A PROJECT REPORT

ON

Analysis of Various Land Use/Land Cover patterns over various regions of India using Machine Learning

Computer Science and Engineering

Submitted By:

Saharsh Sharma (2018kucp1063)

Vishay Yadav (2018kucp1134)

Shaurya (2018kucp1136)

Rajvir Singh (2018kucp1138)

Under the guidance of

Dr. Priyanka Mishra

(Assistant Professor, IIIT Kota)

Department of Computer Science and Engineering

ACKNOWLEDGMENTS

We would like to acknowledge Dr. Priyanka Mishra for his guidance during the course of this project work.

We also thank Dr. Amit Kumar and Dr. Anupam Kumar for their continuous support throughout the timeline of the project.

ABSTRACT

As the development of nations such as India is on a growing trajectory, the urbanization is taking place in a very uneven and undistributed manner, because of which profound changes have been observed in the Land Use/Land Cover (LU/LC) patterns. The study and assessment of these LU/LC patterns has become an important aspect in the early development stages from the perspective of the environment as well as the economy. These studies and assessments demand high quality images.

Traditional methods of extracting information through Remote Sensing Images have been a tedious task for the researchers and geologists. But with the help of high-resolution images and a complete world of Machine Learning, the complexity of the solutions has decreased to many-fold. Google Earth Engine (GEE) is considered as one the best tools which can help to achieve the goal.

In this project, we perform LU/LC Pixel-based Classification of Built-Up and Non Built-Up areas over the urban region of Delhi, India, using the Landsat 7 Remote Sensing Satellite data of the Top of Atmosphere (TOA) Composite of the year 2014 in Google Earth Engine (GEE).

1. Introduction

For some years as of now, the world has experienced a rapid rise in Urbanization of areas with changes in LU/LC cover patterns. Urbanization takes place when cities assimilate the near-by rural regions, mostly in terms of sprawl diffusions appearing out from the middle of the city or directly across major corridors of transportation [1], [2].

But these human settlements have proved to be very uneven and undistributed urbanization. Between 1950 and 2014, the urban area share in the population count increased from 30% to 54% and by 2050, this may increase to an enormous amount of 2500 million in Afro-Asia [3]. In order to achieve sustainable development, it is very necessary to look towards the physical, geographic, social, economic and cultural impact of this urbanization on the human settlements.

Recent times have witnessed rapid development in rural and urban areas of India as well. This is mainly due to the Industrial Revolution and the migration to urban areas in hope of access to better facilities and income. Due to an uneven distribution of population, this urbanization has shown undistributed patterns which can lead to misinformation in the Population data and census and other geographical information covering a particular area. With the availability of satellite data and powerful computational platforms like GEE, it has actually been possible to analyze various regions and study their change in urban sprawl. Although many studies in the past have used GEE for identification of urban areas and LU/LC urban analysis, very few researches have been performed over India.

Therefore, to explore the power of GEE for analysis over the urban regions of India, in this study, we perform the pixel-based classification of areas of Delhi into Built-Up and Non Built-Up respectively. First of all, the dataset mentioned in [3] was acquired, and points over the study area of Delhi were obtained and pre-processed using the Landsat 7 Top of Atmosphere (TOA) percentile composite satellite data of the year 2014, with no negative sun elevation. Additionally, NDVI and NDBI bands were added as features to facilitate the task of classification. We use three most prominent classifiers available in GEE, CART, Linear SVM and Random Forests for the classification task, with 10-fold cross validation for assessment.

2. Related Work

Previously, the literature measured the extent of the urbanized areas by door-to-door household surveys and analyzing its data, nightlights and some mobile phone records. But with the advancements in satellite images and the band availability, the urban research has moved digitally towards the development of the Remote Sensing Data and eventually using them for classification purposes. The conventional techniques of obtaining demographic data and exploration of environmental specimens are not sufficient enough for multi-complex ecological analysis [4]. On the other hand, use of Remote Sensing Satellite data along with Geographic Information Systems (GIS) have expanded as one of the essential technologies that can be used to obtain precise and suitable information on various LU/LC patterns over large areas.

With the rapid developments in the analysis of highly-accurate Machine Learning models, increasing availability of Satellite Images has enabled the extraction of features of various LU/LC classes using highly accurate pixel-based classification models.

Hua, Zhang, Chen, Yin and Tang [5] use many ML techniques including Decision Tree Classification (DTC), CART analysis, SVM etc. and found DTC to be the best method out of these. Jiang, Wang, Yang, Xie and Cheng [6] also proposed classification methods based on DTC for Remote Sensing Images. Ha, Tuohy, Irwin and Tuan [7] use the Random Forests Classification model using Landsat satellite data to monitor the LU/LC variation and Urbanization of Rural areas in Vietnam.

Sidhu, Pebesma and Camara [8] use Google Earth Engine (GEE) to evaluate its utility for performance over vector and raster information on GlobCover, Landsat and MODIS imagery. Shetty [9] conducted LU/LC classification using various Machine Learning Classifiers available in GEE over the region of Dehradun, India and consequently analysed their performances.

Goldblatt, You, Hanson and Khandelwal [3] developed a dataset consisting of about 21,000 polygons, classified manually into ‘Built-Up’ and ‘Non Built-Up’ classes, and which can be used for performing pixel-based classification over different areas of India in GEE using Landsat 7 and Landsat 8 satellite data.

Mutanga and Kumaruse [10] use Google Earth Engine (GEE) to evaluate its utility and application like vegetation mapping and monitoring, land cover mapping, disaster management and Earth science, agricultural application etc. Tamiminia, Salehi, Mahdianpari and Quackenbush [11] used Google Earth Engine for geo-big data applications. Tobón- Marín and Barriga [12] analyse the change in river plan forms using Google Earth Engine.

3. Problem Statement

Although more and more Remote Sensing Satellite data is now being available, researchers often face difficulty in analyzing these data with the help of Machine Learning models, because of the requirement of high-computation processing units. But, with the availability of platforms like the Google Earth Engine (GEE), it has become quite easy to perform geo-spatial analysis, over large areas. Therefore, in this paper, we use the power of GEE to perform a study of the Built-Up patterns of Delhi, India.

The aim of this experiment is to classify the different regions of Delhi, India, into Built-Up and Non Built-Up areas, using Machine Learning models on Landsat 7 TOA composite data of the year 2014 in Google Earth Engine.

4. Methodology

Under this section, we lay an outline of the software and classification models used in this study.

4.1 Google Earth Engine (GEE)

Google Earth Engine (GEE) is a data processing platform based on Google Cloud Platform (GCP) [13] and Geographic Information System (GIS) tool used to display maps and synthesize the geographical data and to run geospatial analysis on Google's infrastructure. It is a huge platform with petabytes of satellite images and geospatial data. The data and images on GEE are the contributions of various satellites. Thus, it has a timeline of satellite imagery.

As the complexity in the technology is increasing with time, Google Earth Engine provides numerous kinds of ready-to-use datasets within a timeline of thirty years of historical images, which are updated and expanded on a daily basis. For example, thermal satellite sensors provide information about emissivity and surface temperature for both land and sea from satellites like MODIS, ASTER, and AVHRR. There are many more other datasets that provide information about Climate, Weather, and Atmosphere. Various other satellites too cover the geophysical information for Terrain, Land Cover, Cropland, and other geophysical data. GEE has an enormous humanitarian effect when it comes to carrying out various scientific analyses using it. Various case studies carried out by different international institutes/organizations are present on GEE for further Research and Development practices [14]. These include remote sensing research, predicting disease outbreaks, natural resource management, and many more. For example, Global Forest Watch provides a real-time system for online monitoring of forests worldwide, for their conservation. It is a project developed by the World Resource Institute [14].

GEE comes along with an Integrated Development Environment (IDE) available online at [15]. The Code Editor features have been designed in such a way that it enables carrying out various geo-spatial studies using JavaScript and develop maps using Google Cloud easily. The Earth Engine API, available for use as a Python module, also comes along with the power of Google Cloud Platform (GCP).

4.2 Classification Models

The task of pixel-based classification is performed using three different Machine Learning models available in GEE-

(I) Classification and Regression Tree (CART)

A binary classifier based on Decision Trees.

(II) Support Vector Machine (SVM)

A classifier used to separate the data points into different classes by identifying the boundaries between them. We have used Linear SVM in this project.

