A Project/ Study on

Atmospheric Lightning measurements from space and its comparison with ground-based data over India.

Project Guide:
Dr. Alok Toari,
Sci./Engr. 'SG',
Dy. Head ACD, ECSA
NRSC,
Hyderabad,

Submitted By:
Aman Agarwal,
1901965,
Priyadarshini Agrawal,
1902004,
Mechanical Engg – IV Year,
Dayalbagh Educational Institute,
Agra-282005

DECLARATION

I hereby declare that I am the sole author of this report. The report represents bona-fide work done during the internship and has not been represented in any other report and/ or publication avenue.

ACKNOWLEDGMENTS

I would like to take this opportunity to express my sincere gratitude and profound feeling of admiration to:

Foremostly, Ms. Jaya Saxena, who provided me with this gracious opportunity of being associated with one of the esteemed organizations of the nation.

Next to my project guide, Dr. Alok Toari and Shri Venkatesh Degala who during the course of this project has guided me with his knowledge, helping in shaping many concepts in my understanding and also with their experiential and invaluable wisdom.

To the Dean, Faculty of Engineering, DEI, Prof V Soamidas for his support and involvement in helping me fetch this internship.

To my project coordinator, Dr Manoj Dixit for his guidance throughout

And, to the Almighty Divine for everything.

TABLE OF CONTENTS

ABSTRACT

Understanding atmospheric lightning flashes and their occurrences is one of the most important aspects of the Earth's climate science. Real-time lightning data have profound importance in climate science and air-quality research, apart from lightning being one of the major natural disasters. Keeping these in view, National Remote Sensing Centre (NRSC), Indian Space Research Organization has established a lightning detection sensor (LDS) network for nationwide detection of lightning occurrences with 42 sensors installed until August 2022. Though there is no space borne measurements over India apart from the Lightning Imaging Sensor (LIS) on the International Space Station (ISS), hereafter referred to as ISS-LIS which detects lightning from space by capturing the optical scattered light emitted from the top of the clouds. The objective of this work is to quantify the similarities and deviations between these two distinct lightning detection technologies by comparing the LDS cloud-to-ground flashes to the ISS-LIS measurements. A full month data collected from these two different instruments during April 2020 is assessed for their efficacies and limitations.

Chapter 1

Introduction

1.1 Understanding the Lightning

Lightning is an electrical discharge between positive and negative charged regions of a thunderstorm and can be deadly if the necessary precautions are not taken.
As the area of negative charge at the base of the thundercloud builds up, it induces a region of positive charge to develop on the ground below. As a result of this, a potential difference or voltage is created across the cloud-to-ground gap. Once the voltage reaches a certain strength, the air between the base of the cloud and the ground develops an electrical conductivity. At first a channel, known as a stepped leader, is formed. Although invisible to the naked eye, this allows electrons to move from the cloud to the ground. [11]
During a storm, colliding particles of rain, ice, or snow inside storm clouds increase the imbalance between storm clouds and the ground, and often negative charge the lower reaches of storm clouds. Pointed objects on the ground, like steeples, trees, and the elevated ground, become positively charged creating an imbalance that nature seeks to remedy by passing current between the two charges.

1.2 Significance & Use of Lightning Data

Lightning data and their usage in operational weather forecasting have shown phenomenal improvements in extreme events. [8] Further, identifying the hot spots of lightning occurrences and early detection of lightning strikes are required to mitigate the loss of life. The lightning strokes produce NOx which is immediately transferred to the ground and has important consequences in surface ozone chemistry. In recent times, it is well understood that lightning activity is an indicator of weather and climate change. Romps et al. have shown that a rise in surface temperature can enhance the occurrences of lighting activity. The global warming becoming prominent in the Antarctica region further supports the above. In short, characterization of lightning activity is important for the lower atmosphere and disaster management, and also to resolve some of the least understood features in the upper atmosphere.

1.3 Datasets

1.3.1 Lightning Imaging Sensor (LIS)

The *International Space Station (ISS)* Lightning Imaging Sensor (LIS) datasets were collected by the LIS instrument mounted on the ISS detect the distribution and variability of total lightning occurring in the Earth's tropical and mid-latitude regions. The LIS uses the 774 nm oxygen airglow emissions and takes images using FOV optical lens. The details of the instrument and its characteristics are elaborated in table 1. These datasets consist of *near-real time* and non-quality controlled science and background data, while the final quality controlled science and background datasets are continually being added as manually reviewed. This data collection can be used for severe storm detection and analysis, as well as for lightning-atmosphere interaction studies. The LIS instrument makes measurements during both day and night with high detection efficiency. The data are available in both HDF-4 and netCDF-4 formats, with corresponding browse images in GIF format and summary images in PNG format. We have used Quality Controlled(QC) Data because QC data have had specific quality control steps applied to ensure that all bad data. are flagged. The data are available in site Reference 4

Characteristic	Description
Platform	International Space Station (ISS)
Instrument	Lightning Imaging Sensor (LIS)
Projection	Centroid
Spatial Coverage	N: 54.0, S: -54.0, E: 180.0, W: -180.0
Spatial Resolution	4-8 km
Temporal Coverage	March 1, 2017 - ongoing
Temporal Resolution	NRT: 2 minutes
Sampling Frequency	Every 2 milliseconds over ~90 seconds
Parameter	lightning, lightning density
Version	V2.1
Processing Level	1B (Background Data) and 2 (Science Data)

Table 1: Data Characteristics

1.3.2 Lightning Detection Sensors (LDS)

NRSC has installed 42 Lightning Detection Sensor of Boltek make in for nationwide monitoring, with each sensor separated by about 200 km of radial distance. These sensors are long-range sensors, which work in the frequency range of 1 Hz to 30 MHz. The location of correlated lightning occurrences is performed with the time of arrival (TOA) algorithm based on the GPS stamped waveforms. More details on the sensors are elaborated elsewhere (e.g., Cummins et al. 1998; Drue et al. 2007; Shlyugaev et al. 2014). As elaborated by Taori et al. (2022), the low-frequency range (1 Hz–5 kHz) is used to detect long-range static discharge pulse/waveform; the 5 kHz–1 MHz signals are used for finding the geolocation of static pulses, while the higher frequency ranges are used for the inter-cloud or cloud-to-cloud pulse detections.

1.4 Purpose of this Project

The purpose of this project is to comparing the data sets of LDS and ISS-LIS network and optimize the data more precisely. Many accidents occur in wide-areas because of the lightning strikes (Over 152 animals killed in lightning strikes at Molakalmuru, Chitradurga district the news article is refer [9]). So, through this more precisely data we can send alerts to those areas which can suffer heavy lightning strikes in future.

1.5 Objective

To develop an approach or methodology to provide an efficacy statement on the space-based information retrieved by the LIS onboard the ISS with the one collected by the ground based LDS lightning datasets.

