

# Predicting Urban trends of growth with Google Earth Engine

by

Rifah Tasmiah Islam  
19101459

Md. Shahriar Mim  
17101463

Nur Fathiha Tahiat Seeum  
19101460

Tahmina Ahmed  
19101479

Tabassum Tanzim Nawar  
19101134

A thesis submitted to the Department of Computer Science and Engineering  
in partial fulfillment of the requirements for the degree of  
B.Sc. in Computer Science

Department of Computer Science and Engineering  
Brac University  
September 2022

© 2015. Brac University  
All rights reserved.

# Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

## Student's Full Name & Signature:



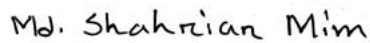
---

Rifah Tasmiah Islam  
19101459




---

Nur Fathiha Tahiat Seeum  
19101460



---

Md. Shahriar Mim  
17101463



---

Tahmina Ahmed  
19101479



---

Tabassum Tanzim Nawar  
19101134

# Approval

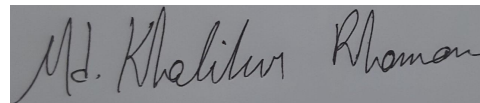
The thesis/project titled “Predicting Urban trends of growth with Google Earth Engine” submitted by

1. Rifah Tasmiah Islam (19101459)
2. Md. Shahriar Mim (17101463)
3. Nur Fathiha Tahiat Seeum (19101460)
4. Tahmina Ahmed (19101479)
5. Tabassum Tanzim Nawar (19101134)

Of Summer, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 20, 2022.

## Examining Committee:

Supervisor:  
(Member)



---

Dr. Md. Khalilur Rahman  
Professor  
Computer Science and Engineering  
Brac university

Program Coordinator:  
(Member)

---

Dr. Md. Golam Rabiul Alam  
Professor  
Department  
Brac University

Head of Department:  
(Chair)

---

Sadia Hamid Kazi  
Chair Person and Associate Professor  
Department of Computer Science and Engineering  
Brac University

## Abstract

Land Usage is one of the most pressing concerns confronting the landscape of Bangladesh due its heavily dense population and limited area. Rapid urbanization has been seen in different parts of Bangladesh, so factors like change in infrastructure, decrease in agricultural land, decrease in greens and water body, as well as a steep increase in built ups are being observed all over the country. Hence, it is critical to have an overall concept of the urbanization trends in order to plan infrastructures, make policies and to conduct large-scale comparison studies. This paper presents a general framework to detect urbanization patterns and transformation of forested areas to residential or commercial developments, specifically in Dhaka division of Bangladesh using Machine Learning Algorithms (MLA). Moreover, for monitoring land coverage change we will be using Google Earth Engine (GEE) data which has a high accuracy record, with accuracy evaluations of 91.21 percent in 2013, 90.46 percent in 2015, and 91.01 percent in 2017. With the help of Landsat archive within GEE, two separate MLA is compared to find the most accurate classification Model. Along with GEE, softwares like QGIS version 3.26, ArcGIS, Terrsat has been used for data cleaning, processing and analysis. Therefore, In this study, the time span of 2015 to 2020 has been considered to create the prediction model and the prediction map of 2025 and 2030 has been obtained using the framework proposed in this work. It is of utmost necessity for the authorities to have optimal data on hand while planning the infrastructure.

**Keywords:** Urbanization Prediction, Google Earth Engine, Land use/land cover (LULC) change, Remote-sensed data, Time series, urban growth, Urban sustainability

## Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.

Secondly, we would like to thank our co-supervisor Shifur Shakil, who had given us the incentive to start our thesis on this topic. And Finally our parents without whose constant support we would not have been able to finish this report.

# Table of Contents

<b>Declaration</b>	<b>i</b>
<b>Approval</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgment</b>	<b>iv</b>
<b>Table of Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Problem Statement . . . . .	2
1.3 Research Objective . . . . .	3
<b>2 Related Work</b>	<b>5</b>
2.1 Literature review . . . . .	5
2.1.1 Machine Learning: . . . . .	5
2.1.2 Urbanization: . . . . .	5
2.1.3 Related Works: . . . . .	6
2.2 Methodology . . . . .	9
2.2.1 Tiff files or Tagged Image File Format . . . . .	10
2.2.2 Machine Learning . . . . .	10
2.3 Research Workflow . . . . .	15
<b>3 Data Collection and Processing</b>	<b>18</b>
3.1 Data Collection . . . . .	18
3.2 Data Pre-Processing . . . . .	18
3.2.1 Collecting Road data . . . . .	19
3.2.2 Slope Data . . . . .	19
3.3 Processing LULC and Feature Collection . . . . .	20
3.3.1 Classification on Google Earth Engine . . . . .	21
3.3.2 Data classification on Qgis . . . . .	22
3.3.3 Data processing on ArcGIS . . . . .	24
3.4 Data Accuracy . . . . .	25

3.5	Data Samples . . . . .	27
<b>4</b>	<b>Model</b>	<b>30</b>
4.1	Cellular Automata . . . . .	30
4.2	Result and analysis . . . . .	33
4.2.1	Future Works Prospects . . . . .	36
4.3	Conclusion . . . . .	37
	<b>Bibliography</b>	<b>40</b>
	<b>Code Snippets for Classifiers</b>	<b>41</b>

# List of Figures

2.1	Processing and Scaling data in raster image. . . . .	11
2.2	SCP Plugin . . . . .	12
2.3	Cellular Automata . . . . .	13
2.4	Decision Trees in Random Forest . . . . .	14
2.5	Data Collection and Prediction of Urban growth. . . . .	16
3.1	Slope and Road Distance of Dhaka. . . . .	20
3.2	Random Forest Classifier workflow. . . . .	21
3.3	SmileCart Classifier workflow. . . . .	21
3.4	RF Model Accuracy (Left) and SmileCart Model accuracy (Right) . . . . .	25
3.5	Data Samples Collected over 2015 to 2018 (Dhaka and Mirpur) . . . . .	29
4.1	CA Model Implementation. . . . .	30
4.2	Change in Land Use over the Year 2015 to 2017. . . . .	31
4.3	Transition Matrix of the Year 2015 to 2017. . . . .	31
4.4	Predicted Map of Mirpur Vs. Actual Map (2020) . . . . .	32
4.5	Predicted Map of Dhaka Vs. Actual Map (2020) . . . . .	32
4.6	Dhaka City Predicted Vs Classified. (2020) . . . . .	34
4.7	LULC 2020(Left), 2025(Center) and 2030(Right). . . . .	34
4.8	LULC 2020(Left), 2025(Center) and 2030(Right). . . . .	35
4.9	Land distribution of the predicted Images. . . . .	35
4.10	Urban Growth Trends in Dhaka. . . . .	36



# List of Tables

3.1	Bands Description of Landsat 8 . . . . .	20
3.2	Five Feature Classes . . . . .	24
3.3	Area Based Error Matrix of Classified 2020 . . . . .	25
3.4	Accuracy of Classified Data . . . . .	26
3.5	Area Based Error Matrix of Mirpur 2015 . . . . .	27
4.1	Accuracy of the Predicted Map in Dhaka and Mirpur . . . . .	33

# Chapter 1

## Introduction

### 1.1 Introduction

Urbanization has emerged as one of the most significant and noticeable global processes over the past several decades. According to [8], urban areas contain more than fifty percent of the world's population, but generate more than ninety percent of the global GDP. The United Nations projects that by 2050, 65 percent of the world's population will reside in cities [20]. The rapid growth of cities around the world is evidence that we have entered the "century of the city" [27]. Regarding the problem of long-term sustainability, rapid urbanization presents both challenges and opportunities. 4 Changes in land use and land cover (LULC) are the result of urbanization and the expansion of industrial activity, which has resulted in the construction of enormous infrastructures. [31]. As a result, the environmental impact of changes in land use is no longer a regional concern, but a global one.

Recent years have witnessed a dramatic acceleration in the rate of urbanization growth in Bangladesh. Expansion of industrial production and urbanization are the primary contributors to the deterioration of the planet's climate. For this reason, decision-makers and urban planners must have a clear understanding of the urban growth rate of cities in order to structure the ongoing unplanned urbanization and industrialization.

Using the cloud-based platform Google Earth Engine, users can easily access high-performance computing resources for the purpose of processing large geographic datasets. This eliminates the requirement for users to download and manage data on their own computers [25]. Using Google Earth Engine simplifies the utilization of remote sensing data when conducting environmental research. GEE compiles a wide range of geospatial datasets, such as data from Landsat satellites Sentinel-1 and Sentinel-2, MODIS, and other widely used geospatial datasets. In addition, the GEE's archival image collections can be used and processed with the aid of two distinct code versions, JavaScript and Python. [27] The spatial and multispectral resolution of 30 meters is ideal for characterizing urban areas, while the revisit time is adequate for monitoring purposes. Because Landsat has maintained temporal continuity from 1972 to the present, it is also the platform of choice for urban development research [11].

Quantum Geographic Information System, also known as QGIS, is a free, cross-platform, open-source desktop Geographic Information System (GIS) program. It

enables users to generate and export graphical maps, as well as edit and analyze spatial data. QGIS can read and write raster, vector, and mesh layers, as well as shapefiles, personal geodatabases, dxf, MapInfo, PostGIS, and other industry-standard formats. It accomplishes this by offering Web services, such as Web Map Service and Web Feature Service, which can be utilized to access and utilize data from external sources. By utilizing the Google Geocoding API, plugins can geocode, perform geoprocessing tasks comparable to those performed by ArcGIS's native tools, and connect to PostgreSQL/PostGIS, SpatiaLite, and MySQL databases [35]

The QGIS Semi-Automatic Classification Plugin enables the semi-automatic classification of remotely sensed data under human supervision. This plugin also provides tools for expediting the classification process [6]. In addition, the group effort known as OpenStreetMap is working toward the creation of a free and editable geographical database that covers the entire planet (OSM). OpenStreetMap's topological data structure is composed of Nodes, Ways, Tags, and Relations, which are the structure's four primary components [34].

ArcGIS is a collection of client, server, and web-based GIS services. ArcGIS, which was created and is maintained by Esri, is based on a geodatabase that stores spatial data using an object-relational database strategy. Known as a "container," a geodatabase is used to associate datasets with geographical characteristics and attributes. Personal geodatabases, enterprise geodatabases, and file geodatabases are the three types of geodatabases that can be stored in ArcGIS [33].

TerrSet is a remote sensing and integrated geographic information system utilized for the processing and presentation of digital geospatial data (GIS). It is widely used for fundamental and advanced spatial analysis, including tools for statistical and surface analysis, time series image data analysis, and decision support [28].

In order to present a general framework to quantitatively detect urbanization trends and the conversion of forested areas to residential or industrial developments in the Dhaka division of Bangladesh, this paper will perform image processing using a variety of different tools. The framework will employ multiple Machine Learning Algorithms (MLA) in the classification and prediction phases to provide an approximation of a growth prediction. In addition, we will use the data from Google Earth Engine (GEE) to monitor the change in land coverage (LULC) due to its large number of time-series datasets, ease of use, and higher level of accuracy. We will use the semi-automatic classification plugin that is available for QGIS to classify the Land Use Land Cover (LULC) change. Consequently, using ArcGIS and TerrSet, the classified data will be re-projected, and following the calculation of the Markov-chain matrix, a CA-Markov Machine learning model will be developed in order to forecast the future urban growth of a particular Region of Interest (ROI), which in this case is the Dhaka division. This will allow us to determine how the population of Dhaka will continue to increase.

## 1.2 Problem Statement

Bangladesh is one of the countries urbanizing at the fastest rate in the world, and the country has experienced an unplanned and rapid rate of urbanization in recent years. Currently, only 34% of the population resides in urban areas [9], but by 2050, this percentage is expected to more than double, reaching 56%. Changes in urban delimitation and the definition of what constitutes a metropolitan area have

been one of the primary drivers of urbanization and population growth over the past several decades. Land use change (LULC) can be monitored through the use of time series GIS (Geographic Information Systems), remote sensing, Landsat images, socioeconomic data, and other types of geographic information [14].

Changing land use is the most blatant indication that humans have left their mark on the planet, and it is widely acknowledged as the leading cause of the decline of biodiversity and the degradation of land [26]. Identifying changes in land use over time is essential to gaining a deeper understanding of the dynamic relationships that exist between humans and their natural environment. It is possible for national, state, and municipal governments to conduct effective urban planning if they have an understanding of how land use has changed over time and adequate information on the current patterns of land use [5].

Imaging with Landsat is the most effective method for determining global urbanization patterns over varying time periods [11]. Environment, economy, and society are the most frequently employed indicators of the urban environment's ability to remain sustainable. To calculate ESV [24], it is necessary to have data on land use and land cover changes (LULC) for a particular research region and study period. As a result, satellite-derived data can be extremely useful for determining patterns in urban sprawl and LULC Changes, as these data are not only inexpensive, but also easy to access and contain long time series. [4] In order to successfully extract accurate LU/LC data from remotely sensed data, it is necessary to have efficient image classification algorithms [32]. Google Earth Engine's spatial and multispectral resolution of 30 meters is ideal for characterizing urban areas, and the revisit time is adequate for monitoring purposes. Because Landsat has maintained temporal continuity from 1972 to the present, it is also the platform of choice for urban development research [11].

The download of remote sensing imagery and a powerful central processing unit (CPU) [26] are required by conventional methods for mapping and monitoring changes in land cover and use, such as the removal of wetlands, urbanization, the growth of agricultural land, and deforestation. As a result, the GEE makes it possible to quickly analyze and categorize satellite data, allowing for the creation of outputs for population distribution that are more accurate and relevant.

Ecological and urban modelers are partially unaware of the new analytical approaches that are becoming available as a result of GEE's extraordinary geospatial processing capabilities and vast collection of enormous geospatial datasets. This is due to GEE's extensive collection of enormous geospatial datasets.

### 1.3 Research Objective

Our group's major purpose is to forecast the urban expansion of cities in Bangladesh using the most recent publicly accessible data and by proposing a broad framework that may be utilized or expanded in the future to forecast urban expansion. We would also like to explore and attempt to ascertain whether or not there is a correlation between the urban development of one region and the urban growth of the next region based on the results we have obtained. To precisely predict the expansion of the city, we will use the number of metal sheds constructed. This will aid in the planning of appropriate infrastructure for rural areas that are still in the process of underdevelopment. To summarize:

- Predicting urban growth by Processing Google Earth Engine Image.
- Considering Tin shed built-ups as a parameter to predict the urban growth.
- Train and test the accuracy with several ML models and compare for the optimum result for geospatial classification.
- Provide an insight on the growth of urban areas as well as predicting development in remote locations.
- Provide authorities with resources needed to make plans of implementing infrastructure in an area that is showing signs of rapid urbanization.
- Correlating urban growth with factors such as deforestation, decrease in agricultural land, alteration in water bodies and rivers, LULC and the change in landscape
- Demonstrating a framework for data classification using Qgis, Arcgis and TerrSet.

# Chapter 2

## Related Work

### 2.1 Literature review

#### 2.1.1 Machine Learning:

According to researchers Jordan Mitchell [10], machine learning tries to answer the question of how to create machines with the ability to learn on their own. Located at the crossroads of computer science and statistics, it is one of the fastest-developing technological domains now. In addition, it is fundamental to artificial intelligence and data science.

The researcher who worked on the Google Brain project, J Dean et al [17], and his colleagues feel that the Machine Learning revolution is ushering in a new Golden Age for computer architecture. They stated that, with the use of machine learning, a car might never be involved in an accident, which was previously considered science fiction. They opted to pursue a profession in machine learning because they were compelled to discover answers to some of the most urgent problems affecting modern society [17]. As student researchers, we were also eager to employ machine learning to address one of Bangladesh's most critical problems. Rapid urbanization growth has a considerable impact on land cover and land use, as well as the destruction of forests and the filling of water bodies and rivers.