(III) Random Forests

A classification model consisting of 'k' decision trees.

4.3 Cross Validation

The probability of a classifier to correctly classify a random set of samples denotes its accuracy [16]. To ensure that the classifier performs fairly, the data is split up into train and test sets, respectively. One such technique, Cross- Validation divides the dataset into k subsets (ideally 5 or 10, in order to have least biased estimation [17]) for k trials. This method makes sure that each subset of data is incorporated once in the test set and is not used for training in that trial. The overall performance is calculated as an average performance of these k trials.

In this experiment, we have used 10-fold Cross Validation by partitioning the data into 10 random stratified folds. In each of the 10 trials, 9 folds are used for training and the remaining fold is used for testing. The aggregate performance is then measured as an average of all the 10 trials.

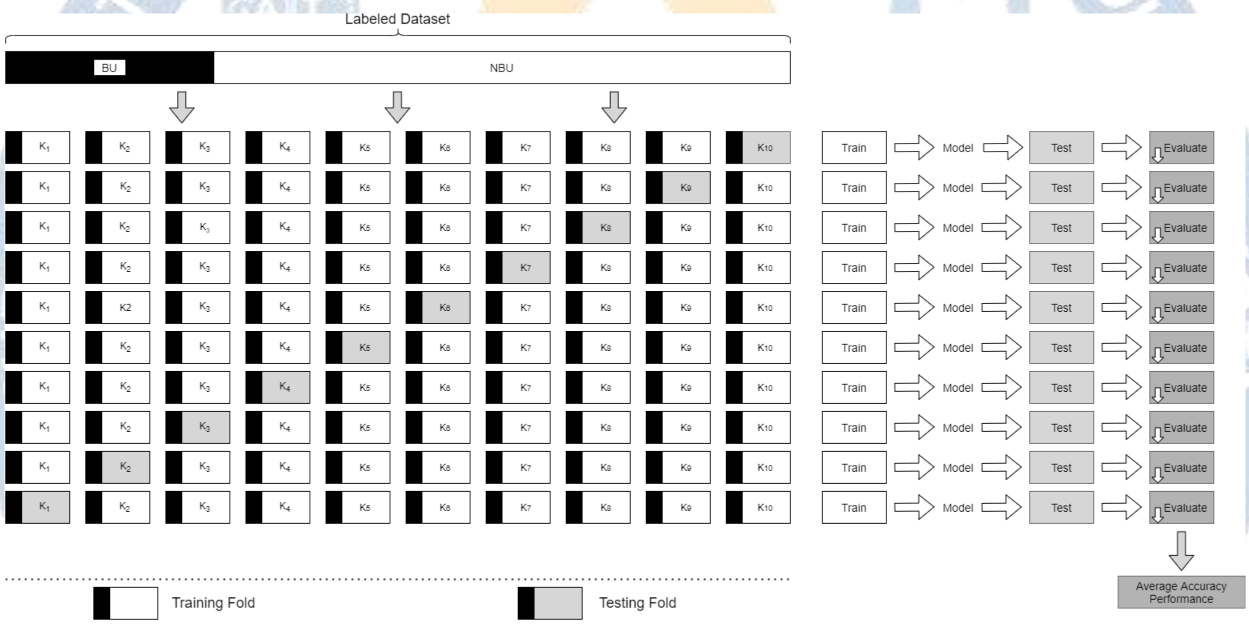


Figure 4.3.1: 10-fold Cross Validation

5. Proposed Method

In this section, we present the proposed method of the experiment. The steps of the proposed method are as shown in Figure 5.1. First, the dataset mentioned in [3] was obtained and was loaded in the Google Earth Engine (GEE). The dataset was filtered to include the data points over the study area. Also, the Landsat 7 Satellite data of the year 2014, with the least cloud cover and no negative sun elevation was loaded in GEE. This was used to create the Top of Atmosphere (TOA) composite image, which was then used to pre-process the data points from the dataset for extracting the per-pixel band values of Landsat 7 as features. Additionally, two bands were added to facilitate the classification task: Normalized Difference Vegetation Index (NDVI) and Normalized Difference Built-Up Index (NDBI), whose values were calculated using the existing bands of Landsat 7 for each data point. Each data point also included an output binary class, representing 'Built-Up' or 'Non Built-Up' respectively.

Once the data points were pre-processed and features were obtained, this set of points obtained was given as input to the three Machine Learning classification models (available in GEE) namely, Classification and Regression Tree (CART) or Decision Tree, Linear SVM and Random Forests. The assessment of the models was done using the technique of Cross Validation with 10 folds. The data was given to Random Forests consisting of 10, 50 and 100 Trees respectively, and out of these the results of the best performing model was then compared with the results of that of CART (Decision Tree) and Linear SVM.

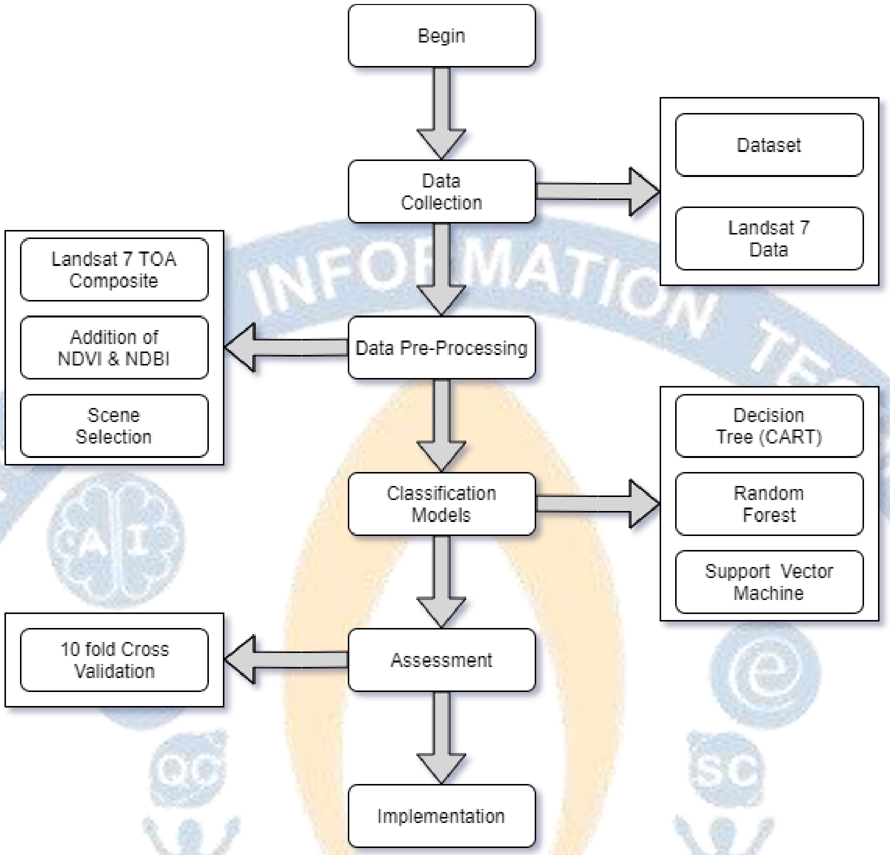


Figure 5.1: Experimental Workflow

6. Experiment

6.1 Study Area

National Capital Territory (NCT) of Delhi is one of the largest metropolitan areas, which consists of New Delhi, the capital city of India [18]. It is located between 28°-24′-17″ N, 76°-50′-24″ E and 28°-53′-00″ N, 77°-20′-37″ E, consisting of an area of about 1484 sq. km. (573 sq. mi.). According to the 2011 census, the total population of NCT of Delhi was about 16.8 million [18]. It is the world's second-largest urban area, as per the United Nations. With the second-highest GDP per capita in India, the NCT of Delhi ranks fifth in the Human Development Index (HDI) among all the states and Union Territories (UTs) of India. It is surrounded by the satellite cities of Gurugram, NOIDA Ghaziabad, Faridabad among others, which has led to an increase in urban migration to the area in the last few years. By 2021, it is expected that the total population of the NCT of Delhi will be 19.5 million. There are 11 districts in the NCT of Delhi, as of 2012.

