Google Earth Engine: Cloud Computing Environment for Land Use Land Cover Classification

SURAJ SAWANT[1,*], JAYANTA KUMAR GHOSH[2] , JINIT TEJAS SANGHVI[3]

[1] Research Scholar, Indian Institute of Technology, Roorkee, India

[2] Faculty of Geomatics Engineering, Indian Institute of Technology, Roorkee, India

[3]Undergrad Student, College of Engineering, Pune, India

*Correspondence: suraj.t.sawant@gmail.com

ORCIDs:

SURAJ SAWANT: https://orcid.org/0000-0001-8526-5734

Abstract: Land Use Land Cover (LULC) Classification has found its utility in multiple areas, from planning, disaster management, an ecosystem to tracking how landforms are changing due to the human activities. Its utility makes it an attractive application to choose from the domain. We can employ machine learning algorithms for this task since we have access to large geo-spatial data sets and high compute power through cloud computing environments. Over the previous many years, even though an enormous number of artificially intelligent classifiers are being developed to improve the exactness and unwavering quality of pixel-wise classification, there is a scope to identify better classifier particularly for LULC analysis. This study deals with the assessment and comparison between three different highly used machine learning algorithms, namely, Classification And Regression Trees (CART), Support Vector Machines (SVM), and Random Forest (RF). The LULC classification is performed on the landscape of Maharashtra's state in India as it covers several classes of land cover and has very undulating terrain. Using Sentinel-2 Imagery provided by Google Earth Engine (GEE) cloud computing platform, the algorithms are fine-tuned and trained to

obtain the best results with Random Forests performing 99.76% (Overall Accuracy), followed by Support Vector Machines 98.55% (Overall Accuracy) and CART 98.05% (Overall Accuracy). RF outperforms the other two mentioned algorithms and classifies most of the individual classes with stability, less computing time and simplicity in tuning the parameters for the selected study area. To summarize, the RF algorithm can be considered as one of the top choices when LULC is concerned.

Key words: Cloud Computing Environment, Google Earth Engine, Sentinel-2, Land Use Land Cover classification, Machine Learning, Accuracy Assessment

1. Introduction

Geospatial Intelligence is a highly interdisciplinary domain involving the acquisition, analysis, and data generation related to geographic features and locations [Council et al. (2003); Wu et al. (2016); Goyal et al. (2020)]. This domain incorporates Big Data, Machine Learning, Computer Vision, Deep Learning, etc. It is estimated that 80% of the data produced daily is geographic in nature, and this domain aims to extract information from this big data acquired every day [Sivarajah et al. (2017); Kong et al. (2020)]. This data is used for multiple applications such as GPS, remote sensing, and geofencing. Geospatial Intelligence tries to understand events and changes about a location by using this data. Geospatial Technology has been further powered by Machine Learning, and Deep Learning techniques as scientific communities now have access to extensive geospatial datasets [Tohidi and Rustamov (2020)]. This field finds itself a multitude of applications in various domains such as urban planning, agricultural monitoring, crisis management, etc. Geospatial technology is being widely used for several purposes such as military advancement, social development, industrial development, etc. and its impacts are overarching and pervasive in nature.

Geospatial artificial intelligence (geoAI) has proved to be an arising discipline combining innovations in spatial science, artificial intelligence techniques in machine learning, data mining, and high-performance computing (HPC) to extricate insightful information from geospatial big data [VoPham et al. (2018)]. Geospatial Artificial Intelligence (geoAI) is where spatial data meets artificial intelligence and its domains, such as machine learning and deep learning [Lunga (2019)]. geoAI is exceptionally interdisciplinary, connecting various fields, including computer science and engineering, geospatial science, and statistics. Conventional methods seem to fall short in dealing with the vast expanse of spatial data available. In contrast, deep learning techniques are being able to thrive now that data and compute power are more accessible. One such application of geoAI is Land Cover Classification, which involves spatial data and computer vision. This task involves pixel-wise satellite imagery classification to identify particular land cover types concerning their locations [Thanh et al. (2020)].

