trees and plants. |
| 7. Shrubs        | Woody perennial plants with persistent and woody stems that are less than 5 m tall and do not have a clear main stem. The shrub’s leaves are either evergreen or deciduous. |
| 8. Water bodies  | These include lakes, reservoirs, and rivers. The water could be fresh or brine. |

3.2. Input Data

The principal input datasets for this study are the same as the dataset that we processed in our previous work, see Adugna, Xu, and Fan [42]; i.e., we employed 10 days of composite images that were acquired by the visible and infrared radiometer (VIRR) sensor carried by FengYun-3C (FY-3C) satellite. We collected one year, April 2019 to the end of March 2020, 10-day composite imageries with 1 km spatial resolution from the Chinese Metrological Administration (CMA).

The FY-3C satellite is a recent Chinese sun-synchronous meteorological satellite that started operation on 23 September 2013. It is a second-generation polar-orbiting, morning or noon orbiting satellite that belongs to FengYun-3 (FY-3, “wind cloud”) series [44]. It operates at an altitude of 836 km with an orbital inclination of 98.75° and crosses the equator at 10:00 a.m. [45]. This modern, mature satellite has reached a stable operation stage [46] and comprises 12 payloads on board, which have different capabilities, functions, and properties [44,45,47]. Moreover, the quality of the data generated by FY-3C and FY-3D, which is a similar instrument to FY-3C except for the number bands, is comparable to that of Moderate Resolution Imaging Spectroradiometer (MODIS) and Advanced Very High-Resolution Radiometer (AVHRR) [47,48].

The data for this work, as mentioned above, were acquired with the Visible and Infrared Radiometer (VIRR), which is one of the twelve instruments mounted on the FY-3C. The VIRR scans the entire Earth daily with a swath width of 2800 km and spatial resolution of 1.1 km at nadir by employing 10 spectral channels that use a wavelength range between 0.44 and 12.50 µm that enables it to gather information about the atmosphere, ocean, and land in the visible and infrared spectral regions [45,47,49].

After acquiring one-year composite imageries, we selected eleven bands including NDVI (see Table 2), and systematically generated seven different input datasets that comprised stacked composite images of 1 month, 3 month, 6 month, 9 month, 12 month, and selected images from 12 months using band/feature importance and selected 9 month to obtain the best input dataset. Then, we evaluated each dataset using the same test data and random forest model that was trained with the same training dataset so that the difference in accuracy is solely due to the input dataset. Comparison of these seven datasets using overall accuracy, kappa value, and individual class accuracy revealed that the selected 9-month input dataset, which is systematically generated data from the one-year data via feature selection technique, is the best input dataset as reported in our previous work, see Adugna, Xu, and Fan [42].
Table 2. Selected scenes/bands and their respective wavelength.

| Band No. | Name of Bands                      | Wavelength Range in µm |
|----------|-----------------------------------|------------------------|
| Band 1   | FY-3C_VIRR_Day_EV_RefSB           | 0.58–0.68              |
| Band 2   | FY-3C_VIRR_Day_EV_RefSB           | 0.84–0.89              |
| Band 3   | FY-3C_VIRR_Day_EV_RefSB           | 1.55–1.64              |
| Band 4   | FY-3C_VIRR_Day_EV_RefSB           | 0.43–0.48              |
| Band 5   | FY-3C_VIRR_Day_EV_RefSB           | 0.48–0.53              |
| Band 6   | FY-3C_VIRR_Day_EV_RefSB           | 0.53–0.58              |
| Band 7   | FY-3C_VIRR_Day_EV_RefSB           | 1.325–1.395            |
| Band 8   | FY-3C_VIRR_Day_EV_Emissive        | 3.55–3.93              |
| Band 9   | FY-3C_VIRR_Day_EV_Emissive        | 10.3–11.3              |
| Band 10  | FY-3C_VIRR_Day_EV_Emissive        | 11.5–12.5              |
| Band 11  | MVC value of NDVI                 | -                      |

Thus, this dataset, which consists of selected 9-month (April, May, June, July, August, September 2019, and January, February, March 2020) multi-temporal data with a size of 11 GB, was selected and manipulated to meet our aims for this work.

3.3. Training and Test Data

Several investigations have indicated that the quantity and quality of training data significantly affect the accuracy of land cover maps (e.g., Mountrakis, Im, and Ogole [13]; Waske, van der Linden, Benediktsson, Rabe, Hostert, and Sensing [17]; and Foody and Mathur [19]), they being even more important than the type of algorithm selected [26]. However, there seems a notable disagreement on the quantity of the training data that should be collected to obtain the required accuracy. For example, Foody and Mathur [19], attained the best accuracy, above 90%, by collecting 100 pixels per class and employing the SVM classifier. On the other hand, Jensen and Lulla [50] argue that training samples should not be fewer than ten times the number of features in the classification model. The other suggestion is that the training data should be 0.25 percent of the total study area, see Thanh Noi and Kappas [36]. Despite these contradictions, there is consensus that employing good quality and a large number of training samples is of paramount importance in finding the best output.

For this study, therefore, we utilized a large quantity of reference data (around 120,000 pixels) that were collected for our previous studies of the same study area, see Adugna, Xu, and Fan [42]; details of the collection procedures were stated in that paper. For completeness, below we summarized the methods employed.

For reference data collection, we acquired atmospherically corrected 12-month Landsat 8 imageries, Collection 1 Level-2 surface reflectance imageries from the United States Geological Survey website (https://earthexplorer.usgs.gov/, accessed on 20 October 2020) where their location and or distributions were determined based on eco-regions of Africa, existing LC maps of Africa, Copernicus global land cover, and personal experience. During image acquisition, we set cloud cover to 10% and date of image acquisition from 1 April 2019 to 30 March 2020, which is the same as the time range used for the input dataset so that land cover variations that might occur due to a mismatch in the image acquisition period will be minimized significantly, if not avoided.

Reference data collection was performed on false-color composites, often composites of bands 5, 4, and 3, Landsat imageries that were manually selected best scenes. A large quantity of training and test data, about 118,874 pixels, were collected from systematically selected sites across the region of interest (see Figure 3).
To identify or label individual land cover classes we employed three methods/references simultaneously. These are Landsat 8 composite image interpretations; frequent consultation to existing land cover map of the target area, i.e., Copernicus global discrete land cover map at 100 m resolution (CGLS LC100 discrete map) for the year 2019 (https://lcviewer.vito.be/download, accessed on 20 October 2020), and Google Earth Pro/Google Maps. To consider the land cover class as an independent class, it should occur homogenously at least for the minimum mapping unit (1 km square, i.e., roughly 40 × 40 Landsat pixels). Employing the three techniques is crucial to make the interpretation more consistent, reliable, and precise than using a single reference.

Once the classes were identified and annotated as separate polygons in XML file format, they were converted to a single shapefile comprising all the polygons using ENVI for that particular scene. Finally, all exported shapefiles from the various locations/scenes were merged by using Arcmap to make them a single shapefile that contain all the reference data and later divided into training and test sample with an approximate ratio of 75/25, respectively. The test samples were randomly but systematically picked with a strategy of 5 samples out of 20 polygons from every land cover type to minimize and/or avoid spatial correlation with the training data, and at the same time to find fairly distributed test data across the study area.