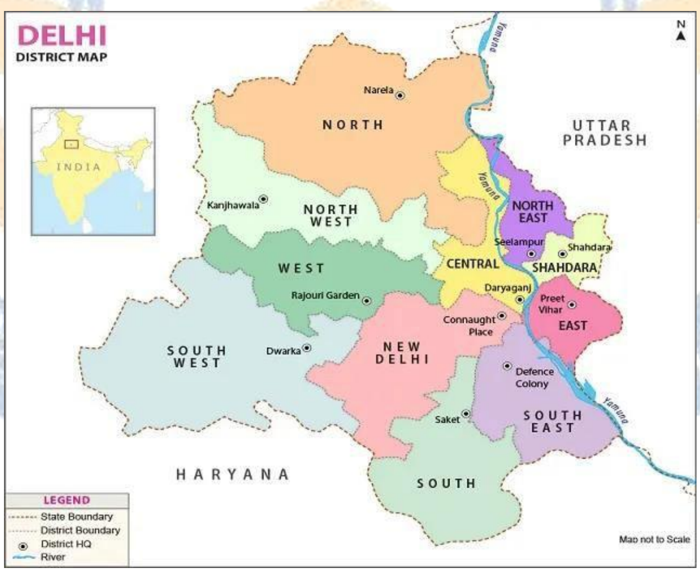


Figure 6.1.1: District Map of Delhi

6.2 Dataset

For this study, the dataset used was as mentioned in [3]. The dataset consists of 21,030 polygons (Geographically mapped areas) of 30 x 30 m size, distributed randomly over mainland India. These polygons have been manually classified into Built-Up (BU) and Non Built-Up (NBU) classes, respectively [3]. The polygons with more than 50% area consisting of man-made structures are marked as Built-Up and the remaining as Non Built-Up. The dataset has been generated using WorldPop [19], a per-pixel population estimation of 2010-2011 India Census dataset to create the sample polygons. The whole procedure of construction of the dataset as mentioned in [3] is shown in Figure 6.2.1. The Satellite View and Map View of the Polygons are shown in Figure 6.2.2. The dataset consists of 286 polygons over our study of area as seen in Figure 6.2.3.

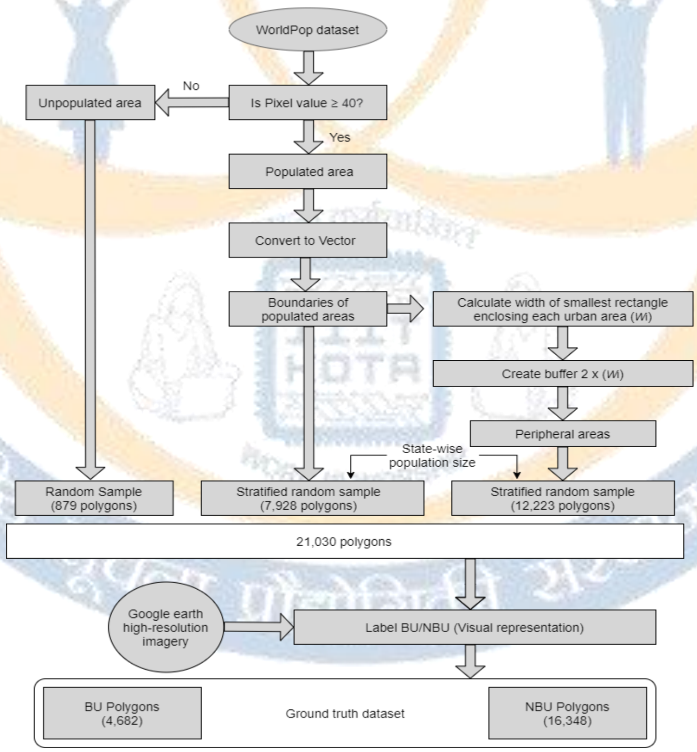


Figure 6.2.1: The procedure used to generate the Dataset

Figure 6.2.2: Polygon representation as viewed in Google Earth Engine
(a) Satellite View: (i) Built-Up example. (ii) Non Built-Up example. (b) Map View

Figure 6.2.3: Satellite View of Built-Up and Non Built-Up Polygons mapped over an area of Delhi (as viewed in Google Earth Engine)

6.3 Pre-processing and Scene Selection

After retrieving the dataset, for scene selection, Landsat 7 Surface Reflectance (TOA-1) images of year 2014 (available in GEE) were used as input for image classification which consists of 8 different bands of different wavelengths (micrometers), as shown in Table 6.3.1. Landsat 7 annual TOA percentile composite (2014), which is a composite of pre-processed scenes, filtered with lowest cloud cover and no negative sun elevation, was given as input feed. The composite consists of pixels, evaluated as percentile estimates of each band scaled to 8 bits (for bands 1 to 5 and band 7 to 8) and in the units of Kelvin-100 (for band 6).

Two additional indices are added as bands to the Landsat 7 data, so that it can improve the classification: Normalized Difference Vegetation Index (NDVI) [20] and Normalized Difference Built-Up Index (NDBI) [21].

(I) NDVI

Normalized Difference Vegetation Index (NDVI) communicates the correlation of the visible red light (which is absorbed by the chlorophyll of plant) and near-infrared wavelength (which is dispersed by the mesophyll of a plant leaf) [22]. It is the most-widely used vegetation index to observe greenery globally. Usually, healthy plants have high reflectance in Near Infrared (NIR) in the range of 0.7 to 1.3 µm. High reflectance for NIR and high absorption for the Red spectrum, helped to estimate the formula for Normalized Difference Vegetation Index (NDVI) [22].

$$NDVI = (NIR - Red) / (NIR + Red)$$

Here, for Landsat 7, NIR falls in Band 4 and Red falls in Band 3 [22]. Therefore,

$$NDVI = (B4 - B3) / (B4 + B3)$$

(II) NDBI

Normalized Difference Built-Up Index communicates the correlation of the medium-infrared (MIR) and near-infrared (NIR) wavelengths. The bare soil and built-up structures reflect more MIR than NIR [22].

$$NDBI = (MIR - NIR) / (MIR + NIR)$$

Here, for Landsat 7, MIR falls in Band 5 and NIR falls in Band 4 [22]. Therefore,

$$NDBI = (B5 - B4) / (B5 + B4)$$

The NDBI value lies in the range of –1 to 1. Higher the value of NDBI, it denotes built-up area and values closer to -1 denote water bodies [22].

Table 6.3.1: Landsat 7 Bands used as features for Classification

Spectral Band Name	Resolution Pixel Size (m)	Wavelength (µm)	Description
B1	30	0.45 - 0.52	Blue
B2	30	0.52 - 0.60	Green
B3	30	0.63 - 0.69	Red
B4	30	0.77 - 0.90	Near Infrared (NIR)
B5	30	1.55 - 1.75	Medium Infrared (MIR) 1
B6_VCID_1	30	10.40 - 12.50	Low-gain Thermal Infrared (TIR)
B6_VCID_2	30	10.40 - 12.50	High-gain Thermal Infrared (TIR)
B7	30	2.08 - 2.35	Medium Infrared (MIR) 2
B8	15	0.52 - 0.90	Panchromatic
NDVI	30		(B4–B3)/(B4+B3)
NDBI	30		(B5–B4)/(B5+B4)

6.4 Detection of Built-Up Regions

In order to detect the Built-Up regions using GEE, first, we overlaid the labeled polygons from the dataset over the region of interest, the NCT of Delhi. We collected all Landsat 7 points within the area of these polygons along with per-band reflectance values and index values. These sampled points differ from the original number of polygons used, as they entirely do not overlap with the pixels of the Landsat data points. Each sample point includes an output binary class: Built-Up or Non-Built Up. This set of points, as seen in Figure 6.4.1 and 6.4.2, is used as input to the different classification models.