Human activities and natural phenomena have changed landscapes, leading to profound effects on surrounding ecosystems and environments [Nilsson and Grelsson (1995)]. These changes can be biophysical or biogeochemical in nature. For sustainable development, we need to recognize and identify these transformations to plan how we conduct human activities that impact a location’s geography [Hopwood et al. (2005)]. Thus, Land Cover Classification is vital to our understanding of landforms and terrains concerning their locations. Using satellite imagery, we can classify landforms according to their type, such as vegetation, crops, residential, etc., and this allows us to understand the physical state of locations better. Hence, our knowledge of landforms can greatly help us in urban and agricultural planning, policy formation, sustainable development, etc. Since we can also monitor such areas, the classification will allow us to see how landforms are changing and to measure the effects of human activities and climate change. Besides, variety of applications, such as forest ecosystems, agroecosystems, grassland ecosystems and aquatic ecosystems, desertification monitoring, forest inventories, and so on, are being carried out on the basis of LULC Classification. Hence, valid and apt LULC Classification becomes necessary to monitor and assess the environment. Because of the fast advancements in the development of remote sensing methods, day by day, increasing numbers of satellite imageries with resolution of high intensity, capacious area-inclusion, and multiband data have given profused crude information to acquire significant spatiotemporal data on Land Cover Classification [Lira Melo de Oliveira Santos et al. (2019)]. As of now, we have limited and crude knowledge about landcover maps because of ground constraints, which can be largely mitigated by geoAI. Pixel wise classification of satellite imagery can help us segment areas of a particular landcover, and this approach can help us in different ways.

LULC classification strategies according to satellite imagery are classified as supervised or unsupervised techniques [Li et al. (2014)]. The previous perceives unclassified data by utilizing qualities found out from the training sets of output classes. All the while, the last doesn’t require prior information of classes before classifying, and the class is assigned to every group of pixels via ocular observation [Ge et al. (2020)]. Now that we have access to bigger datasets and more compute power, supervised algorithms have become easier to use and have proven to be effective and robust [Hansen and Loveland (2012)]. Supervised methods include machine learning algorithms like CART, Support Vector Machines, Random Forests and many more [Maxwell et al. (2018)]. For Land Cover Classification, these algorithms are gaining popularity in the scientific community as they are efficient and effective because they can be trained from scratch and can adapt according to the application. This allows them to perform better than conventional classifiers. These benefits also come with caveats as the complexity of certain Machine Learning Algorithms may cause it to overfit, so it is important to choose the appropriate algorithm and finetune it accordingly.

All of this can be achieved by using Google Earth Engine [Gorelick et al. (2017); Kumar and Mutanga (2018)], an integrated platform providing the ease of access to data, and the convenience of deploying algorithms and applications. Google Earth Engine was designed while keeping scientists and researchers in mind, facilitating the deployment of applications and algorithms without requiring expertise in web development or application development. Google Earth Engine (GEE) is a cloud-based stage for planetary-scale geospatial analysis that offers Google's gigantic computational abilities as a powerful influence for an assortment of high-sway societal issues, including deforestation, dry season, calamities, illness, food scarcity, climate change, and environmental protection [Gorelick et al. (2017)]. Its impact and productivity can be best understood because this integrated platform benefits a huge spectrum of users, even those who do not have access to powerful computational resources. GEE simplifies the task of accessing HPC resources for computing extensive geospatial data-sets without going in to the details of its actual implementations [Sakr and Liu (2012)].

Following the Landsat series's free availability in 2008, GEE has archived all the data sets and connected them to the cloud computing environment for open source use [Li et al. (2019)]. Currently, other satellites and Geographic Information Systems (GIS) based vector data sets, digital elevation models, social, demographic, weather, and climate data layers are available [Mutanga and Kumar (2019)]. Along with the Landsat series, GEE also provides Sentinel 2 imagery of 10m resolution free of cost [Barboza Castillo et al. (2020)], enabling researchers worldwide to use Sentinel 2 data for their work and application.

Main objective of this work is to conduct a comparative analysis of three supervised machine learning classifiers used for pixel-wise classification (LULC) of Sentinel-2 images for Maharashtra State, India. The study aims to consider seven classes for LULC maps and summarize the advancement in geoAI and its application in the development of geospatial tools and systems. Seven classes considered for this study are Urban, Agriculture, Fallow, Barren, Forest, Water, and Wetland. The outcomes from this investigation can give insights into the classifier choice for the LULC analysis of Maharashtra state and other similar regions in the western parts of India. This study will also highlight the application of the cloud computing environment and platform, GEE, which provides scientists and researchers an opportunity to work on extensive computational problems without possessing the expensive hardware configuration free of cost.

