Analyzing Historical Turbidity in the Tennessee River Using Google Earth Engine

Connor B. Firat*, AKM Azad Hossain*, and Xiaobo Chao**

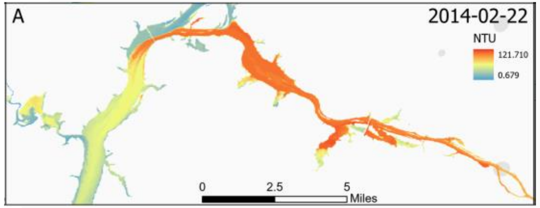
*Department of Biology, Geology, and Environmental Science, The University of Tennessee at Chattanooga

**The National Center for Computational Hydroscience and Engineering
The University of Mississippi

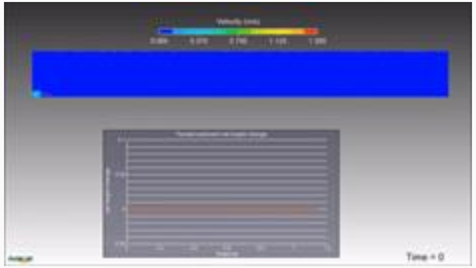
ASPRS Mid-south Regional Conference, Oak Ridge National Laboratory (ORNL)
April 21, 2023

Study Plan

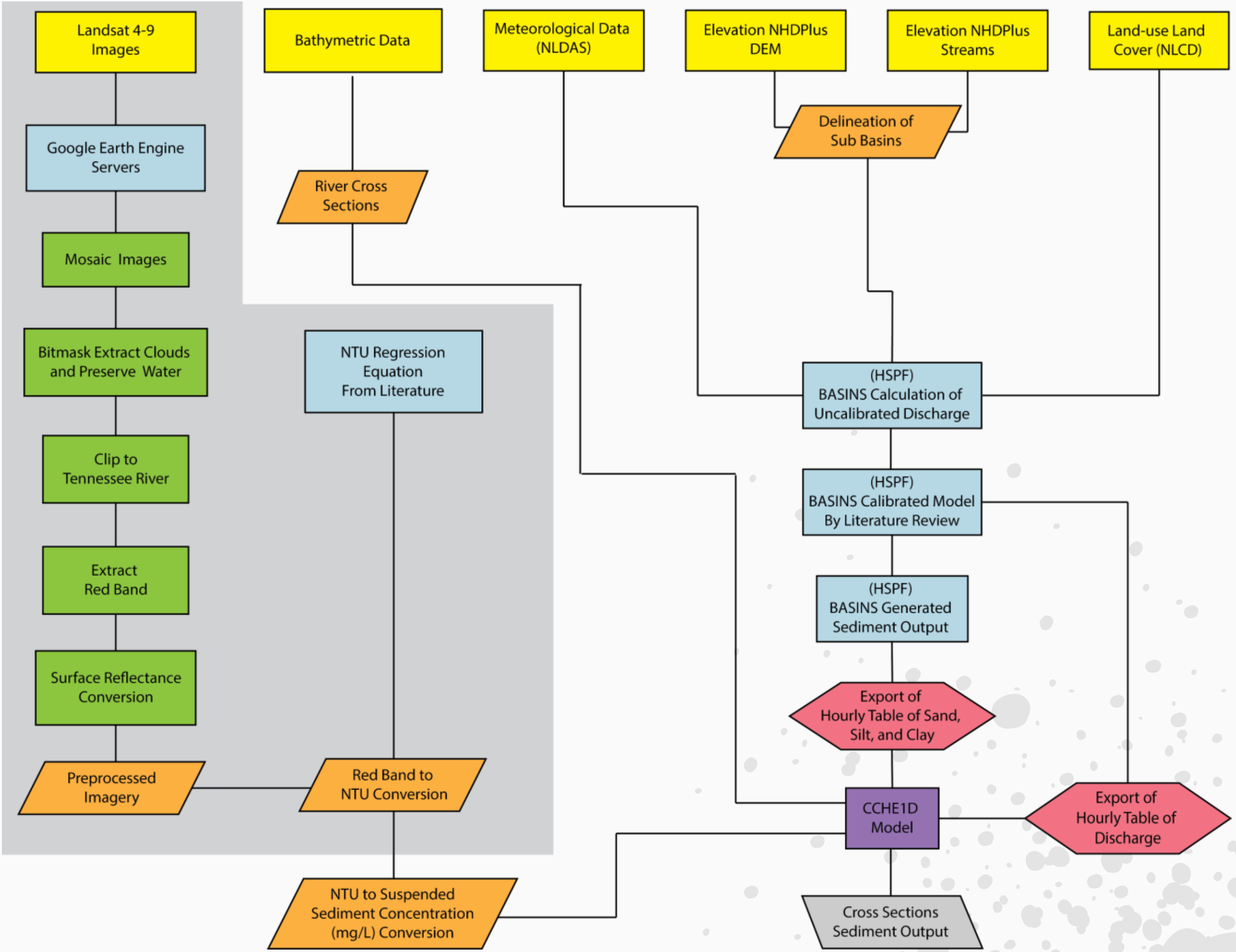
- Incorporate remote sensing data into hydrologic models to understand the transport of suspended sediment