Figure 6.4.1: Satellite View of Training Points after being mapped over points of Landsat 7 composite (as viewed in Google Earth Engine)

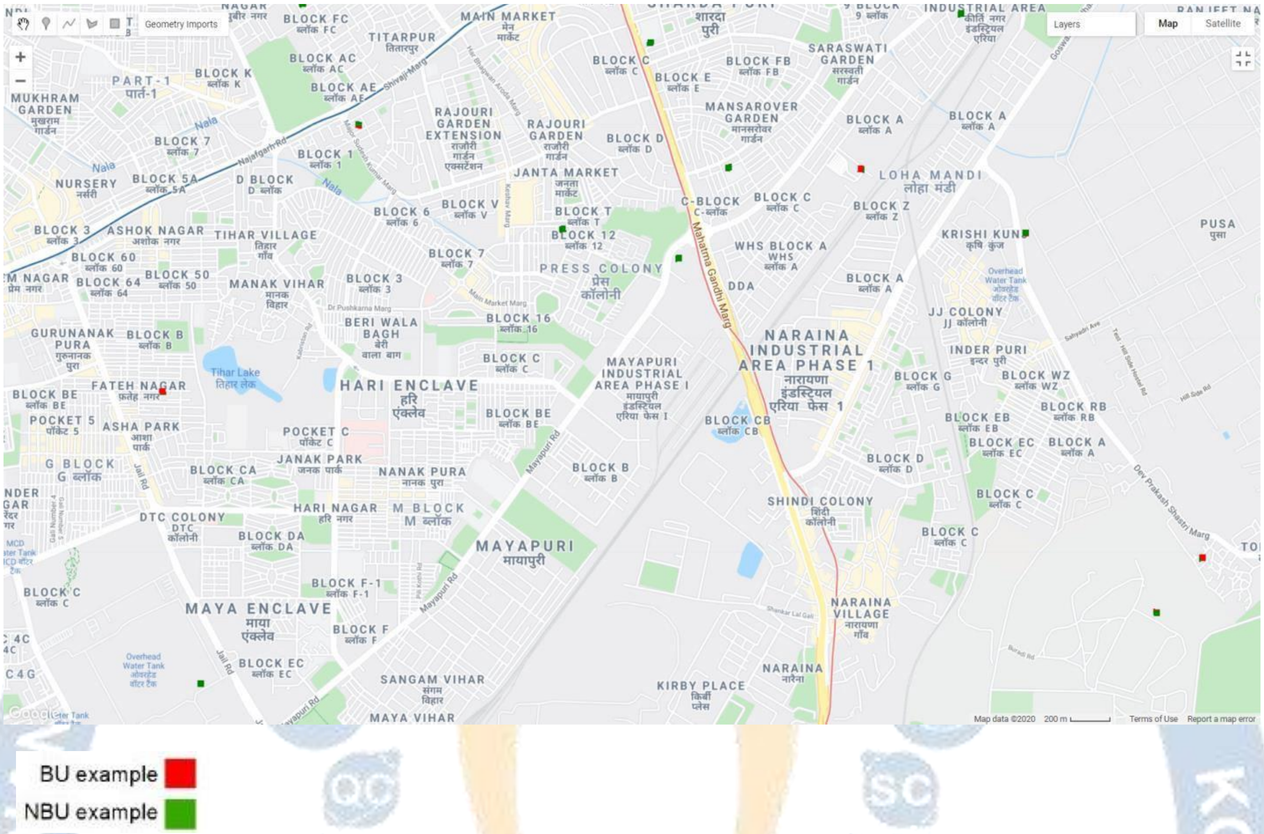


Figure 6.4.2: Map View of Training Points after being mapped over points of Landsat 7 composite (as viewed in Google Earth Engine)

7. Results and Analysis

In GEE, various models are available for pixel-based classification. We compare the results obtained by three most prominent models available: CART, Linear SVM and Random Forests. First, we performed 10-fold cross validation for Random Forests consisting of 10, 50 and 100 trees, and the average testing accuracy achieved was 64.9%, 65.8% and 65% respectively. The highest performance was therefore obtained with 50 Trees, as seen in Figure 7.1 and Table 7.1.

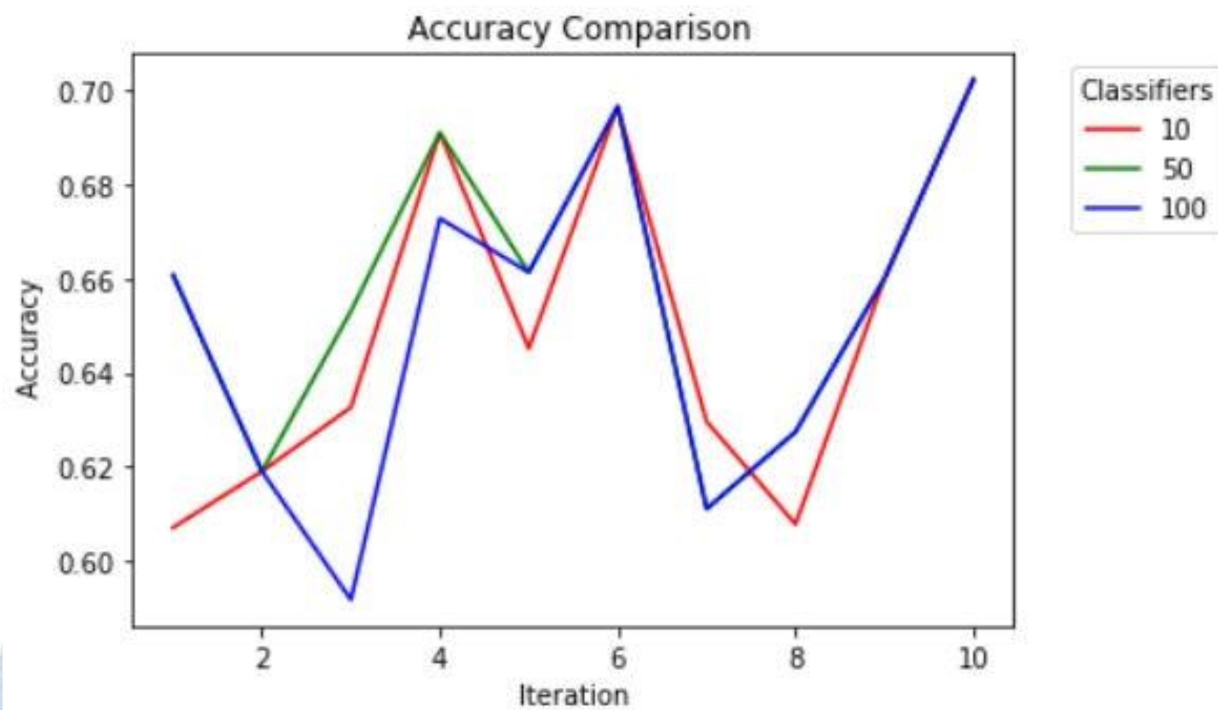


Figure 7.1: Accuracy Comparison Graph between Random Forests consisting of 10, 50 and 100 Trees

Table 7.1: Average Accuracy Comparison between Random Forests consisting of 10, 50 and 100 Trees

Number of Trees in Random Forests	Average Accuracy
10	0.6490942916534208
50	0.6582140865247743
100	0.6502734557270007

We then performed 10-fold cross validation using Linear SVM and CART (Decision Tree), and compared their results with that of Random Forests consisting of 50 Trees, as shown in Figure 7.2 and 7.3, and Table 7.2 and 7.3. The average testing accuracy achieved for CART (Decision Tree) was 65.7%, while the Linear SVM performed with an average accuracy of 63.6%. The Ground Truth and Predicted Output Points for one of the folds for comparison is shown in Figure 7.4, 7.5, 7.6 and 7.7.

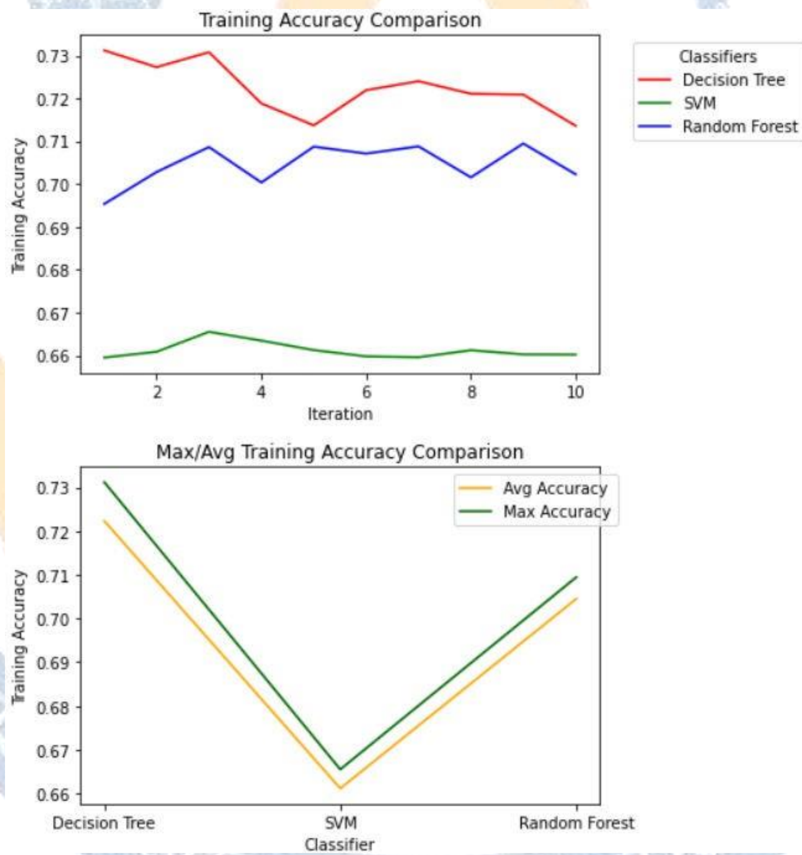


Figure 7.2: Training Accuracy Comparison Graphs between Decision Tree, Linear SVM and Random Forest (consisting of 50 Trees)