2. **Material and Methods**

2.1. Study Area

Maharashtra is an enormous state in India's western central part and the north-western part of the Indian Subcontinent. Study area is shown in Fig.1. It has an expanse of approximately 120 thousand square miles, and the coast of the Indian Ocean form the western borders of the state. To the north, Maharashtra has borders with the states of Gujarat and Madhya Pradesh. The Maharashtra state has one of the longest coastlines of approximately 450 miles long. Maharashtra's central space is occupied by the Deccan plateau, with an abundance of woodlands and outstanding productive soil. A series of the mountain range is located in the southern and the eastern parts of Maharashtra. The rives passing by the state include Godavari, Bhima, Krishna, Tapi-Purna, and others. Maharashtra is located in latitude 19.66° N and longitude 75.30° E. Maharashtra is a key economic contributor and industrial region of the country, and this makes the state one of the wealthiest and the most developed among the other states. Hence, the study area covers ethnicities within the North-West and Central-West regions of the country. The region has a mean temperature of 26.25° C, annual rainfall of 5822 mm in overall Maharashtra and relative humidity of 66% in overall Maharashtra. The study area is full with variety of land cover types, mainly including agriculture land, forest and barren land, which account for very undulating terrain of the entire region.

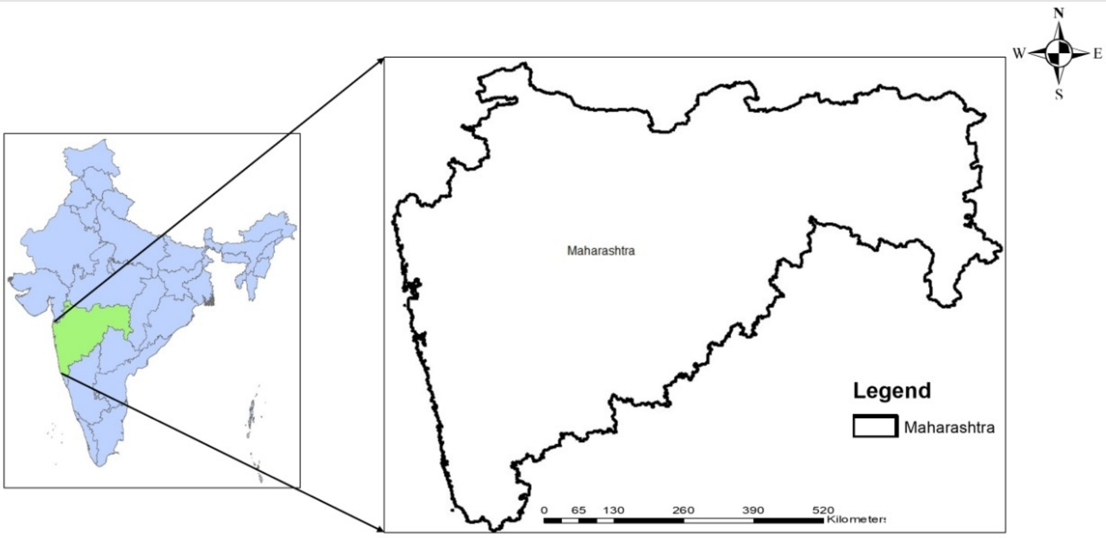

Figure. 1 Study Area.

2.2. **Pre-processing**

Steps of the carried out study is represented in Fig.2 and inspired by [Ge et al. (2020)]. Implementation of code for the Machine Learning Algorithms is done in Javascript of the GEE code editor. Libraries for the said algorithms are available in GEE. The source code may be shared as per the request from the reader.