Remote Sensing Turbidity Data



Hydrodynamic Model Data

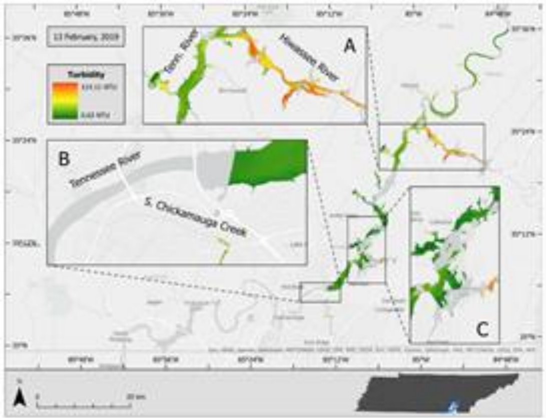


Previous Study

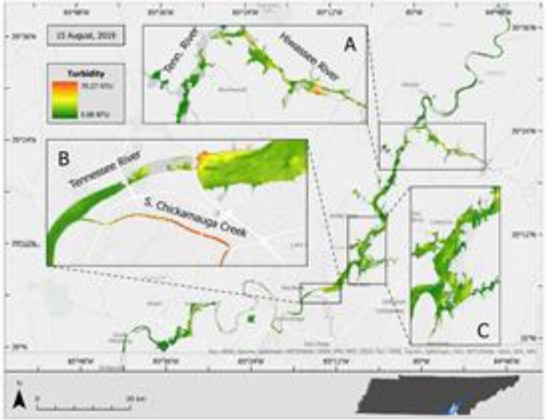
- Quantitative Suspended Sediment Equation
  - Field data collected on the Tennessee River and tributaries with turbidity measuring instruments.
  - NTU (Nephelometric Turbidity Units)
- Correlation yielded this equation:

$$T = 2677.2\rho_{\text{red (Landsat 8)}}^{1.856}$$

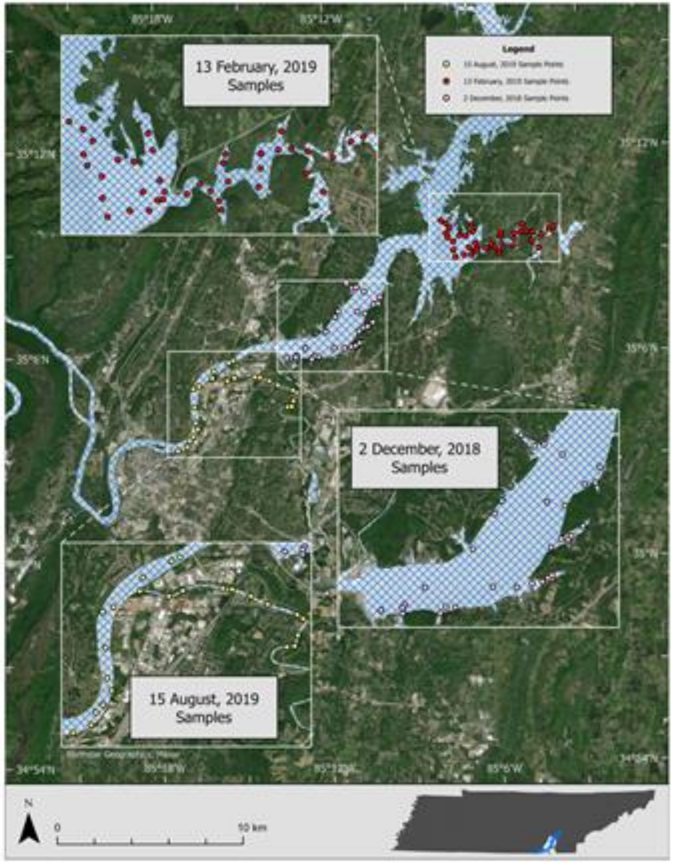
$R^2 = 0.95$



(Hossain et. al. 2021)



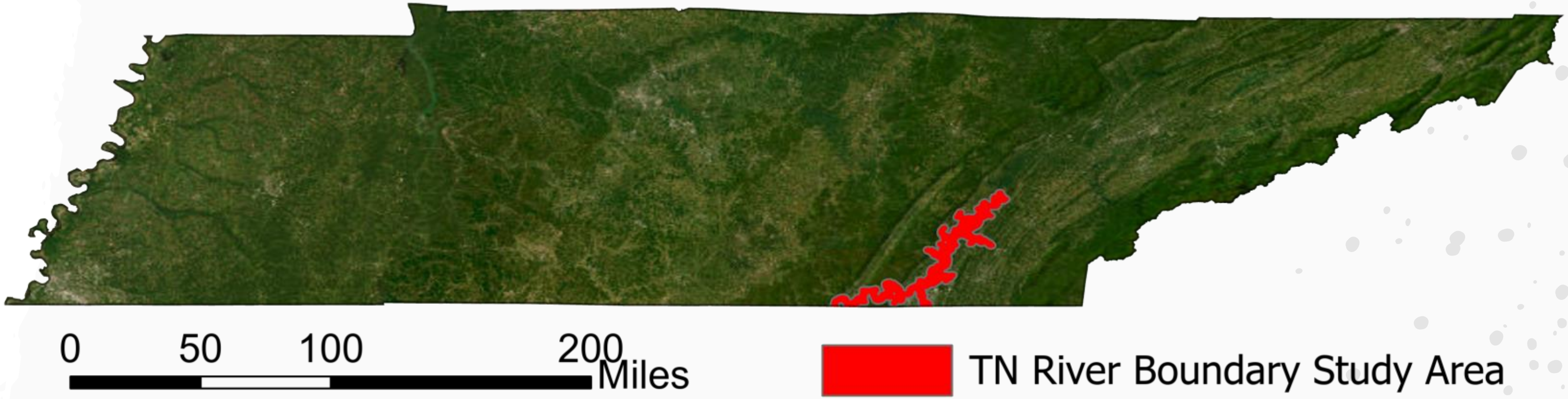
(Hossain et. al. 2021)



(Hossain et. al. 2021)