Finally, two separate shapefiles, training samples with 91,207 pixels and test data with 27,667 pixels, were created to train and test our models. Moreover, the quantity of training and test pixels for the individual land covet types is different (Table 3). That is primarily/majorly due to variations in their prevalence and/or the difficulty to obtain homogenous sample mapping units of the minimum size in rare cases. The technique
of allocating the reference samples based on the extent of areal coverage yields better results [51].

Table 3. Training and Test data per individual classes.

| Class (Name)                          | Training Data | Test Data |
|---------------------------------------|---------------|-----------|
| Class 1 (Bare/Sparse vegetation)      | 20,585        | 6424      |
| Class 2 (Built-up)                    | 1620          | 575       |
| Class 3 (Cropland)                    | 15,472        | 4246      |
| Class 4 (Forest)                      | 11,378        | 3101      |
| Class 5 (Herbaceous wetland)          | 1686          | 602       |
| Class 6 (Herbaceous vegetation)       | 12,394        | 3641      |
| Class 7 (Shrubs)                      | 18,956        | 5283      |
| Class 8 (Water bodies)                | 9116          | 3795      |
| Total number of pixels                | 91,207        | 27,667    |

3.4. Algorithms

Machine learning algorithms are nonparametric supervised approaches that have gained a wide acceptance among the remote sensing community in recent years due to their great success in remote sensing image classifications (e.g., Belgiu and Drăguț [21]; Wulder, Coops, Roy, White, and Hermosilla [16]; Maxwell, Warner, Strager, Conley, and Sharp [20]; Mountrakis, Im, and Ogole [13]; and Pal [25]).

Machine learning algorithms have several advantages, such as the capability of handling complicated class patterns and embracing a wide range of input variables; in addition, as opposed to traditional supervised algorithms they are not influenced by the statistical distribution of the data [20]. In other words, they do not assume input datasets are Gaussian. Furthermore, these techniques typically outperform standard para-metric classifiers, specifically for complicated datasets and input datasets with high dimensional parameters, i.e., many predictors [14,24–26,29,30,52].

In this study, two competitive machine learning algorithms, random forest and support vector machine, were used to find out how they perform on large area land cover classification where high dimensional input datasets were considered. According to previous works, both are regarded as the most effective and widely applied classifiers for satellite image classifications. However, the performance of these algorithms is substantially affected by the values of the parameters they employ. In this regard, the number of trees (Ntree) and the number of variables (Mtry) are the two parameters that greatly impact the performance of the RF model [21,35]. Whereas penalty function/cost value (C) and gamma (γ) are the two essential parameters that control the performance of SVM when the radial basis function (RBF) is considered as the kernel function [19,26,53].

To conduct performance comparisons between the best models of each algorithm, we conducted several trial-and-error experiments in order to determine the best model of each classifier by varying the magnitudes of the two most influential parameters of each classifier. In other words, comparisons of the two algorithms were carried out after obtaining two optimum values of the parameters for each model by testing ranges of values.

3.4.1. Random Forest (RF)

The random forest (RF) is an ensemble classifier that comprises a large number of decision trees created by randomly chosen predictors from randomly selected data that is a subset of the training dataset, and the final classification/prediction decision is made based on a majority vote [21,54]. It is among the best and most powerful machine-learning classifiers [21].

A random forest is often generated using bagging and random subspace techniques [55]. The trees are created by replacing a subset of training samples (a bagging technique in which the same sample could be picked frequently at the expense of the other samples).
The trees are trained with about two-thirds of the samples (referred to as in-bag samples). The other one-third, also known as out-of-the-bag samples, is utilized for internal cross-validation to evaluate the RF model or to estimate the out-of-bag (OOB) error [55,56]. Every decision tree is created separately, with no pruning, and each node is divided using randomly selected variables from the number of features (Mtry) defined by a user. The technique permits us to produce trees with high variance and low bias, and that ultimately leads to the creation of the forest consisting of several trees (Ntree), estimators, as specified by the user [54]. Finally, the model uses majority votes from many predictors (trees) to classify new samples [56].

In the literature, a number of benefits of using the RF algorithm have been highlighted. RF results in high accuracy output [21,23], outperforms other powerful machine learning models such as discriminant analysis, support vector machines, and neural networks [56], and is resilient to overfitting [21,54]. It is computationally fast relative to other ML algorithms such as SVM. In addition, it allows us to select important variables [23] that allow us to eliminate the least important attributes [18,21,56].

Although the RF is one of the robust ML classifiers, it has certain notable drawbacks such as difficulty to visualize the trees as it uses several trees to make predictions [54], and split rules for classification are mysterious/unclear, and hence it is considered as black-box type classifier [23].

Furthermore, as mentioned earlier the performance of the RF algorithm is highly impacted by Ntree and Mtry, the two user-defined parameters [18,21]. According to Ghosh, Fassnacht, Joshi, and Koch [22] and Kulkarni and Sinha [57] the level of the impact of these parameters on accuracy is different, claiming the Mtry parameter is more significant than the Ntree parameter. However, employing optimum values for both parameters is critical to achieving higher accuracy. In this regard, Belgiu and Drăgut [21] and Gislason, et al. [58] proposed 500 as a default number for Ntree. Guan, et al. [59], on the other hand, suggested as many Ntree values as feasible, stating that the RF classifier is efficient/robust and resistant to overfitting. In terms of the Mtry parameter, it is commonly defined as the square root (sqrt) of the number of input variables [21,58]. Some researchers, however, define Mtry to be equal to the entire number of available variables (e.g., Ghosh, Fassnacht, Joshi, and Koch [22]). According to Belgiu and Drăgut [21], setting such value, however, might affect the speed of the algorithm because the method must compute the information gained from all of the parameters considered to split the nodes.

In this work, therefore, we considered ranges of Ntree (‘n_estimators’) and Mtry (‘max_features’) values to tune the parameters and obtain the optimum magnitudes of the parameters that yield the best result in terms of overall and individual accuracy and Kappa coefficient. We started the experimentation with default values, for all parameters but random state, as given in sklearn library (https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html, accessed on 15 December 2021) by Pedregosa, et al. [60]. Where Ntree (‘n_estimators’) =100 and Mtry (‘max_features’) =auto, i.e., is the square root of the number of variables but the random state was converted to 42 from “None“. Then, we increased the value of Ntree to 300, 500, 700, and 1000 and tested them by maintaining the Mtry at default values (auto). Once the best Ntree value was determined, we kept it constant and conducted further experiments to find the best value of Mtry by considering a range of magnitudes, i.e., Mtry = 10, 40, 100, and 200. In total, nine tests were carried out to decide the two best values of the two most important parameters of RF.