Table 7.2: Training Accuracy Comparison between Decision Tree, Linear SVM and Random Forest (consisting of 50 Trees)

Classifier	Average Accuracy	Maxmium Accuracy
Decision Tree	0.7223108322725234	0.7311827956989247
SVM	0.661146076403116	0.6655011655011654
Random Forest	0.7044972999807453	0.7094488188976378

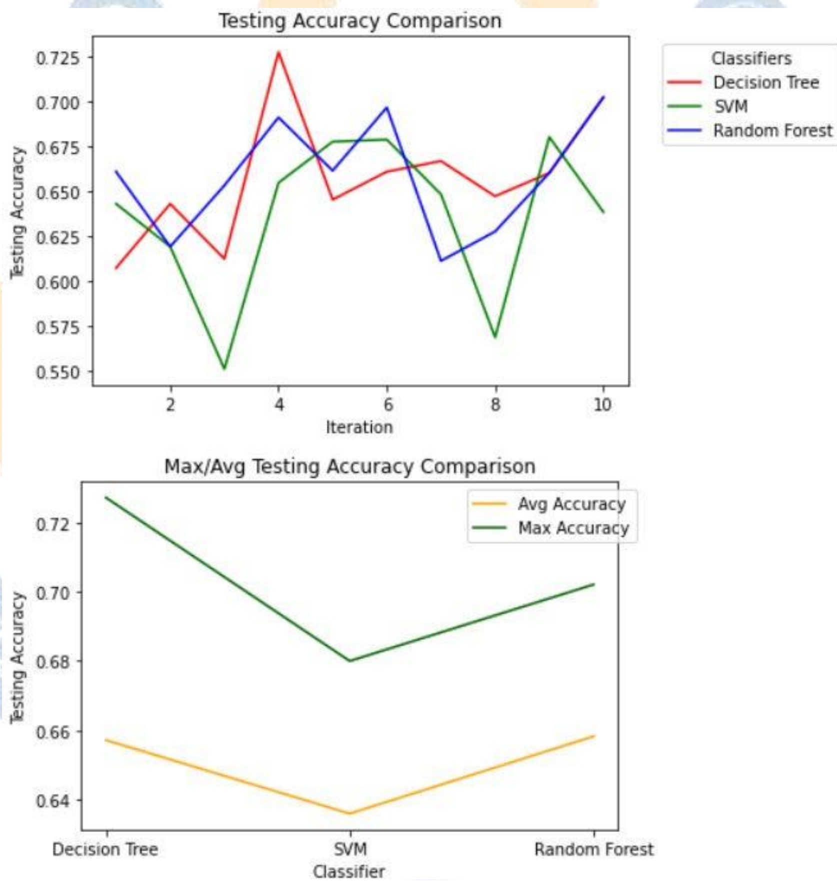
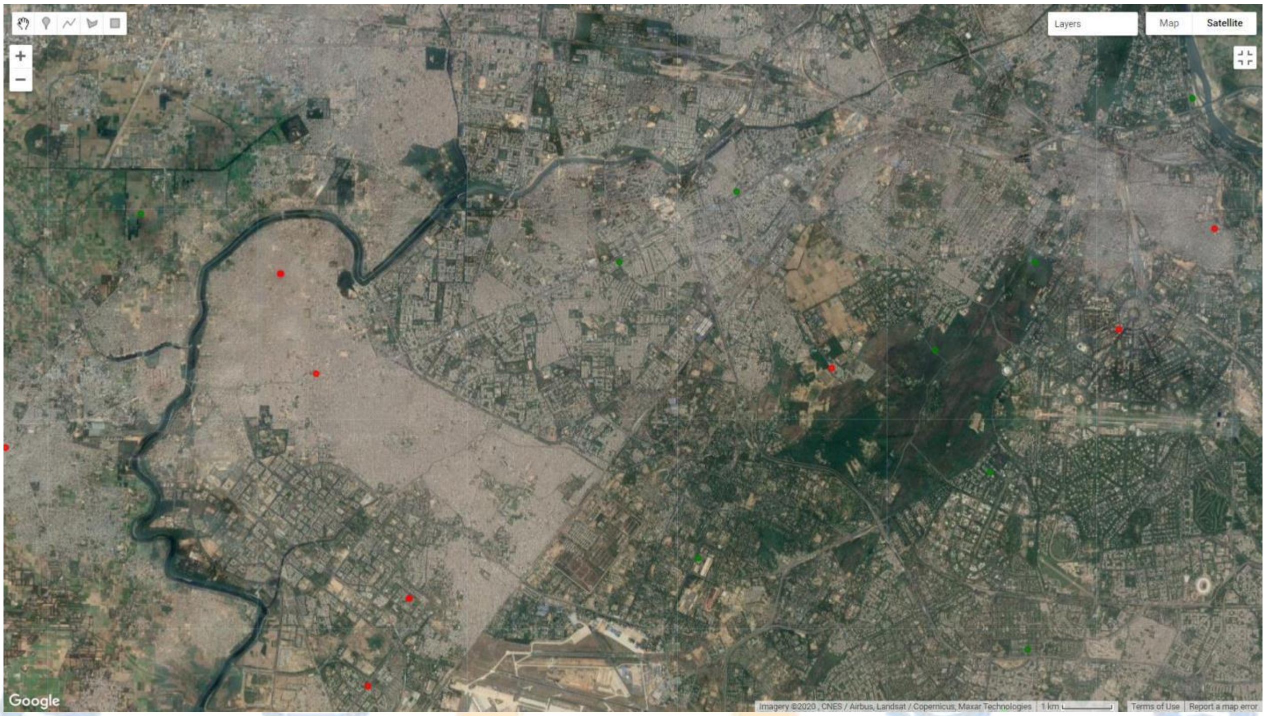