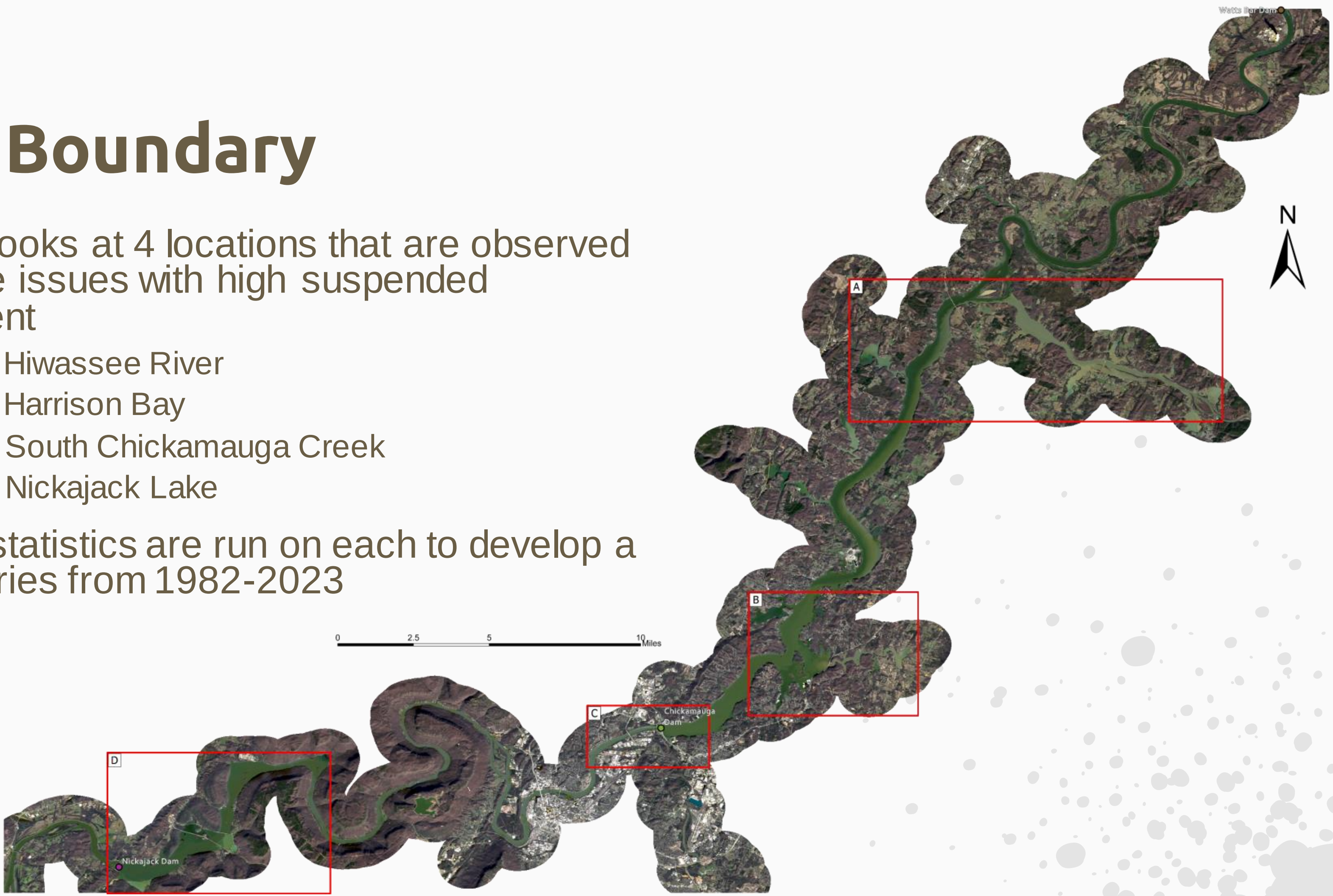
Study Area

- Selected part of the Tennessee River
    - Between Watts Bar and Nickajack Dam
    - Runs through Chickamauga Dam
- River Runs North to South and Travels through the City of Chattanooga, TN



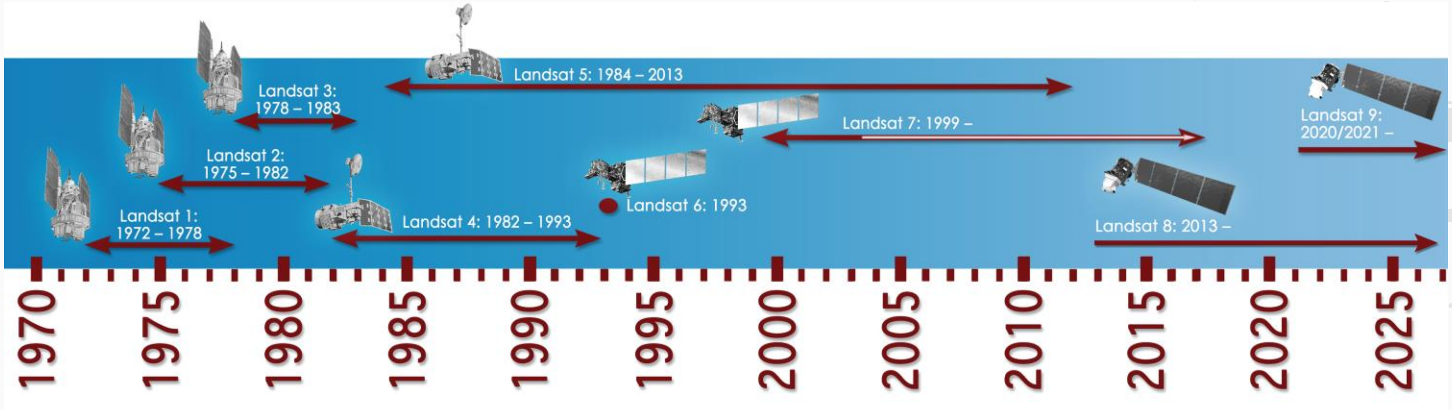
River Boundary

- Study looks at 4 locations that are observed to have issues with high suspended sediment
  - A) Hiwassee River
  - B) Harrison Bay
  - C) South Chickamauga Creek
  - D) Nickajack Lake
- Mean statistics are run on each to develop a timeseries from 1982-2023