3.4.2. Support Vector Machine (SVM)

The support vector machine (SVM) is a collection of theoretically powerful machine learning algorithms [26]. The fundamental principle of SVM lies in creating an optimum hyperplane also referred to as a decision boundary or optimal boundary that maximizes the distance between the nearest samples (support vectors) to the plane and effectively separates classes [19,26,53,61]. The model seeks to find the optimal separating hyperplane
between classes by focusing on the training cases that occur at the edge of the class distributions, the support vectors, with the other training cases effectively discarded [62,63]. As a result, the approach can yield high accuracy with small training datasets that cut the costs of training data acquisition, this is considered as one of the pros of employing the algorithm. The basis of the SVM approach to classification is, therefore, the notion that only the training samples that lie on the class boundaries are necessary to separate classes [19].

The construction/mathematical definition of optimum hyperplane is significantly determined by the nature of the distribution of the training samples, i.e., whether the datasets are discernible effectively or separable with certain inevitable errors. When two classes are completely separable the decision boundary between them is represented by two equations \( w \cdot x_i + b \geq +1 \) (for \( y_i = +1 \)), and \( w \cdot x_i + b \leq -1 \) (for \( y_i = -1 \)); otherwise, it is defined by \( w \cdot x \geq 1 - \xi_i \) (for \( y_i = +1 \)) and \( w \cdot x_i + b \leq -1 + \xi_i \) (for \( y_i = -1 \)). Where \( w \) is the norm to the optimal plane, \( x \) is training data (points) on hyperplane, \( b \) represents the bias, \( (\xi_i) \) is the slack variable that is the offset of the misclassified data from the optimal plane, and \( "y_i" \) is labeled data/classes. Moreover, in cases of indiscernible samples, the slack variable \( (\xi_i) \) and penalty value also known as cost value (C) are introduced to penalize the outliers, i.e., to regularize/compensate for misclassification/errors. Finally, the best hyperplane that classifies the data with the largest gap between the support vector and the plane is obtained by minimizing the norm \( (w) \) function \( F(w) \), Equation (1), for purely separable samples; and \( F(w, \xi) \), Equation (2), for non-differentiable data, which are the most common data type in remote sensing [19,26]. Accordingly, they are expressed mathematically as follow [26]:

\[
F(w) = (w'w)^{\frac{1}{2}} \tag{1}
\]

\[
F(w, \xi) = (w'w)^{\frac{1}{2}} + C(\sum_{i=1}^{k} \xi_i) \tag{2}
\]

In the second function (Equation (2)), the variable C is used to control the extent of the penalty to regularize misclassified training datasets, i.e., data lie on the wrong side of the optimal hyperplane, and hence they play a significant role in influencing the accuracy and/or the capacity of the algorithm to generalize [19,26,53,61]. If the value of C is set to be high, the penalty factor is high, which leads to overfitting and diminishing the power of the model to generalize unseen data. On the other hand, if it is small, it can result in a smoothing, i.e., a biased model or underfitting [13,19,26,53,61]. Therefore, setting an appropriate value of this parameter is crucial. In this regard, Yang [53] suggests a moderate value to overcome the trade-off.

The constant C is not the only parameter that affects the performance of the SVM algorithm. There exist other important parameters that profoundly control its effectiveness [13,53]; these include the type of kernel functions, functions that are used to project/map linearly inseparable samples to higher dimensional space so that they can be classified linearly, and their parameters [17,19,26,27].

Concerning kernel functions, despite the existence of numerous different types of kernels, such as linear, polynomial, radial basis function, and sigmoid, the radial basis function (RBF) kernel is the most effective and commonly used parameter in remote sensing image classifications [17,19,35,53,61], and hence we adopted it for this work. For the RBF kernel to perform well, two parameters, namely, penalty value (C) and gamma (\( \gamma \)), should be meticulously chosen [26,34,53]. The effect of gamma (\( \gamma \)), a parameter that controls the width of the kernel, on the SVM model using the RBF kernel is similar to C, in that if a high value is assigned to it, the model will overfit and does not generalize well [19]. Detailed theoretical and mathematical explanations of SVM have been given by Huang, Davis, and Townshend [26] and Foody and Mathur [19].

In addition to being the most widely used and robust classifier for remote sensing image classification [19], SVM has several advantages. As it is a non-parametric ML algorithm it efficaciously handles multi-modal datasets with hundreds of bands/channels [17], i.e., it
is insensitive to problems associated with the dimensionality of data [17,27]. It also works well with a smaller amount of training samples [13,17,19,26], given that good representative data that lie at the boundary of the class distributions are fed in and allow us to define the optimum hyperplanes [19,62]. Moreover, unlike other advanced ML algorithms such as the neural network, it does not fall in local minimum at times, it always results in the global maximum [13].

Although the SVM classifier has considerable attributes that make it a superior ML algorithm, it has also certain limitations apart from setting its parameters. Its performance can drastically drop even if relatively small wrongly labeled training samples are incorporated, i.e., more sensitive to noisy data than other algorithms [13]. In addition, according to Waske, van der Linden, Benediktsson, Rabe, Hostert, and Sensing [17] it can be affected by the curse of dimensionality; although, it performs quite well in handling high-dimensional input datasets most of the time. Furthermore, SVM was originally developed to separate only two classes, i.e., for binary classification, and employing it for multiclass classifications could be problematic [19]. However, it can be successfully enhanced to classify datasets consisting of several categories/classes, such as remote sensing data, by employing a variety of techniques; for instance: one-against-one, one-against-others, directed acyclic graph (DAG) strategies, and multiclass SVM [55].

As mentioned earlier, the performance of the SVM model using the RBF kernel is greatly influenced by values assigned to the two important parameters, C and $\gamma$. In this study, therefore, to figure out the optimum values for these parameters we conducted 23 experiments. Similar to the techniques we employed for RF, we started with default parameter values given in the sklearn python library, (https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html, accessed on 22 December 2021), by Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, and Dubourg [60], but the ‘random_state’ = 0 in all tests we conducted. That is, we started the experiment with the following values: penalty value (C) = 1, gamma = scale ($1/(n_features*X.var()$)), where $n_features$ represents number of features, $X$ is pixels’ values (reflectance in the mxn dimension (matrix)), “.” is dot product and var() is a variance, and kernel = RBF. Then, we executed 10 tests by varying the magnitude of C step by step from 50 to 2000, i.e., we tried a range of values (50, 100, 300, 600, 900, 1000, 1200, 1500, 1800), to obtain the best C value while keeping the $\gamma$ and other values unchanged, i.e., fixing them to the default value. After determining the optimum value C, we conducted five tests to determine the best $\gamma$ from a range of values ($10^{-3}$, $10^{-6}$, $10^{-8}$, $10^{-10}$, $10^{-12}$) by maintaining the best C value fixed, the other parameters at default values. Finally, the two best values were selected to build the best SVM model and compared with the best RF model for which the optimum parameters were found via a similar method, systematic trial-and-error techniques, as employed by numerous investigators (e.g., Liaw and Wiener [56]; Li, Wang, Wang, Hu, and Gong [34]; Qian, Zhou, Yan, Li, and Han [61]; and Raczko and Zagajewski [33]).