#### 2.1.2 Urbanization:

Urbanization is the process of massive amounts of people living partially or completely in relatively small areas, forming cities. According to the Ritchie Roser [18], in terms of overall population, several cities throughout the world have risen quickly in the last 50 years. Among them the urbanization growth of Dhaka city is noticeable. This number is predicted by them to rise to 25 million by 2035. Dhaka grew from a population of fewer than half a million in 1950 to about 18 million in 2015 and is projected to reach 31 million by 2030. Urban development, among the many human activities that promote habitat degradation [1], causes some of the highest local extinction rates and typically destroys the majority of native species [2]. In addition, urbanization has a longer lifespan than other forms of habitat degradation.

### 2.1.3 Related Works:

In 2020, Hossain, Zarin, Sahriar Haque [30] predicted the land use and land cover change in Gopalganj District using ML models. Their research predicted that urban areas in the district’s center and north-western regions would develop by 10.88 percent by 2050, using an Artificial Neural Network (ANN) with integrated cellular automation (CA) simulation. There will be a 4.09 percent drop in urban vegetation, as well as a large reduction in bare land and aquatic bodies. According to their research, based on the validation result, the projected Land Use Land Cover (LULC) has an accuracy of 89.48 percent. Between 2000 and 2020, multi-temporal and multi-spectral Landsat data covering the Gopalganj district were obtained for this research from the USGS website.

In 2015, researchers Hassan Nazem [12] did a study on Environmental Sustainability, LULC changes, Urban growth dynamics of Chittagong City of Bangladesh. Their research objective was to lessen environmental effects, and future urban expansion planning should be made easier. They collected their data from Chittagong Development Authority (CDA) for Base Map from the year 2001 and 2006, Chittagong Metropolitan Master Plan, Road, Network. Moreover, they collected data from UDD, CUS, BBS and CUS. They did the processing using GIS information and Landsat pictures, a themed land cover map was generated. The land cover maps and corresponding transition matrix were created using a temporal series of Landsat data, comprising imagery from MSS 1977, TM 1989, ETM 1999, and the most current Landsat-8 (OLI–TIRS)2013 sensors. Topographic maps are used. To save computing and image analysis time, Landsat pictures were chosen to investigate urban growth during the pre-satellite period. Other photos were registered to this OLI–TIRS image as a reference after a reasonably high-resolution image from OLI to TIRS was geo-referenced largely using GIS employing road network data. Overall, they were between 85% and 93% accurate.

In their study on the urban development of Isfahan city in 2021, Gao et al. [29] utilized Google Earth Engine’s remote sensing data (GEE). The city’s name is Isfahan, and it is located in the middle of the Iranian plateau. The distance to Tehran, the nation’s capital, is 424 kilometers. Their major purpose was to evaluate Isfahan’s urbanization and calculate the overall quantity of green area that had been lost as a result. By examining the GEE database, it was possible to locate satellite pictures of the city of Isfahan. In these images, the years 1986 and 2019 are depicted. Both the Landsat 5 satellite in 1986, which gathered data for the USGS Landsat 5 Surface Reflectance Tier 1, and the Landsat 8 satellite in 2019, which gathered data for the USGS Landsat 8 Surface Reflectance Tier 1, provided photos for this investigation. In addition to a per-pixel saturation mask, Landsat 5 pictures were atmospherically adjusted with LEDAPS and included a cloud, shadow, water, and snow mask made with CFMASK. In contrast, Landsat 8 pictures were atmospherically adjusted with LaSRC and contained a cloud, shadow, water, and snow mask generated with CFMASK, as well as a per-pixel saturation mask. Landsat 5 satellite images were captured between 5 and 10 January 1986, while Landsat 8 satellite images were captured between 5 and 1 January 2019. Both sets of photographs were captured in January. The photographs have the least amount of cloud cover possible at this time. To remove clouds from images in GEE, we utilized the Median method. NDVI (Normalized Difference Vegetation Index) [13], NDWI (Normalized Difference Water Index) [7], NDBI (Normalized Difference Built-up Index), and Urban Index (UI)

[19] were established as part of their processing technique. The Maximum Value Composite Algorithm was applied for the purpose of computing each index. After traversing each index value, the data is sorted so that only the highest value for each pixel is taken into account. SVM was applied because it is a non-parametric and supervised technique. Even though they only had a little amount of training data, their accuracy ranged between 95 and 96 percent.

A 2019 study by H. A. Zurqani et al.[26] utilized Google Earth Engine to map urbanization patterns in a forested environment. Their research concentrated on the upstate region of South Carolina, which comprised roughly 2114.47 km<sup>2</sup> of total land area. The primary goals of the study were to: (1) use high-resolution satellite imagery to determine the land cover classes and distribution for the upstate region of South Carolina; (2) identify areas that have been converted from forested to residential or industrial constructions; and (3) discuss the potential consequences of land use change in this region. For each research year, high-resolution NAIP aerial photos were classified using the unsupervised clustering technique. In addition, the land use change detection approach was created and assessed using the GEE platform to generate code. The NAIP imagery was also incorporated into the development of the reference dataset in order to validate the classifications. The excellent spatial resolution of the imagery made this possible. The unsupervised classification k-mean algorithm was used to preprocess the data. Chlorophyll reflectance was evaluated to distinguish between forest and non-forest environments. Other indices derived from different combinations of spectral bands, such as the Normalized Difference Vegetation Index (NDVI), the Enhanced Vegetation Index (EVI), the Green Ratio Vegetation Index (GRVI), and the Modified Soil- Adjusted Vegetation Index (MSAVI), were also used to identify features that are more indicative of the greenness of vegetation. The Normalized Difference Water Index (NDWI) was utilized to identify water spots more precisely. In order to build a map of South Carolina's urban zones, calculations and layering of all of these indices were performed on every image. In 2013, the overall accuracy was 91.21 percent, in 2015 it was 90.46 percent, and in 2017 it was 91.01 percent. (2017). 3). The amount of land covered in forest during the first year of the study (2013) was 1402.52 km<sup>2</sup> (66.33%) and the amount of land covered in non-forested areas was 532.85 km<sup>2</sup> (25.20%); during the second year of the study (2015), the amount of land covered in forest decreased to 1354.31 km<sup>2</sup> (64.05%), and the trend continued. In some regions, the National Aeronautics and Space Administration (NASA) Imagery is not accessible annually, which was one of the limits of this method.

Numerous academics have utilized Google Earth Engine and its interfaces in the course of their work, and the platform is finding special relevance in the field of urban growth forecasts due to the availability of high-resolution images and data. In the year 2020, [27] provided a basic example of how to apply GEE to image-based ecological and urban modeling. The investigation was conducted in Hohhot, IMAR, China (at coordinates N39°61– N41°38, 110°51– 112°31 E). As the major data source, Google Earth Engine provided us with a rich geographical dataset. This collection contained Landsat5 TM, Landsat7 ETM+, and Landsat8 OLI satellite imagery. After the image data had been atmospherically adjusted using the LaSRC method of atmospheric correction, the cloud, shadow, water, and snow patches were masked using CFMASK. Using the "Reducer" function in GEE, approximately 82 distinct Landsat SR image scenes were combined into a single annual image. Each year,



these scenes were featured in the compilation. After retrieving images and completing preprocessing, which included the standardization of data and the stacking of time-series images, LULC conducted preliminary classification, which included the selection of classification systems, appropriate classifiers, as well as training and testing samples. During the final stage of processing, known as "post-classification," LULC confirms the temporal consistency of the classified images and removes errors and noises from the initial classification. When it comes to the post-classification phase of LULC, there are two key goals to keep in mind: removing errors and noise from the initial classification, and ensuring that the temporal presentation of the classified images is consistent. The modeling of urban sustainability was conducted within the context of ESV, and both the GDP and the per capita net income were included (PCNI). In this method, the Random Forest classifier had the best overall performance throughout all years, with an average overall accuracy (OA) of around 75.76 percent. This was in contrast to the MIN classifier, which was just 50.69% accurate, and the CART classifier, which was only 67.46% accurate. In addition, a greater value for the kappa coefficient suggests that the performance of RF classifiers is superior. While the RF classifiers' kappa values ranged from 0.669 to 0.738, with an average kappa value of 0.701, indicating strong consistency in the results, the other two classifiers' kappa statistics exhibited only moderate consistency, ranging from 0.40 to 0.625. However, ESV and SI (sustainability index) investigations are constrained to a LULC perspective. Because GEE APIs and functions can only be enhanced using JavaScript or Python, the number of individuals who can utilize the interface is limited.

Furthermore, a research conducted by Gómez et al., 2020 [22] on spatiotemporal modelling of Urban Growth; In terms of population distribution, binary urban footprint, and colored urban footprint, it represented a broad framework for forecasting the growth of three important city factors. They focused on Valledupar and Rionegro, two Colombian cities in Latin America with various geomorphological traits. Datasets from Landsat 4TM, 5TM, 7TM, and 8TM were gathered. Again, the data sources of population distribution, Urban footprint, Binary urban footprint, Official population projection, Water bodies were taken from GHSL pop, through a binary classifier, Landsat, the National Bureau of Statistics, and IGAC each determined the urban footprint. The raster library, the RStoolbox toolkit, and R were used to preprocess the Landsat pictures, mask and fill clouds and shadows, and apply relative radiometric normalization. To classify the images into urban and non-urban categories for this study, a random forest model was trained using the caret library. After that, a two-step temporal correction was performed using the raster library. QGIS was used to take random multitemporal samples of the Landsat images with seven bands, label them, evaluate the perceived quality of the urban footprint images in RGB, and rasterize vector datasets. Linear regression, lasso regression, Bayesian regression, trees, random forests, gradient boost regressors, support vector machines, ridge regression, and multilayer perceptrons were among the machine learning models utilized in the study. A multi-temporal collection of pixels with four categories—built-up, vegetation, bare soil, and water—has been compiled by the researchers. Performance metrics for the reclassification process included Valledupar's overall accuracy, which was 0.99, as well as the Kappa statistic, precision, recall, and F1-score. In Rionegro, the F1-score is 0.96, the precision is 0.96, the recall is 0.97, the Kappa statistic is 0.96, and the total accuracy is 0.98. The following flaws

were identified: (1) A semantic segmentation module based on deep learning could have been used to improve the urban/non-urban classifier; (2) the initial estimate of the binary urban footprint in the preprocessing stage might have been improved. (3) The urban footprint in color might be enhanced using a photo-realistic semantic inpainting that was developed using satellite photos of urban and non-urban areas from various geographies.

In their 2017 study using Landsat to track changes in land use across continental Africa, Midekisa et al. [15], who addressed the computational difficulties of managing enormous amounts of earth observation data using cloud computing for those scientists and practitioners lacking powerful computing facilities. The continental area of Africa was the focus of the study. Data were gathered using the Landsat 7 Enhanced Thematic Mapper Plus (ETM+) and surface reflectance data from the United States Geological Survey (USGS). Later, a cloud screening technique was used to eliminate pixels from 1999 and 2016 that had been tainted by clouds. The 2015 layers' NDVI, NDWI, and nighttime light were utilized as covariates in the model together with the Landsat spectral bands. To model and predict the 7 LULC classes on GEE, a random forest decision tree classification approach was employed. On the GEE cloud platform, all analyses were completed. A total accuracy of 88% was reached by the model. Over the course of the seven sessions, a total of 7,084 data points—5,664 for training and 1,420 for validation were gathered. Based on the random forest classification algorithm, annual LULC projected maps are created. With an AUC value of 0.98, it displayed extremely strong predicting ability. One limitation of the researchers' methodology and findings was that only 2015's training and validation data were available. (2) It depended on eye assessment of high-resolution photography to produce training data, (3) It only looked at seven LULC classes, which may have limited the applicability of the current LULC product in some circumstances.

We have found that most of the researchers have not used the Tin Shed built-ups as their feature for collecting, training, and predicting data. Tin Shed built-ups are one of the most significant features of the urban growth of Bangladesh, especially Dhaka city. In our work, we have used Tin Shed built-ups as a feature to collect data and predict its future states. Moreover, we trained our 5 features (Tin Shed, Concrete, Vegetation, Water Body, Unused Land) and collected our noble data. We have used the Landsat-8 satellite which launched in 2013 for collecting our data.

## 2.2 Methodology

The area we will take under consideration here will be Dhaka division in Bangladesh. The dataset we have chosen to use is the Landsat 8 Tier 1 data that is available from the year 2013. The plan is to select the location as ROI and then predict the growth from images collected from GEE (Google Earth Engine). We have taken a pixel based approach, where each pixel was classified into five distinct categories based on the values of the 7 bands on these Tagged Image Files. The indices NDVI, NDWI and NDBI are used to determine the pixel to pixel NDSV value by an approach developed by Pesaresi et al. and Xu [3]. Taking incentive from their research we have also taken into consideration the values of Vegetation Index, Built-up Index and Water Index for the classified images where the vegetation or water or built-up were seen to have been under-recorded by our classification model. In performing

the task we had tried to classify our data by following three separate approaches, two of which were classifiers on Google Earth Engine and another approach was on the Qgis Software version 3.22. A brief introduction to the models, geospatial files and softwares used in our work is provided below:

### 2.2.1 Tiff files or Tagged Image File Format

Tagged Image File Format, also known simply as TIFF, is a file format for images that is capable of storing the data associated with geographical information. The term "geospatial imagery" refers to a wide range of graphical products that communicate information regarding natural occurrences and human actions on the surface of the globe. Data from geographic information systems (GIS) as well as satellite photos can be retrieved from the Google Earth Engine and saved as.tif files. The information collected from the satellite includes a number of bands, beginning with B2 and going all the way up to B10. These bands, each of which stands for a distinct value and expresses a distinct object on the surface of the earth, include, for instance, the B3 band, which symbolizes the vegetation that can be seen on the surface from space. What the satellite sees as surface reflectance is used to assign a value to each pixel or square on the map. This value is determined by what the satellite sees. After that, these data are saved in each band, and then each band is layered on top of the other to produce a monochromatic image. However, by running a machine learning model on the portion of the map that has been clipped away, it is feasible to categorize each pixel and assign them a label. This may be accomplished by using the band values. Therefore, each pixel contains many values that are classified under a variety of different bands.

Since the bands express very different values on the map, they will have very different ranges. In order to tackle this it is possible to scale the values. This can be achieved by a number of ways including:

#### 6.1.1 Normalization:

$$Normalization[0, 1] = \frac{x - x_{min}}{x_{max} - x_{min}}$$

However for bands that have very consistent values most of the pixels will be assigned either 0 or 1 entirely. In order to avoid that the second method is used.

#### 6.1.2 Standardisation:

$$Standardisation[-1, 1] = \frac{x - x_{mean}}{x_{std}}$$

Here as the value is not binary these is a higher possibility of getting better scaled values for closely clustered features.