1.) Landsat 4-9 Images

- The time frame
  - From 1982-12-07 to 2023-01-06 = around 40 years
- Landsat 4, 5, 7, 8, and 9 sensors
  - Landsat 6 excluded due to no images



1.) Landsat 4-9 Images

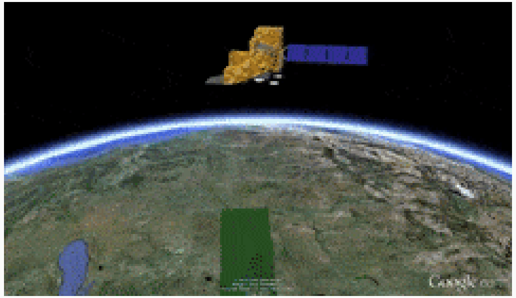
- Google Earth Engine Python API
- Code snippet of Landsat 4 Image Collection.
  - All images of Landsat 4 are collected and filtered out based on the dimensions of the Tennessee River.
  - Only bands 1-4 are obtained with a bitmask pixel identifier.
  - Images are sorted by date for organization.

```
#Call in Landsat 4 Level 2, Collection 2, Tier 1 dataset
LS4_SR = (
    ee.ImageCollection("LANDSAT/LT04/C02/T1_L2")
    .filterBounds(TN_River) #Filter only swath grids that cover the TN River Boundary
    .select(['SR_B1', 'SR_B2', 'SR_B3', 'SR_B4', 'QA_PIXEL'],['Blue', 'Green', 'Red', 'NIR', 'QA_PIXEL'])
    .sort('system:time_start') #Sort collection by acquisition time
)
```

Code Snippet of Landsat 4 Data Collection. Code is repeated for Landsat 5, 7, 8, and 9

2.) Mosaic Images

- Code snippet of a custom mosaic function in python.
- Images are merged by:
    - Unique date of image
    - Images that occur on the same date are images that are on the same orbital path and usually two are needed to cover the entire study area



```python
def mergeByDate(imgCol):
    #Convert the image collection to a list.
    imgList = imgCol.toList(imgCol.size())

    # Driver function for mapping the unique dates
    def uniqueDriver(image):
        return ee.Image(image).date().format("YYYY-MM-dd")

    uniqueDates = imgList.map(uniqueDriver).distinct()

    # Driver function for mapping the moasiacs
    def mosaicDriver(date):
        date = ee.Date(date)

        image = (imgCol
                 .filterDate(date, date.advance(1, "day"))
                 .mosaic())

        return image.set(
                        "system:time_start", date.millis(),
                        "system:id", date.format("YYYY-MM-dd"),)

    mosaicImgList = uniqueDates.map(mosaicDriver)

    return ee.ImageCollection(mosaicImgList)
```

Code snippet of mosaicking images together based on the same dates

3.) Bitmask Extract Clouds and Preserve Water

- Pixels are either excluded or included based on “QA_PIXEL” Band
  - Clouds pixels are excluded
  - Cloud shadows pixels are excluded
  - Water pixels are included

- Not all images are included
  - If 30% or more of the of study area has cloud pixels, the whole whole mosaic image is excluded

- Images are clipped to a vectorized Tennessee River boundary

```
# A function to mask out cloud shadow pixels.
def cloud_shadows(image):
    QA = image.select(['QA_PIXEL'])
    # Get the internal cloud shadow algorithm flag bit.
    return getQABits(QA, 4,4, 'Cloud_shadows').eq(0)

# A function to mask out cloudy pixels.
def clouds(image):
    # Select the QA band.
    QA = image.select(['QA_PIXEL'])
    # Get the internal cloud algorithm flag bit.
    return getQABits(QA, 3,3, 'Cloud').eq(0)

# A function to preserve only water pixels.
def water(image):
    QA = image.select(['QA_PIXEL'])
    # Get the internal water algorithm flag bit.
    return getQABits(QA, 7,7, 'Water').neq(0)
```

Code snippet of eliminating cloud and cloud pixels and preserving water pixels

4.) Extract Red Band and Run Surface Reflectance Conversion and NTU Conversion

- Isolation of the red band is needed to run the NTU conversion equation later
- DN integer values are converted to decimal (float) values
- Decimal red reflectance values are converted to NTU units

```
#Create fucntion that calcualtes the Quantitative index
def QUANT(image):
    red = image.select('Red') #Create a variable that selects the red band
    #Run the '2677.2 * (pow(Red, 1.856))' equation
    scale = red.multiply(0.0000275).add(-0.2)
    quant = (((scale.pow(1.856)).multiply(2677.2)).toFloat()).rename('QUANT')
    #return quant
    return quant
```