3.5. Accuracy Assessment

In algorithm comparison works, it is a common practice to use overall accuracy, user’s (also known as precision), producer’s accuracy (recall), and kappa coefficient to determine which classifier achieves a better accuracy (e.g., Li, Wang, Wang, Hu, and Gong [34]; Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and Rigol-Sanchez [23]; Waske, van der Linden, Benediktsson, Rabe, Hostert, and Sensing [17]; and Yang [53]). It is also common to employ only overall accuracy and/or kappa along with producer’s accuracy, for instance, Ghosh, Fassnacht, Joshi, and Koch [22], Raczko and Zagajewski [33]; and Huang, Davis, and Townshend [26].

In this work, we generated overall accuracy (OA), user’s (also known as precision), producer’s accuracy (recall), f1-score, and kappa coefficient (k), using the same test set sample (see Table 3). Employing all these accuracy-measuring methods permits us to assess the performance of the model based on the overall and individual class accuracy. Moreover,
conducting comparisons of classifiers only based on only overall accuracy (OA) and/or kappa (k) could lead to drawing a wrong conclusion as discussed below.

4. Results

As shown in Tables 4 and 5, several (21) experimentations were carried out using ranges of parameter values to determine the optimum parameter values of two of the most crucial parameters of the classifiers and to find the best model of each algorithm. Comparisons of these models reveal that RF outperformed SVM. The best results, highest overall accuracy (0.86) accuracy, kappa score (0.83), and generally higher individual class accuracy were achieved when the two important RF model parameters, Ntree and Mtry, were set to be 100 and ‘auto’, respectively, which are default parameter values as given in sklearn library of Python platform (see Table 4, exp. 1). Moreover, the RF model is fast and more effective in handling large datasets.

Table 4. Accuracy of the various RF models.
Table 4. Cont.

| Experiment No. | Input Data | Class Name | Bare/Sparse Vegetation | Built-Up | Cropland | Forest | Herbaceous Wetland | Herbaceous Vegetation | Shrub | Water Bodies |
|----------------|------------|------------|------------------------|----------|----------|--------|-------------------|-----------------------|-------|--------------|
|                |            | Class ID   | 1                      | 2        | 3        | 4      | 5                  | 6                     | 7     | 8            |
|                |            | No. of Pixels | 6424                 | 575      | 4246     | 3101   | 602               | 3641                 | 5283  | 3795         |
| 8              | Parameter  | Ntree      | default (100)         |          |          |        |                    |                       |       |              |
|                |            | Mtry       | 100                   |          |          |        |                    |                       |       |              |
|                |            | precision  | 0.94                  | 0.53     | 0.79     | 0.91   | 0.68              | 0.79                 | 0.76  | 0.91         |
|                |            | recall     | 0.88                  | 0.58     | 0.85     | 0.84   | 0.62              | 0.71                 | 0.83  | 0.98         |
|                |            | f1-score   | 0.91                  | 0.55     | 0.82     | 0.87   | 0.65              | 0.75                 | 0.79  | 0.94         |
|                |            | OA         | 0.84                  |          |          |        |                    |                       |       |              |
|                |            | k          | 0.81                  |          |          |        |                    |                       |       |              |
| 9              | Parameter  | Ntree      | default (100)         |          |          |        |                    |                       |       |              |
|                |            | Mtry       | 100                   |          |          |        |                    |                       |       |              |
|                |            | precision  | 0.94                  | 0.43     | 0.78     | 0.90   | 0.69              | 0.78                 | 0.76  | 0.89         |
|                |            | recall     | 0.84                  | 0.58     | 0.87     | 0.83   | 0.59              | 0.67                 | 0.83  | 0.97         |
|                |            | f1-score   | 0.89                  | 0.49     | 0.82     | 0.86   | 0.63              | 0.72                 | 0.79  | 0.93         |
|                |            | OA         | 0.83                  |          |          |        |                    |                       |       |              |
|                |            | k          | 0.79                  |          |          |        |                    |                       |       |              |

Table 5. Accuracy of the various SVM models.

| Experiment No. | Input Data | Class Name | Bare/Sparse Vegetation | Built-Up | Cropland | Forest | Herbaceous Wetland | Herbaceous Vegetation | Shrub | Water Bodies |
|----------------|------------|------------|------------------------|----------|----------|--------|-------------------|-----------------------|-------|--------------|
|                |            | Class ID   | 1                      | 2        | 3        | 4      | 5                  | 6                     | 7     | 8            |
|                |            | No. of Pixels | 6424                 | 575      | 4246     | 3101   | 602               | 3641                 | 5283  | 3795         |
| 1              | Parameter  | cost       | 300 default           |          |          |        |                    |                       |       |              |
|                |            | gamma      | Default (scale (1/(n_features * X.var())) as value of gamma, where X.var is variance) |          |          |        |                    |                       |       |              |
|                |            | precision  | 0.94                  | 0.78     | 0.78     | 0.87   | 0.80              | 0.76                 | 0.75  | 0.97         |
|                |            | recall     | 0.96                  | 0.44     | 0.89     | 0.84   | 0.59              | 0.64                 | 0.81  | 0.97         |
|                |            | f1-score   | 0.95                  | 0.56     | 0.83     | 0.85   | 0.68              | 0.70                 | 0.77  | 0.97         |
|                |            | OA         |                       |          |          |        |                    |                       |       |              |
|                |            | k          | 0.85                  |          |          |        |                    |                       |       |              |
| 2              | Parameter  | cost       | 100 default           |          |          |        |                    |                       |       |              |
|                |            | gamma      | Default (1)           |          |          |        |                    |                       |       |              |
|                |            | precision  | 0.94                  | 0.78     | 0.78     | 0.87   | 0.74              | 0.76                 | 0.75  | 0.97         |
|                |            | recall     | 0.95                  | 0.48     | 0.89     | 0.83   | 0.65              | 0.63                 | 0.80  | 0.98         |
|                |            | f1-score   | 0.95                  | 0.60     | 0.83     | 0.85   | 0.69              | 0.69                 | 0.77  | 0.97         |
|                |            | OA         |                       |          |          |        |                    |                       |       |              |
|                |            | k          | 0.85                  |          |          |        |                    |                       |       |              |
| 3              | Parameter  | cost       | 600 default           |          |          |        |                    |                       |       |              |
|                |            | gamma      | Default (1)           |          |          |        |                    |                       |       |              |
|                |            | precision  | 0.95                  | 0.73     | 0.78     | 0.86   | 0.72              | 0.78                 | 0.74  | 0.96         |
|                |            | recall     | 0.95                  | 0.54     | 0.89     | 0.83   | 0.69              | 0.62                 | 0.79  | 0.98         |
|                |            | f1-score   | 0.95                  | 0.62     | 0.83     | 0.84   | 0.71              | 0.69                 | 0.76  | 0.97         |
|                |            | OA         |                       |          |          |        |                    |                       |       |              |
|                |            | k          | 0.84                  |          |          |        |                    |                       |       |              |
| 4              | Parameter  | cost       | 900 default           |          |          |        |                    |                       |       |              |
|                |            | gamma      | Default (1)           |          |          |        |                    |                       |       |              |
|                |            | precision  | 0.95                  | 0.67     | 0.77     | 0.85   | 0.74              | 0.78                 | 0.74  | 0.96         |
|                |            | recall     | 0.93                  | 0.57     | 0.89     | 0.83   | 0.70              | 0.63                 | 0.78  | 0.97         |
|                |            | f1-score   | 0.94                  | 0.62     | 0.82     | 0.84   | 0.72              | 0.69                 | 0.76  | 0.97         |
|                |            | OA         |                       |          |          |        |                    |                       |       |              |
|                |            | k          | 0.84                  |          |          |        |                    |                       |       |              |
| 5              | Parameter  | cost       | 1500 default          |          |          |        |                    |                       |       |              |
|                |            | gamma      | Default (1)           |          |          |        |                    |                       |       |              |
|                |            | precision  | 0.94                  | 0.64     | 0.74     | 0.84   | 0.78              | 0.78                 | 0.73  | 0.96         |
|                |            | recall     | 0.91                  | 0.61     | 0.88     | 0.83   | 0.70              | 0.62                 | 0.77  | 0.97         |
|                |            | f1-score   | 0.93                  | 0.62     | 0.80     | 0.83   | 0.74              | 0.69                 | 0.75  | 0.97         |
|                |            | OA         |                       |          |          |        |                    |                       |       |              |
|                |            | k          | 0.83                  |          |          |        |                    |                       |       |              |
### Table 5. Cont.