Chapter 2

Data & Resources

2.1 Study Area

India

2.2 Software

QGIS 3.10

2.3 Data Collection & Analysis

ISS-LIS Data sets .nc/.hdf files from GHRC websites, LDS Datasets .txt files from NRSC

2.4 Coding

Python 3.0

2.5 Stats

Kappa Stats

2.6 Libraries

GDAL, NumPy, Cartopy, Pandas, netCDF4, Scikit, OpenCV, Matplotlib

Chapter 3

Methodology

3.1 Collected the International Space Station (ISS) Lightning Imaging Sensor (LIS) datasets which are available in GHRC website and our lightning detection sensor (LDS) data set from 1 April to 30 April 2020 data.

3.2 Using Python Programming and Data Science techniques, extract the data of ISS-LIS in CSV files. After that plot the LDS present in .txt files and LIS data sets in QGIS for further Observation.

The images available beforehand had been:

- ISS-LIS_data_5_April_raw_yellow_dot
- LDS_data_5_April_raw

Figure 1: Point data(LIS) for one orbit on 5th April 2020

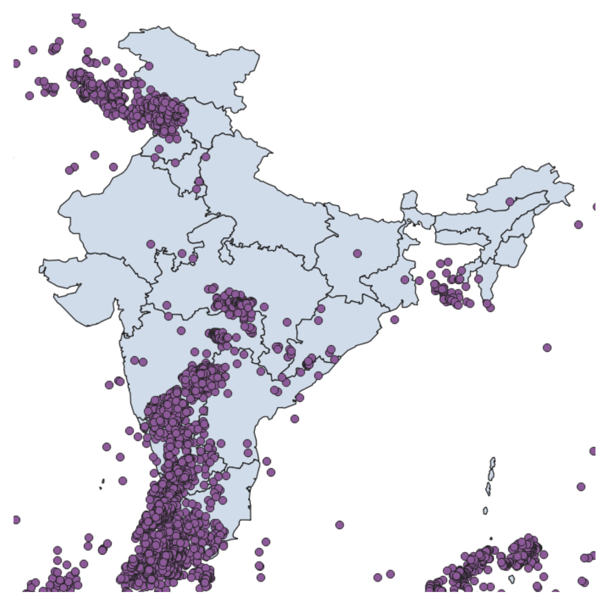


Figure 2 : Point data of LDS for complete day (5th April 2020)

Hence, the procedure deals with generating a single image for the India region using multiple methods/ techniques. Those are further analyzed and compared with each other according to the same flash time of both the data and that image available to decide on the best output out of all those generated. These images were generated with QGIS software.

3.3 **Flash time parameters**:

The LDS Flash occurrence time is given in Indian Standard Time(IST) and the LIS Flash occurrence time is available in Seconds since 1993-01-01 00:00:00. Hence, we need to convert both the dataset at 12:00 AM to 11:59 PM, and then the further operation was applied to compare both the data sets. The table shows the converted flash time as per IST.

flash_time_since 1993-01-01	Converted_flash_time IST
860256599.7	4:10:00 PM
860256606.8	4:10:07 PM
860256627.1	4:10:27 PM
860256636.4	4:10:36 PM

3.4 **Subset LIS datasets over India:**

The LIS network is spread over the world but we need the data only over India because LDS network is operational over India. Hence, we need to separate the LIS data over India with the help of python. The dimensions of India had been taken for separation i.e.,west (55°), east (105°), north (40°), and south (0°).

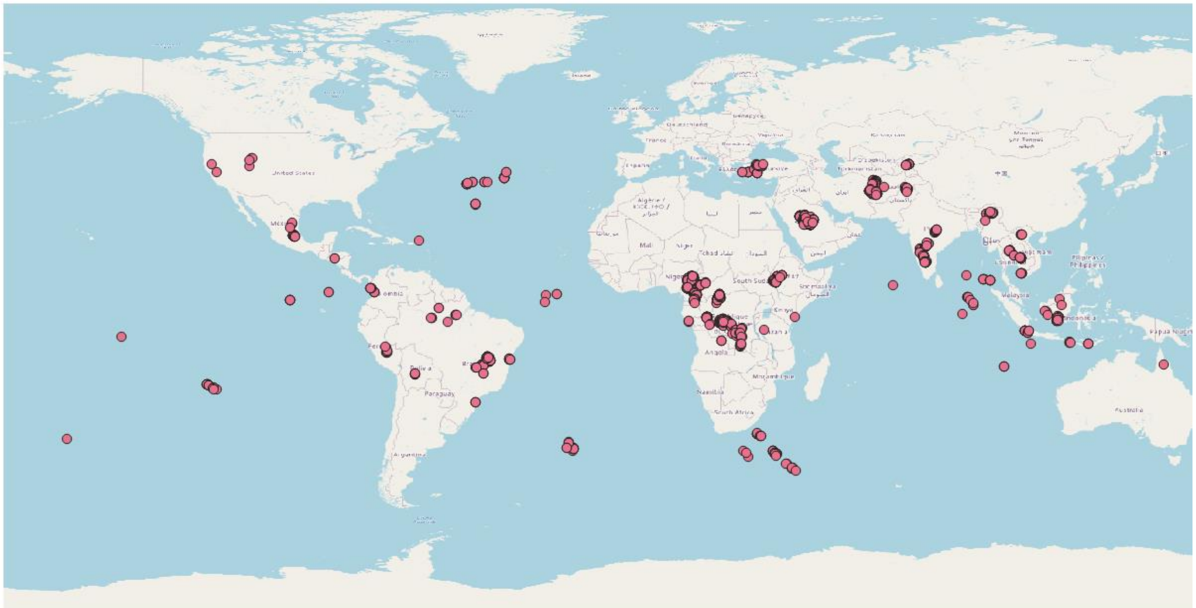
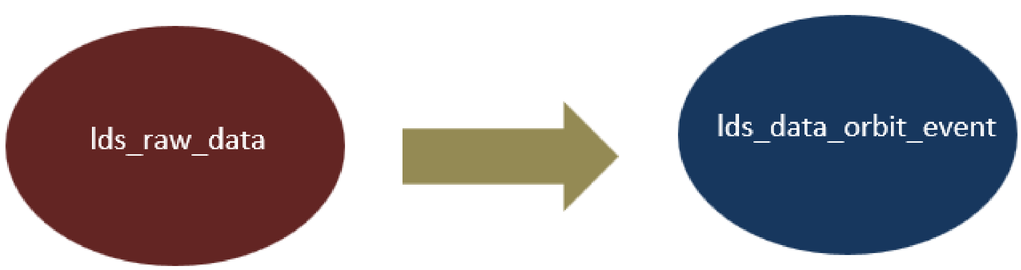


Figure 3 : LIS payload complete coverage data points on 5th April 2020

3.5 LDS data sets according to LIS orbit event over India:

In an image named LDS_data_5_April_raw_violet_dot, there had been full 5 April Data from 12:00 AM to 11:59 PM and in ISS-LIS_data_5_April_raw_yellow_dot image there had been some particular time data over India because the ISS-LIS data examine all over world data. So, the satellite of the LIS network over India has come for a particular time. Therefore, the ISS-LIS points are very less as compared to the LDS points as shown in the image named LIS_data_5_April_raw_yellow_dot.