### 2.2.2 Machine Learning

Machine learning (ML) is a branch of research concerned with comprehending and developing methods that 'learn,' that is, methods that use data to improve performance on a set of tasks. There exists several Machine Learning models that use

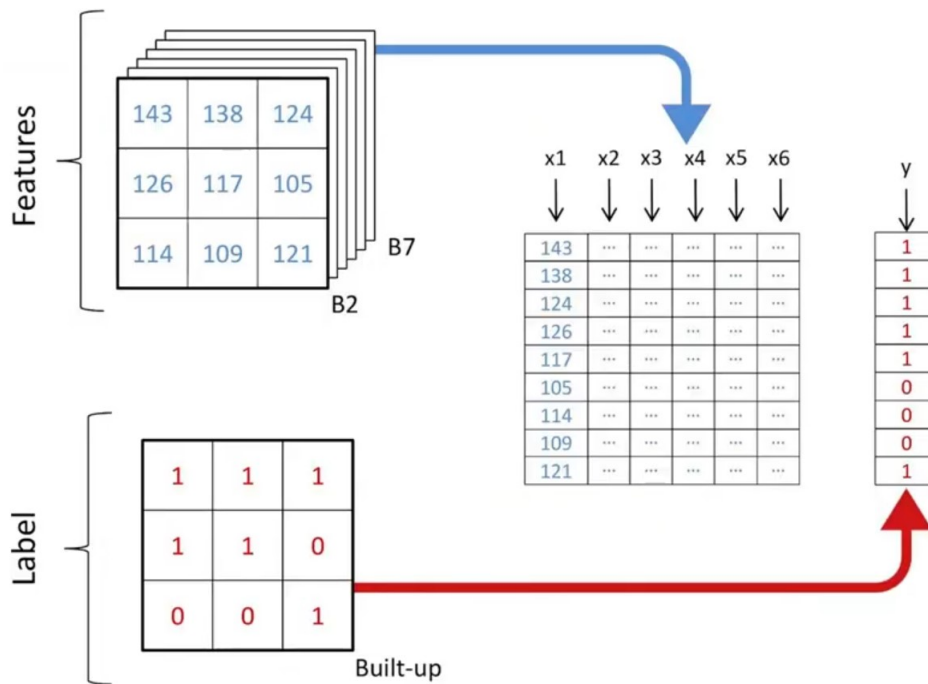


Figure 2.1: Processing and Scaling data in raster image.

systemic learning procedures from supervised, unsupervised, semi-supervised or reinforcement learning data. In our study we have used a model named SmileCART to Train-Test and classify the data. Then from the data procured we have used an ML models named Cellular Automata to train-test and finally predict the growth in urbanization in the divisions of Bangladesh.

## SmileCart

SmileCart is the name of a classifier that can be found on the Google Earth Engine. The raster data can be classified using the classifier package and its sub-package smileCart. This will allow you to label the areas as having vegetation, a body of water, urban areas, or barren land. Smiling face classifiers are a relatively new addition to Google Earth Engine, but they have already been demonstrated to enhance the system's performance and stability. Using the points on the map, a training dataset must be compiled as the initial step. This is accomplished in a sensitive manner by selecting and appropriately labeling at least 25–30 points on the map. If all of the first points selected are from the same type of terrain, the first classifier will only be able to identify the vegetation on the map based on the values of the bands selected from the training points. Currently under consideration are the Normalized Difference Vegetation Index, also known as NDVI, the Normalized Difference Water Index, also known as NDWI, and the Normalized Difference Built-up Index, also known as NDBI. The SmileCart classifier handles all of these tasks in the background, and the resulting tiff file can be extracted to a specific folder on

Drive before proceeding to the next step.

## Semi-Automatic Classification Plugin

The Semi-Automatic Classification Plugin (SCP) is a Python plugin for the QGIS software that was designed to make monitoring land cover easier for non-remote sensing specialists who could benefit from remote sensing analysis. The Semi-Automatic Classification Plugin (SCP) was designed to make monitoring land cover in developing nations simpler (QGIS Development Team, 2021). SCP, created by Luca Congedo, is a set of tools that enables the supervised classification of remote sensing images. These instruments feature download, pre-processing, and post processing capabilities. You can search for and download satellite images from ASTER, GOES, Landsat, MODIS, Sentinel-1, Sentinel-2, and Sentinel-3. When classifying various types of land cover, there are numerous available classification algorithms. Installing GDAL, OGR, Numpy, SciPy, and Matplotlib is required for this plugin to function correctly. This instrument was developed for the express purpose of categorizing various land cover and land use types.

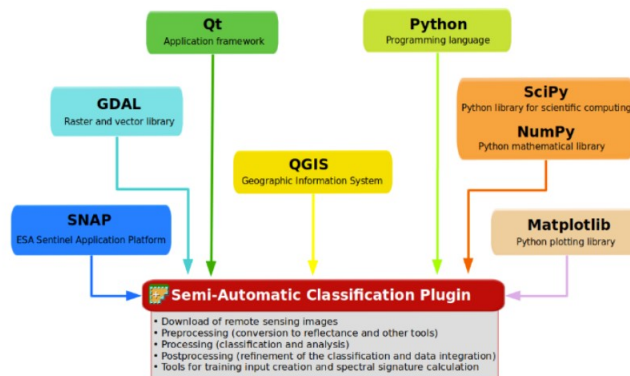


Figure 2.2: SCP Plugin

## Markov Chain

Modeling random processes statistically can be done in a straightforward manner using the Markov Chain. [Case in point:] Because they do not require any difficult mathematical equations or calculations to be carried out, putting them into practice is not an overly challenging endeavor to undertake. It then generates a matrix known as the transition matrix based on the probabilities that it determines by performing a calculation to determine the probability of one event changing to all of the other events that are available in that space and then generating the matrix. The matrix will be a  $N \times N$  matrix if there are  $N$  possible states in the Markov chain. The entry  $(I, J)$  will represent the probability of transitioning from state  $I$  to state  $J$ . If there are  $N$  possible states, the matrix will be a  $N \times N$  matrix. In a Markov chain with  $N$

potential states, the matrix will have the same number of rows and columns as there are potential states. The transition matrix must also be in the form of a stochastic matrix, which requires that the sum of the values in each row's cells be exactly 1, and this requirement stipulates that the matrix must be in a row-by-row format. Everything makes perfect sense when you take into account the fact that each row reflects a unique probability distribution. In relation to the work that we do, there are five distinct classes, which are as follows: C1 (Tin Shed), C2 (Concrete), C3 (Vegetation), C4 (Water Body), and C5 (Unused Land). As a consequence of this, the transition matrix that is generated by the Markov chain would look like this as a result:

	Class 1	Class 2	Class 3	Class 4	Class 5
Class 1 :	0.3460	0.2099	0.0694	0.0875	0.2871
Class 2 :	0.0644	0.3920	0.2570	0.0383	0.2484
Class 3 :	0.0070	0.1124	0.7223	0.0348	0.1235
Class 4 :	0.0351	0.0597	0.2736	0.5022	0.1293
Class 5 :	0.0332	0.1807	0.3238	0.0229	0.4394

This transition matrix is later used to predict the values of the current cells based on its neighboring cell.

### Cellular Automata

The Cellular Automata model works on raster files like a matrix and it goes through the cells in the form of a kernel. It then goes through the raster image and decides the future value of the cell based on the values of the cells that are neighboring it. The user is responsible for determining the size of the kernel. The checking begins

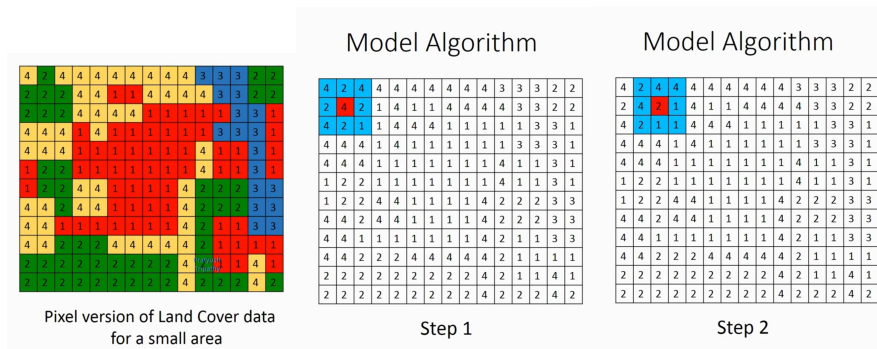


Figure 2.3: Cellular Automata

in the second cell from the left, and the kernel continues to advance to the right from there. The middle cell will be examined to see if it satisfies all of the criteria necessary to be considered urban. In order for the middle pixel to be classified as an urban area, it is necessary to take into account the fixed threshold values of the cells that surround it. Then, if it is successful in passing all of the checks, it is thought that urban growth will take place.

The checking begins in the second cell from the left, and the kernel continues to advance to the right from there. The middle cell will be examined to see if it satisfies all of the criteria necessary to be considered urban. In order for the middle

pixel to be classified as an urban area, it is necessary to take into account the fixed threshold values of the cells that surround it. Then, if it is successful in passing all of the checks, it is assumed that urban growth will take place.

## Random Forest

Random forest classification works on the basis of several decision trees working one after another to successfully classify the data on hand. Decision trees are like branches on a tree that ask a question and according to that move on to the next step for example when identifying a human and an object the question can be does it have eyes and according to that answer there can be two branches that one points to human and other to an object. This model after getting the training data selects these questions, so as to say the questions can be compared to thresholds. So the labels from the training data set according to their weights are assigned threshold values by the classifier. And the classifier also selects which labels to consider first and the sequence. This sequential comparison with a threshold classifies each pixel on the raster files as urban/vegetation/water/ barren land. In our case this random forest model can be used in predicting growth by comparison with previous years and checking if an area was converted to urban or not. The primary working of a random forest model;

## Random Forest Classifier

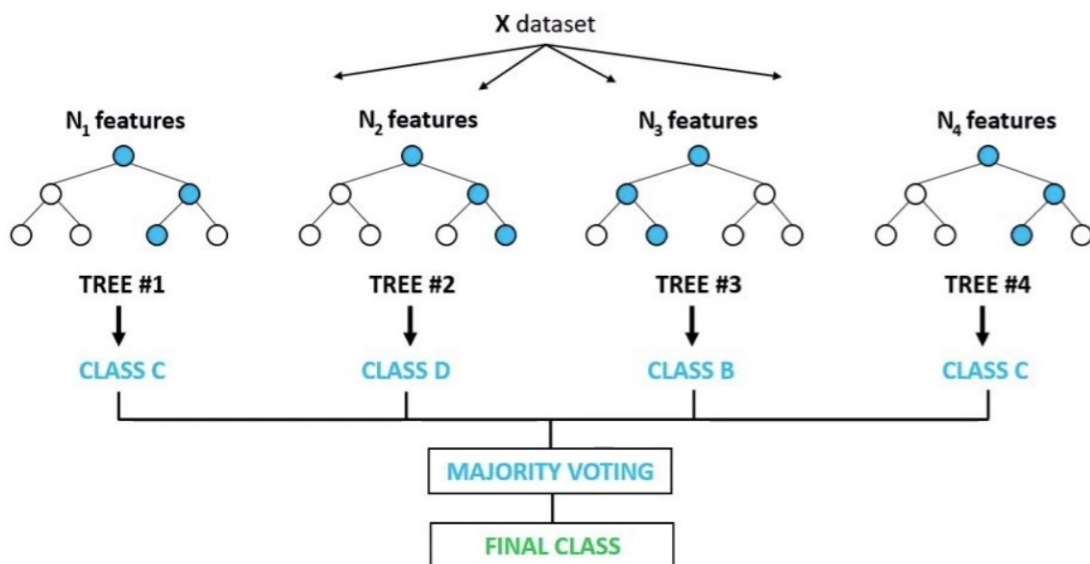


Figure 2.4: Decision Trees in Random Forest

## 2.3 Research Workflow

Before we could collect, classify, and use the Landsat 8 datasets to make growth predictions, we were required to complete a number of steps. Before the collected raw data could be utilized, it required processing and correction. For this purpose, the SmileCart classifier was utilized on the Google Earth Engine Platform. After being identified by this classifier, our data's pixels were categorized. Downloading the data with the first seven bands and then classifying it with the aid of a Qgis plugin for semi-automatic classification was a second method we employed. This plugin was able to analyze the collected data and divide it into five distinct regions labeled "Tin Shed," "Concrete," "Vegetation," "Water Body," and "Unused land," respectively. After the data processing was completed, the Google Earth, Earth Engine, and Qgis tif files needed to be recategorized. On ARCGis, the reclassification was performed, and then the pertinent data was imported into the TerrSet application. TerrSet gives users access to a land change modeler plugin, which was ultimately used to compute the evolution of land use over time. To generate the anticipated change in land use, we first employed the Markov chain to develop a Transition matrix. On the generated Matrix, an ML prediction model was applied to make predictions, which were then compared to the actual data. Cellular Automata was the preferred method for training and evaluating the ML model. TerrSet contains a CA Markov Plugin, and it was this plugin that was utilized. To test the data, however, will require a very consistent and continuous data set. Given that the population census occurs only once every ten years, while we are attempting to map data annually, this may prove difficult. When a single, highly distinct data set is available, regression models perform exceptionally well. A simplified flowchart of our work process would look as follows:

The authors of a recently published article [29] used SVM, which turned out to be the most successful machine learning model, to make predictions regarding the growth of the city of Isfahan in Iran. Because SVM classifies data by locating hyper planes that most effectively partition classes, it's possible that this approach will prove more effective. If performed on such a large volume of data, neural networking may require a more powerful central processing unit (CPU), but if quality can be guaranteed, it may provide the highest rate of accuracy. Cellular automata is another prevalent method for urban growth forecasting. SLEUTH is a cellular automaton computer simulation model that utilizes historical land use data in addition to other information in order to make predictions regarding future population growth and changes in land use [16]. The model was first presented twenty years ago, and given that it makes use of historical data, it has been implemented in a great deal of research ever since. There is a plugin available on Qgis that helps in performing this method of prediction. However, the SLUETH plugin has not been updated in the last 3 years and so it is not supported by any of the newer versions of Qgis and is even seen to malfunction on the older versions of Qgis available in the archives. Furthermore it is much more time-consuming than the other machine learning (ML) algorithms; consequently, it can be combined with GA (Genetic Algorithm) to engineer chromosomes with nodes that contain the values of the indices, and then the best chromosomes are selected. This results in a significant reduction in the amount of time consumed. Taking in account all of these factors our team decided to proceed with the simple Cellular Automata Model, by incorporating the



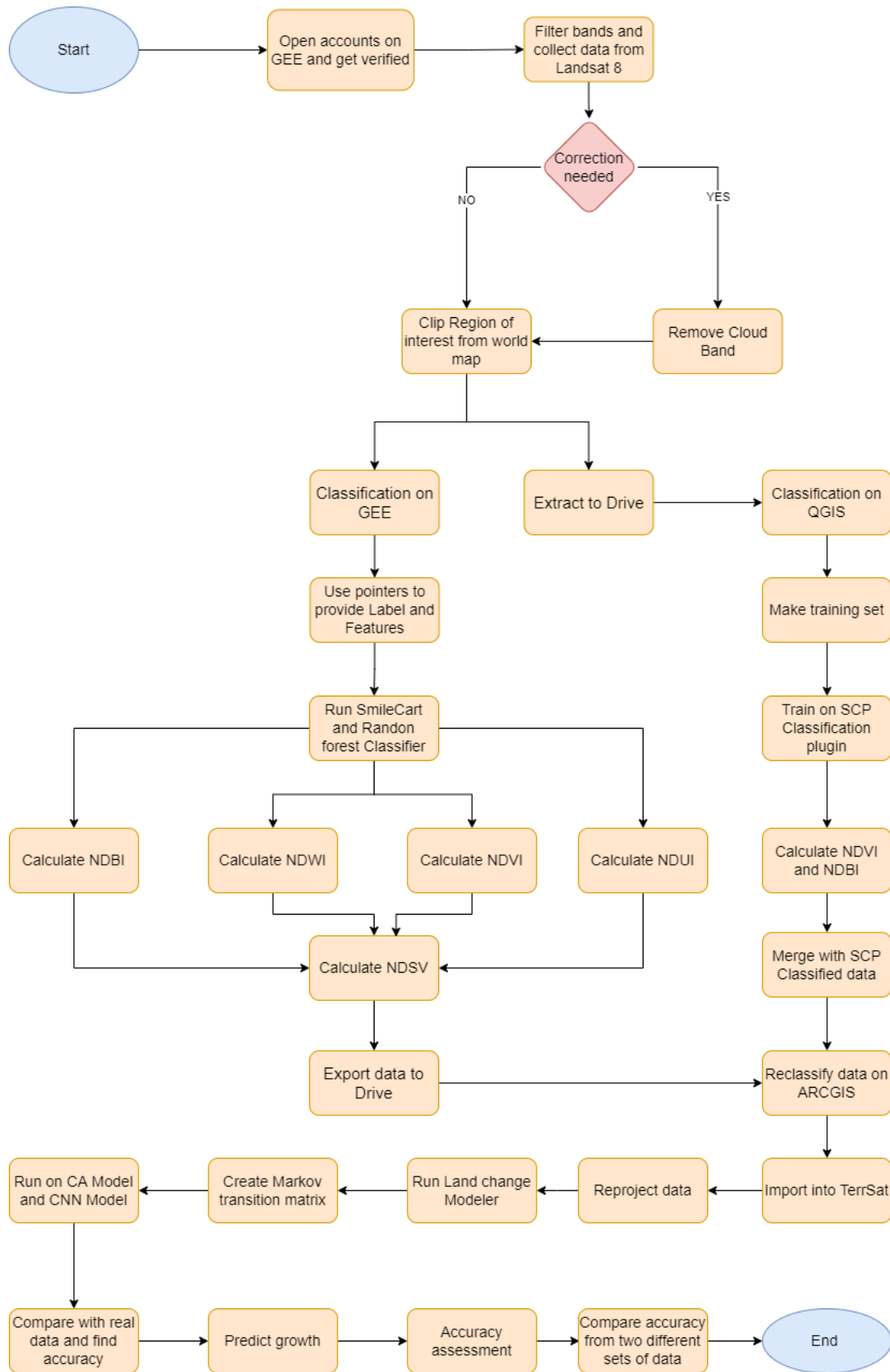


Figure 2.5: Data Collection and Prediction of Urban growth.

transitional matrix generated from a markov chain. Following the application of the model to the data, an accuracy evaluation will be carried out based on confusion matrix to evaluate the accuracy of the generated image.