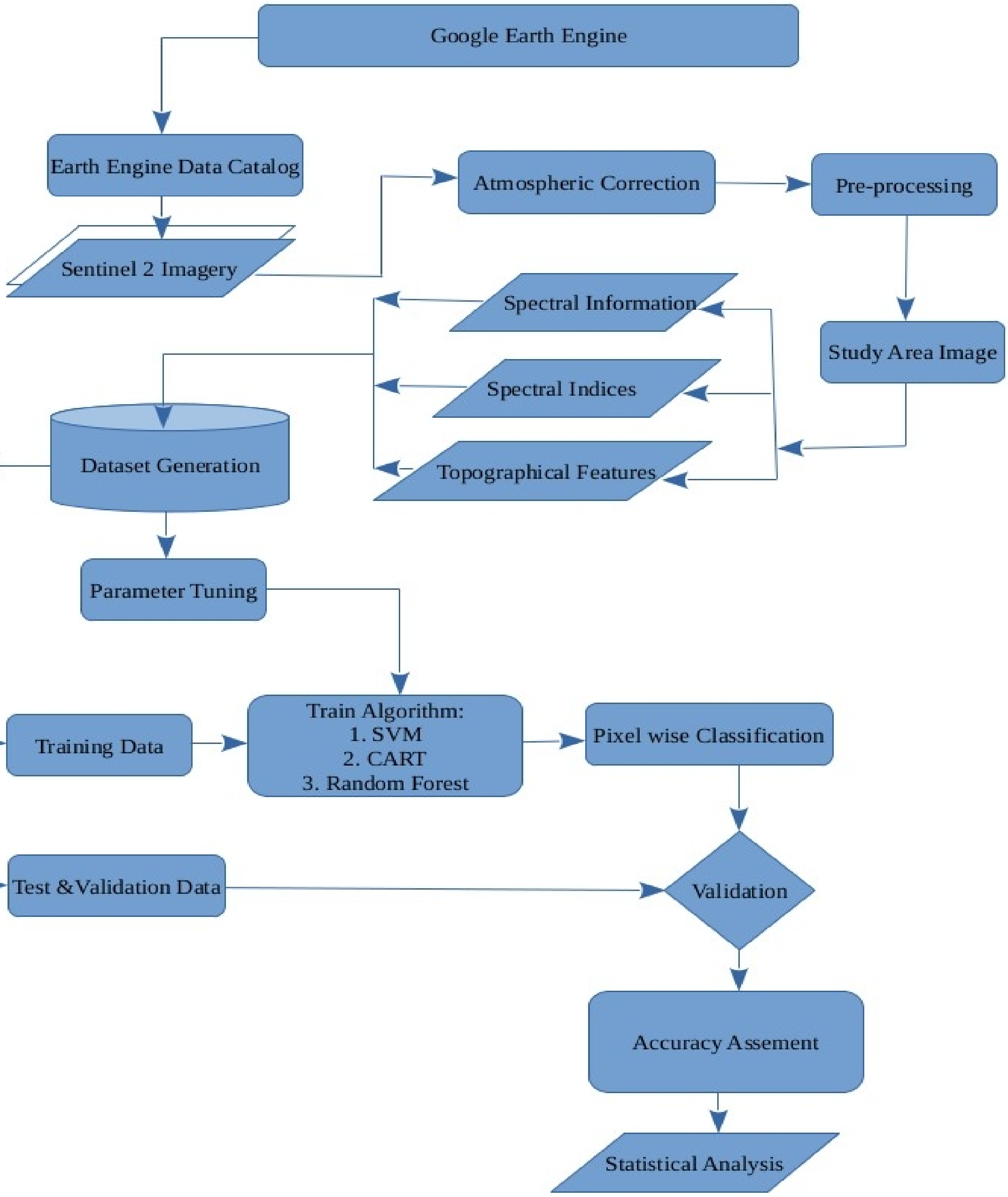

Figure. 2 Workflow Diagram.

2.2.1 Dataset Preparation

Since the launch of Sentinel-2 satellite in June, 2015, Sentinel-2 imagery is extensively used for LULC classification at regional level. In this study, images are considered from the duration of March to May, 2020 as the cloud cover percentage is very low and Vegetation is also clearly visible. The images are downloaded from the GEE Data Catalog. As the images were already geometrically corrected and orthorectified, only atmospheric correction was required to be done. Study area images are being atmospherically corrected using SEN2COR [Main-Knorn et al. (2017)] tool provided by European Space Agency (ESA). Then the images were layer stacked, mosaicked and clipped as per the study area boundary. The processed images are layer stacked with total nine bands, namely Bands 2, 3, 4, 5, 6, 7, 8, 11, and 12. In order to increase the accuracy of the implemented algorithms, spectral indices namely of the normalized difference vegetation index (NDVI) and modified normalized water index (MNDWI) [Han-Qiu (2005)]. These spectral indices are considered to be secondary data for classification. Topographic features such as elevation and slope is considered as input to the dataset. All the spatial layers were transformed to same geographical co-ordinate system, World Geodetic System i.e. WGS-84 and to Universal Tansverse Mercator (UTM) projection co-ordinate system. As the availability of Ground Truth Data for the accuracy assessment of the study area is not available or the LULC maps are available but they are very old, We have considered the Google Earth images. Randomly uniform number of sample pixels are selected from the total number of classes and they are used for classification and accuracy assessment. As per the requirement stated by [Ge et al. (2020)] for LULC classification, all other criteria are being satisfied to best of our knowledge. The classes considered for classification are explained in Table 1.

70% of the Data set is used for training, 15% for testing and remaining 15% for validation.