Hence, Using the various libraries of python and data science customize the LDS data and both the data sets plot in the India map with the help of Cartopy Library in python. The output comes in multiple images i.e., orbit-wise. So, we examine the images that orbit seen over India.



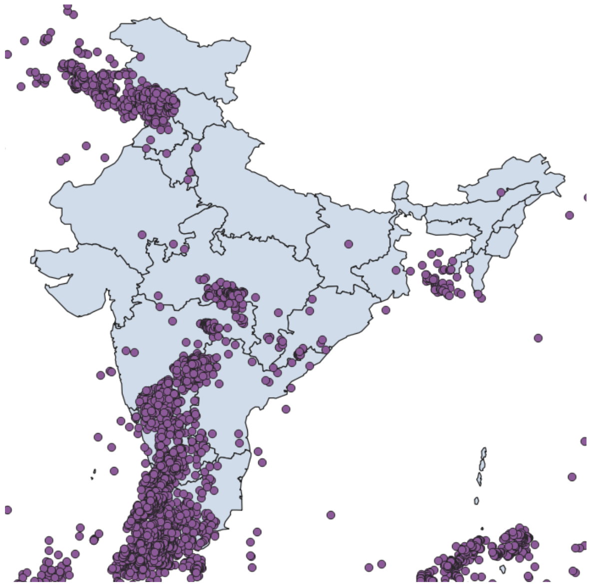


Figure 4 : LDS data on 5th April 2020

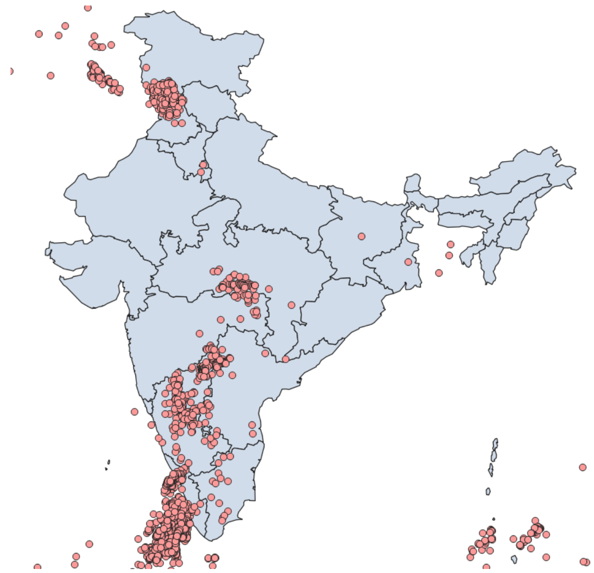
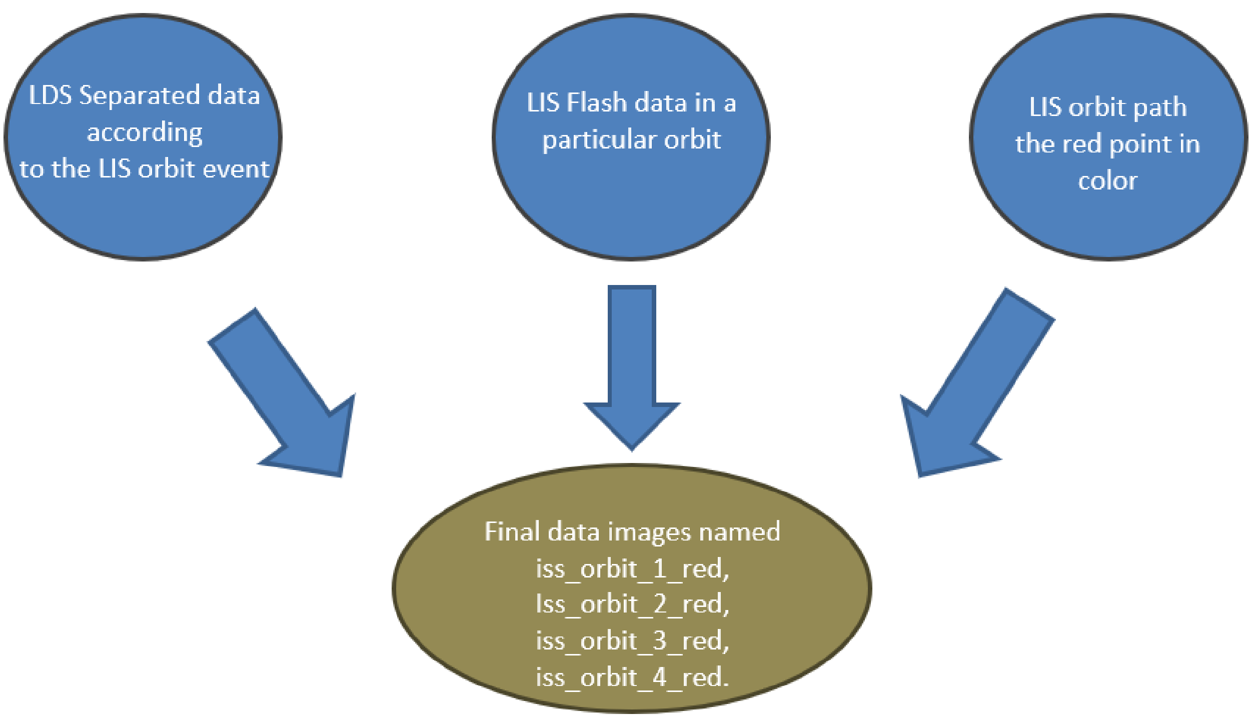


Figure 5 : LDS points falling in the orbit of LIS on 5th April 2020

With the help of cartopy library in python plot the orbit over India. There had been four-orbit instance in % April 2020 which shown in image named _image_orbit_1, _image_orbit_2, _image_orbit_3, _image_orbit_4. The red dots show the orbit covering the path over India.



