

# Mapping Biogeophysical Parameters for Indian Cities Using Long-Term Satellite Data and Geospatial Technologies

Swarnalee Mazumder<sup>1</sup>, Manpa Barman<sup>2</sup>, Santonu Goswami<sup>3</sup>, and Harini Nagendra<sup>3</sup>

<sup>1</sup>University of Hamburg

<sup>2</sup>University of Stuttgart

<sup>3</sup>Azim Premji University

February 1, 2023

# Mapping Biogeophysical Parameters for Indian Cities Using Long-Term Satellite Data and Geospatial Technologies



Swarnalee Mazumder [1][2], Manpa Barman [1][3], Santonu Goswami [1], Harini Nagendra [1]

Azim Premji University, Centre for Climate Change and Sustainability, Bangalore, India [1]; University of Hamburg, Germany [2]; University of Stuttgart, Germany [3]

PRESENTED AT:

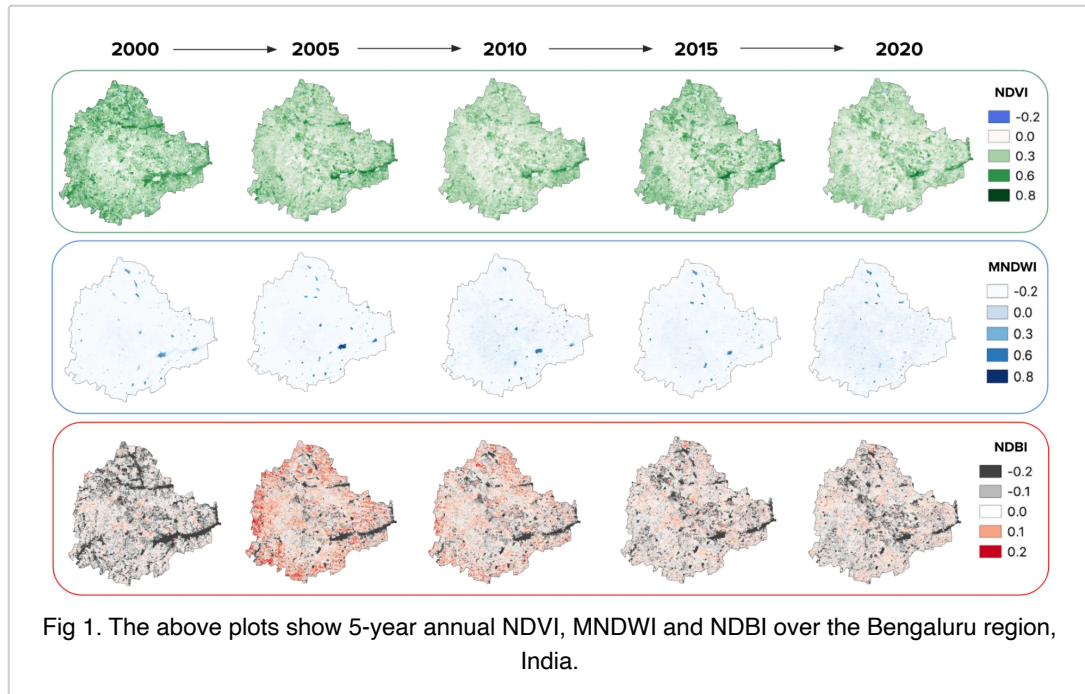


## INTRODUCTION

With the growing population and expanding economies, the world's landscape is continuously at change. To identify this change and eventually find out the underlying causes, this work has attempted to develop an **automation pipeline** that uses long-term Landsat datasets to create maps of crucial biophysical parameters.

## INDEXES AS A MEASURE OF BIOPHYSICAL PARAMETERS

The red and near-infrared radiation reflected (spectral reflectance) by terrestrial surfaces are measured by a variety of sensors carried aboard satellites. Using mathematical equations (algorithms), raw satellite data can be transformed into indices which can extract information on the state of various physical parameters on Earth.