Class	Description
Urban Land	Urban/ Rural residential , development and construction areas
Agriculture Land	Area used for Agriculture
Fallow Land	Agriculture land but currently not under cultivation
Barren Land	Bare areas which are not being cultivated
Forest Area	Tree covers including forests
Water Bodies	Area where wate is found
Wet Land	Marshy land

Table 1. Class Description.

2.3 Algorithms

2.3.1 Classification and Regression Trees (CART)

CART or Classification and Regression Trees are simple and interpretable models which try to split observations and classify them by taking decisions or satisfying conditions in a hierarchical fashion. CART is a standard based strategy that produces a twofold tree via paired recursive partitions that parts a subset (leaf) of training samples into two classes (sub-leaves) as indicated by the minimization of a heterogeneity basis calculated on the subsequent sub-leaves [Bel et al. (2009)]. This split is made with respect to a particular variable and the splitting is continued till the purity of the split is increasing. By purity, we mean the ability of the partitioning to split observations of distinct classes. In order to measure this purity, we can use the Gini Index [Ceriani and Verme (2012)]. A Gini Index of 0 indicates that the split is perfect and our aim is to minimize the Gini index. The Gini index can be computed by summing of square of probabilities of all classes minus one. CART algorithm is available in GEE in form of library and ee.Classifier.smileCart() is a fuction to invoke the classifier.

2.3.2 Support Vector Machine

A support vector machine (SVM) is a supervised non-parametric classifier that is frequently applied in applications related to remote sensing [Mountrakis et al. (2011); Ge et al. (2020)]. A nonlinearly distinct dataset that comprises of multiple points from two classes can be isolated from those of the other class by utilizing many numbers of hyperplanes, and the best hyperplane with the biggest margin between the two classes is chosen by utilizing a subset of training sets that are known as support vectors. SVM aims to distinguish the target that are classified by the most suitable hyperplane into one of the given classes. Four kernel functions are available in SVM. They are linear, radial basis function (RBF), polynomial, and sigmoid kernels. For our purpose, we identified RBF to work the best as it is powerful for higher dimensions too and is known to fit complex datasets.

SVMs have two hyperparameters, control error (C) and Gamma. C represents the amount of misclassifications we can allow our model to make in order to find a better classifier whereas gamma represents the extent of curvature of the classifier. Optimizing these two hyperparameters is a classic example of the bias-variance tradeoff. To find the optimal values, we use cross-validation along with a grid search for C and Gamma. In this study, we have used ten different values of C and gamma. For more insights on SVM, please refer to [Suthaharan (2016)].

2.3.3 Random Forest

The random forest classifier comprises of a blend of tree classifiers where each classifier is created utilizing a random vector sampled independently from the input vector, and each tree makes a unit choice for the most famous class to classify an input vector [Breiman (2001)]. This approach where we factor in the output of multiple tree classifiers requires these different classifiers to be distinct and uncorrelated. The reason behind is that we need to shield a particular classifier from the errors of the other classifiers, otherwise multiple homogeneous trees will amplify a common error which will crowd out the predictions of trees that don't share this error. In order to ensure, we can use bagging and feature randomness. Bagging is a common ensemble method that uses bootstrap sampling. Since random forest classifiers are sensitive to even the smallest changes in the dataset, bagging exploits this very property to produce uncorrelated decision trees. Bootstrap sampling creates multiple samples with replacement of data and we train different decision trees on different samples. Feature randomness refers to the process of randomly selecting features for partitioning. Random Forest being an ensemble method using decision trees works better for classification tasks and gives us better results for LULC classification too.

2.4 Implementation using Google Earth Engine

Google Earth Engine provides inbuilt packages to perform supervised classification using algorithms such as CART, Support Vector Machines and Random Forest. Using the classifier package, we can easily implement these algorithms by following a series of steps.

1. To start off, first acquire training data and prepare the features to be used for classification. Training data can be acquired from numerous sources, and Google Earth Engine provides a collection of datasets that you can have access to by using packages such as ImageCollection. Using FeatureCollection, you can store all the labels as you'll need them for supervised classification.

2. Once you've prepared your dataset, initialize a classifier of your choice using the Classifier package. smileCart, libSVM and smileRandomForest can be used to access CART, Support Vector Machines, and Random Forest respectively from the Classifier package. Upon choosing an algorithm, you can set hyperparameters for the algorithm and start training.

3. Once trained, one can try out own model on an image by using the classify method. One can also use own validation data to estimate classification error.