# Chapter 3

## Data Collection and Processing

### 3.1 Data Collection

Surplus amounts of EOD (Earth Observation Data) confining of satellite images are taken from a popular platform called Landsat as well as other geographic data which are stored in the cloud-based GEE platform that have been used here. The Landsat program managed by NASA and the U.S geological survey is a succession of earth observing satellites expeditions. For this research, we have used “USGS Landsat 8 Collection 2 Tier 1 TOA Reflectance” which is considered suitable for time-series process analysis. The transition delay time from Real-Time to Tier 1 or Tier 2 is approximately between 14-28 days.

So far data images ranging from 2013-2021 keeping the division of Dhaka as well as Mirpur area which consisted of Wards-11, 12, 13 and 14 as our regions of interest have been gathered annually. We also collected data on a monthly basis, however as monthly data tends to miss several thousands of pixels we are using annual data that are complete.

### 3.2 Data Pre-Processing

Firstly we clipped shape files of 7 divisions from the Bangladesh administrative-1 map with the help of ArcGIS and set it as our region of interest(ROI). Seven shapefiles from 7 divisions of Bangladesh have been clipped for this purpose from the administrative-1 map of Bangladesh and the Mirpur wards were collected from administrative-5 map. Then we did a supervised classification of 5 features using three separate algorithms. We collect the training data in the ROI using the GUI of GEE. To train our data we used GEE’s built-in Classifier named SmileCart and Random Forest Classifiers. SmileCart is the updated version of the CART model that came in the GEE’s classifier package [23]. GEE’s JavaScript code does the work of training and classification so conveniently and easily. General Work-flow of GEE’s Classifiers:

- Gather training information. Assemble features with a property that stores the known class name and properties that store the predictors’ numeric values.
- Create a classifier object. If required, change the settings.

- Training the classifier using the training data
- Sort an image or feature collection into categories.
- Using separate validation data to calculate classification error.

We used the data from Google earth engine with 7 bands on Qgis to classify using a plugin named "Semi-automatic Classification Plugin" or SCP. The steps of classifying data in the SCP was:

- Import .tif files on Qgis.
- Select the desired year on the Qgis, in our case 2015.
- Overlay Google Street map form 2015 using Street view API.
- Select regions of Tin shed, Concrete, Vegetation, Water Body and Unused Land by using the overlay as a reference.
- Create adequate training ROIs under each feature and start the plugin.
- Create a NDVI and NDBI map of the tiff file.
- Merge the NDVI and NDBI map with the classified map..
- Using confusion matrix by random ROI creation to assess the accuracy of the classified data.

### 3.2.1 Collecting Road data

The road data was collected from OpenStreetMap as a .osm file. Initially, the data was fetched from OpenStreetMap in rectangular shape. Therefore, using the QuickOSM plugin in QGIS, the data was cleaned. The inclusion criteria that has been maintained are as follows:

- The highway road networks were added
- Only line vector data was added as roads. Polygon, strings, and nodes were excluded.
- The bicycle roads were added as well.
- Networks like riverbank line and airway lines were excluded.

Therefore, after cleaning the road network layer, the rectangle layer was clipped using the shape file of our desired area Dhaka division. Therefore, this osm vector layer was exported as a shape file raster data.

### 3.2.2 Slope Data

In case of the slope data the Digital Elevation Model of 2000 of the region of Bangladesh was collected and then the data was clipped by the ROIs. The slope was later calculated on the Arcgis software to obtain the slope raster file. The collected data are shown below:

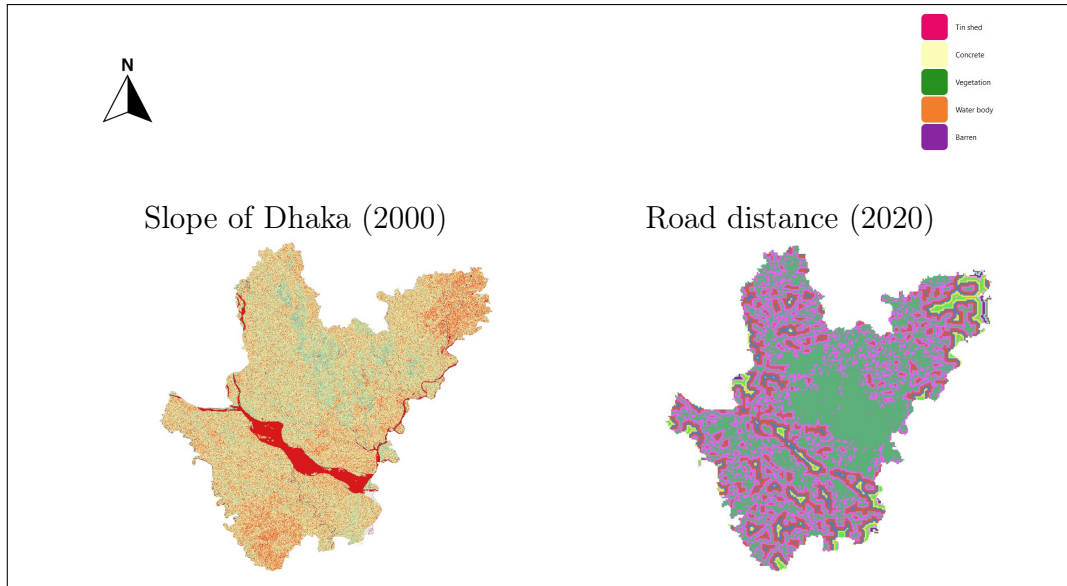


Figure 3.1: Slope and Road Distance of Dhaka.

These were collected from open street maps and from google earth engine as mentioned before.

### 3.3 Processing LULC and Feature Collection

For predicting urbanization, five factors (Tin shed, Concrete, water body, vegetation, bareland) were considered and those were taken as feature collection to enable operation on the entire set as rendering, sorting, filtering and others. The landsat data are filtered by collections with `.filterBounds`, time period set through `.filterDate`, taking minimal cloud effects from satellite images by `.filterMetadata`. Afterwards layers are added to the map through bands(B2,B3,B4). Required bands are further selected for training the dataset with SmileCart Classifier. A brief description of the bands selected are followings:

Bands	Description
B1	Coastal AAerosol
B2	Blue
B3	Green
B4	Red
B5	Near infrared
B6	Shortwave infrared 1
B7	Shortwave infrared 2

Table 3.1: Bands Description of Landsat 8

### 3.3.1 Classification on Google Earth Engine

We used two different models on Google Earth Engine when classifying the data, one was the Random forest model and the other was the SmileCart model. In order to classify Landsat daytime image of Dhaka, some pointers were used to select places of each class. We selected 25-30 points for each of the classes. These selected points were used by the RF model to predict the thousands of other pixels based on the values of the bands in the selected pixels. The code snippet for the random forest model is given below:

#### Flow chart for RF Classifier

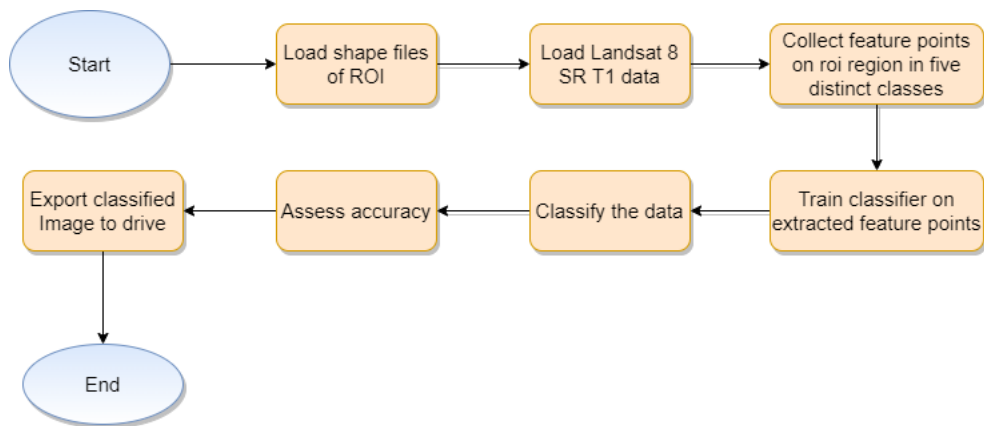


Figure 3.2: Random Forest Classifier workflow.

Another classifier exclusively found on GEE is the SmileCart classifier. We also used this classifier to identify and classify the five different features, Tin-shed, Concrete, Vegetation, Water body and Unused Land. The process was the same but in this case the model used was different. As CART classifiers are considered a more stable version of regression models, we were interested to know how well it would perform on geospatial data. The code snippet for the SmileCart Classifier used was:

#### Work flow for SmileCart

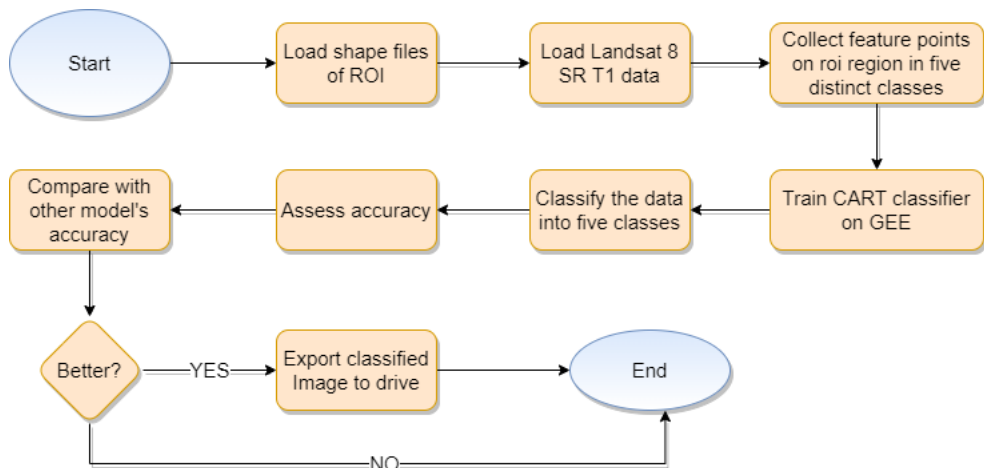


Figure 3.3: SmileCart Classifier workflow.

Lastly, after classifying images, images are exported to drive using `Export.image.toDrive()` in .tiff format in both of these cases.

### 3.3.2 Data classification on Qgis

Another approach to handle the data classification used was using the Semi-automatic Classification Plugin (SCP) on Qgis. To use this method of classification we only download data from the google earth engine of Landsat 8 with 7 bands. The code segment used for that purpose was:

#### Code Snipet

```

1  **** Start of imports.****/
2  var roi = //load feature map/ roi shape file of Dhaka,
3      L8 = //load landsat 8 dataset,
4      mirpur = //load feature map/ roi shape file of Mirpur;
5
6  var dataset = //Filtering landsat images by year, filtering cloud coverage and clipping to roi
7
8  var visualization = {
9      //Visualize the data on map to check
10 };
11 Map.setCenter(90.4123, 23.8026, 8);
12 Map.addLayer(dataset, visualization, "L8_Composite")
13 //Map.addLayer(roi)
14 //Map.addLayer(dataset2, visualization2, "L8_Composite2")
15 Export.image.toDrive({
16     //Export to drive folder to download later
17 });

```

After collecting this data we had to perform the classification on Qgis. Importing the tiff file on Qgis, we created a training dataset. There were five classes and we selected several samples from our ROI under each class. After collecting sufficient samples we proceeded to use the SCP to classify the data. After classification as it was that vegetation and built up was under recorded on the data, we had to calculate the NDVI, NDWI and NDBI using the formula:

$$NDVI = \frac{NIR - RED}{NIR + RED} = \frac{B5 - B4}{B5 + B4} \quad (3.1)$$

$$NDWI = \frac{G - NIR}{G + NIR} = \frac{B3 - B5}{B3 + B5} \quad (3.2)$$

$$NDBI = \frac{SWIR - NIR}{SWIR + NIR} = \frac{B6 - B5}{B6 + B5} \quad (3.3)$$

Here the NIR indicates Band 5, R indicates Band 4, G indicates Band 3 and SWIR indicates Band 6. After calculating the values of each pixel, we identify an appropriate filter value for each of these indexes. The raster calculated assigns all the pixels a value of NDVI, NDWI and NDBI however, not all pixels can be vegetation or water or built-up. Therefore we select a threshold value by trial and error by comparing the value with the overlay of google streets as well as the previously classified raster.

Year: 2015

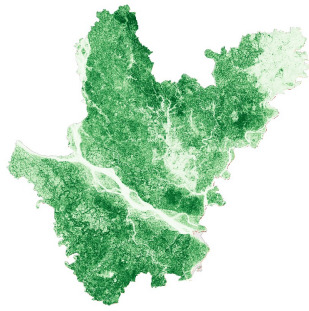


Figure 3.1: Initial NDVI

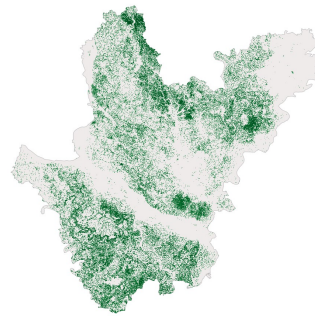


Figure 3.2: Considered NDVI

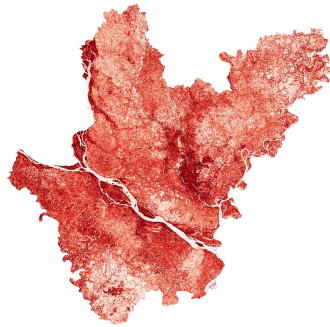


Figure 3.3: Initial NDBI



Figure 3.4: Considered NDBI

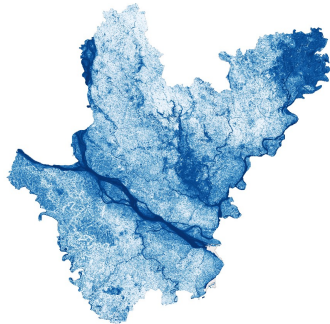


Figure 3.5: Initial NDWI



Figure 3.6: Considered NDWI

These NDVI, NDWI and NDBI values are then incorporated into the previously classified image on SCP to obtain the desired output of classified data by merging the four separate rasters. The change in the values recorded on the raster map is clearly evident.