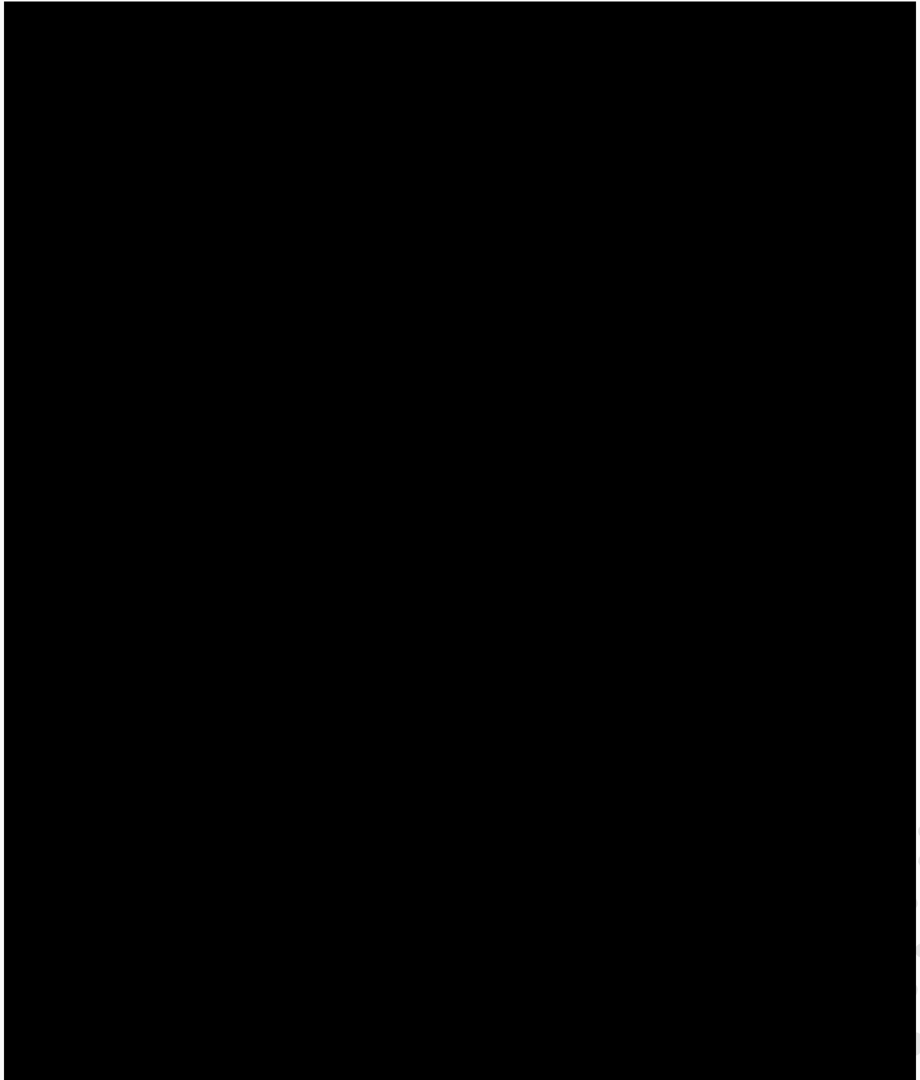
Code snippet of selecting the red band, converted it from integer to DN values and converting reflectance to NTU

Result

- 1,196 scenes processed
- Video shows true color image over river NTU image

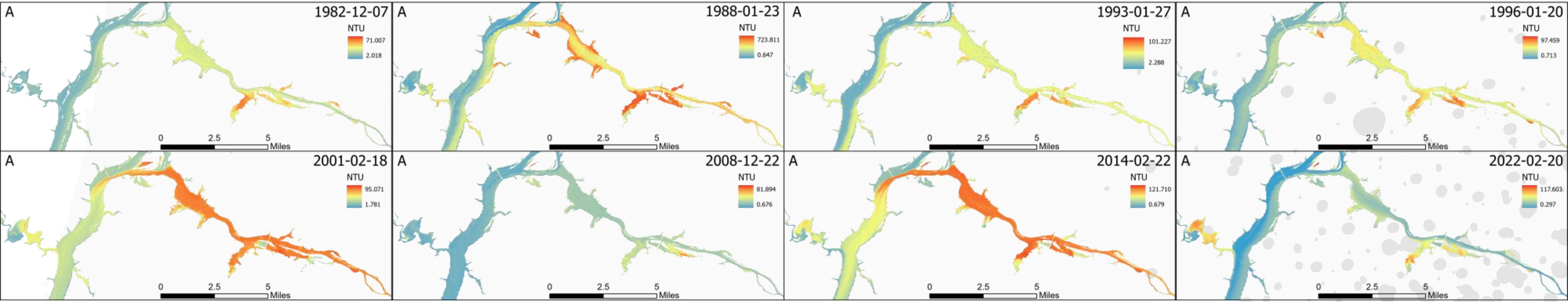
Minimum: 0.11
Maximum: 176.98
N Pixels: 222575
Mean: 16.1
Median: 13.79