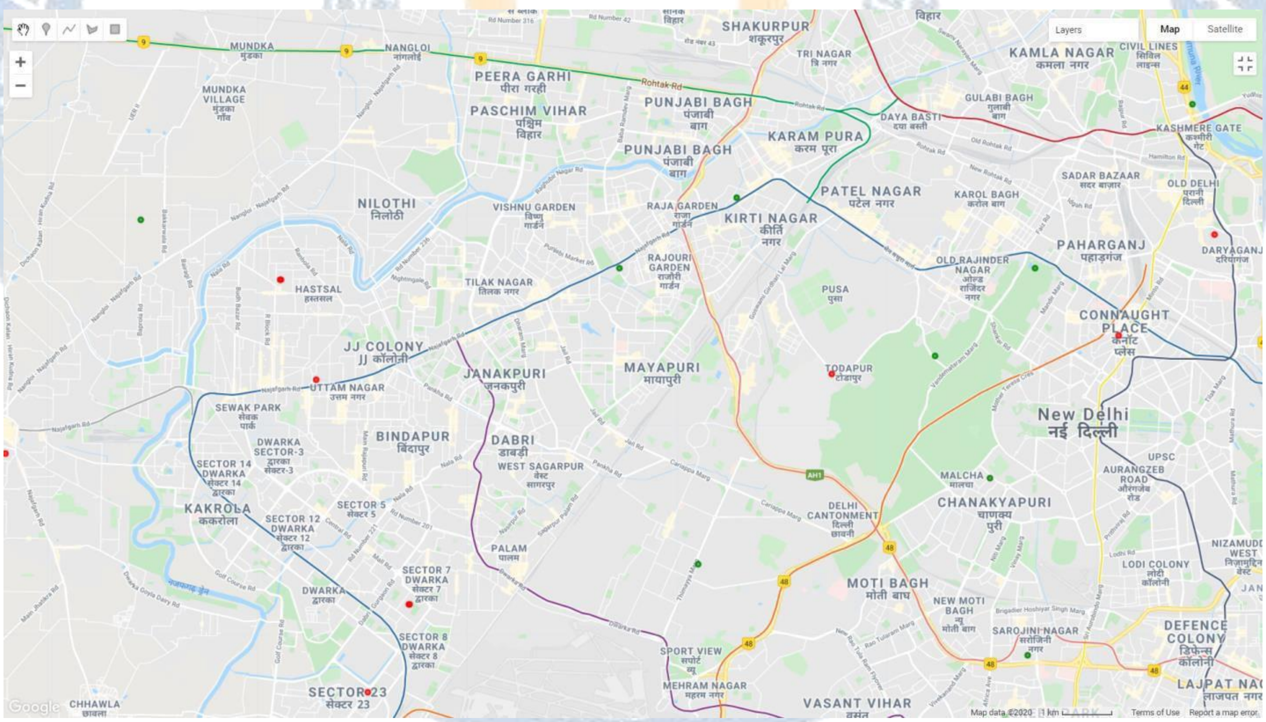
Figure 7.3: Testing Accuracy Comparison Graphs between Decision Tree, Linear SVM and Random Forest (consisting of 50 Trees)

Table 7.3: Testing Accuracy Comparison between Decision Tree, Linear SVM and Random Forest (consisting of 50 Trees)

Classifier	Average Accuracy	Maxmium Accuracy
Decision Tree	0.6571246351039324	0.7272727272727273
SVM	0.6358534879492584	0.68
Random Forest	0.6582140865247743	0.7021276595744681



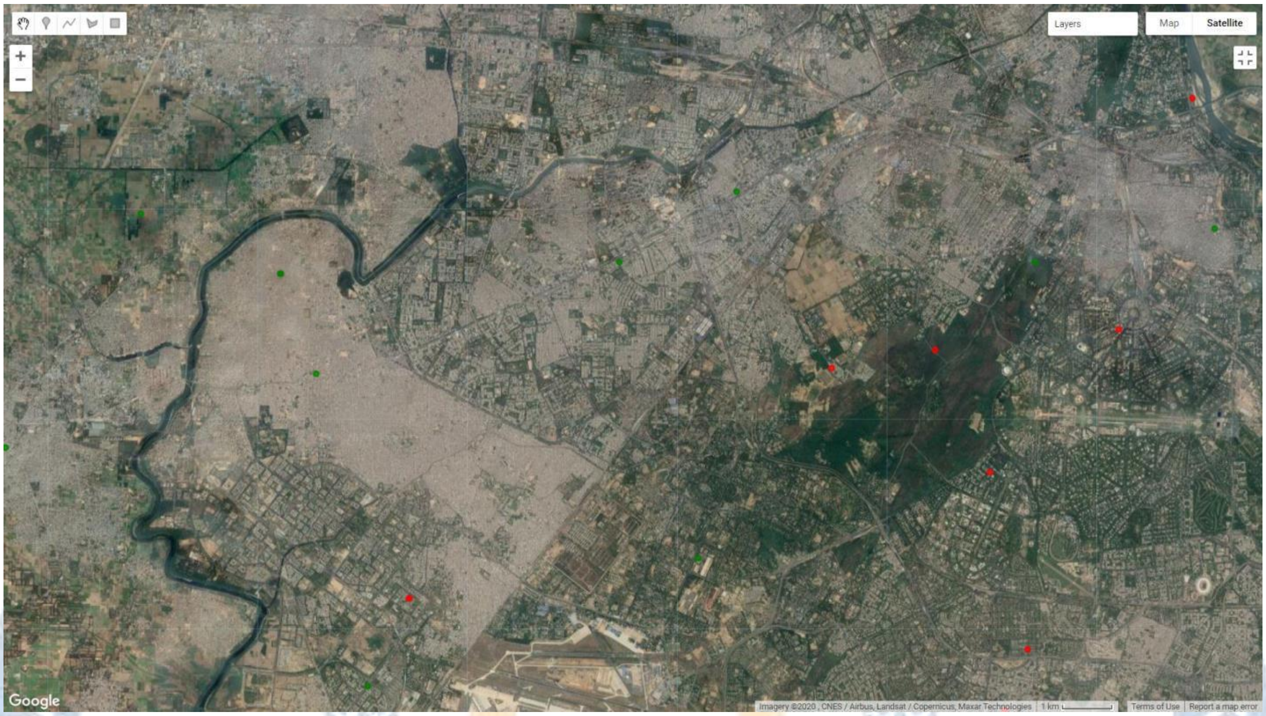
(a)



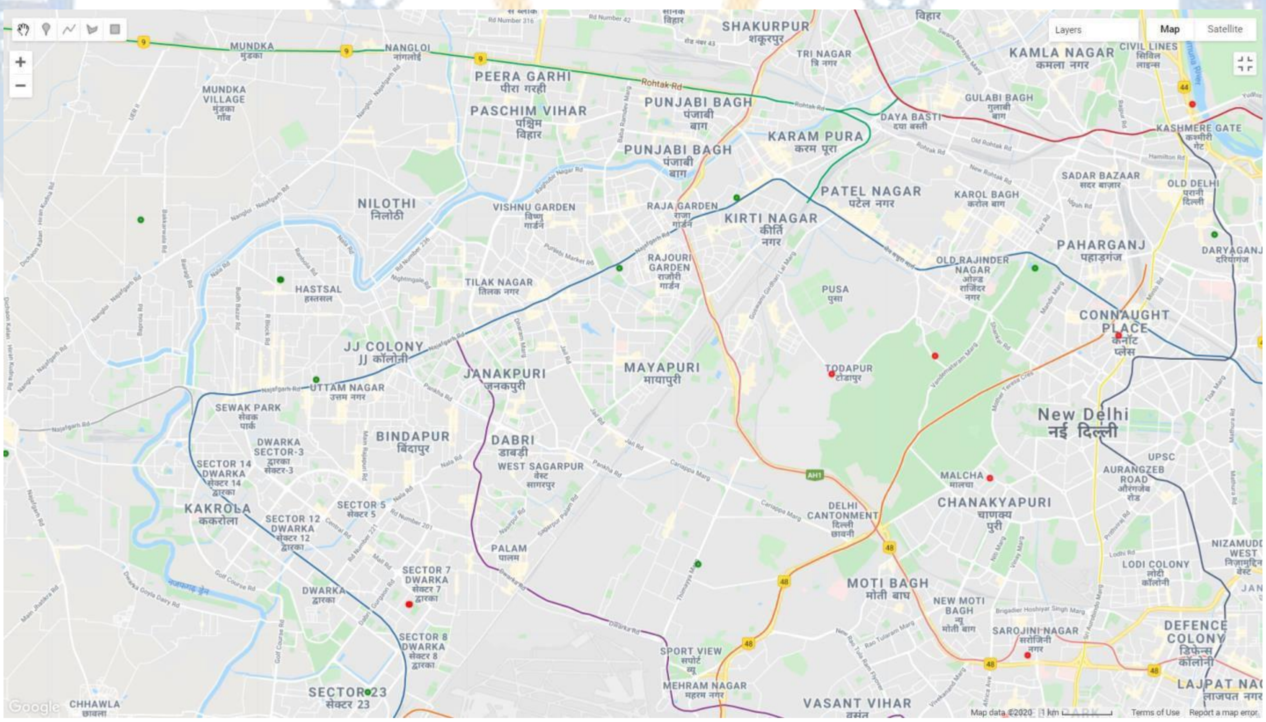
(b)

BU example
NBU example

Figure 7.4: Ground Truth (for one of the folds): (a) Satellite View (b) Map View



(a)