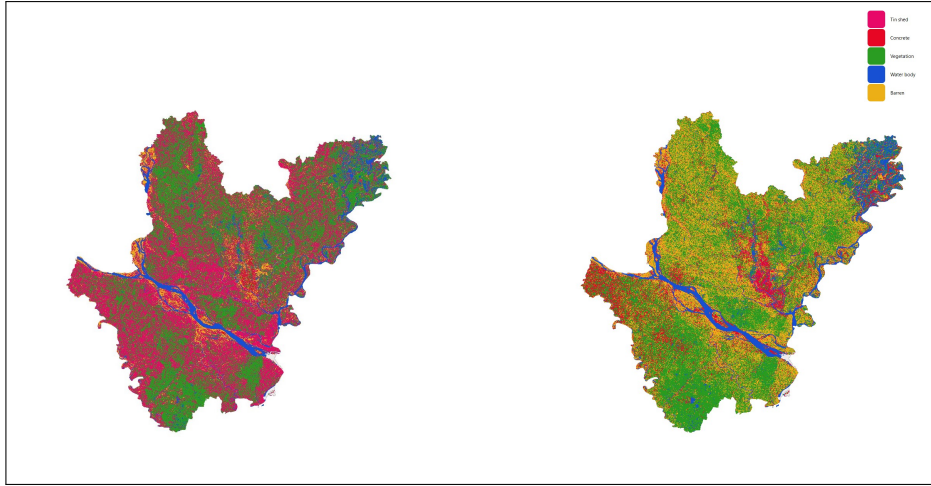


Figure 3.7: Initial Classification Vs. After merging the index values

As it can be seen that a lot of areas that are vegetation and unused were recorded as Tin Shed by the semi-automatic classification model, one key reason behind this is the color of tin sheds used in Bangladesh. From our work we have presumed that as the color of the tins used in Bangladesh closely resemble the color of dark greens and red, the surface reflectance value can be misread by the ML models. This was also the case when identifying water bodies in Dhaka as the color of the water is green. This confuses the model and some areas that are farming land are classified as water bodies (Seen on the North east part of Figure 14). Although slight modification and correction was possible in these cases by incorporating the index values, some of the inaccuracies still remained.

### 3.3.3 Data processing on ArcGIS

After all these were collected they had to be re-classified into 5 distinct classes. The five classes represented: At the very last stage of processing these data were

Bands	Description
1	Tin Shed
2	Concrete
3	Vegetation
4	Water Body
5	Unused Land
0	Background

Table 3.2: Five Feature Classes

converted to ascii format to be imported into TerrSet. The TerrSet software imports ascii files and converts them to idrisi format. The same processes were followed to process data of a much smaller region in Dhaka city, Mirpur.

### 3.4 Data Accuracy

While processing data for collection we classified the data using three different methods. All these methods are suitable for their specific fields. The abundance of data on each of these raster images gave a very high accuracy on the Random Forest Classified data and the Qgis classified data. We obtained an accuracy of 50% on the random forest model while the Qgis SCP plugin classified files showed an accuracy close to 87%. On the other hand the data that was classified by the SmileCart classifier showed an accuracy of 100% and the Kappa Coefficient was 1. Although SmileCart showed a far greater accuracy, it is very unlikely that a data this huge will have a perfect accuracy of 100. There are thousands of pixels and it is unrealistic to take this result as an actual accuracy score. That is why we chose to disregard SmileCart Classified Images here.

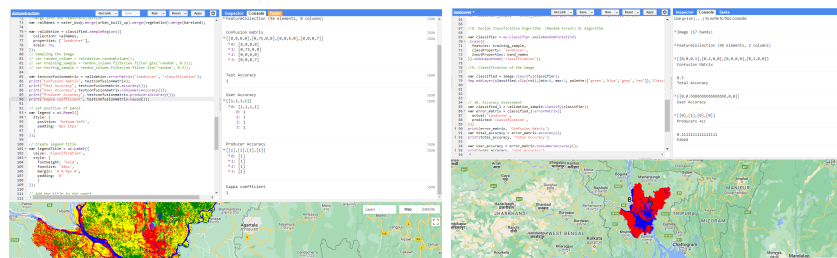


Figure 3.4: RF Model Accuracy (Left) and SmileCart Model accuracy (Right)

The error matrix generated by the SCP plugin accuracy assessment showed:

V_Classified	-999	1	2	3	4	5	Area	Wi
-999	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.03	0.00	0.00	0.00	0.00	0.06	0.03
2	0.00	0.00	0.16	0.00	0.00	0.00	0.30	0.16
3	0.00	0.00	0.00	0.35	0.00	0.13	0.88	0.48
4	0.00	0.00	0.00	0.00	0.07	0.00	0.13	0.07
5	0.00	0.00	0.00	0.00	0.00	0.24	0.44	0.24
Total	0.00	0.03	0.16	0.36	0.07	0.37	1.81	
Area	0	0	0	1	0	1	1	
SE	0.00	0.00	0.00	0.00	0.00	0.00		
SE area	0	0	0	0	0	0		
95% CI area	0	0	0	0	0	0		
PA [%]	nan	99.99	98.95	100.00	99.99	64.64		
UA [%]	nan	100.00	100.00	72.71	100.00	100.00		
Kappa hat	nan	1.00	1.00	0.58	1.00	1.00		

Table 3.3: Area Based Error Matrix of Classified 2020

Overall accuracy [%] = 86.6374  
 Kappa hat classification = 0.8097  
 Area unit = degree<sup>2</sup>  
 SE = standard error

CI = confidence interval  
 PA = producer's accuracy  
 UA = user's accuracy

For this reason we decided to use the data obtained from using the Qgis SCP classified data for further analysis and prediction. Another reason for not using the random forest classifier was that the data exported to google drive was very large. All the files were split into 2 or 3 parts and these could not be imported to ARCGis or TerrSet for further research work. This was a huge barrier for us, as we needed to use these softwares to predict future land use. Therefore we focused our approach on the Semi-automatic Classification Model available on Qgis. The user interface on this plugin was extremely easy to navigate and this also gave us room to train and test data well.

Model	Data Accuracy	Kappa Coefficient
SmileCart	100%	1
Random Forest	50%	0.74
Semi-automatic Classification	87%	0.8097

Table 3.4: Accuracy of Classified Data

The work flow we followed for generating the LULC map is a very solid framework for the process. The steps are efficient and does not require high efficiency computing capabilities. In the paper by Liang,J et al.[27], their data classification showed the best performance random forest, whose average overall accuracy was approximately 75.76%, only 50.69% for MIN classifier and 67.46% for CART classifier, respectively. However they did not implement the Semi-automatic classification therefore it is not available for comparison. According to google they have updated their RF and CART for better and efficient classification of the geospatial image and we theorize that being the reason behind getting higher accuracy scores on the two models compares to [27]. Again when we used these steps to classify a smaller area like Mirpur in Dhaka city, we obtained a far better result on our try. The error matrix shows;

Overall accuracy [%] = 97.0759  
 Kappa hat classification = 0.9522  
 Area unit = degree<sup>2</sup>  
 SE = standard error  
 CI = confidence interval  
 PA = producer's accuracy  
 UA = user's accuracy

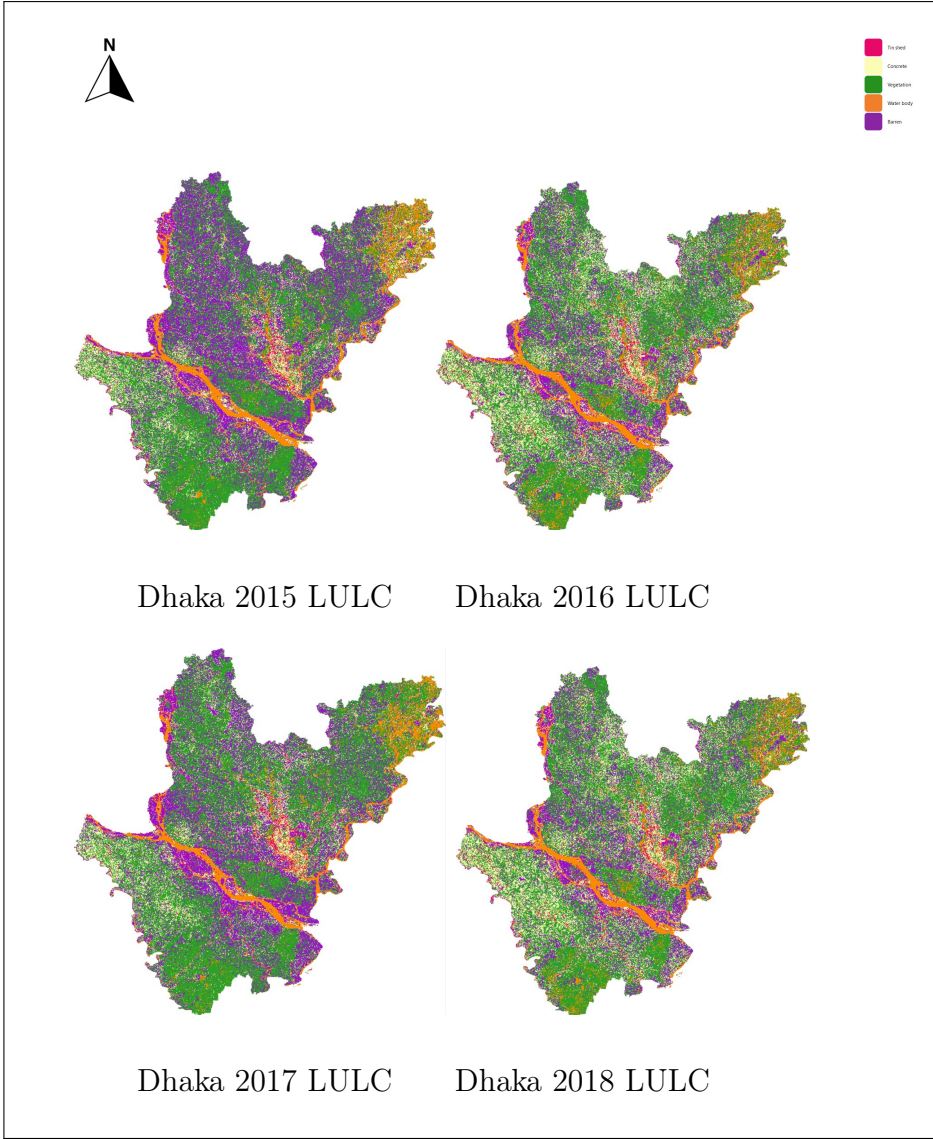
As we can see the overall accuracy of Mirpur region shows up to be 97% accurate and is a much more reliable figure compared to a larger area like the Dhaka division. Using the three software that have very user friendly interfaces provides a well structured approach in to swiftly generating a LULC map of any region of interest. If the training dataset is created with precision on SCP then we speculate that the accuracy of this framework can go even higher. As all these software have plugins that enable the overlay of street data, the procedure of collecting training ROIs become easier and more efficient. That being said, we proceeded to predict future land use and analyse the urban growth of Dhaka city.

V_Classified	-999	1	2	3	4	5	Area	Wi
-999	0	0	0	0	0	0	0	0
1	0	0.1574	0	0	0	0	0.0001	0.1574
2	0	0	0.5734	0	0	0	0.0003	0.5734
3	0	0	0	0.1406	0.0146	0.0146	0.0001	0.1698
4	0	0	0	0	0.0061	0	0	0.0061
5	0	0	0	0	0	0.0934	0	0.0934
Total	0	0.1574	0.5734	0.1406	0.0207	0.108	0.0005	
Area	0	0	0	0	0	0	0	
SE	0	0	0	0.0019	0.0014	0.0014		
SE area	0	0	0	0	0	0		
95% CI area	0	0	0	0	0	0		
PA [%]	nan	100	100	100	29.3706	86.4611		
UA [%]	nan	100	100	82.7792	100	100		
Kappa hat	nan	1	1	0.7996	1	1		

Table 3.5: Area Based Error Matrix of Mirpur 2015

### 3.5 Data Samples

The data were collected from the Region of Dhaka Division and an area inside Dhaka city in Dhaka division called Mirpur (Ward- 11, 12, 13 and 14). The Longitude and latitude of Dhaka division is 23.8103° N, 90.4125° E and of Mirpur is 23.8223° N, 90.3654° E. The Data was collected at an interval of one year to maintain data integrity to avoid collecting data with missing values. Some of the sample collected are shown below:



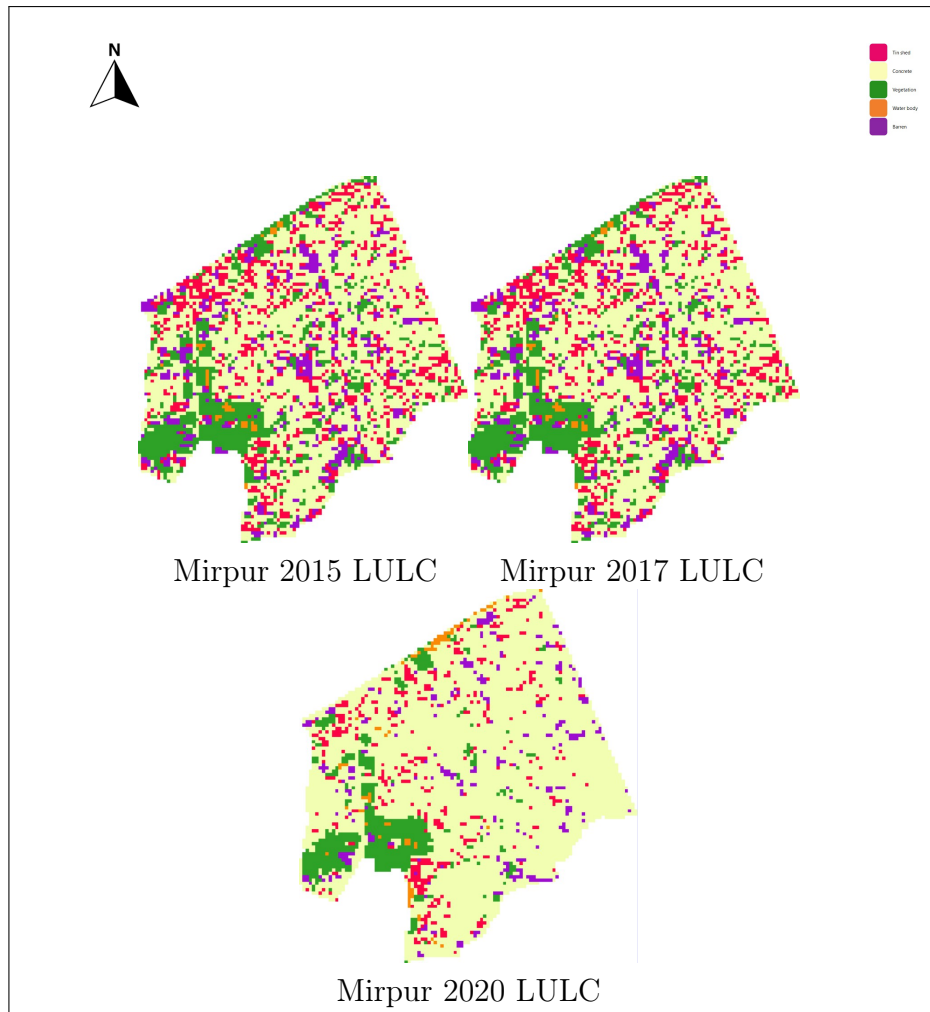


Figure 3.5: Data Samples Collected over 2015 to 2018 (Dhaka and Mirpur)

# Chapter 4

## Model

We ran our classified LULC data on the Cellular Automata Model. It was a two step procedure. The first step we had to do was generate a Markov transition Matrix. Markov chain matrix was the transitional matrix that was used as a reference for how much an area grew in the lapse of the year. The transitional matrix was then used in the cellular automata model to analyze the future change and predict the future land use map. The generated data used its neighbor cells information to change its current value and predict whether the current cell values would change to a different state in the future based on the transition matrix.