| Experiment No. | Input Data | Class Name | Bare/Sparse Vegetation | Built-Up | Cropland | Forest | Herbaceous Wetland | Herbaceous Vegetation | Shrub | Water Bodies |
|----------------|------------|------------|------------------------|----------|----------|--------|--------------------|-----------------------|-------|-------------|
| 8              | Parameter  | cost       | 300                    |          |          |         |                    |                       |       |             |
|                |            | gamma      | 0.24                   | 1.00     | 0.00     | 0.00   | 0.00              | 0.00                  | 0.00  | 0.00        |
|                |            | precision  | 1.00                   | 0.02     | 0.00     | 0.00   | 0.00              | 0.00                  | 0.00  | 0.00        |
|                |            | recall     | 1.00                   | 1.00     | 0.00     | 0.00   | 0.00              | 0.00                  | 0.00  | 0.00        |
|                |            | f1-score   | 0.38                   | 0.04     | 0.01     | 0.00   | 0.00              | 0.00                  | 0.00  | 0.26        |
|                |            | OA         | 0.25                   | 0.03     |          |        |                    |                       |       |             |
| 9              | Parameter  | cost       | 300                    |          |          |         |                    |                       |       |             |
|                |            | gamma      | 0.99                   | 1.00     | 1.00     | 0.00   | 0.00              | 0.00                  | 0.00  | 0.00        |
|                |            | precision  | 0.05                   | 0.02     | 0.01     | 0.00   | 0.00              | 0.00                  | 1.00  | 0.15        |
|                |            | recall     | 0.00                   | 0.00     | 0.00     | 0.00   | 0.00              | 0.00                  | 0.00  | 0.00        |
|                |            | f1-score   | 0.10                   | 0.04     | 0.02     | 0.00   | 0.00              | 0.00                  | 0.00  | 0.26        |
|                |            | OA         | 0.22                   | 0.04     |          |        |                    |                       |       |             |
| 10             | Parameter  | cost       | 300                    |          |          |         |                    |                       |       |             |
|                |            | gamma      | 0.94                   | 0.61     | 0.78     | 0.88   | 0.77              | 0.80                  | 0.78  | 0.80        |
|                |            | precision  | 0.85                   | 0.71     | 0.83     | 0.82   | 0.71              | 0.72                  | 0.78  | 0.99        |
|                |            | recall     | 0.89                   | 0.65     | 0.80     | 0.84   | 0.74              | 0.75                  | 0.78  | 0.88        |
|                |            | f1-score   | 0.82                   |          |          |        |                    |                       |       |             |
|                |            | Overall accuracy |              |          |          |        |                    |                       |       |             |
|                |            | Kappa      | 0.79                   |          |          |        |                    |                       |       |             |
| 11             | Parameter  | cost       | 300                    |          |          |         |                    |                       |       |             |
|                |            | gamma      | 0.95                   | 0.73     | 0.78     | 0.86   | 0.72              | 0.78                  | 0.74  | 0.96        |
|                |            | precision  | 0.95                   | 0.54     | 0.89     | 0.83   | 0.69              | 0.62                  | 0.79  | 0.98        |
|                |            | recall     | 0.95                   | 0.62     | 0.83     | 0.84   | 0.71              | 0.69                  | 0.76  | 0.97        |
|                |            | f1-score   | 0.84                   |          |          |        |                    |                       |       |             |
|                |            | OA         | 0.84                   |          |          |        |                    |                       |       |             |
|                |            | k          | 0.81                   |          |          |        |                    |                       |       |             |
| 12             | Parameter  | cost       | 300                    |          |          |         |                    |                       |       |             |
|                |            | gamma      | 0.92                   | 0.85     | 0.66     | 0.78   | 0.92              | 0.67                  | 0.74  | 0.96        |
|                |            | precision  | 0.96                   | 0.15     | 0.84     | 0.84   | 0.24              | 0.47                  | 0.77  | 0.97        |
|                |            | recall     | 0.94                   | 0.26     | 0.74     | 0.81   | 0.39              | 0.55                  | 0.76  | 0.97        |
|                |            | f1-score   | 0.80                   |          |          |        |                    |                       |       |             |
|                |            | OA         | 0.80                   |          |          |        |                    |                       |       |             |
|                |            | k          | 0.76                   |          |          |        |                    |                       |       |             |

In terms of individual class accuracy, the RF model classified four classes (Built-up, Forest, Herbaceous vegetation, and Shrub) with higher users’ accuracy and F1-score than the respective best SVM model, but the two best models of RF and SVM produced high and similar user’s accuracy and F1-score for two cover types (Bare/Sparse vegetation, water body). Only on one occasion, the best SVM model generated slightly better user’s accuracy and F1-score, i.e., in classifying herbaceous wetland (Table 4 exp. 1 and Table 5 exp. 4). The best SVM algorithm attained an overall accuracy (0.84) and a kappa value of (0.81) that are 2% and less than the results achieved by the best RF classifier. Consequently, the two best models generated slightly different land cover maps (see Figure 4). Moreover, the map generated by SVM looks a bit noisy, which is due to the presence of more confused classes, a common phenomenon in course resolution imageries, as is exhibited by the confusion matrix (Figure 5).
Figure 4. Land cover generated by the best RF model (a) and the best SVM model (b).