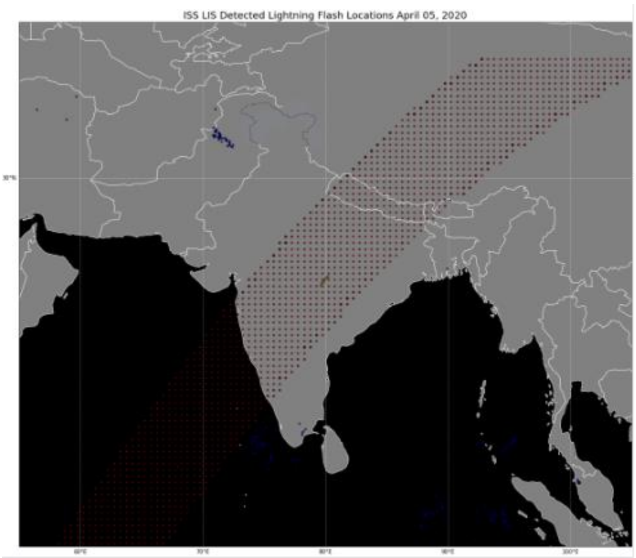


Figure 6 : LIS Orbit 18497 on 5th April 2020 over ROI

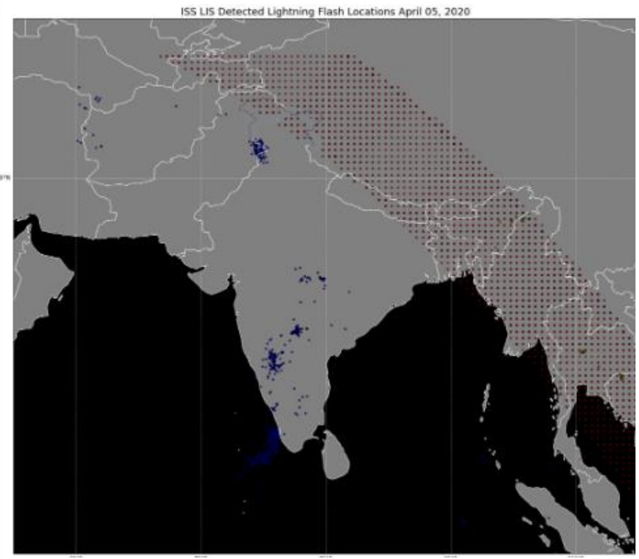


Figure 7 : LIS Orbit 18499 on 5th April 2020 over ROI

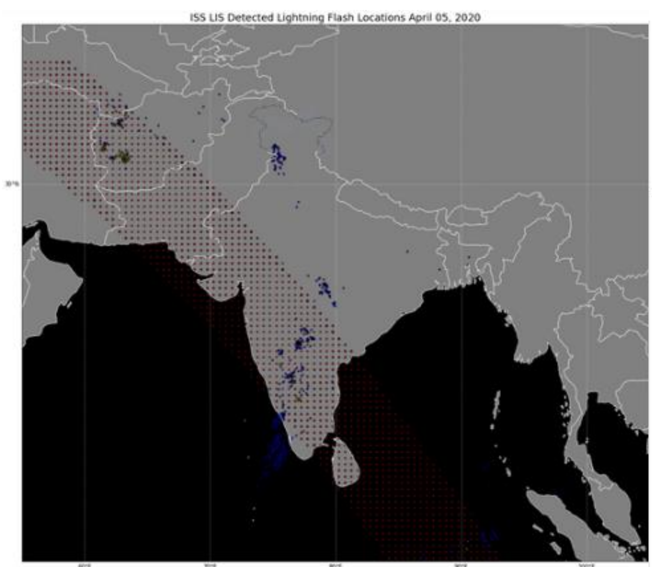


Figure 8: LIS Orbit 18506 on 5th April 2020 over ROI

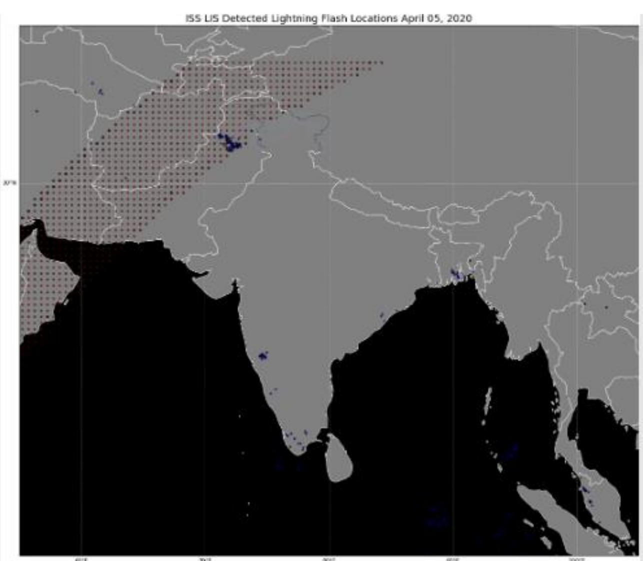


Figure 9: LIS Orbit 18509 on 5th April 2020 over ROI

For our observation, we take the image named iss_orbit_3_red orbit data because in the above four images almost high flash occurrences occur in this orbit. So, we examined iss_orbit_3_red image data for comparing both datasets.