### 4.1 Cellular Automata

For the Cellular Automata approach along with the land cover dataset we also needed data on the roads and slope of the region. A simple flowchart of the process would look like:

The data was then first used in a plugin available on the TerrSet software named,

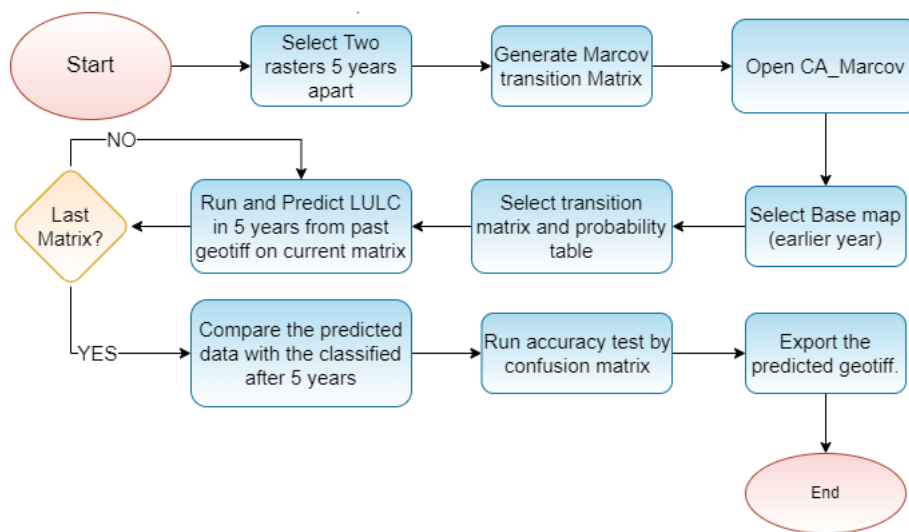


Figure 4.1: CA Model Implementation.

“Land Change Modeler”. After inputting the two data of different years, the plugin

generates graphs and heat maps that represent the change in each of the categories. A sample of these graph from changes observed over the year of 2015 to 2017 looked like:

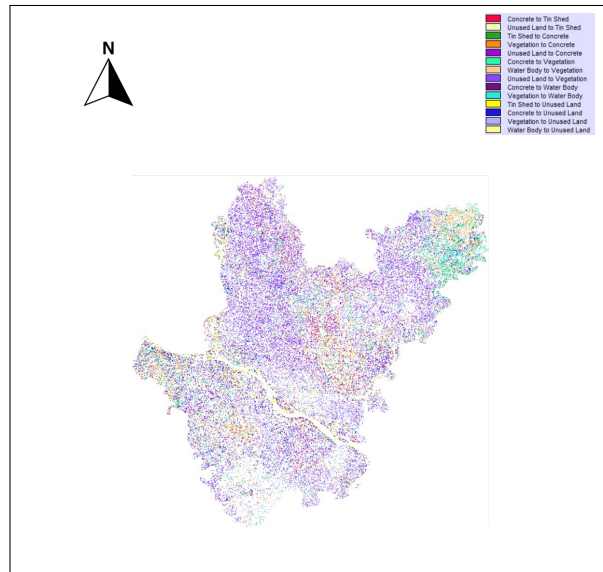


Figure 4.2: Change in Land Use over the Year 2015 to 2017.

It is interesting to observe that some areas were marked to have been converted from Concrete to Tin shed. Our hypothesis is that some rooftop tin sheds were identified as tin shed build up instead of concrete built-up, however field level data is required to solidify that was a claim. It is also seen that some areas that were concrete were changed to vegetation over the span of two years. But this can very well be rooftop-gardens that become a trend in city-dwellers in recent years, to be recorder as vegetation in the years going forward. Moving on to the next step, in order to generate a prediction map based on these data we use the Markov transitional matrix. Based on the features from the Road distance map, the slope map and the two LULC maps which are years apart from each other we generate this matrix. A transition matrix that looks like this was created as output:

Given :	Probability of changing to :				
	Tin Shed	Concrete	Vegetation	Water Body	Unused Land
Tin Shed	0.2700	0.2392	0.1311	0.0921	0.2676
Concrete	0.0514	0.3196	0.3147	0.0446	0.2697
Vegetation	0.0058	0.0819	0.7066	0.0405	0.1651
Water Body	0.0280	0.0566	0.3004	0.5323	0.0827
Unused Land	0.0259	0.1396	0.3577	0.0183	0.4585

Figure 4.3: Transition Matrix of the Year 2015 to 2017.



This transition matrix was then used in generating the probable map of Dhaka city in 2020. The data on land use of Dhaka in 2020 was already available for use and so we used that data to compare and see the difference between the predicted image and the actual land use image.

Now we also looked at an area that was much smaller in size instead of considering the area of the whole Dhaka Division. We looked at a very rapidly growing region in Dhaka, Mirpur. We took Wards- 11, 12, 13 and 14 as our region of interest this time. We classified the data again into five features, in the same way as the Dhaka region. Then Moved on to use The CA Markov model on this data. The data generated the following outcome:

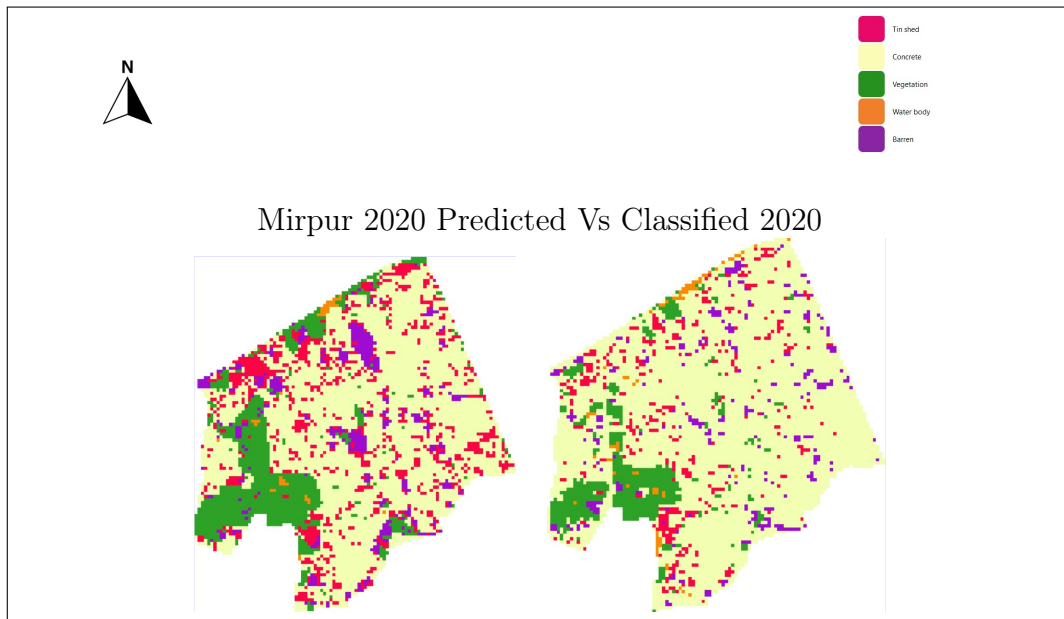


Figure 4.4: Predicted Map of Mirpur Vs. Actual Map (2020)

While in case of Dhaka this was the resultant output of the prediction model:

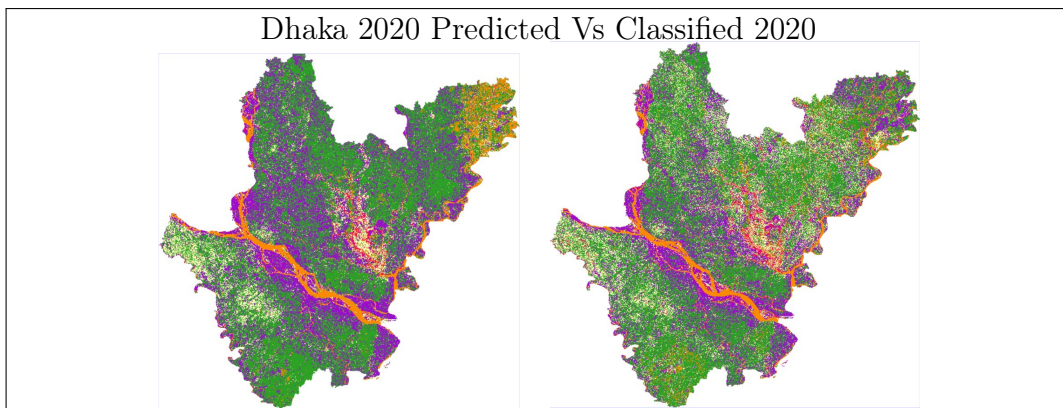


Figure 4.5: Predicted Map of Dhaka Vs. Actual Map (2020)

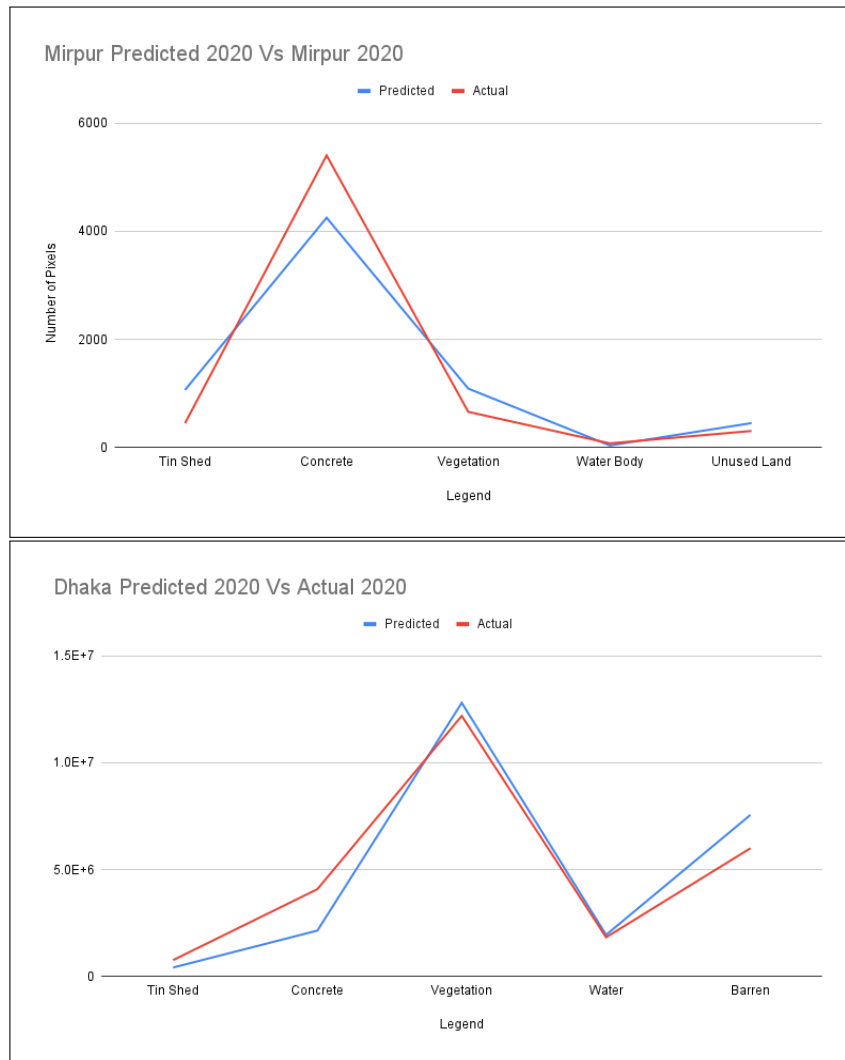
## 4.2 Result and analysis

Not only the Cellular Automata but we tried using Multilayer perceptron(MLP) on our data as well, however the accuracy generated in that case was only 23%. While Cellular Automata gave an accuracy of 80% on data of Dhaka and 86% on the data of Mirpur.

Region	Accuracy
Dhaka	80%
Mirpur	86%

Table 4.1: Accuracy of the Predicted Map in Dhaka and Mirpur

The comparison graphs of Mirpur and Dhaka show this result vividly:



Both of these show some similar trends. While both the predicted image recorded lower amount of Concrete built-up compared to the classified image, It is noticed that the Mirpur data generated more numbers of Tin shed Built-up. The number of Vegetation and Water body generated were close to the actual data. In case of Barren or unused land the data was close to accurate on the Mirpur's predicted Map while it was recorded in very high number in the Dhaka Division data. We

zoom in on the central Dhaka or the Dhaka city part of the Dhaka division to find the following: As mentioned earlier in the Predicted image the amount of Unused/

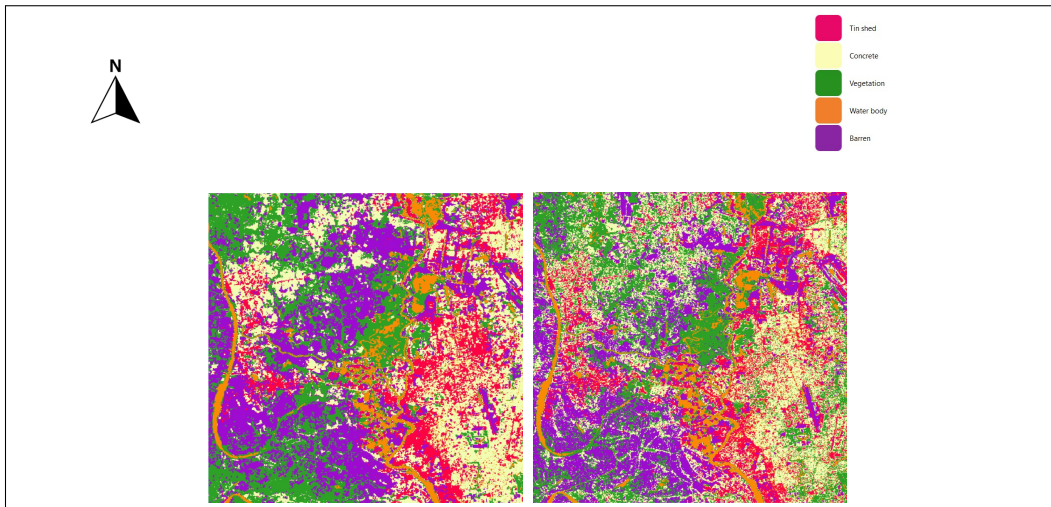


Figure 4.6: Dhaka City Predicted Vs Classified. (2020)

Barren land is much higher and this takes away form the credibility of the data. As our aim for using this data is to help in efficient planning of the city infrastructure, therefore if land ares that will be converted to concrete or other used land is recorded as unused area then that will pose as an issue when the plan in implemented. On the contrary in case of the Mirpur data the barren lands recorded were slightly less than the amount on the predicted Map. In order to generate the data we used the LULC maps from year 2015 and 2017 for the 2020 data and used 2015 as the base area map. For the data of 2025 and 2030 we used the LULC from the years 2015 and 2020 and 2015 as the base map again. The generated predicted LULC maps of Dhaka and Mirpur are given below:

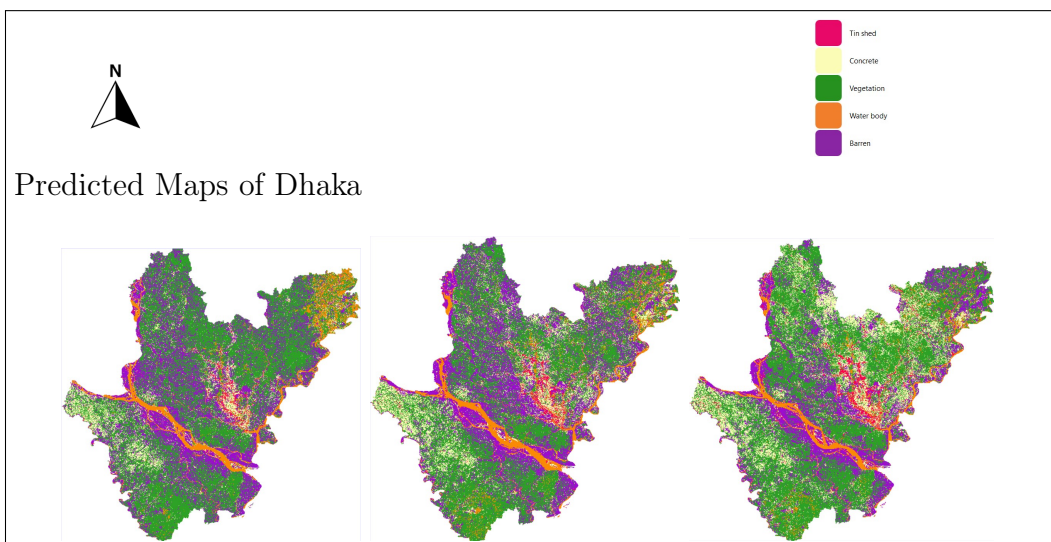


Figure 4.7: LULC 2020(Left), 2025(Center) and 2030(Right).