Vegetation, for example, can be assessed using the **Normalized Difference Vegetation Index (NDVI)**. NDVI helps in i. evaluating the general health (state) of vegetation and ii. distinguishing it from other artificial land covers. NDVI defines values from -1.0 to 1.0, where negative values are mainly formed from clouds, water and snow, and values close to zero are primarily formed from rocks and bare soil. Particularly small values of 0.1 or less correspond to rocks, sand or snow.

To detect the changes in water bodies we used, **Modified Normalized Difference Water Index (MNDWI)** since it improves built-up area attributes that are frequently connected with open water in other indices and detracts from both of them. To detect the built-up areas the **Normalized Difference Built-up Index (NDBI)** has been used. NDBI uses the NIR and SWIR bands to emphasize artificial built-up areas.

The equations for the indices are as below:

$$NDVI = \frac{\text{Near Infrared} - \text{Red}}{\text{Near Infrared} + \text{Red}}$$

$$MNDWI = \frac{\text{Green} - \text{SWIR 1}}{\text{Green} + \text{SWIR 1}}$$

$$NDBI = \frac{\text{SWIR 1} - \text{Near Infrared}}{\text{SWIR 1} + \text{Near Infrared}}$$

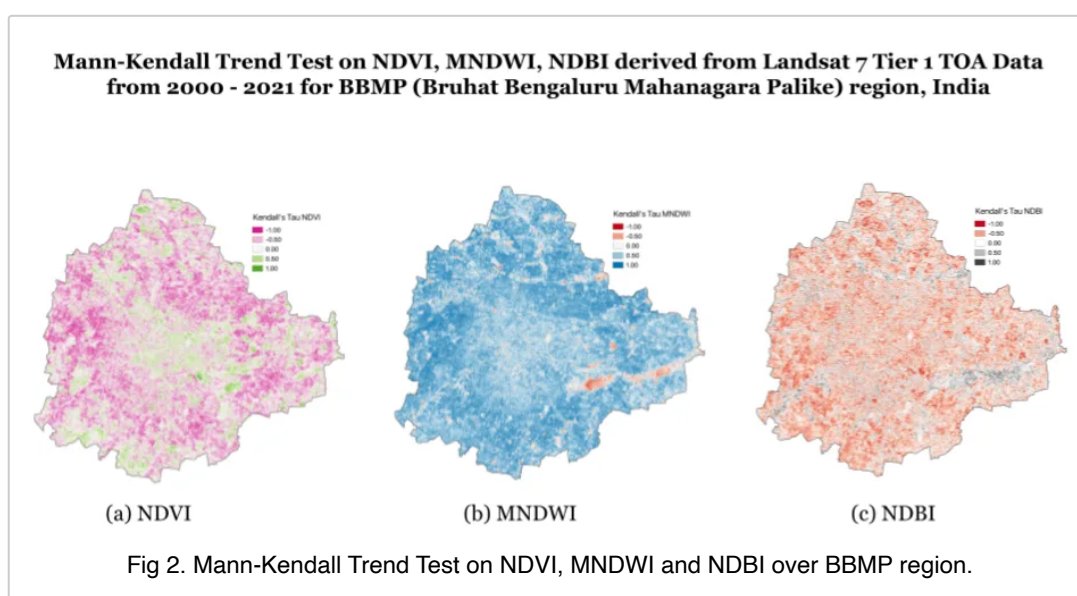
## ANALYZING THE INDICES

### Data

For initial analysis and for building the pipeline, we have used **USGS Landsat 7 Collection 2 Tier 1 TOA Reflectance** and **FAO GAUL 500m** dataset in **Google Earth Engine**.

### Creating the Indices

The analysis to map all the above-mentioned spectral indices consisted of first masking the cloud cover in the satellite images using bitwise operations. Then getting the mean image for the specified time period. Trend analysis helps in determining where some parameter of interest is rising or declining and by how much. Monotonic trends in imagery have been identified using the non-parametric Mann-Kendall test for the presence of an increasing or decreasing trend.