Example of statistics of the whole image



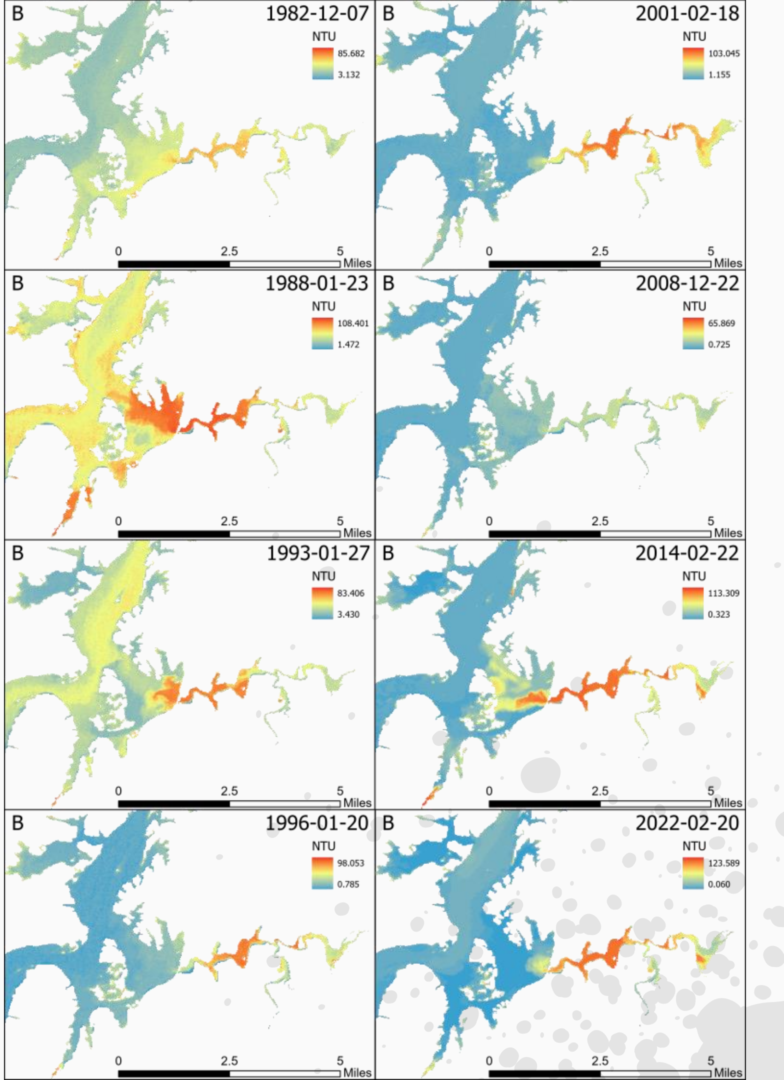
Hiwassee River

- High variations in turbidity feeding into the Tennessee River
    - Seems to be one of the biggest contributors of sediment for our area
- The site shows what happens to high concentrations of suspended sediment when they meet a higher velocity current.



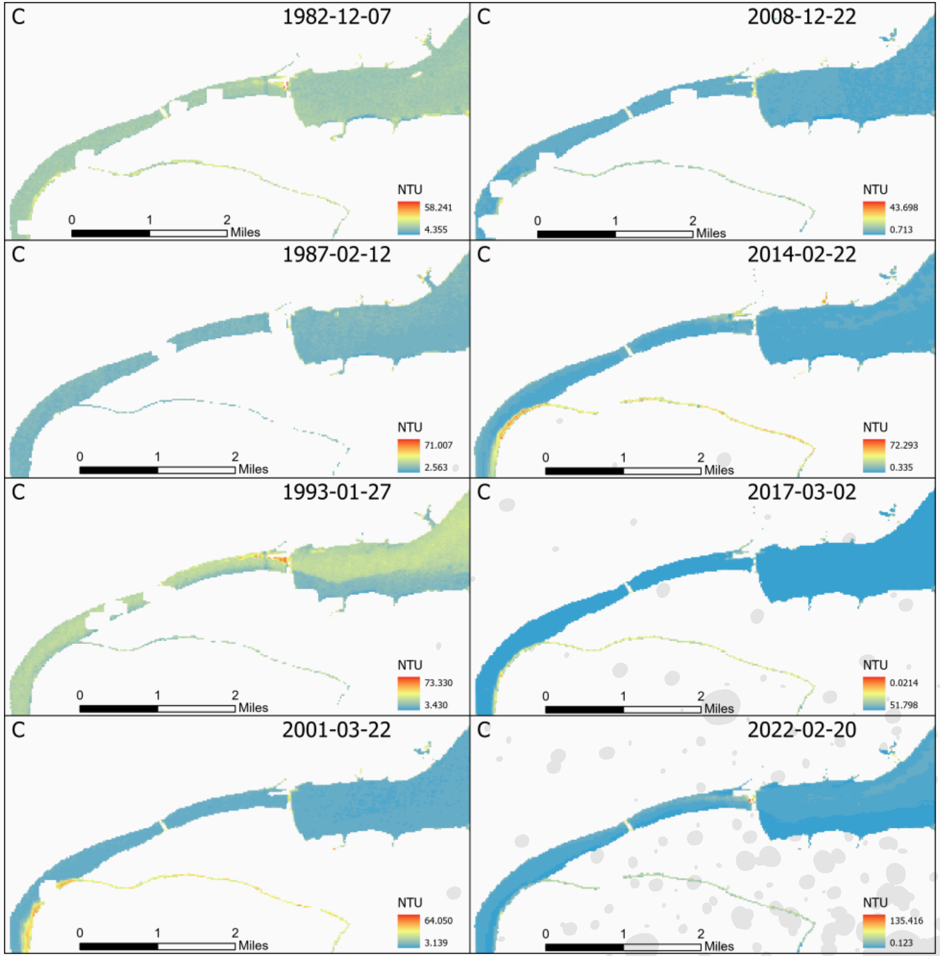
Harrison Bay

- Seen to be a large contributor of sediment to the Tennessee River as well
- The site shows what happens to the sediment when it meets the low flow velocities in the lake
    - Seems to create a billowing cloud effect when sediments are dropping out of suspension.