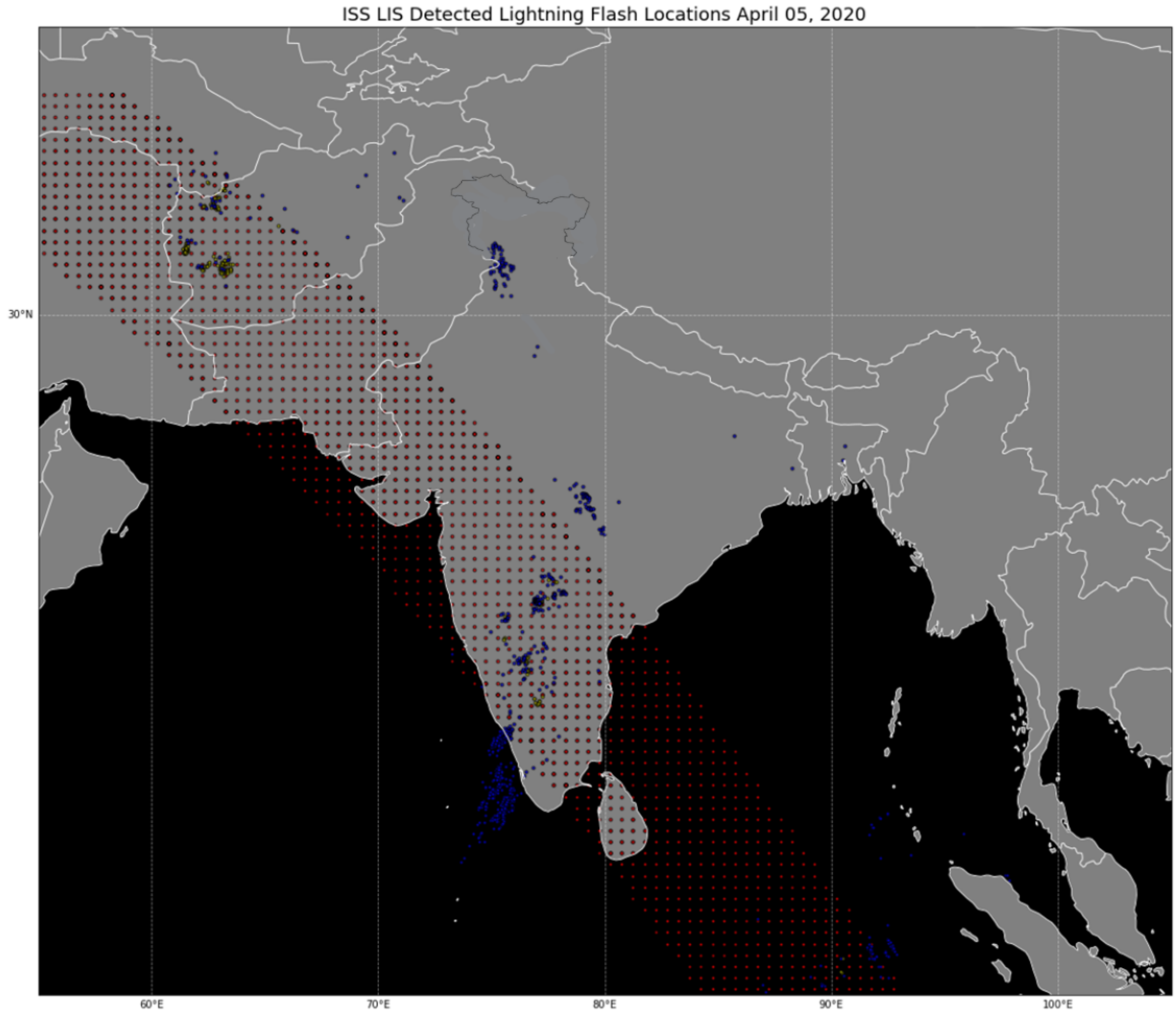


Figure 10: Detailed representation of Orbit 18506

The image named larger_view_orbit_3 represents the combined and customize data with orbit events of LIS and LDS data sets. Blue dots represent the LDS data, the lesser yellow point shows the LIS data, and the red dot represents the LIS network orbit over India as I already discussed in this report. We need only the data inside the orbit path. The LDS data outside the orbit path did not consider in our observation. Hence, Consider the blue points inside the orbit path only.

3.6 Compared both the final datasets using kappa statistics:

The Cohen’s kappa is a statistical coefficient that represents the degree of accuracy and reliability in a statistical classification. It measures the agreement between two raters (judges) who each classify items into mutually exclusive categories.

This statistic was introduced by Jacob Cohen in the journal Educational and Psychological Measurement in 1960.

$$k = \frac{p_o - p_e}{1 - p_e}$$

where p_o is the relative observed agreement among raters, and p_e is the hypothetical probability of chance agreement. [10]

Steps in kappa Stats:

- Making Squares of dimension 55.5km X 55.5km with the reference of the LIS path.
- Divide whole grids into equal rectangles. One of the rectangles shown in the image named rectangle_grid_kappa_stats
- Give the tolerance of 25 Km in LIS and 100-200 m in LDS
- Then find the separate value of squares in which both the dataset and individual datasets are available.
- Find the Cohen Kappa value for measuring the accuracy of data’s

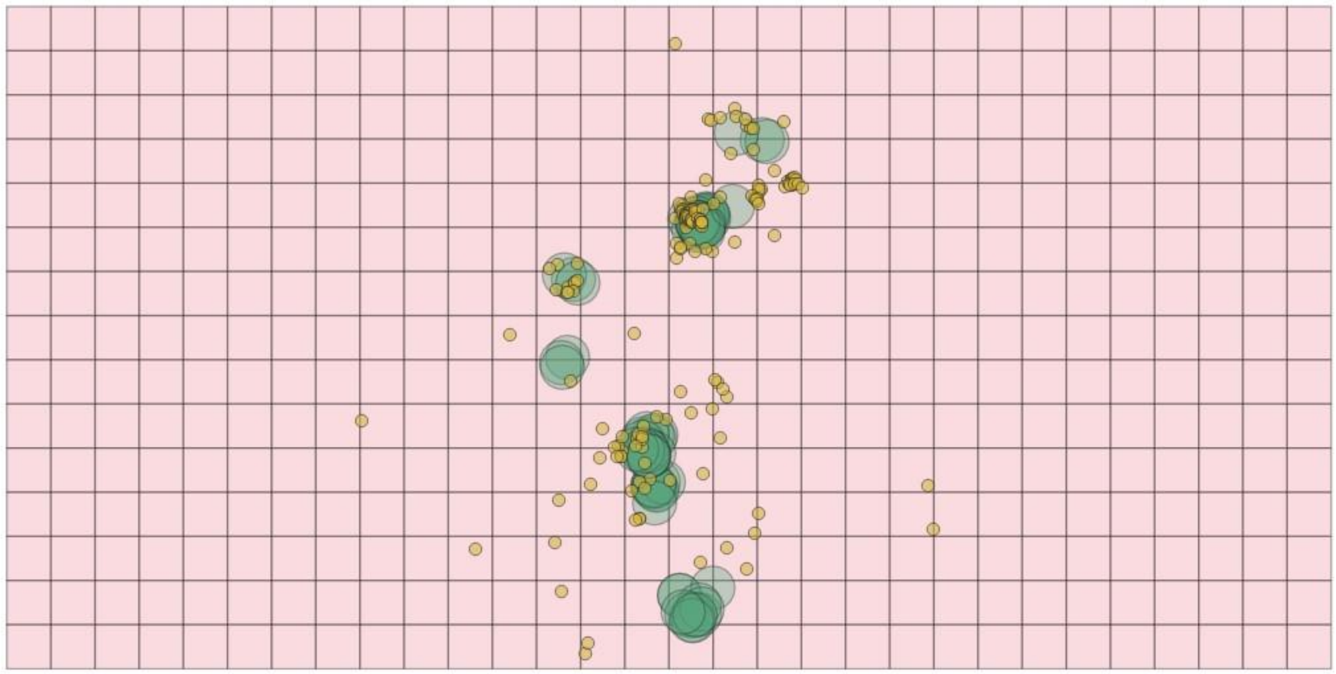


Figure 11 : Comparison of LDS with LIS data

The above image has a total of 450 grid squares in which 20 grids are in which both the datasets include, 395 grids- both the data's excluded, 14 grids- only LIS data included, 21 grids- only LDS data included.