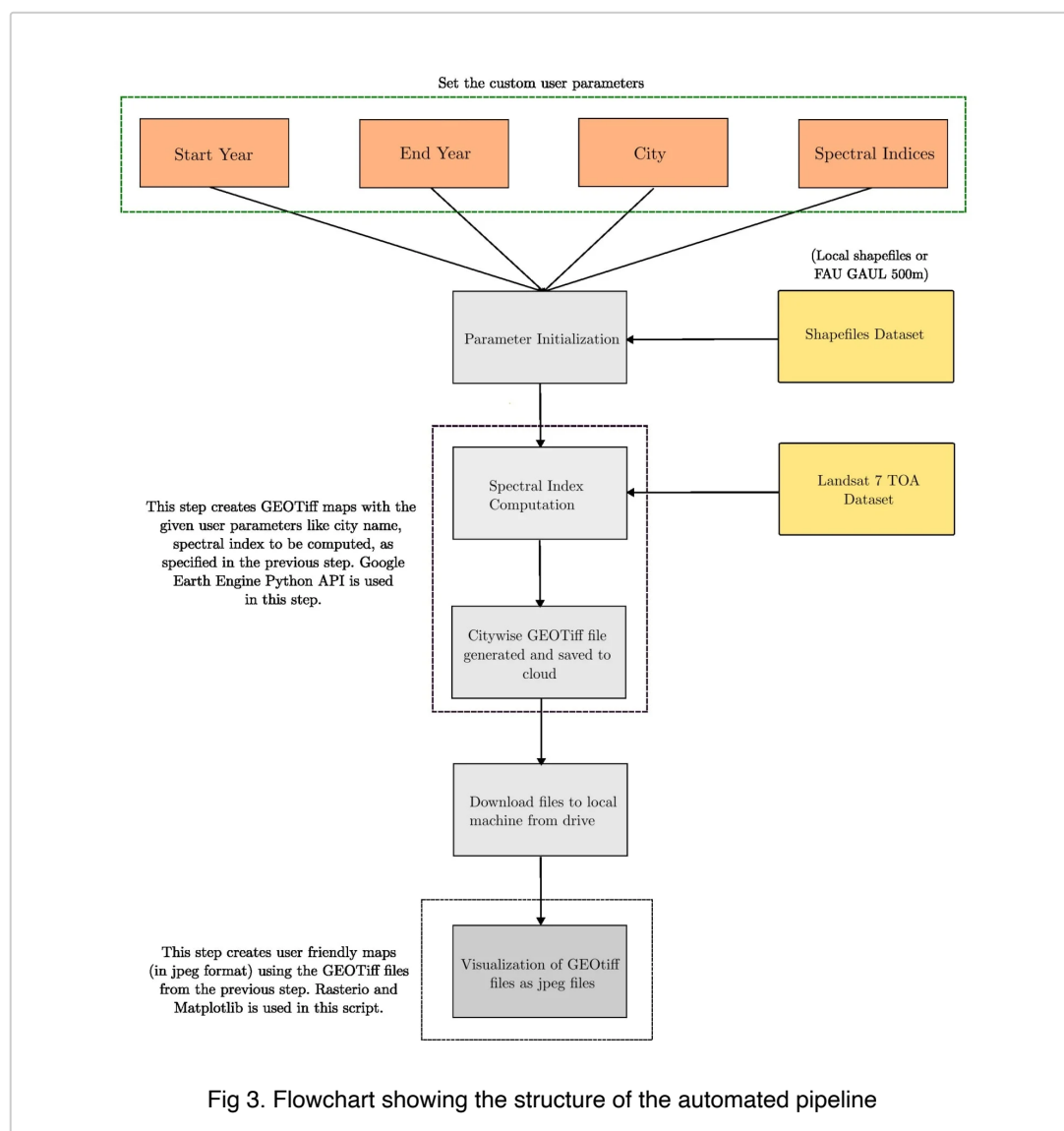


# BUILDING THE AUTOMATION PIPELINE

## Building the automation pipeline

The automation pipeline involves three major steps - Parameter Initialization, Spectral Indices Computation, and Map Visualization.