2.5 Statistical Analysis

Once the classifiers are applied to the study area, the accuracy is computed with help of Confusion Matrix. In depth information regarding the confusion matrix may be obtained from [Lewis and Brown (2001)]. In this study, three algorithms are compared on the basis of Overall Accuracy (OA) Equation (1), Kappa Coefficient Equation (2), Producer's Accuracy Equation (3), and User's Accuracy, Equation (4). The equations for the metrices are taken as it is from [Ge et al. (2020)]. Equations are mentioned below.

$$Overall\,Accuracy = \frac{1}{N}\sum_{i=1}^{r} X_{ii} \qquad (1)$$

$$Kappa\,Coefficient = N\sum_{i=1}^{r} X_{ii} - \sum_{i-1}^{r} \langle X_{i+.} * X_{+i} \rangle \qquad (2)$$

$$Producer's\,Accuracy\,(Class_i) = X_{ii} / X_{i+.} \qquad (3)$$

$$User's\,Accuracy\,(class_i) = X_{ii} / X_{+i} \qquad (4)$$

Where N: number of observations,

r: number of rows in the matrix,

X_{ii} : numberof observations in row i and column i (i.e. diagonal elements), and

X_{+i} and X_{i+} : marginal total of row (r) and column (i), respectively[Geet al. (2020); Congalton (1991)].

3 Results

LULC classification results are represented in Figure 3. After the stable results are obtained, parameters are set and then the overall accuracy, kappa coefficient, and user's and producer's accuracies are computed to compare the performances of the three algorithms. From Table 2, it is observed that the overall Accuracy and Kappa Coefficient of the three classification algorithms are above 95% and 0.85.

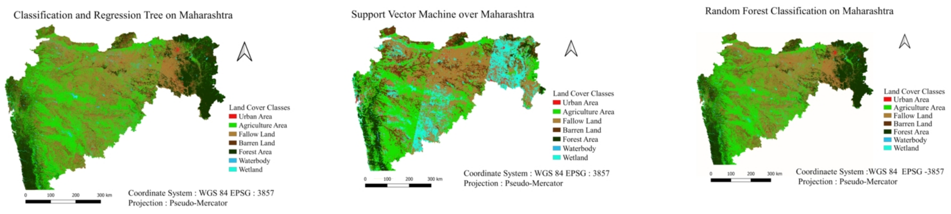

Figure. 3 LULC classification output: CART, SVM and RF.

CART		SVM		RF	
Overall Accuracy	Kappa Coefficient	Overall Accuracy	Kappa Coefficient	Overall Accuracy	Kappa Coefficient
98.05	**0.87**	**98.85**	**0.92**	**99.76**	**0.98**

Table 2. Summary of Overall Accuracy and Kappa Coefficient.

4 Performance Analysis

In terms of accuracy metrics, Random Forest perform the best followed by Support Vector Machines followed by CART. CART's performance can be justified by the fact that it is a non-robust algorithm as it significantly changes even upon the slightest modification of training data. This problem is largely solved by Random Forests and since Random Forests rectify this problem, it outperforms CART and gives more consistent results. To compare the performance of Random Forests and Support Vector

Machines, we need to understand when these algorithms perform well. Support Vector Machines perform well when number of features outnumber observations but tend to fail when the training data is too vast. Random Forests is an ensemble method which can generalize well to large datasets as it combines numerous decision trees, making it robust and relatively error free. It also reduces overfitting by pruning. Thus, Random Forests outperform Support Vector Machines. Confusion Matrices for three classifiers is shown in Figure. 4. After comparing the OA of SVM and RF, it is clear that, finetuning the SVM parameters may result into getting very near to the OA obtained by RF. After various iterations, it can be said that, there is clear difference between the performance of the three implemented algorithms for the study area. RF proved to be the best among the three algorithms, SVM the second best and CART being the third one.

For all the classifiers, misclassifications are mainly because of commission and omission errors because of the land cover types forest, agriculture and urban. Agriculture and forest shows almost similar spectral properties, which makes difficult to classify correctly. Elevation data seems to be very useful for land cover types When individual land cover type classification accuracy is concerned, such as barren land, urban area and fallow land as they have spatial distributions, conditioned by their relief [Ge et al. (2020); Rodriguez-Galiano and Chica-Rivas (2014)].