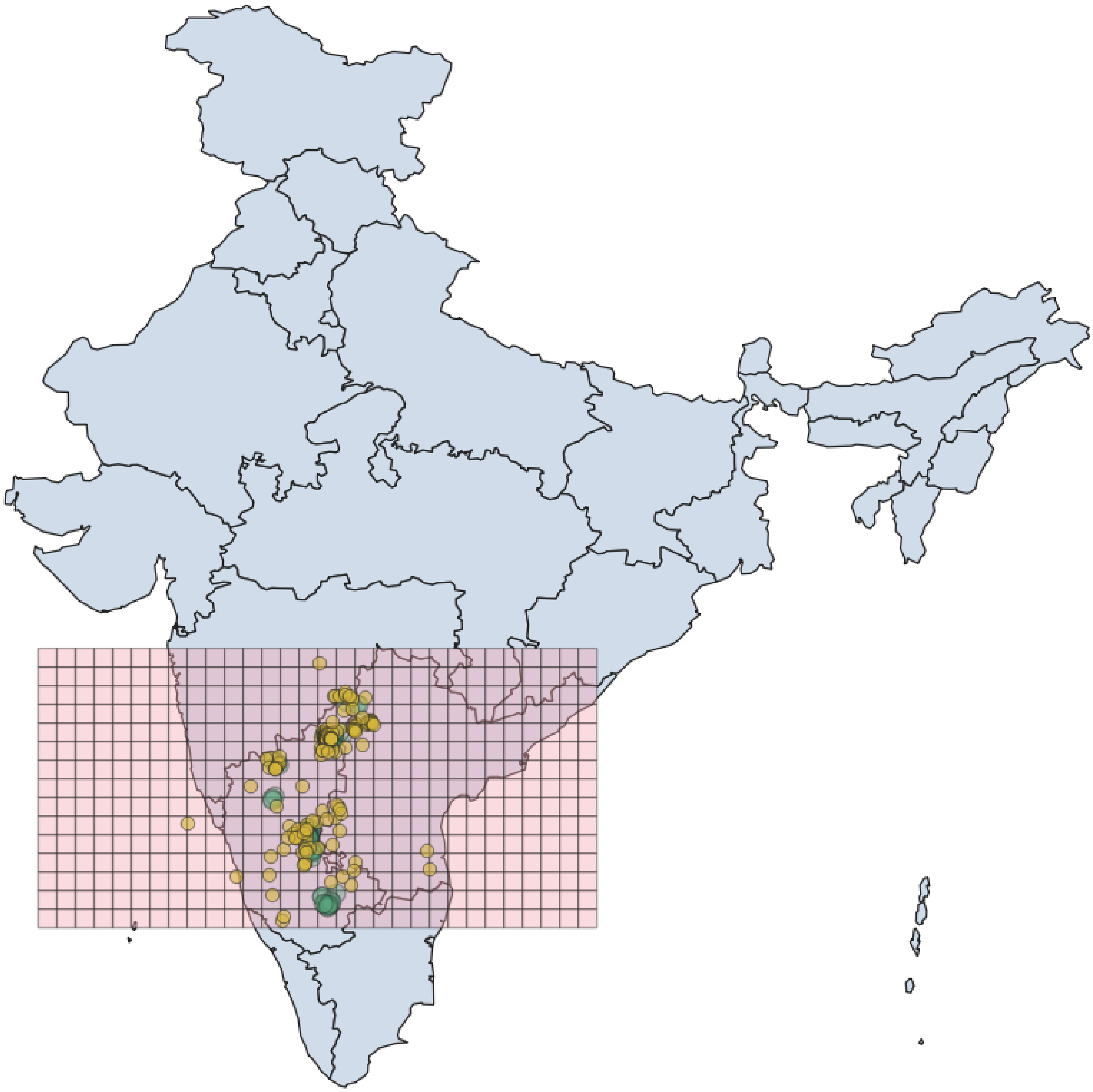


Figure 12 : Rectangular grid based Kappa Statistics

The images named final_examined_rec_5A, and rectangle_grid_kappa_stats have one of the rectangles over India according to the path. This rectangle had almost all LIS points in a particular orbit. So this rectangle comes with the best result of accurate data. The green dots show ISS-LIS data and the yellow dots show the LDS data with given tolerance as we already discussed above.