Figure 5. Cont.
5. Discussion

5.1. Effectiveness and Efficiency of the Algorithms

As stated in the Section 4, the RF model classified four land cover types (Built-up, Forest, Herbaceous vegetation, and Shrub) with better accuracy than SVM. However, both algorithms performed almost equally on two occasions, i.e., separating two classes, bare/sparse vegetation, and water body, with high accuracy (greater 0.90) (see Table 4 exp. 1, Table 5 exp. 4, and Figure 5). This indicates that both perform almost equally when the classes are relatively pure with distinct spectral variation, i.e., consisting of fewer mixed pixels. However, the RF is more effective when classes comprise mixed pixels as in the case of the four classes. In this regard, Mountrakis, Im, and Ogole [13] also pointed out that the performance of SVM is significantly affected by the occurrence of mixed pixels and/or wrongly labeled training samples even in relatively small quantities, i.e., it is more sensitive to noisy data than other algorithms.

Moreover, the RF algorithm is computationally less expensive, and faster to train the model and to perform prediction/classification, which is conformable with the finding of Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and Rigol-Sanchez [23]. When the RF was trained, using the best parameter values, it took 35 min less than the best SVM model, and it required only two hours to predict/classify the unknowns; that is much less time than the SVM consumed, i.e., 56 h (more than two days). In addition, the RF is quite fast to carry out modal filtering, the removal of salt and paper effects, during post-processing.

In terms of memory consumption, RF requires lesser storage space, i.e., less memory consumption, during the classification operation. Especially, RF is more effective in handling big input datasets. For instance, the RF model effectively classified and generated a land cover map of the study area when we experimented with a larger dataset that consisted of a full years’ worth of data comprising 396 composite images. However, the SVM model failed to classify/generate a land cover map of the study area using the same one-year data; it told us, “cannot allocate memory, insufficient memory”. All these attributes, therefore, suggest that RF is a more robust ML algorithm than SVM.

This computational efficiency and capacity to classify big datasets or the main reason for robustness of the RF algorithm could fundamentally be related to the ways/logics that decisions are made, i.e., the RF uses simple conditional codes to generate the trees/forest
that decide where a feature belongs on the basis the majority vote. In other words, the algorithm does not require complex mathematical operations, i.e., computation of large dimensional matrices, unlike SVM. The SVM, on the other hand, involves repetitive/iterative computations involving multiplication of large dimensional matrices (pixel values, i.e., spectral values), which demand huge computational time and storage space, to define the best hyperplanes that separate the different classes.

5.2. Impact of Parameter Tuning

Tables 4 and 5 show the performance of both algorithm various with the magnitudes of the parameters. Although the impact of these parameters on the accuracy is considerable, the level of importance of each parameter is notably different. For instance, in the case RF, the step-by-step increment of Ntree from 100 (default value), 300, 500, 700, to 1000 by fixing the Mtry at the default value (auto) of the software, which is the square root of the variables, has little impact on the overall accuracy, kappa score, and individual class accuracy (see Table 4. exp. 1–4). This implies the default value is already optimum and increasing the number of trees has no significance. Likewise, changing the Mtry parameter from the default value (approximately 20) to lower or higher values (10, 40, 100, 200) while maintaining the Ntree at the best value (100), decreases the overall accuracy, kappa, users’ accuracy, and F1-score (see, Table 4. exp. 5–9). Especially, when higher Mtry values (100, 200) were used, the accuracy and kappa values declined by 1–2% and 2–3%, respectively, relative to the result found when the default value was used. That proves that the influence of the Mtry parameter is stronger compared to the impact of Ntree on accuracy as reported by Ghosh, Fassnacht, Joshi, and Koch [22] and Kulkarni and Sinha [57].

In addition to affecting the accuracy negatively, using a higher Mtry value took a longer time to train the model, i.e., computationally expensive, the same effect as increasing the number of trees, which is conformable with the observation made by Belgiu and Drăguț [21]. Therefore, the results demonstrated that setting both Ntree and Mtry values to default, i.e., 100 and auto (sqrt of the number of variables) as it is given in sklearn library, yielded the best RF model, and hence this model was used to produce the land cover map of the study area (Figure 4).

The number of trees (number of estimators = 100) used in our model is less than the values recommended by Belgiu and Drăguț [21] and Gislason, Benediktsson, and Sveinsson [58], who suggested 500 as a default value, and an even higher value [59]. In our case, however, growing several trees greater than the optimum value (100) had no/little impact on the results.

Regarding the Mtry, which is a more important parameter than Ntree, several researchers recommended setting its value to the square root of the number of variables yields a better output, e.g., Belgiu and Drăguț [21] and Gislason, Benediktsson, and Sveinsson [58], which accords with our findings.

Similarly, it was observed that the magnitude of the impacts of the two most important parameters of RBF kernel SVM, C, and $\gamma$, on the accuracy, are different (Table 5). Systematically altering the value of C from default value (1) as given in the platform, to higher values (50, 100, 300, 600, 900, 1200) by maintaining the gamma ($\gamma$) value to default, i.e., scale = (1/(n_features * X.var())), affected the performance of the model in two ways. In the first phase, Table 5, exp. 1 to 3, the increment of C from 1 to 300 generally improves the accuracy. Although the maximum overall accuracy (0.85) and kappa value (0.81) were obtained when C = 100, we opted to use C = 300 because of two reasons. The first reason is that using C = 300 instead of C = 100 had no impact on kappa value but minor decrement (1%) in overall accuracy, which is not as significant as compared to the benefit of setting C = 300. That is, using C = 300 greatly improves the user’s accuracy and f-score of the minority classes, classes that cover a small area and are represented by fewer pixels, such as a built-up and herbaceous wetland. The other reason is that further step-by-step increment of its value from 600 to 1200 decreases the accuracy. For instance, when we used C = 1200,
the overall accuracy and kappa score decreased by 1% and 2%, respectively, relative to the result obtained when $C = 300$ was used; although, it improves the accuracy of the minor classes at the expense of the other. Moreover, using larger $C$ values could lead to overfitting and compromise the capacity of the model to generalize [19], and a moderate $C$ value is recommended to manage the trade-off [53].

In a number of earlier works, however, we have observed that the values of $C$ considered had a great variation ranging from 0 to $10^8$ (e.g., Yang [53]; Raczko and Zagajewski [33]; Huang, Davis, and Townshend [26]; and Qian, Zhou, Yan, Li, and Han [61]) that could be related to the type and amount of input data. To mention a few, Yang [53] tested values ranging between 0 and 300 using 50 steps and reported that moderate $C$ value (100) resulted in the best accuracy in classifying eight land cover classes by the manipulating Landsat-5 Thematic Mapper (TM) scene. In the contrary, Qian, Zhou, Yan, Li, and Han [61] obtained $C = 1,000,000$ as the optimum value after evaluating a range of values ($10^{-1}$ to $10^9$) to generate a local land cover map using a very high-resolution imageries of WorldView-2.