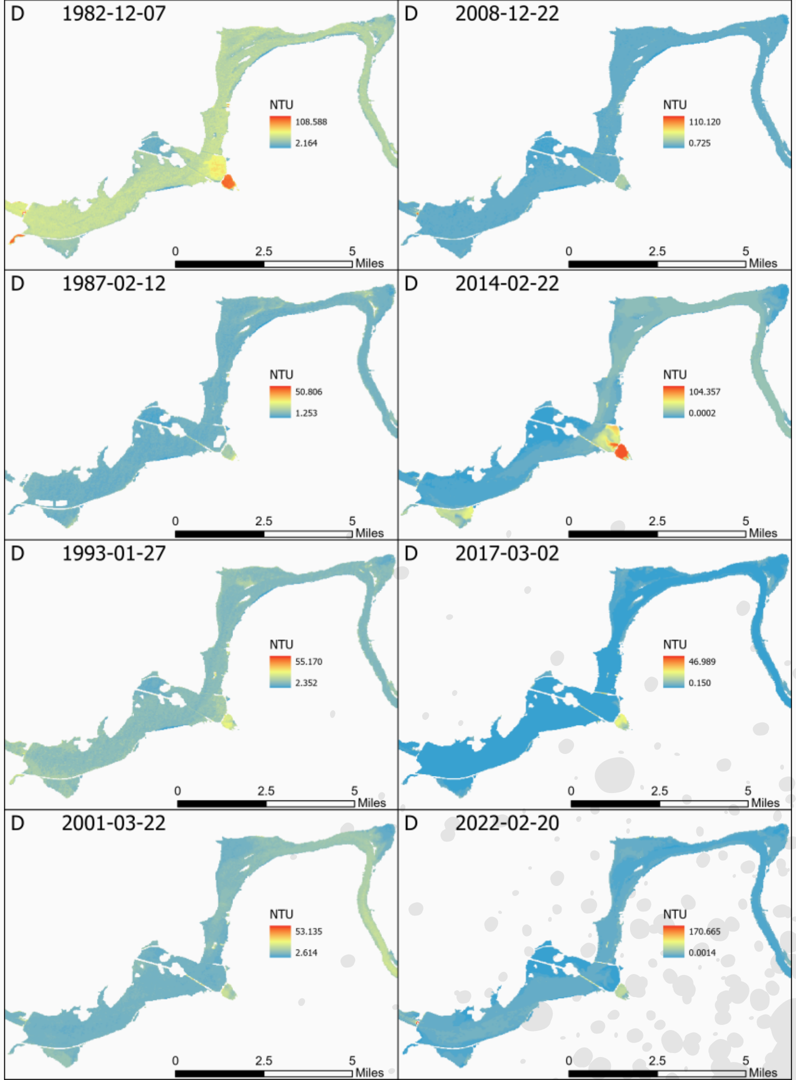
South Chickamauga Creek

- Shows the effect of what a smaller tributary has on the Tennessee River.
  - The site seems to have a substantial amount of sediment feeding into the Tennessee River for a very small tributary.
  - Current velocities show how higher turbidity waters are forced to the left side of the riverbank when they meet high velocity water.



Nickajack Lake

- Shows the effect off how point bars and cut banks have on the suspended sediment concentrations.
- Suspended sediment seem to be lower at the point bars and higher in the cut bank sections.

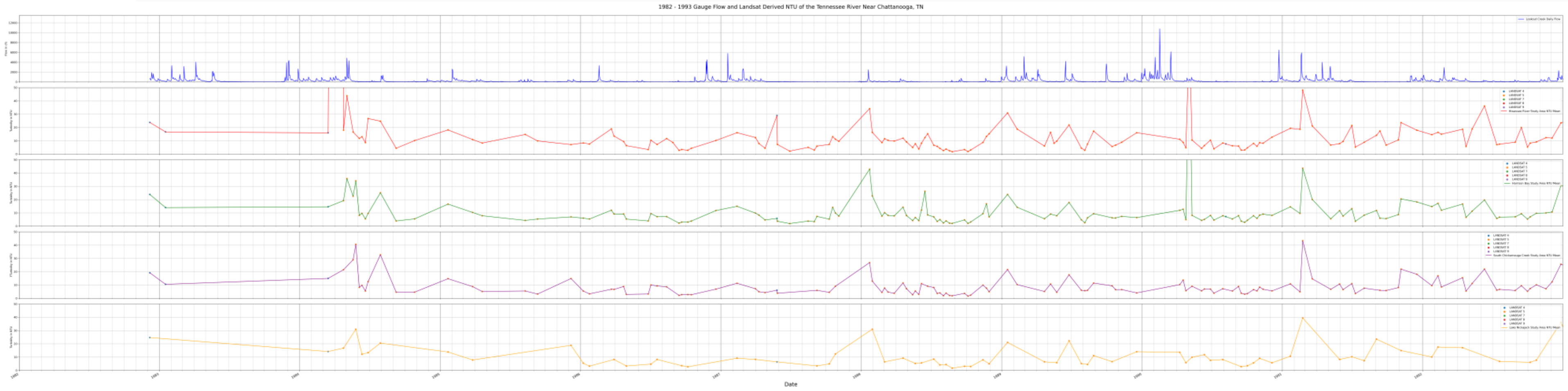


Time Series of Each Study Area

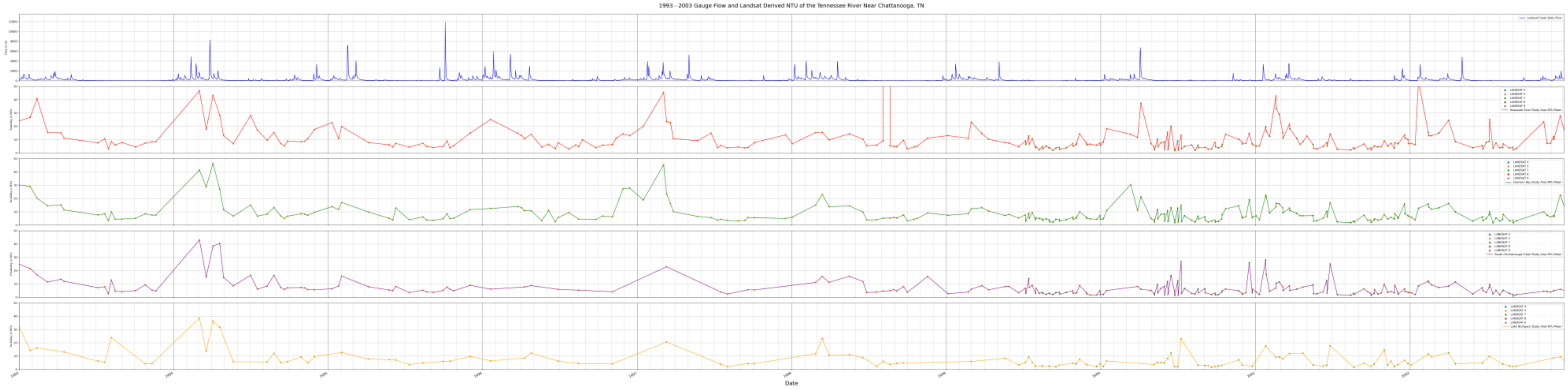
- The mean NTU of each study area was calculated and put into a timeseries plot.
- A discharge hydrograph within the study area was added to the timeseries plot
  - There are 12 discharge gages in the study area. We choose one that had data available for Landsat 4-9 timeframe.
- Shows Relationship between high discharge and high NTU fluctuations.