3.6.1 Python Project Code:

```
#### Import Python packages ####
import sys
import os
import glob
from netCDF4 import Dataset, num2date
import numpy as np
import pandas as pd
import csv
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import cartopy.feature as cfeature
import matplotlib.ticker as mticker
from cartopy.mpl.gridliner import LONGITUDE_FORMATTER, LATITUDE_FORMATTER
import re

#Initial file path. It can be changed by passing a different path as an argument
#to the main() function
file_path = 'C:/Users/Dell/Downloads/isro internship/28 April/'

lon_W = int(55)
lon_E = int(105)
lat_S = int(0)
lat_N = int(40)


#Define the directory of the files
dataDir = os.path.join(file_path, '')

#Identify all the ISS LIS NetCDF files in the directory and their paths
raw_files = glob.glob(dataDir+'Data_28/ISS_LIS_*.nc')

files = [os.path.normpath(i) for i in raw_files]

lds_file = os.path.join(dataDir, 'LDSN_28Apr2020.txt')

df = pd.read_csv(lds_file, sep=",")

#Extract the dates for the files
#Create empty lists to hold the orbit start and end times
orbit_start = []
orbit_end = []
orbit_start_nadir = []
end_time_nadir_dt = []
lds_lat = df['Latitude'].tolist()
lds_lon = df['Longitude'].tolist()
lds_time = df['Time'].tolist()
lds_time_sec_final = []
lds_lat_final = []
lds_lon_final = []
```

```
flash_lat_final = []
flash_lon_final = []
iss_lat_final = []
iss_lon_final = []
flash_time_final = []
flash_time_final_final = []
flash_time_dt = []
file_no = 0

for i in files:
    datafile = Dataset(i, 'r')
    file_no = file_no + 1
    start_value = datafile.variables['orbit_summary_TAI93_start'][:].data.tolist()
    start_value_units = datafile.variables['orbit_summary_TAI93_start']
    end_value = datafile.variables['orbit_summary_TAI93_end'][:].data.tolist()
    end_value_units = datafile.variables['orbit_summary_TAI93_end']
    orbit_start.append(start_value)
    orbit_end.append(end_value)

    #From the start and end times, calculate the minimum and maximum date of the files

    start_dates = num2date(orbit_start[:], start_value_units.units)
    stop_dates = num2date(orbit_end[:], end_value_units.units)

    begin_date_value = min(start_dates)
    end_date_value = max(stop_dates)

    #Create text and numerical dates to use in file names and plot title
    begin_date = begin_date_value.strftime("%B %d, %Y")
    end_date = end_date_value.strftime("%B %d, %Y")
    begin_int = begin_date_value.strftime("%Y%m%d")
    end_int = end_date_value.strftime("%Y%m%d")

    end_time_nadir_dt.clear()
    orbit_start_nadir.clear()

    end_value_nadir = datafile.variables['viewtime_TAI93_end'][:].data.tolist( )
    end_value_units_nadir = datafile.variables['viewtime_TAI93_end']
    orbit_start_nadir.append(end_value_nadir)
    stop_time_nadir = num2date(orbit_start_nadir[:], end_value_units_nadir.units)
```

```
stop_time_nadir = num2date(orbit_start_nadir[:], end_value_units_nadir.units)
end_time_nadir_value = max(stop_time_nadir)
for v in range(0, len(end_time_nadir_value)):
    end_time_nadir_dt.append(int(end_time_nadir_value[v].strftime("%H%M%S")))



#strftime(end_time_nadir_dt,26397,"%m/%d/%Y %H:%M:%S",end_time_nadir_value)
#Create CSV file and destination

csvfile = os.path.join(dataDir, 'isslis_flashloc_' +begin_int + '_' + end_int + str(file_no) + '.cs
csvfileFinal = os.path.join(dataDir, 'isslis_flashloc_Final' + begin_int + '_' + end_int + str(file_
csvfileFinal_lds = os.path.join(dataDir, 'LDS_Final' + begin_int + '_' + end_int + str(file_no) + '.
csvfilepath = os.path.join(dataDir, 'iss_path' + begin_int + '_' + end_int + str(file_no) + '.csv')
# csvfilelds_h = os.path.join(dataDir, 'lds_h' + begin_int + '_' + end_int + '.csv')
```

```
lds_time_sec_final.clear()
lds_lat_final.clear()
lds_lon_final.clear()
for l in range(0, len(lds_time)):
    lds_time_split = re.split(':| ', lds_time[l])

    if lds_time_split[3] == "PM" and lds_time_split[0] != 12:
        lds_time_split[0] = int(lds_time_split[0]) + 12
    if lds_time_split[3] == "AM" and lds_time_split[0] == 12:
        lds_time_split[0] = int(lds_time_split[0]) - 12
    else:
        lds_time_split[0] = int(lds_time_split[0])

    lds_time_final = (int(lds_time_split[0])*10000) + (int(lds_time_split[1])*100) + int(lds_time_s
    #lds time in seconds
    lds_time_sec = (int(lds_time_split[0])*3600) + (int(lds_time_split[1])*60) + int(lds_time_split

    #iss_time_final = end_time_nadir_value[l].strftime("%H%M%S")
    if min(end_time_nadir_dt) < lds_time_final < max(end_time_nadir_dt):
        if lon_W < lds_lon[l] < lon_E and lat_S < lds_lat[l] < lat_N:
            lds_lat_final.append(lds_lat[l])
            lds_lon_final.append(lds_lon[l])
            lds_time_sec_final.append(lds_time_sec)

flash_lat.clear()
iss_lat.clear()
flash_lon.clear()
iss_lon.clear()
flash_time.clear()
flash_lat_final.clear()
flash_lon_final.clear()
iss_lat_final.clear()
iss_lon_final.clear()
flash_time_final.clear()
flash_time_final_final.clear()
flash_time_dt.clear()

flash_lat_value = datafile.variables['lightning_flash_lat'][:].data.tolist()  # add to array
for n in range(0, len(flash_lat_value)):
    flash_lat.append(flash_lat_value[n])
iss_lat_value = datafile.variables['viewtime_lat'][:].data.tolist()
for n in range(0, len(iss_lat_value)):
    iss_lat.append(iss_lat_value[n])
flash_lon_value = datafile.variables['lightning_flash_lon'][:].data.tolist()
for n in range(0, len(flash_lon_value)):
    flash_lon.append(flash_lon_value[n])
iss_lon_value = datafile.variables['viewtime_lon'][:].data.tolist()
for n in range(0, len(iss_lon_value)):
    iss_lon.append(iss_lon_value[n])
```

```
flash_time = datafile.variables['lightning_flash_TAI93_time'][:].data.tolist()
time_value_units = datafile.variables['lightning_flash_TAI93_time']
stop_time = num2date(flash_time[:], time_value_units.units)
final_flash_time_value = max(stop_time)
for s in range(0, len(stop_time)):
     flash_time_dt.append(stop_time[s].strftime("%H:%M:%S"))
for t in range(0, len(flash_time_dt)):
    flash_time_split = re.split(':', flash_time_dt[t])
    flash_time_sec = (int(flash_time_split[0])*3600) + (int(flash_time_split[1])*60) + int(flash_ti
    flash_time_final.append(flash_time_sec)


for g in range(0, len(flash_lat)):
    if lon_W < flash_lon[g] < lon_E and lat_S < flash_lat[g] < lat_N:
        flash_lon_final.append(flash_lon[g])
        flash_lat_final.append(flash_lat[g])
        flash_time_final_final.append(flash_time_final[g])
for h in range(0, len(iss_lat)):
    if lon_W < iss_lon[h] < lon_E and lat_S < iss_lat[h] < lat_N:
        iss_lon_final.append(iss_lon[h])
        iss_lat_final.append(iss_lat[h])



plt.figure(figsize=((20, 20)))  # Set plot dimensions
map = plt.axes(projection=ccrs.PlateCarree(central_longitude=0.0))
gl = map.gridlines(crs=ccrs.PlateCarree(central_longitude=0.0),draw_labels=True, linewidth=0.8, alp
#lightning = map.hexbin(flash_lon, flash_lat, gridsize=300, bins='log',cmap='jet', mincnt=1 ,zorder

#Draw geographic boundaries and meridians/parallels
map.set_extent([lon_W, lon_E, lat_S, lat_N])
#map.set_extent([55, 105,0, 40])
#map.set_extent([73, 78,15, 18])
map.coastlines(color='white')
map.add_feature(cfeature.LAND, facecolor='gray')
map.add_feature(cfeature.BORDERS, edgecolor='white')
map.add_feature(cfeature.OCEAN, facecolor='black')
gl.ylocator = mticker.FixedLocator([-90, -60, -30, 0, 30, 60, 90])
gl.xformatter = LONGITUDE_FORMATTER
gl.yformatter = LATITUDE_FORMATTER
gl.top_labels = False
gl.right_labels = False

map.scatter(iss_lon_final, iss_lat_final, s=9, c='r', edgecolor='black',alpha=0.75, transform=ccr
map.scatter(flash_lon_final, flash_lat_final, s=9, c='y', edgecolor='black',alpha=0.75, transform
map.scatter(lds_lon_final,lds_lat_final,s=9,c='b',edgecolor='black',alpha=0.75, transform=ccrs.Pl
eate CSV files of values from the populated flash_lat/lon arrays
============================================================================
with open(csvfile, 'w', newline='') as myfile:
    writer = csv.writer(myfile)# Define headers in row (zip creates columns)
    writer.writerows(zip(["flash_lat"], ["flash_lon"], ["flash_time_dt"], ["flash_time_final"]))#
    writer.writerows(zip(flash_lat, flash_lon, flash_time_dt, flash_time_final))
```

```
with open(csvfileFinal, 'w', newline='') as myfile:
    writer = csv.writer(myfile)
    writer.writerows(zip(["flash_lat_final"], ["flash_lon_final"], ["flash_time_final_final"]))
    writer.writerows(zip(flash_lat_final, flash_lon_final, flash_time_final_final))
with open(csvfileFinal_lds, 'w', newline='') as myfile:
    writer = csv.writer(myfile)
    writer.writerows(zip(["lds_lat_final"], ["lds_lon_final"], ["lds_time_sec_final"]))
    writer.writerows(zip(lds_lat_final, lds_lon_final, lds_time_sec_final))
with open(csvfilepath, 'w', newline='') as myfile:
    writer = csv.writer(myfile)
    writer.writerows(zip(["iss_lon_final"], ["iss_lat_final"]))
    writer.writerows(zip(iss_lon_final, iss_lat_final))
```

```
if begin_date != end_date:
    plot_title = 'ISS LIS Detected Lightning Flash Locations ' + begin_date + ' - ' + end_date
else:
    plot_title = 'ISS LIS Detected Lightning Flash Locations ' + end_date

plt.title(plot_title, fontsize=18)

#Save the plot as an image
plt.savefig(os.path.join(dataDir, 'isslis_flashloc_' + begin_int + '_' + end_int + str(file_no)
```