In this regard, our results uncovered that a variation in $C$ value has a minor impact on the overall accuracy and Kappa value, but has a considerable effect on certain individual class accuracy, particularly the rare classes. This could be due to the quality of the training data, i.e., less overlap among the different land cover types and/or the choice of the correct value for $\gamma$ parameter Foody and Mathur [19]. According to Qian, Zhou, Yan, Li, and Han [61], if an inappropriate value of gamma is chosen, no values of $C$ can give us the intended result. This implies how detrimental parameter $\gamma$ is. Our result also revealed that $\gamma$ is the most essential parameter of the SVM model. As exhibited in Table 5, exp. 8 and 9, where $\gamma$ is set to $10^{-3}$ and $10^{-6}$, the performance of the model is very poor despite setting the optimum $C$ value (300).

Further testing with lesser $\gamma$ magnitudes ($10^{-8}$, $10^{-10}$, $10^{-12}$), while $C$ is fixed to 300 somewhat improved the performance of the model, but the result is not as good as the accuracy of the model generated when $\gamma$ was set to the default value (scale). Moreover, if the magnitude of $\gamma$ is too small, the model becomes highly constrained and unable to capture the complexity of the datasets [60]. Therefore, we selected and employed the default $\gamma$ value.

Searching for the optimum parameter, also known as parameter tuning or optimization, using the trial-and-error technique is somewhat challenging and requires systematic selection of the parameter value based on the result from the earlier tests to determine the next value.

However, the other option to find more fine-tuned parameters is employing the GridSearchCV in the sklearn library (https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html, accessed on 25 December 2021) [61]; however, for this study, we skipped using it because it became computationally so expensive. Hence, we concluded that it is infeasible to implement in a scenario where large datasets are involved and a decent computer, specifically with a capacity of the processor: Intel(R) Core(TM) i3-7020U CPU @ 2.30 GHz 2.30 GHz; RAM: 16.0 GB) as in our case, is used. In any case, applying the GridSearch method could give us more optimized parameters as compared to the best result we obtained after testing ranges of values. However, the result is less likely to bring significant differences that affect our overall interpretations as the values we considered for the trial-and-error experiments were selected systematically and cover wide ranges; hence, it is highly likely that the fine-tuned parameters obtained by the GridSearch will fall in the ranges.

Finally, in this finding, the agreement of three parameters with the default values out of four indicates the quality of our training data and the effectiveness of the default parameter values that are given in the sklearn library for remote sensing image classification without much effort to optimize the parameters. Moreover, our result disclosed that these parameter values hugely rely on the type input datasets and the quality of training data; especially, in the case of SVM it leads to incorrect results to assume a fixed magnitude, i.e.,
C between 1,000,000 and 100,000,000, and gamma between 0.00001 and 0.001, as suggested by Qian, Zhou, Yan, Li, and Han [61].

6. Conclusions

In this paper, we compared the performance of the RF and SVM algorithms, the two most commonly used and effective machine learning algorithms, to produce regional land cover maps using a big dataset of coarse resolution images, unlike the previous studies in which the comparisons were made by employing either high or medium resolution imageries for local scale map generation. The study area covers about one-third of Africa, and FY-3C composite imageries were employed as principal input datasets. We systematically selected 9 months 10-day composite images, 297 images metrics, out of one-year data, and stacked them to form a single image composed of eleven different bands including the NDVI where each scene has 1 km spatial resolution. Reference data were collected across the study area, based on the ecological, preexisting maps and personal experience, from atmospherically corrected, Landsat 8 collection 1 level 2 imageries by combining three techniques: Landsat image interpretation, continues consultation of Copernicus global land cover map with 100 m resolution, and Google Earth/Map. Then, the reference data were randomly split into two independent datasets, shapefiles, which contain 91,000 pixels of training data and 27,000 pixels of test data. To determine optimum values of the two most important parameters of each classifier, i.e., Ntree and Mtry for RF and C and γ for SVM, and to find the best model of each algorithm we tested several models by considering ranges of values of the parameters including the default values as defined in the sklearn library of the Python platform. We found that Ntree = 100 and Mtry (‘max_features’) = ‘auto’ (square root of variables) as the best parameter values of RF; in other words, the best model of RF was obtained when both parameters were set to the default values as defined in the sklearn library of Python platform. In the case of SVM, the best model of SVM, which gives the best result, was found when penalty/cost value (C) is equal to 300 and gamma (γ) is set to default, i.e., scale (1/(n_features * X.var())).

Comparison of the two best models, we found out that RF is the superior algorithm provided higher overall accuracy, kappa value, and individual accuracy (user’s, producer’s, and f1 accuracy). For large area/regional land cover mapping involving big input datasets of coarse resolution imageries, RF outperformed SVM. In addition, RF is effective in classifying mixed classes such as built-up, forest, herbaceous vegetation, and shrub resulting in a smoother LC map indicating less mixing up among classes, the SVM generated a slightly noisy product due to more confusion among classes. Furthermore, SVM is computational and memory expensive; especially, when the size of the input data is so large the algorithm fails to allocate memory and is unable to generate the land cover product as it runs out of memory. That could be due to the technique and mathematical equation being used by the algorithm to determine the optimum boundary. In other words, when the study area becomes vast, and big input dataset is used, the algorithm needs to perform several large dimension matrix multiplications to obtain the best hyperplane that demands considerable space and time, unlike RF, where the decision is made quite fast based on a majority vote with no iterative matrix multiplication involved.

Hence, SVM is less feasible for heterogeneous, large area LC mapping using big dataset of coarse resolution imageries that contain more mixed pixels than higher resolution images; however, the computational speed issue can be addressed if cloud computing (accessibility is an issue) or a graphical processing unit (GPU), which is incompatible with sklearn library, is employed.

Conducting parameter optimization by using the GridSearchCV module in the sklearn Python library could somewhat improve the performance of both algorithms; however, the method is too computationally expensive to implement in big dataset scenarios. Moreover, even if we could succeed using it, there would only be little/no fundamental impact on our findings because we tested a wide range of values, and hence it is very less likely for these values to be out of the range of values that could be picked by the GridSearch method.
Finally, the majority of the default parameter values given in the sklearn library of the Python platform, regarding the two algorithms under consideration, work very well for satellite image classification with a little or no adjustment.

**Author Contributions:** Conceptualization, T.A., W.X. and J.F.; Data curation, T.A., W.X. and J.F.; Formal analysis, T.A.; Investigation, T.A., W.X.; Methodology, T.A.; Project administration, T.A., W.X. and J.F.; Resources, T.A., W.X. and J.F.; Software, T.A.; Supervision, W.X.; Validation, T.A. and W.X.; Visualization, T.A. and W.X.; Writing—original draft, T.A.; Writing—review & editing, T.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** FY3C/VIRR data can be requested from the CMA NSMC website (http://satellite.nsmc.org.cn/, accessed on 19 September 2021) and Landsat 8 imageries can be obtained from (https://earthexplorer.usgs.gov, accessed on 20 October 2020). Copernicus global discrete land cover map at 100 m resolution (CGLS LC100 discrete map) for the year 2019 can be acquired from (https://lcviewer.vito.be/download, accessed on 20 October 2020).