This shows that the model is more accurate when a smaller area of interest is considered like Mirpur instead of Dhaka. While Dhaka as ROI generated a result that

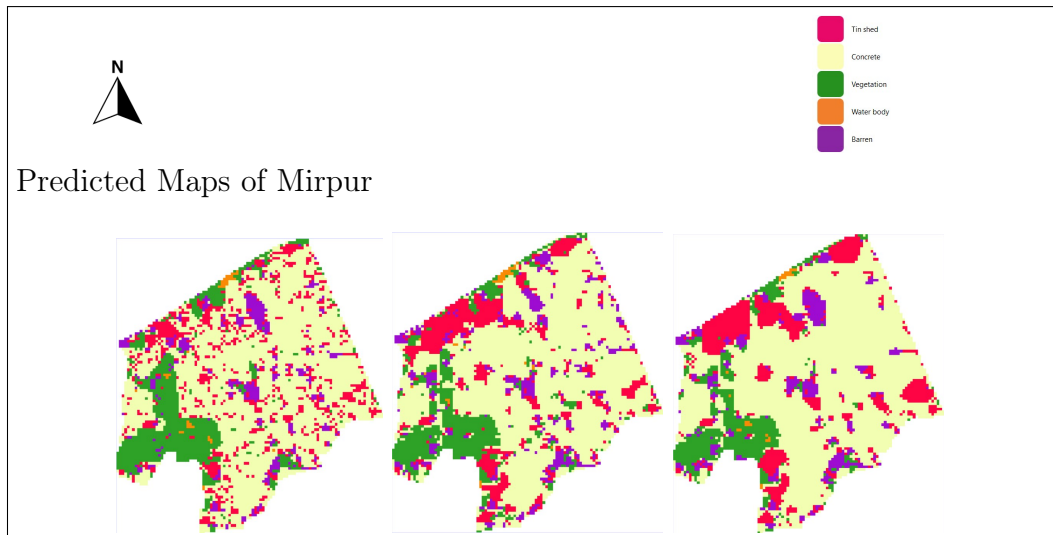


Figure 4.8: LULC 2020(Left), 2025(Center) and 2030(Right).

is 80% accurate this increased significantly in case of Mirpur. Smaller working area generated a better classified image that also has a great effect on the prediction performed by the model. Here are the pictures in numbers shows:

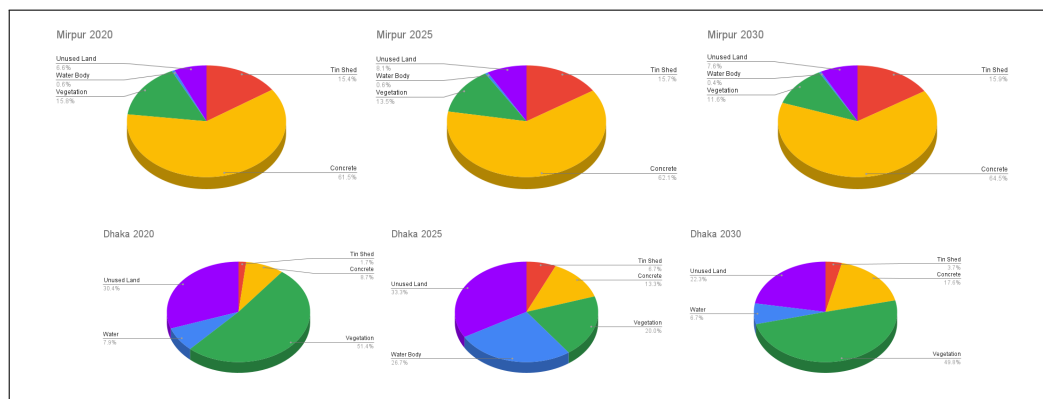


Figure 4.9: Land distribution of the predicted Images.

The data shows a trend of the concrete areas increasing steadily. The Barren land on the other hand is increasing with the decrease in vegetation. However looking at the map some areas that were concrete were later converted to vegetation on the predicted map, one reason of it could be the new urban trend of making rooftop gardens. While the water bodies remained almost the same in the coming years. In case of Gao Z, et al.,[21] they used the monocentric urban model which is more of a number driven approach that used income and other economic factors and they got a stunning accuracy of 95-96%. However, in the case of the cellular automata model, we had to use three different types of geotiff files that had varying characteristics. There was the LULC data, slope of the roi(region of interest) and Road distance in roi. All these data, although collected with well performing algorithms, have inaccuracies due to cloud coverage and position of satellite at certain time, which might be seen as a factor that contributed to the CA model giving an accuracy result

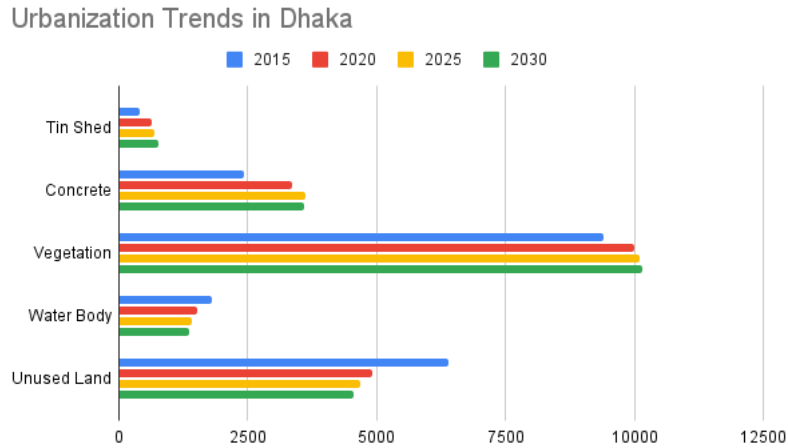


Figure 4.10: Urban Growth Trends in Dhaka.

of 86%. While in a similar ML implementation by Gomez J.A et al [22] saw a kappa efficiency score of 0.96-0.99. But their research area was Latin America and the data on Google Earth Engine of the regions under United States of America are of higher resolution and are far more frequent. To overcome our lower accuracy score, it is necessary to collect data from other sources which are not available as public property. That being the case from our study we speculate that choosing to work with a smaller ROI will provide future researchers with better results considering our goal is to work with data available at hand without any barrier.

#### 4.2.1 Future Works Prospects

This study mainly focuses on prediction of urban growth in small areas of Dhaka city for that we initially classified our collected data using smile cart and random forest in Google Earth Engine. However, for this approach needed an unimaginable accuracy rate to be assembled and worked on, in TerrSet Software along with. Thus we moved forward with LULC Classification of the composite images using the semi-automatic classification plugin on QGIS. Even though our paper gives a justified accuracy rate on the work, some areas can be explored further in future research works on similar topic. The first of which could be classifying data with the aforementioned models with relevant accuracy which was lacking in ours. Secondly, for training and testing, Cellular Automata has been used ;integrating with the Markov chain to generate a probability matrix to get prediction precision which leaves future researchers the option to try and use some other models like layered ML models for instance Neural Networks, ANN and CNN.

Thirdly, in this study unpaid dataset has been used for the work frame being accessible for beginners or freelance researchers working in the relevant fields. However there is a scope for working with the paid satellite imagery, however the cost of these high resolution daytime imagery can go up to thousands of dollars. While we did most of our work on the TerrSet software for prediction, the whole workflow could also be done on Tensorflow integrated API of Google for geospatial analysis. However, a lot of the functions and libraries available on Tensorflow are still under experimental runs.

Lastly, our research found the land use land cover change of Dhaka district over the time span 2013 to 2020. Thus other cities of different divisions could be taken into consideration. In these cases, the region of interest will have to be classified based on training set collected from that specific region.

### 4.3 Conclusion

Urbanization is a global phenomena. With long-term urban and regional development adverse effects if no thorough planning and management strategy is in place to keep track of it. However, there has been debate about how to quantify the impact of urban growth on the natural environment and what amount of urbanization is appropriate in the context of attaining environmental sustainability. Our goal through this paper is to ultimately provide relevant jurisdictions and policy makers with resources to implement proper infrastructure. In our research, we have used three ML models for processing the land use land cover data. We have gotten 100%, 50%, and 88% accuracy from SmileCart, Random Forest, and Semi-automatic Classification models. For the prediction of future land use and land cover which ultimately shows the urban growth, Cellular Automata gave an accuracy of 80% on data of the Dhaka division and 86% on the data of Mirpur. For the accuracy measurement, we compare the predicted 2020 year's data with the actual data for the year 2020. From our study, we have found that CA works better on smaller areas for prediction. Based on the LULC changes from 2015 to 2020, we have predicted that in the year 2025 Dhaka division will have 6.7% Tin Shed Build-ups, 13.3% Concrete built-ups, 20.0% Vegetation lands, 26.7% Water body, and 33.3% Unused Lands. Moreover, in 2030, the Dhaka division will have 3.7% Tin Shed built-ups, 17.6% Concrete built-ups, 49.8% Vegetation lands, 6.7% Water bodies, and 22.3% Unused Lands. In the year 2025, the Mirpur area(Ward- 11, 12, 13 and 14) of Dhaka City will have 15.7% Tin Shed built-ups, 62.1% Concrete built-ups, 13.5% Vegetation lands, 0.6% Water bodies, and 8.1% Unused Lands. Furthermore, in 2030 Mirpur area will have 15.9% Tin Shed built-ups, 64.5% Concrete built-ups, 11.6% Vegetation lands, 0.4% Water body, and 7.6% Unused Lands

In addition to, the comparisons between algorithms will help future endeavors in taking the best approach and this will save their time and reduce CPU usage. The prevalent research has shown that there exists limitations in the consistency of the data where some regions did not have images available and in other cases the data were only available annually in reduced quantity. To reduce these limitations continuous monitoring of land use change will be required in future study to better understand its effects in the regions and to develop more effective management measures.

# Bibliography

- [1] B. Czech, P. R. Krausman, and P. K. Devers, “Economic associations among causes of species endangerment in the united states: Associations among causes of species endangerment in the united states reflect the integration of economic sectors, supporting the theory and evidence that economic growth proceeds at the competitive exclusion of nonhuman species in the aggregate,” *BioScience*, vol. 50, no. 7, pp. 593–601, 2000.
- [2] J. M. Marzluff, “Worldwide urbanization and its effects on birds,” in *Avian ecology and conservation in an urbanizing world*, Springer, 2001, pp. 19–47.
- [3] H. Xu, “Modification of normalised difference water index (ndwi) to enhance open water features in remotely sensed imagery,” *International journal of remote sensing*, vol. 27, no. 14, pp. 3025–3033, 2006.
- [4] A. Schneider, M. A. Friedl, and D. Potere, “Mapping global urban areas using modis 500-m data: New methods and datasets based on ‘urban ecoregions’,” *Remote sensing of environment*, vol. 114, no. 8, pp. 1733–1746, 2010.
- [5] T. Ramachandra and H. Bharath, “Spatio-temporal pattern of landscape dynamics in shimoga, tier ii city, karnataka state, india,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 9, pp. 563–576, 2012.
- [6] L. Congedo, “Semi-automatic classification plugin for qgis,” *Sapienza Univ*, vol. 1, p. 25, 2013.
- [7] W. Li, Z. Du, F. Ling, *et al.*, “A comparison of land surface water mapping using the normalized difference water index from tm, etm+ and ali,” *Remote Sensing*, vol. 5, no. 11, pp. 5530–5549, 2013.
- [8] W. Solecki, K. C. Seto, and P. J. Marcotullio, “It’s time for an urbanization science,” *Environment: Science and Policy for Sustainable Development*, vol. 55, no. 1, pp. 12–17, 2013.
- [9] G. Lamb and N. DiLorenzo, “United nations, department of economic and social affairs, population division,” *World Urbanization Prospects: The*, 2014.
- [10] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science*, vol. 349, no. 6245, pp. 255–260, 2015.
- [11] N. N. Patel, E. Angiuli, P. Gamba, *et al.*, “Multitemporal settlement and population mapping from landsat using google earth engine,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 35, pp. 199–208, 2015.
- [12] M. M. Hassan and M. N. I. Nazem, “Examination of land use/land cover changes, urban growth dynamics, and environmental sustainability in chit-tagong city, bangladesh,” *Environment, development and sustainability*, vol. 18, no. 3, pp. 697–716, 2016.

- [13] T. Hu, J. Yang, X. Li, and P. Gong, “Mapping urban land use by using landsat images and open social data,” *Remote Sensing*, vol. 8, no. 2, p. 151, 2016.
- [14] M. M. Hassan, “Monitoring land use/land cover change, urban growth dynamics and landscape pattern analysis in five fastest urbanized cities in bangladesh,” *Remote Sensing Applications: Society and Environment*, vol. 7, pp. 69–83, 2017.
- [15] A. Midekisa, F. Holl, D. J. Savory, *et al.*, “Mapping land cover change over continental africa using landsat and google earth engine cloud computing,” *PloS one*, vol. 12, no. 9, e0184926, 2017.
- [16] K. Clarke, “Land use change modeling with sleuth: Improving calibration with a genetic algorithm,” in *Geomatic approaches for modeling land change scenarios*, Springer, 2018, pp. 139–161.
- [17] J. Dean, D. Patterson, and C. Young, “A new golden age in computer architecture: Empowering the machine-learning revolution,” *IEEE Micro*, vol. 38, no. 2, pp. 21–29, 2018.
- [18] H. Ritchie and M. Roser, “Urbanization. our world in data,” *Retrieved August*, vol. 7, p. 2020, 2018.
- [19] T. Dammalage and N. Jayasinghe, “Land-use change and its impact on urban flooding: A case study on colombo district flood on may 2016,” *Engineering, Technology & Applied Science Research*, vol. 9, no. 2, pp. 3887–3891, 2019.
- [20] Z. Gao, M. Kii, A. Nonomura, and K. Nakamura, “Urban expansion using remote-sensing data and a monocentric urban model,” *Computers, environment and urban systems*, vol. 77, p. 101 152, 2019.
- [21] Z. Gao, M. Kii, A. Nonomura, and K. Nakamura, “Urban expansion using remote-sensing data and a monocentric urban model,” *Computers, environment and urban systems*, vol. 77, p. 101 152, 2019.
- [22] J. A. Gómez, J. E. Patiño, J. C. Duque, and S. Passos, “Spatiotemporal modeling of urban growth using machine learning,” *Remote Sensing*, vol. 12, no. 1, p. 109, 2019.
- [23] Google, *Announcement: Old classifiers will stop working on march 1*. 2019. [Online]. Available: [https://groups.google.com/g/google-earthengine-announce/c/rCu4FP\\_Cn08/m/DqC192X9BAAJ](https://groups.google.com/g/google-earthengine-announce/c/rCu4FP_Cn08/m/DqC192X9BAAJ).
- [24] Á. Kertész, L. A. Nagy, and B. Balázs, “Effect of land use change on ecosystem services in lake balaton catchment,” *Land Use Policy*, vol. 80, pp. 430–438, 2019.
- [25] H. Li, W. Wan, Y. Fang, *et al.*, “A google earth engine-enabled software for efficiently generating high-quality user-ready landsat mosaic images,” *Environmental Modelling & Software*, vol. 112, pp. 16–22, 2019.
- [26] H. A. Zurqani, C. J. Post, E. A. Mikhailova, and J. S. Allen, “Mapping urbanization trends in a forested landscape using google earth engine,” *Remote Sensing in Earth Systems Sciences*, vol. 2, no. 4, pp. 173–182, 2019.
- [27] J. Liang, Y. Xie, Z. Sha, and A. Zhou, “Modeling urban growth sustainability in the cloud by augmenting google earth engine (gee),” *Computers, Environment and Urban Systems*, vol. 84, p. 101 542, 2020.
- [28] Wikipedia contributors, *Terrset — Wikipedia, the free encyclopedia*, <https://en.wikipedia.org/w/index.php?title=TerrSet&oldid=965296137>, [Online; accessed 15-September-2022], 2020.