		Truth data								
		Urban	Agriculture	Fallow	Barren	Forest	Water	Wetland	Classification Overall	Producer's Accuracy (Precision)
CART Result	Urban	1022	1	1	0	0	0	0	1024	99.81
	Agriculture	1	1944	15	4	23	0	0	1981	97.84
	Fallow	1	19	1610	7	11	0	0	1648	97.7
	Barren	0	2	11	126	19	0	0	158	79.75
	Forest	2	14	12	4	60233	2	0	60267	99.95
	Water	0	3	1	0	0	86	3	93	92.47
	Wetland	1	1	1	0	0	3	57	63	90.48
	Truth Overall	1027	1984	1651	141	60286	91	60	65238	
	User's Accuracy (Recall)	99.51	97.98	97.52	84.36	99.91	94.51	95		

(a)

		Truth data								
		Urban	Agriculture	Fallow	Barren	Forest	Water	Wetland	Classification Overall	Producer's Accuracy (Precision)
SVM Result	Urban	799	1	16	29	19	73	57	994	80.38
	Agriculture	0	1830	10	73	13	25	27	1978	92.52
	Fallow	0	16	1529	0	13	33	37	1628	93.97
	Barren	5	2	9	106	11	11	42	186	56.99
	Forest	3	7	9	4	60089	44	28	60184	99.84
	Water	4	0	0	3	15	88	12	122	72.13
	Wetland	2	38	1	3	29	23	50	146	34.24
	Truth Overall	813	1894	1574	218	60189	297	253	65238	
	User's Accuracy (Recall)	98.28	96.62	97.14	48.62	99.84	29.63	19.76		

(b)

Random Forest Result	Truth data								
	Urban	Agriculture	Fallow	Barren	Forest	Water	Wetland	Classification Overall	Producer's Accuracy (Precision)
Urban	1022	1	1	0	0	0	0	1024	99.81
Agriculture	1	1944	15	4	23	0	0	1987	97.84
Fallow	1	19	1610	7	11	0	0	1648	97.69
Barren	0	2	11	126	19	0	0	158	79.75
Forest	2	14	12	4	60233	2	0	60267	99.95
Water	0	3	1	0	0	86	3	93	92.47
Wetland	1	1	1	0	0	3	57	63	90.48
Truth Overall	1027	1984	1651	141	60286	91	60	65238	
User's Accuracy (Recall)	99.51	97.98	97.52	89.36	99.91	94.51	95		

(c)

Figure 4. Confusion Matrix of (a) CART (b) Support Vector Machines and (c) Random Forest.

5 Conclusion

In this study, we have compared and evaluated namely three machine learning algorithms, CART, Support Vector Machine and Random Forest, for the land use land cover classification of the state of Maharashtra in India. In order to achieve the best and most comparative results, the hyperparameters of these algorithms are finetuned. It is observed that Random Forest performs the best at 99.76% Overall Accuracy and a Kappa Coefficient of 0.98, followed by Support Vector Machine at 98.85% Overall Accuracy and 0.92 Kappa Coefficient and lastly CART at 98.05% Overall Accuracy and 0.87 Kappa Coefficient. All three algorithms perform exceptionally well on classes such as Urban and Forest, as these land covers are very distinct and hence easy to classify. However, CART and Support Vector Machines do not perform well for Barren Land class and Wetland class while Random Forests perform well for these classes too, causing the better performance achieved by Random Forests. Apart from the barren class and wetland class, all algorithms perform well when tuned properly, making these algorithms a good fit for the task.

References

Council NR, et al. (2003) IT roadmap to a geospatial future. National Academies Press

Wu J, Guo S, Li J, Zeng D (2016) Big data meet green challenges: Big data toward green applications. IEEE Systems Journal 10(3):888–900

Goyal MK, Sharma A, Surampalli RY (2020) Remote sensing and gis applications in sustainability. Sustainability: Fundamentals and Applications pp 605–626

Sivarajah U, Kamal MM, Irani Z, Weerakkody V (2017) Critical analysis of big data challenges and analytical methods. Journal of Business Research 70:263–286

Kong L, Liu Z, Wu J (2020) A systematic review of big data-based urban sustainability research: State-of-the-science and future directions. Journal of Cleaner Production p 123142

Tohidi N, Rustamov RB (2020) A review of the machine learning in gis for megacities application. Geographic Information Systems in Geospatial Intelligence

VoPham T, Hart JE, Laden F, Chiang YY (2018) Emerging trends in geospatial artificial intelligence (geoai): potential applications for environmental epidemiology. Environmental Health 17(1):40

Lunga DD (2019) Artificial intelligence. Geographic Information Science & Technology Body of Knowledge 2019(Q4)