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Beaubien, J.; Cihlar, J.; Simard, G.; Latifovic, R. Land cover from multiple thematic mapper scenes using a new enhancement-classification methodology. *J. Geophys. Res. Atmos.* 1999, 104, 27909–27920. [CrossRef]
2. Latifovic, R.; Othlof, I. Accuracy assessment using sub-pixel fractional error matrices of global land cover products derived from satellite data. *Remote Sens. Environ.* 2004, 90, 153–165. [CrossRef]
3. Tchuenté, A.T.K.; Roujean, J.-L.; De Jong, S.M. Comparison and relative quality assessment of the GLC2000, GLOBCOVER, MODIS and ECOCLIMAP land cover datasets at the African continental scale. *Int. J. Appl. Earth Obs. Geoinf.* 2011, 13, 207–219. [CrossRef]
4. Cihlar, J. Land cover mapping of large areas from satellites: Status and research priorities. *Int. J. Remote Sens.* 2000, 21, 1093–1114. [CrossRef]
5. Feng, M.; Li, X. Land cover mapping toward finer scales. *Sci. Bull.* 2020, 65, 1604–1606. [CrossRef]
6. Giri, C.P. *Remote Sensing of Land Use and Land Cover: Principles and Applications*; CRC Press: Boca Raton, FL, USA, 2012; p. 477.
7. Friedl, M.A.; Sulla-Menashe, D.; Tan, B.; Schneider, A.; Ramankutty, N.; Sibley, A.; Huang, X. MODIS collection 5 global land cover: Algorithm refinements and characterization of new datasets. *Remote Sens. Environ.* 2010, 114, 168–182. [CrossRef]
8. Mayaux, P.; Bartholomé, E.; Fritz, S.; Belward, A. A new land-cover map of Africa for the year 2000. *J. Biogeogr.* 2004, 31, 861–877. [CrossRef]
9. Tchuenté, A.T.K.; De Jong, S.; Roujean, J.-L.; Favier, C.; Mering, C. Classification of African ecosystems at 1 km resolution using multiannual SPOT/VEGETATION data and a hybrid clustering approach. In Proceedings of the Les Satellites Grand Champ Pour le Suivi de L’environnement, des Ressources Naturelles et des Risques, Clermont-Ferrand, France, 21–22 January 2010.
10. Loveland, T.R.; Belward, A. The IGBP-DIS global 1 km land cover dataset, DISCover: First results. *Int. J. Remote Sens.* 1997, 18, 3289–3295. [CrossRef]
11. Hansen, M.C.; Defries, R.S.; Townshend, J.R.; Sohlberg, R. Global land cover classification at 1 km spatial resolution using a classification tree approach. *Int. J. Remote Sens.* 2000, 21, 1331–1364. [CrossRef]
12. Arino, O.; Bicheron, P.; Achard, F.; Latham, J.; Witt, R.; Weber, J.-L. The most detailed portrait of Earth. *Eur. Space Agency* 2008, 136, 25–31.
13. Mountrakis, G.; Im, J.; Ogole, C. Support vector machines in remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* 2011, 66, 247–259. [CrossRef]
14. Ghimire, B.; Rogan, J.; Galiano, V.R.; Panday, P.; Neeti, N. An evaluation of bagging, boosting, and random forests for land-cover classification in Cape Cod, Massachusetts, USA. *Gisience Remote Sens.* 2012, 49, 623–643. [CrossRef]
15. Campbell, J.B.; Wynne, R.H. *Introduction to Remote Sensing*; Guilford Press: New York, NY, USA, 2011.
16. Wulder, M.A.; Coops, N.C.; Roy, D.P.; White, J.C.; Hermosilla, T. Land cover 2.0. *Int. J. Remote Sens.* 2018, 39, 4254–4284. [CrossRef]
17. Waske, B.; van der Linden, S.; Benediktsson, J.A.; Rabe, A.; Hostert, P.; Sensing, R. Sensitivity of support vector machines to random feature selection in classification of hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* 2010, 48, 2880–2889. [CrossRef]
18. 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]
19. Foody, G.M.; Mathur, A. A relative evaluation of multiclass image classification by support vector machines. *IEEE Trans. Geosci. Remote Sens.* 2004, 42, 1335–1343. [CrossRef]
48. Han, X.; Yang, J.; Tang, S.; Han, Y. Vegetation products derived from Fengyun-3D medium resolution spectral imager-II. *J. Meteorol. Res.* **2020**, *34*, 775–785. [CrossRef]
49. Wang, L.; Hu, X.; Chen, L.; He, L. Consistent calibration of VIIRR reflective solar channels onboard FY-3A, FY-3B, and FY-3C using a multisite calibration method. *Remote Sens.* **2018**, *10*, 1336. [CrossRef]
50. Jensen, J.R.; Lulla, K. Introductory digital image processing: A remote sensing perspective. *Geocarto Int.* **1987**, *2*, 65. [CrossRef]
51. Colditz, R.R. An evaluation of different training sample allocation schemes for discrete and continuous land cover classification using decision tree-based algorithms. *Remote Sens.* **2015**, *7*, 9655–9681. [CrossRef]
52. Hansen, M.C.; Reed, B. A comparison of the IGBP DISCover and University of Maryland 1 km global land cover products. *Int. J. Remote Sens.* **2000**, *21*, 1365–1373. [CrossRef]
53. Yang, X. Parameterizing support vector machines for land cover classification. *Photogramm. Eng. Remote Sens.* **2011**, *77*, 27–37. [CrossRef]
54. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
55. Tso, B.; Mather, P. *Classification Methods for Remotely Sensed Data*; CRC Press: Boca Raton, FL, USA, 2009; p. 367.
56. Liaw, A.; Wiener, M. Classification and regression by random Forest. *R News* **2002**, *2*, 18–22.
57. Kulkarni, V.Y.; Sinha, P.K. Pruning of random forest classifiers: A survey and future directions. In Proceedings of the International Conference on Data Science & Engineering (ICDSE), Cochin, India, 18–20 July 2012; IEEE: Manhattan, NY, USA, 2012.
58. Gislason, P.O.; Benediktsson, J.A.; Sveinsson, J.R. Random forests for land cover classification. *Pattern Recognit. Lett.* **2006**, *27*, 294–300. [CrossRef]
59. Guan, H.; Li, J.; Chapman, M.; Deng, F.; Ji, Z.; Yang, X. Integration of orthoimagery and lidar data for object-based urban thematic mapping using random forests. *Int. J. Remote Sens.* **2013**, *34*, 5166–5186. [CrossRef]
60. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
61. Qian, Y.; Zhou, W.; Yan, J.; Li, W.; Han, L. Comparing machine learning classifiers for object-based land cover classification using very high-resolution imagery. *Remote Sens.* **2015**, *7*, 153–168. [CrossRef]
62. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
63. Shi, D.; Yang, X. Support vector machines for land cover mapping from remote sensor imagery. In *Monitoring and Modeling of Global Changes: A Geomatics Perspective*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 265–279.