Chapter 4

Results and Interpretations

4.1 Observations

4.1.1 Daily Data Analysis

Date	Both Data include	Only LIS Data	Only LDS Data	Both data exclude
6 April 2020	32	8	28	382
7 April 2020	27	9	25	419
8 April 2020	27	4	34	385

(i)data_table_kappa_stats

Date	Cohen_kappa_value	Agreement
6 April 2020	0.597	Moderate
7 April 2020	0.576	Moderate
8 April 2020	0.545	Moderate

(ii)data_table_cohen_value

Daily data analysis from 1 April 2020 to 28 April 2020 had been done in our project and majorly dates come with a result of Moderate Agreement i.e., Cohen kappa value 0.41 to 0.60. All the rectangles have 450 grids (15X30). A table named data_table_kappa_stats contains grids values out of 450. Both Data include refers to the grid which both the datasets had available. Similarly, Both data exclude refers to the grid which both the datasets had absent.

Cohen's Kappa Interpretation

Cohen_kappa_Value	Interpretation
0	No Agreement
0.10 - 0.20	Slight Agreement
0.21 - 0.40	Fair Agreement
0.41 - 0.60	Moderate Agreement
0.61 - 0.80	Substantial Agreement
0.81 – 0.99	Near Perfect Agreement
1.00	Perfect Agreement

4.1.2 Weekly Data Analysis

Week	Both Data include	Only LIS data	Only LDS data	Both data exclude
Week 1(1-7 April)	91	69	74	2943
Week 2(8-14 April)	78	97	72	2452
Week 3(15-21 April)	118	136	103	2793
Week 4(22-28 April)	126	179	82	2805

(iii)data_table_weekly

Week	Cohen_kappa_value	Agreement
Week 1(1-7 April)	0.536	Moderate
Week 2(8-14 April)	0.446	Moderate
Week 3(15-21 April)	0.456	Moderate
Week 4(22-28 April	0.448	Moderate

(iv)data_table_cohen_kappa_weekly

4.1.3 Monthly Data Analysis

Month	Both Data include	Only LIS data	Only LDS data	Both data exclude
April	413	481	331	10993

(v)data_table_monthly

Month	Cohen_kappa_value	Agreement
April	0.4715	Moderate

(vi)data_table_cohen_kappa_monthly

All the data tables had been designed on daily basis and that process continues for to whole year. The kappa value result is 0.41 - 0.60 which means a good agreement in both the data sets. Still, there have certain limitations in these examinations. We note that often the ground-based network monitors more lightning occurrences compared to the space-based data. The present setup has a limitation with some datasets which can be upscale in the future by improving the network and technology.

Chapter 5

Conclusion

Two datasets namely Lighting detection sensor(LDS) and Lighting imaging sensor (LIS) are compared using kappa stats that have been analyzed and compared. It is concluded that from 1 April to 28 April 2022 the agreement of stats is moderate which means 40-60 percent of data are accurate in LDS data if we consider the ISS-LIS Data to be more accurate. However, if we consider the ground data to be the reference, the LIS data misses about 40-60% of the data.
The final image shows the data according to the path of the satellite. In some areas the path of the satellite not covering the portion of India so on that part, we assuming the no flash occurrence of ISS-ISS points.
The main objective of this study was to investigate how well the observations of a ground-based LDS, are linked to space-based optical lightning signatures of the LIS on the ISS over a part of India.

Bibliography/ References

1. ISS LIS Data Sets
   [https://ghrc.nsstc.nasa.gov/lightning/data/data_lis_iss.html]

2. Quality Controlled Lightning Imaging Sensor (LIS) on International Space Station (ISS) Science Data.
   [https://cmr.earthdata.nasa.gov/search/concepts/C2015490897-GHRC_DAAC.html]

3. Christian Hugh J., Richard J. Blakeslee, Steven J. Goodman, Douglas A. Mach, et al. (1999): The Lightning Imaging Sensor.
   [https://www.researchgate.net/profile/Dennis_Buechler/publication/4667066_The_Lightning_Imaging_Sensor/links/00b495284ee208717c000000/The-Lightning-Imaging-Sensor.pdf]

4. ISS LIS Lightning Flash Location Quickview using Python 3.0 and GIS.
   [https://ghrc.nsstc.nasa.gov/home/data-recipes/iss-lis-lightning-flash-location quickview-using-python-30-and-gis]

5. Pinto, Jr, O., Lightning in the Tropics: From a Source of Fire to a Monitoring System of Climate Changes, Nova Science Publishers, 2009.

6. Sivakandan, M., Paulino, I., Taori, A. and Niranjan, K., Mesospheric gravity wave characteristics and identification of their sources around spring equinox over Indian low latitudes. Atmos. Meas. Techn., 2016, 9, 93–102; doi:10.5194/amt-9-93-2016.

7. Atmospheric Measurement Techniques, Investigations on the atmospheric lightning: establishment of network, geolocation and identification of vulnerable ones. NRSC Technical Report, NRSC-ECSA-ACSG-ACSD-MAR-2018-1126-1.0, 2018.

8. Atmospheric Lightning Essential Climate Variable (ECV) mapping over India using NRSC/ISRO Lightning Detection Sensor Network- by Alok Taori, Arun Suryavanshi2, Biswadip Gharai1, M. V. R. Seshasai1.

9. https://kannada.oneindia.com/news/chitradurga/154-animals-killed-by-lightning-at-molakalmuru-255462.html

10. https://idostatistics.com/cohen-kappa-free-calculator/

11. https://www.sciencelearn.org.nz/resources/239-lightning-explained