Thanh HNT, Doan TM, Tomppo E, McRoberts RE (2020) Land use/land cover mapping using multitemporal sentinel-2 imagery and four classification methods—a case study from dak nong, vietnam. Remote Sensing 12(9):1367

Nilsson C, Grelsson G (1995) The fragility of ecosystems: a review. Journal of Applied Ecology pp 677–692

Hopwood B, Mellor M, O’Brien G (2005) Sustainable development: mapping different approaches. Sustainable development 13(1):38–52

Lira Melo de Oliveira Santos C, Augusto Camargo Lamparelli R, Kelly Dantas Araújo Figueiredo G, Dupuy S, Boury J, Luciano ACdS, Torres RdS, Le Maire G (2019) Classification of crops, pastures, and tree plantations along the season with multi-sensor image time series in a subtropical agricultural region. Remote Sensing 11(3):334Li C, Wang J, Wang L, Hu L, Gong P (2014) Comparison of classification algorithms and training sample sizes in urban land classification with landsat thematic mapper imagery. Remote sensing 6(2):964–983

Ge G, Shi Z, Zhu Y, Yang X, Hao Y (2020) Land use/cover classification in an arid desert-oasis mosaic landscape of china using remote sensed imagery: Performance assessment of four machine learning algorithms. Global Ecology and Conservation 22:e00971

Hansen MC, Loveland TR (2012) A review of large area monitoring of land cover change using landsat data. Remote sensing of Environment 122:66–74

Maxwell AE, Warner TA, Fang F (2018) Implementation of machine-learning classification in remote sensing: An applied review. International Journal of Remote Sensing 39(9):2784–2817

Gorelick N, Hancher M, Dixon M, Ilyushchenko S, Thau D, Moore R (2017) Google earth engine: Planetary-scale geospatial analysis for everyone. Remote sensing of Environment 202:18–27

Kumar L, Mutanga O (2018) Google earth engine applications since inception: Usage, trends, and potential. Remote Sensing 10(10):1509

Sakr S, Liu A (2012) Large scale data management techniques in cloud computing platforms. Data-Intensive Computing: Architectures, Algorithms, and Applications p 8

Li H, Wan W, Fang Y, Zhu S, Chen X, Liu B, Hong Y (2019) A google earth engine enabled software for efficiently generating high-quality user-ready landsat mosaic images. Environmental Modelling & Software 112:16–22

Mutanga O, Kumar L (2019) Google earth engine applications

Barboza Castillo E, Turpo Cayo EY, de Almeida CM, Salas López R, Rojas Briceño NB, Silva López JO, Barrena Gurbillón MÁ, Oliva M, Espinoza- Villar R (2020) Monitoring wildfires in the northeastern peruvian amazon using landsat-8 and sentinel-2 imagery in the gee platform. ISPRS International Journal of Geo-Information 9(10):564

Main-Knorn M, Pflug B, Louis J, Debaecker V, Müller-Wilm U, Gascon F (2017) Sen2cor for sentinel-2. In: Image and Signal Processing for Remote Sensing XXIII, International Society for Optics and Photonics, vol 10427, p 1042704

Han-Qiu X (2005) A study on information extraction of water body with the modified normalized difference water index (mndwi). Journal of remote sensing 5:589–595

Bel L, Allard D, Laurent J, Cheddadi R, Bar-Hen A (2009) Cart algorithm for spatial data: Application to environmental and ecological data. Computational Statistics & Data Analysis 53(8):3082–3093

Ceriani L, Verme P (2012) The origins of the gini index: extracts from variabilità e mutabilità (1912) by corrado gini. The Journal of Economic Inequality 10(3):421–443

Mountrakis G, Im J, Ogole C (2011) Support vector machines in remote sensing: A review. ISPRS Journal of Photogrammetry and Remote Sensing 66(3):247–259

Suthaharan S (2016) Support vector machine. In: Machine learning models and algorithms for big data classification, Springer, pp 207–235

Breiman L (2001) Random forests. Machine learning 45(1):5–32

Lewis H, Brown M (2001) A generalized confusion matrix for assessing area estimates from remotely sensed data. International journal of remote sensing 22(16):3223–3235

Congalton RG (1991) A review of assessing the accuracy of classifications of remotely sensed data. Remote sensing of environment 37(1):35–46 Rodriguez-Galiano VF, Chica-Rivas M (2014) Evaluation of different machine learning methods for land cover mapping of a mediterranean area using multiseasonal landsat images and digital terrain models. International Journal of Digital Earth 7(6):492–509