- [29] M. Aghlmand and G. Kaplan, “European journal of geosciences,” *Journal reference: Eur. J. Geosc*, vol. 3, no. 01, pp. 01–08, 2021.
- [30] M. T. Hossain, T. Zarin, M. R. Sahriar, and M. N. Haque, “Machine learning based modeling for future prospects of land use land cover change in gopalganj district, bangladesh,” *Physics and Chemistry of the Earth, Parts A/B/C*, p. 103022, 2021.
- [31] A.-A. Kafy, M. S. Rahman, M. Islam, *et al.*, “Prediction of seasonal urban thermal field variance index using machine learning algorithms in cumilla, bangladesh,” *Sustainable Cities and Society*, vol. 64, p. 102542, 2021.
- [32] A. Shetty, P. Umesh, and A. Shetty, “An exploratory analysis of urbanization effects on climatic variables: A study using google earth engine,” *Modeling Earth Systems and Environment*, vol. 8, no. 1, pp. 1363–1378, 2022.
- [33] Wikipedia contributors, *Arcgis — Wikipedia, the free encyclopedia*, <https://en.wikipedia.org/w/index.php?title=ArcGIS&oldid=1110384349>, [Online; accessed 15-September-2022], 2022.
- [34] Wikipedia contributors, *Openstreetmap — Wikipedia, the free encyclopedia*, <https://en.wikipedia.org/w/index.php?title=OpenStreetMap&oldid=1110470839>, [Online; accessed 15-September-2022], 2022.
- [35] Wikipedia contributors, *Qgis — Wikipedia, the free encyclopedia*, <https://en.wikipedia.org/w/index.php?title=QGIS&oldid=1104710526>, [Online; accessed 15-September-2022], 2022.

# Appendix A

## Code Snippets for Classifiers on Google Earth Engine

### Random Forest Classifier

The code used on Google Earth Engine Code editor:

```
1 var geometry =
2   /* color: #fffd1c */
3   /* shown: false */
4   //Collecting feature for the five classes.
5   ee.FeatureCollection(
6     [ee.Feature(
7       ee.Geometry.Point([90.44169485478632, 23.812011747517595]),
8       {
9         "landcover": 3,
10        "system:index": "0"
11      }
12    ),
13     ...
14   ],
15   geometry2 = /* color: #d63000 */ ee.Feature(
16     ee.Geometry.Point([90.46862775293837, 23.881929466350826]),
17     {
18       "landcover": 2,
19       "system:index": "0"
20     }
21   ),
22   ...
23   geometry9 = /* color: #bf04c2 */ ee.Geometry.Point([90.39726480617124,
24     ↪ 23.796378230546267]),
25   DEM = ee.Image("CGIAR/SRTM90_V4"),
26   viz =
27     {"bands":["B4","B3","B2"],"gamma":1.7610000000000001,"max":2206,"min":711,"opacity":1};
28 var dhaka = ee.FeatureCollection("users/shahriarmim01/Dhaka_Division")
29
30 var image = ee.Algorithms.Landsat.simpleComposite({
31   collection: imageCollection2.filterDate('2020-01-01', '2020-12-30'),
32   asFloat: true
33 });
34
35 var dataset = ee.ImageCollection('LANDSAT/LC08/C02/T1_L2')
36   .filterDate('2021-01-01', '2022-01-01');
37
38 Map.centerObject(dhaka, 11);
39
40 //Merge featurecollection
41 var classNames =
42   ↪ urban.merge(water).merge(forest).merge(agriculture).merge(tinshed_builtups).merge(road);
43 var bands = ['B2', 'B3', 'B4', 'B5', 'B6', 'B7'];
44 var training = image.select(bands).sampleRegions({
45   collection: classNames,
```

```

41   properties: ['landcover'],
42   scale: 30
43 });
44 var random_column = dataset.randomColumn(); // is used to split data
45 var training_sample = random_column.filter(ee.Filter.gte('random',0.7)); // 70% (70/100) gte
↪   training
46 var validation_sample = random_column.filter(ee.Filter.lte('random',0.45)); // 30%
47
48
49 //8. Decide Classification Algorithm (Random Forest) ML Algorithm
50
51 var classifier = ee.Classifier.smileRandomForest(50)
52 .train({
53   features: training_sample,
54   classProperty: 'landcover',
55   inputProperties: bands
56 }).setOutputMode('classification');
57
58 //9. Classification of the image
59
60 var classified = image.classify(classifier);
61 Map.addLayer(classified.clip(roi),{min:0, max:3,
↪   palette:['green','blue','grey','red']},'Classified Image RF');
62
63
64 // 10. Accuracy Assessment
65 var classified_1 = validation_sample.classify(classifier);
66 var error_matrix = classified_1.errorMatrix({
67   actual:'landcover',
68   predicted:'classification',
69 });
70 print(error_matrix, 'Confusion Matrix')
71 var total_accuracy = error_matrix.accuracy();
72 print(total_accuracy, 'Total Accuracy')
73
74 var user_accuracy = error_matrix.consumersAccuracy();
75 print(user_accuracy, 'User Accuracy')
76
77 var pro_acc = error_matrix.producersAccuracy();
78
79 print(pro_acc, "Producers Acc")
80
81 var kappa = error_matrix.kappa();
82 print(kappa, 'Kappa');
83
84 Map.centerObject(roi)
85 var dhakaraw=dataset.median().clip(dhaka)
86 var visualization = {
87   bands: ['SR_B4', 'SR_B3', 'SR_B2'],
88   min: 0.0,
89   max: 0.3,
90 };
91
92 Map.setCenter(90.3996, 23.8224, 8);
93
94 // Map.addLayer(dhakaraw, visualization, 'True Color (432)');
95 Map.addLayer(classified, {min: 0, max: 3, palette: ['red', 'blue', 'green', 'cyan']});
96
97 Export.image.toDrive({
98   image: dhakaraw,
99   description: 'Dhaka_2021-2022RAW',
100   scale:30,
101   region:dhaka
102 });

```

# SmileCart Classifier

The code used on Google Earth Engine Code editor:

```
1 var geometry = /* color: #fffd1c */ ee.FeatureCollection(
2   [ee.Feature(
3     ee.Geometry.Point([90.44169485478632, 23.812011747517595]),
4     {
5       "landcover": 3,
6       "system:index": "0"
7     }
8   ),
9   ...
10  ee.Feature(
11    ee.Geometry.Point([90.47167417263384, 23.877352734569566]),
12    {
13      "landcover": 3,
14      "system:index": "6"
15    }
16  )
17  ],
18  geometry7 = /* color: #ffc82d */ ee.Feature(
19    ee.Geometry.Point([90.38167044295865, 23.802925821748882]),
20    {
21      "landcover": 2,
22      "system:index": "0"
23    }
24  ),
25  tinshed = /* color: #00ffff */ ee.Feature(
26    ee.Geometry.MultiPoint(
27      [[90.41122838630277, 23.80660685627567],
28      ...
29      [90.41200766045628, 23.78632216170665]]),
30    {
31      "landcover": 2,
32      "system:index": "0"
33    }
34  ),
35  geometry9 = /* color: #bf04c2 */ ee.Geometry.Point([90.39726480617124,
36    ↪ 23.796378230546267]),
37  DEM = ee.Image("CGIAR/SRTM90_V4"),
38  imageVisParam = {"opacity":1,"bands":["B1"],"gamma":1},
39  imageVisParam2 = {"opacity":1,"bands":["B1"],"gamma":1},
40  imageVisParam3 = {"opacity":1,"bands":["B1"],"gamma":1},
41  imageVisParam4 = {"opacity":1,"bands":["classification"],"min":0,"max":3,
42    ↪ "palette":["0000ff","ff0000","008000","ffff00"]};
43
44 var roi = ee.FeatureCollection("users/shahriarmim01/Dhaka_Division"),
45 L8 = ee.ImageCollection("LANDSAT/LC08/C02/T1_TOA"),
46 urban_built_up =
47   /* color: #ff0000 */
48   /* shown: false */
49   geometry5,
50 water_body =
51   /* color: #00cad9 */
52   /* shown: false */
53   geometry6,
54 vegetation =
55   /* color: #20ff74 */
56   /* shown: false */
57   ee.FeatureCollection(
58     [tinshed,
59     ee.Feature(
60       geometry9,
61       {
62         "landcover": 2,
63         "system:index": "1"
64       }
65     ),
66     geometry7,
67     geometry4,
68     geometry3,
69     geometry2]),
70 bareland =
71   /* color: #fffd1c */
```

```

65     /* shown: false */
66     geometry;
67
68     var image = L8.filterBounds(roi)
69         .filterDate('2021-01-01','2022-01-01')
70         .filterMetadata('CLOUD_COVER','less_than',1)
71         .mean()
72         .clip(roi);
73
74 Map.addLayer(image, {bands:['B1']},'TrueColor');
75
76
77 //Merge into one FeatureCollection and print details to console
78 var classNames =
79 ↪ water_body.merge(urban_built_up).merge(vegetation).merge(bareland).merge(tinshed);
80 print(classNames);
81
82 //Extract training data from select bands of the image, print to console
83 var bands = ['B2', 'B3', 'B4', 'B5', 'B6', 'B7'];
84 var training = image.select(bands).sampleRegions({
85     collection: classNames,
86     properties: ['landcover'],
87     scale: 30
88 });
89 print(training);
90
91 //Train classifier - e.g. cart,
92 var classifier = ee.Classifier.smileCart().train({
93     features: training,
94     classProperty: 'landcover',
95     inputProperties: bands
96 });
97
98 //Run the classification
99 var classified = image.select(bands).classify(classifier);
100
101 //Centre the map on your training data coverage
102 Map.centerObject(classNames, 11);
103 //Add the classification to the map view, specify colours for classes
104 Map.addLayer(classified,
105 {min: 0, max: 3, palette: ['0000FF', 'FF0000', '008000','FFFF00']},
106 'classification');
107
108 //Merge into one FeatureCollection
109 var valNames = water_body.merge(urban_built_up).merge(vegetation).merge(bareland);
110
111 var validation = classified.sampleRegions({
112     collection: valNames,
113     properties: ['landcover'],
114     scale: 30,
115 });
116 // Sampling the image
117 // var random_column = validation.randomColumn();
118 // var training_sample = random_column.filter(ee.filter.gte('random', 0.3));
119 // var testing_sample = random_column.filter(ee.filter.lte('random', 0.3));
120
121 var testconfusionmatrix = validation.errorMatrix('landcover', 'classification');
122 print("Confusion matrix", testconfusionmatrix);
123 print("Test Accuracy", testconfusionmatrix.accuracy());
124 print("User Accuracy", testconfusionmatrix.consumersAccuracy());
125 print("Producer Accuracy", testconfusionmatrix.producersAccuracy());
126 print("Kappa coefficient", testconfusionmatrix.kappa());
127
128 // set position of panel
129 var legend = ui.Panel({
130     style: {
131         position: 'bottom-left',
132         padding: '8px 15px'
133     }
134 });
135

```

```

136 // Create legend title
137 var legendTitle = ui.Label({
138   value: 'Classification',
139   style: {
140     fontWeight: 'bold',
141     fontSize: '18px',
142     margin: '0 0 4px 0',
143     padding: '0'
144   }
145 });
146
147 // Add the title to the panel
148 legend.add(legendTitle);
149
150 // Creates and styles 1 row of the legend.
151 var makeRow = function(color, name) {
152
153   // Create the label that is actually the colored box.
154   var colorBox = ui.Label({
155     style: {
156       backgroundColor: '#' + color,
157       // Use padding to give the box height and width.
158       padding: '8px',
159       margin: '0 0 4px 0'
160     }
161   });
162
163   // Create the label filled with the description text.
164   var description = ui.Label({
165     value: name,
166     style: {margin: '0 0 4px 6px'}
167   });
168
169   // return the panel
170   return ui.Panel({
171     widgets: [colorBox, description],
172     layout: ui.Panel.Layout.Flow('horizontal')
173   });
174 };
175
176 // Palette with the colors
177 var palette = ['0000FF', 'FF0000', '008000', 'FFFF00', '00FF80'];
178
179 // name of the legend
180 var names = ['Water-body', 'Urban Builtup', 'Vegetation', 'BareLand', 'Tinshed'];
181
182 // Add color and names
183 for (var i = 0; i < 4; i++) {
184   legend.add(makeRow(palette[i], names[i]));
185 }
186
187 // add legend to map (alternatively you can also print the legend to the console)
188 Map.add(legend);
189 //Exporting to gDrive
190 Export.image.toDrive({
191   image: classified.visualize({min: 0, max: 3, palette: ['0000FF', 'FF0000',
192     ↵ '008000', 'FFFF00', '00FF80']}),
193   description: 'Dhaka_2021-2022',
194   scale: 30,
195   region: roi
196 });

```

# Semi-automatic Classification

The code used to download data from Google Earth Engine was:

```
1  /* Start of imports. If edited, may not auto-convert in the playground. */
2  var roi = ee.FeatureCollection("users/rifahtasmiahislam/Dhaka"),
3      L8 = ee.ImageCollection("LANDSAT/L008/C01/T1_RT"),
4      mirpur = ee.FeatureCollection("projects/ee-rifahtasmiahislam/assets/Mirpur");
5  /* End of imports. If edited, may not auto-convert in the playground. */
6  var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1_SR')
7      .filterDate('2017-01-01', '2017-12-31')
8      .filterBounds(mirpur)
9      .filter(ee.Filter.lt('CLOUD_COVER',10))
10     .median()
11     .clip(mirpur)
12     ;
13
14 var visualization = {
15   bands: ['B5', 'B4', 'B3'],
16   min: 0.0,
17   max: 3000,
18   gamma:1.2,
19 };
20 Map.setCenter(90.4123, 23.8026, 8);
21 Map.addLayer(dataset, visualization, "L8_Composite")
22 //Map.addLayer(roi)
23 //Map.addLayer(dataset2, visualization2, "L8_Composite2")
24 Export.image.toDrive({
25   image: dataset.select(["B1","B2","B3","B4","B5","B6","B7"]),
26   description: "Mirpur_Landsat_Composite17",
27   folder: "Thesis",
28   region: mirpur,
29   scale: 30,
30   maxPixels: 10e11,
31   fileDimensions: 7680,
32   crs: "EPSG:4326"
33 });
```