Sample of the timeseries. Lookout Creek Daily Discharge, Hiwassee River, Harrison Bay, South Chickamauga Creek, and Lake Nickajack mean NTU

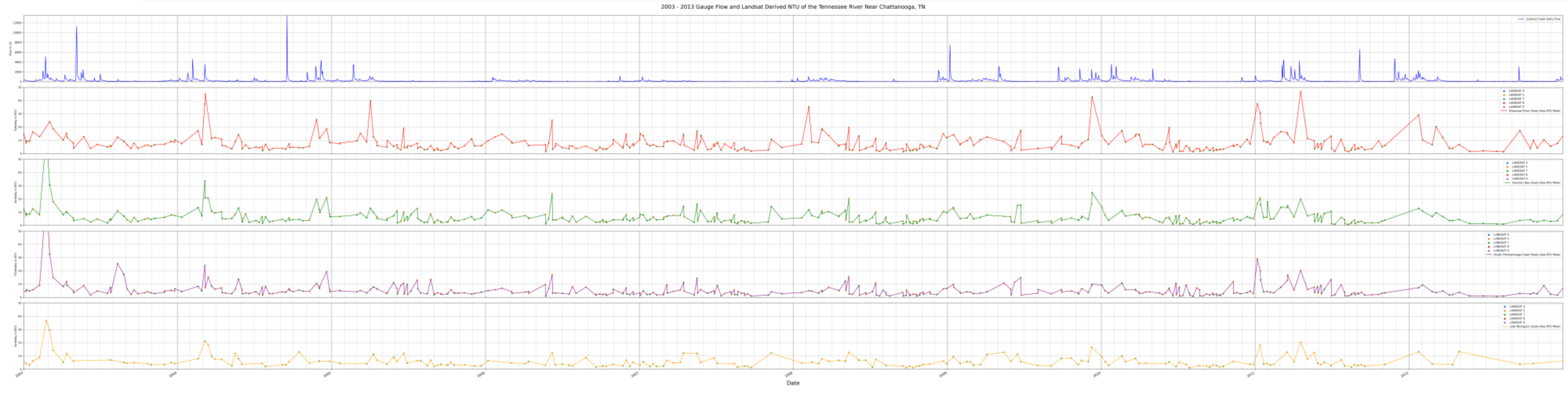
1983-1993 Timeseries



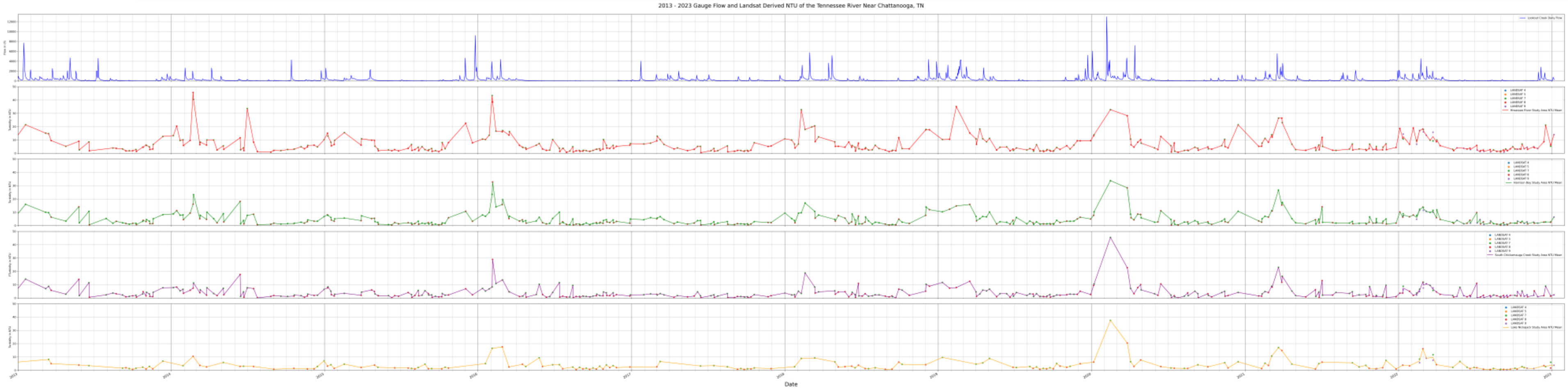
1993-2003 Timeseries



2003-2013 Timeseries



2013-2023 Timeseries



Future Research

- Use more sensors like:
  - Sentinel 2 (10m) 5-day acquisitions
  - PlanetScope (3m) almost daily acquisitions
  - This will smooth out the NTU line plot
- Convert NTU to Suspended Sediment Concentration (SSC) in mg/L units with field collected samples

Questions?