**1. Parameter Initialization:** In this step, the user specifies the inputs like the start year and end year of computation, the target city/cities, and the spectral index to be monitored. Other parameters like the colormaps of the final map to be created or the source of shapefiles of the target region can also be specified. The output of this step is a mapping of the target city name and the corresponding shapefile location.



**2. Spectral Index Computation:** The next step involves taking all the user parameters and the mapping as mentioned in the previous step and computing the specified spectral index for the given region. The dataset used in this step for the spectral index computation is the **USGS Landsat 7 Collection 2 Tier 1 TOA** Reflectance dataset. This step computes the seasonal and annual spectral index values and saves them in a city-wise folder arrangement in the cloud. The files saved in this step are in the **GeoTIFF** format. **Google Earth Engine Python API** [1] is used for this step.

**3. Map Visualization:** The Map Visualization step includes downloading all the **GeoTIFF** files and creating analytical maps according to the colormaps and specifications given by the users in the first step. This step produces user-friendly jpeg maps which can be further used for research purposes. **GDAL, rasterio, and matplotlib** libraries are used for this purpose.

The above pipeline is used to produce a huge number of maps in one iteration, given all the user inputs.

## CONCLUSION

In this work, we have carried out an analysis to characterize the long-term changes in biophysical parameters such as **vegetation, water bodies and built-up area** for any specified target city or region, given the geographical boundaries.

The final goal of the project aimed at building an **automated pipeline** which can now process long-term remotely sensed data and plot various biophysical indices eventually making the process of preliminary investigations of biophysical parameters *fast* by manifolds as well as *efficient*.



## DEMONSTRATION

The plot here shows the **automation pipeline** in action.

[VIDEO] [https://res.cloudinary.com/amuze-interactive/image/upload/f\\_auto,q\\_auto/v1670871433/agu%20fm2022/c5-13-53-03-34-ba-5f-20-3a-45-5a-9b-62-96-f3-2e/image/intro-gif\\_bmxxf1.mp4](https://res.cloudinary.com/amuze-interactive/image/upload/f_auto,q_auto/v1670871433/agu%20fm2022/c5-13-53-03-34-ba-5f-20-3a-45-5a-9b-62-96-f3-2e/image/intro-gif_bmxxf1.mp4)

Fig 4. Plot showing the **automation pipeline** in action.

*Background Image: Photo by NASA (<https://unsplash.com/@nasa>) on Unsplash (<https://unsplash.com/>).*

---

## ABSTRACT

Climate change decisions are complex, expensive, and have long-term consequences. As a result, it is critical that such judgments are based on the best available data. In our proposed work, an analysis is carried out to characterize the long term changes in biophysical parameters such as vegetation, wetland and built up area for any specified target city. A trend analysis of various spectral indices such as NDVI; to detect the change in vegetation, NDBI; to detect the change in the built index, and MNDWI; to detect the change in water bodies and the effect of climate variation has been explored for the years from 2000 to 2021. In order to perform this analysis, an automation pipeline is designed that makes use of the Earth Engine Python API. The pipeline accepts the target area and time of interest as the input and generates a map highlighting the trend in the specified spectral index over the years. The Mann-Kendall (MK) trend test and Sen's slope estimates are employed as an evaluation metric to analyze the city specific trends based on the maps obtained. Google Earth Engine datasets USGS Landsat-7 and Sentinel-2 MSI have been leveraged for the analysis. The trends have been analyzed over multiple time intervals; annual, 5-year seasonal (MAM, JJAS, SON, DJF), and a 10-year seasonal interval. Our work is an attempt to map any biogeophysical parameters and create a database for the cities of India using our automation pipeline. Furthermore, an investigation has been carried out to understand how this observational trend has been influenced by the changing temperature over the region and based on the past temperature trend, how the projected trend of NDVI, MNDWI, and NDBI is going to change under various future climate scenarios.

## REFERENCES

- [1] Wu, Q. (2020). geemap: A Python package for interactive mapping with Google Earth Engine. *Journal of Open Source Software*, 5(51), 2305.