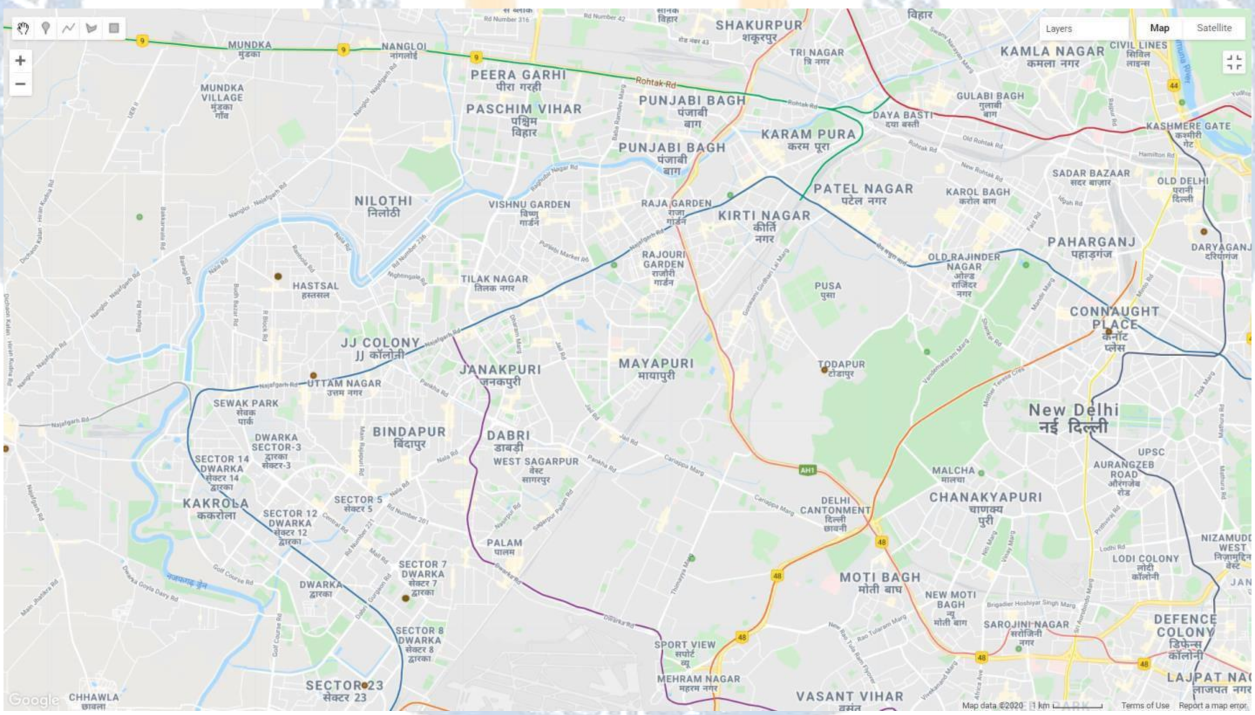


(b)

BU example
NBU example

Figure 7.5: Prediction Output of Decision Tree (CART)

(a)

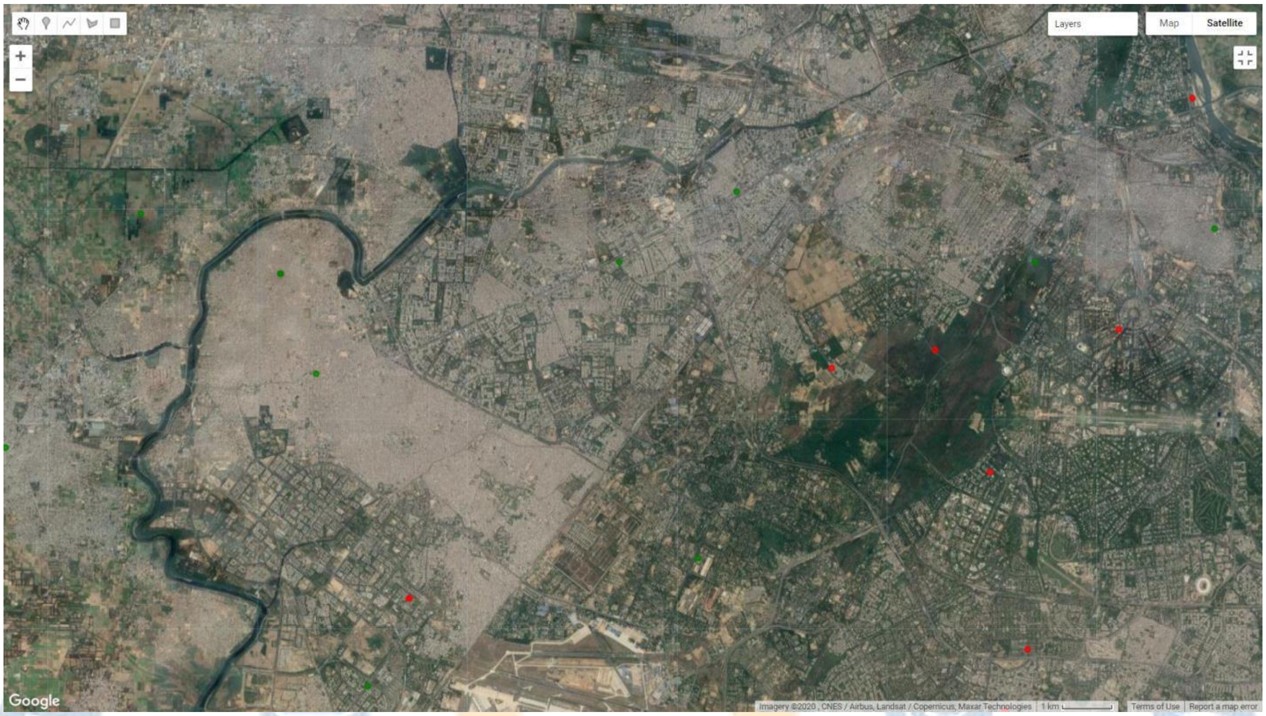


(b)

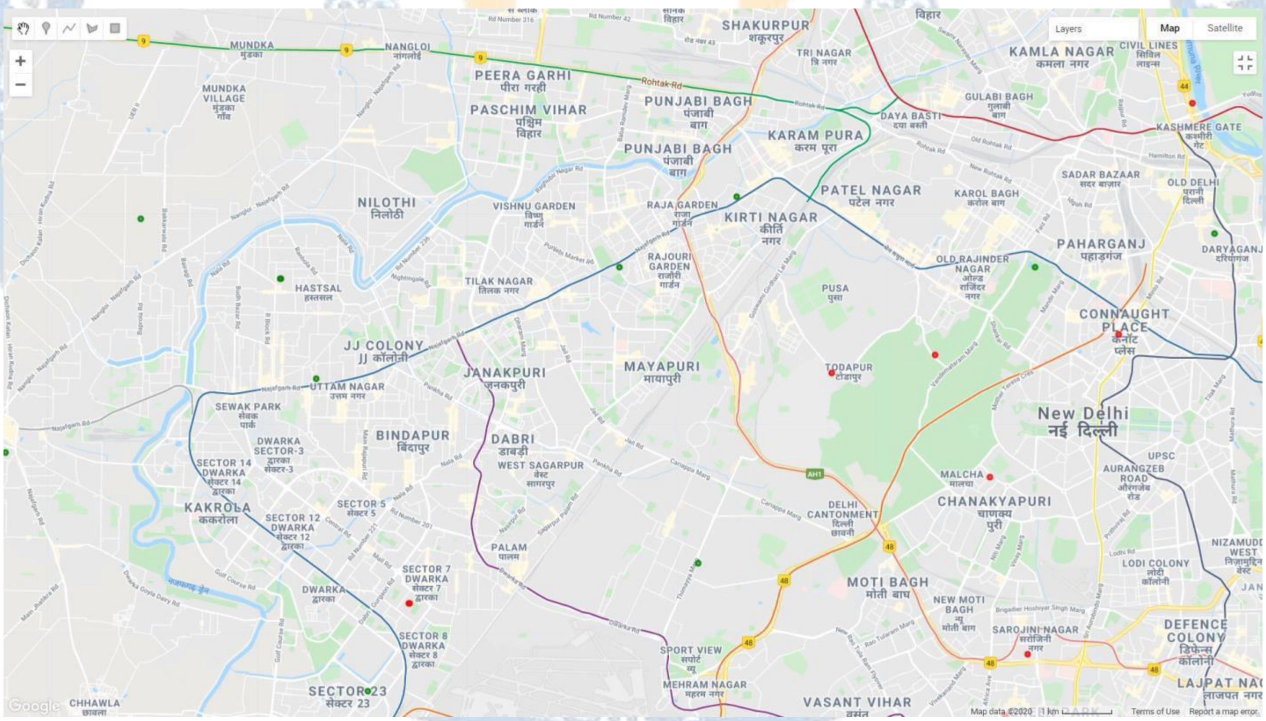
BU example

NBU example

Figure 7.6: Prediction Output for SVM



(a)



(b)

BU example

NBU example

Figure 7.7: Prediction Output for Random Forest (consisting of 50 Trees)

8. Discussion and Conclusion

Various studies have shown that a Random Forests classifier's accuracy is proportional to the number of trees [23]. But, at a certain point, this improvement in performance can decrease as the number of trees is increased if the performance of prediction from learning is less than the computation time for learning with these additional trees [24], [25]. This was the case when we performed 10-fold cross validation on Random Forests consisting of 10, 50 and 100 Trees, and the highest average accuracy was achieved for that with 50 Trees.

Out of all the models used, Linear SVM performs the worst with average testing accuracy of 63.6%. The CART (Decision Tree) and Random Forests (consisting of 50 Trees) gave almost the same performance, with average testing accuracy of 65.7% and 65.8% respectively. This can be because Random Forests generally build trees, with each tree including some of the features and so it can be possible that some of the trees do not consist of some of the important distinguishing features, and thus it performs similar to a single Decision Tree [26].

9. Future Work Recommendations

The task presented in this study can be extended by incorporatingsocio-economic and other geographic features over large areas to study the extension of the urban sprawl in a limited timeframe. The performance of the classifiers can also be improved by further tuning their parameters using various techniques, other than cross validation.

10. References

[1] H.S. Sudhira, T.V. Ramachandra, K.S. Jagadish, “Urban sprawl: Metrics, dynamics and modelling using GIS,” Int. J. Appl. Earth Obs. Geoinf., vol. 5, pp. 29-39, 2004, doi: 10.1016/j.jag.2003.08.002.

[2] N. Baum-Snow, “.Did Highways Cause Suburbanization?” The Quarterly Journal of Economics, vol. 122, no. 2, May, 2007, doi: 10.1162/qjec.122.2.775.

[3] R. Goldblatt, W. You, G. Hanson and A. Khandelwal, “Detecting the Boundaries of Urban Areas in India: A Dataset for Pixel-Based Image Classification in Google Earth Engine,” Remote Sens., vol. 8(8), no. 634, Aug., 2016, doi: 10.3390/rs8080634.

[4] P.K. Mallupattu and J.R.S. Reddy, “Analysis of Land Use/Land Cover Changes Using Remote Sensing Data and GIS at an Urban Area, Tirupati, India,” The Scientific World Journal, vol. 2013, May, 2013, doi: 10.1155/2013/268623.

[5] L. Hua, X. Zhang, X. Chen, K. Yin and L. Tang, “A Feature-Based Approach of Decision Tree Classification to Map Time Series Urban Land Use and Land Cover with Landsat 5 TM and Landsat 8 OLI in a Coastal City, China,” ISPRS Int. J. Geo-Inf., vol. 6(11), no. 331, Oct., 2017, doi: 10.3390/ijgi6110311.

[6] L. Jiang, W. Wang, X. Yang, N. Xie and Y. Cheng, “Classification Methods of Remote Sensing Image Based on Decision Tree Technologies,” IFIP Advances in Information and Communication Technology, vol. 344, pp. 353-358, 2011, doi: 10.1007/978-3-642-18333-1_41.

[7] V.T. Ha, M, Tuohy, M. Irwin and P.V. Tuan., “Monitoring and mapping rural urbanization and land use changes using Landsat data in the northeast subtropical region of Vietnam,” The Egyptian Journal of Remote Sensing and Space Sciences, vol. 23, no. 1, April, 2020, doi: 10.1016/j.ejrs/2018.07.001.

[8] N. Sidhu, E. Pebesma and G. Camara, “Using Google Earth Engine to detect land cover change: Singapore as a use case,” European Journal of Remote Sensing, vol. 51, no. 1, April, 2018, doi: 10.1080/22797254.2018.1451782.

[9] S. Shetty, “Analysis of Machine Learning Classifiers for LULC Classification on Google Earth Engine,” M.S. thesis, Dept. Geo-Information Science and Earth Observation, Univ. of Twente, Enschede, The Netherlands, 2019. [Online]. Available: http://library.itc.utwente.nl/papers_2019/msc/gfm/shetty.pdf.

[10] L. Kumar and O. Mutanga, “Google Earth Engine Applications Since Inception: Usage, Trends, and Potential,” Remote Sens., vol. 10(10), no. 1509, Sept., 2018, doi: 10.3390/rs10101509.

[11] H. Tamiminia, B. Salehi, M. Mahdianpari and L. J. Quackenbush, “Google Earth Engine for geo-big data applications: A meta-analysis and systematic review,” ISPRS Jr. Photogrammetry Rem. Sens., vol. 164, pp. 152-170, June, 2020, doi: 10.1016/j.isprsjprs.2020.04.001

[12] A. Tobón-Marín and J. C. Barriga, “Analysis of changes in rivers planforms using Google Earth Engine,” Int. Jr. Rem. Sens., vol. 41, no. 22, pp. 8654-8681, May, 2020, doi: 10.1080/01431161.2020.1792575

[13] R. Goldblatt, K. Deininger, G. Hanson, “Utilizing publicly available satellite data for urban research: Mapping built-up land cover and land

use in Ho Chi Minh City, Vietnam," Development Engineering, vol. 3, pp. 83-89, Mar, 2018, Pages 83-99, doi: 10.1016/j.deveng.2018.03.001.

[14] "Case Studies." earthengine.google.com. http://earthengine.google.com/case_studies/ (accessed Oct. 15, 2020).

[15] Google Earth Engine (2020), Google. [Online]. Available: http://code.earthengine.google.com.

[16] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in IJCAI'95, Montreal, QC, Canada, vol 2, pp. 1137–1143, 1995, doi: 10.5555/1643031.1643047.

[17] J.D. Rodriguez, A. Perez, J.A. Lozano, "Sensitivity analysis of k-fold cross validation in prediction error estimation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 48, no. 9, pp. 569-575, Sept., 2015, doi: 10.1016/j.patcog.2015.03.009.

[18] "Delhi." Wikipedia. http://en.wikipedia.org/wiki/Delhi.

[19] "Open Spatial Demographic Data and Research." WorldPop.org. http://www.worldpop.org.uk.

[20] N. Pettorelli, J.O. Vik, A. Mysterud, J.-M. Gaillard, C.J. Tucker, N.C. Stenseth, "Using the satellite-derived NDVI to assess ecological responses to environmental change," Trends Ecol. Evol., vol. 20, no. 9, pp. 503-510, Sept., 2005, doi: 10.1016/j.tree.2005.05.011.

[21] Y. Zha, J. Gao, S. Ni, "Use of normalized difference built-up index in automatically mapping urban areas from TM imagery," Int. J. Remote Sens., vol. 24, no. 3, pp. 583-594, 2003, doi: 10.1080/01431160304987.

[22] NDVI, NDBI & NDWI Calculation Using Landsat 7, 8, LinkedIn. [Online]. Available: http://www.linkedin.com/pulse/ndvi-ndbi-ndwi-calculation-using-landsat-7-8-tek-bahadur-kshetri/.

[23] V.F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo and J.P. Rigol-Sanchez, "An assessment of the effectiveness of a random forest classifier for land-cover classification," ISPRS J. Photogramm. Remote Sens., vol. 67, pp. 93-104, Jan., 2012, doi: 10.1016/j.isprsjprs.2011.11.002.

[24] J. Ramon, "How to determine the number of trees to be generated in the Random Forest algorithm?" ResearchGate.net. http://www.researchgate.net/post/How_to_determine_the_number_of_trees_to_be_generated_in_Random_Forest_algorithm#view=52b6a9eed4c118a6208b4855.

[25] T.M. Oshiro, P.S. Perez and J.A Baranauskas. (2012). How Many Trees in a Random Forest? [Lecture Notes in Computer Science]. Available: https://link.springer.com/chapter/10.1007/978-3-642-31537-4_13

[26] William, "When Would You Prefer a Decision Tree?" Kaggle. http://www.kaggle.com/learn-forum/60582#370873.