# POLSAR Data Processing and relation with Global forest height

# (Haldwani,Uttarakhand )

Group 1

Sai Susrush

Abhishek Kushwah

Harshavardhan Thorat

Submitted to

Prof. Gulab Singh


GNR 401

November '22

# INTRODUCTION

- Location: Haldwani Uttarakhand
- Data collected from 2010
- Data was taken from the Japanese satellite ALOS
- Sensor used: PALSAR
- Guide: Prof. Gulab Singh, TA. Konica Bhandari

# Aim Of The Project

- POLSAR Scattering data processing and relation with global forest height
- To perform regression analysis on Global forest height recorded from polsar scattering

The PolSAR sensor aboard the ALOS Satellite is a Synthetic Aperture Radar (SAR), which emits microwaves and receives the reflection from the ground to acquire information. Since it does not need other light sources, such as the sun, SAR has the advantage of providing satellite images regardless of day or night. The frequency of transmitting and receiving microwave is L-band, which is less affected by clouds and rain. This all-weather observing capability is suitable for monitoring disasters rapidly. In addition, L-band microwaves can reach the ground by partially penetrating through vegetation to obtain information of vegetation and ground surface.

# Scattering Mechanisms

- Surface Scattering: Scattering which takes place only on the border surface between two different but homogeneous media, from one of which electromagnetic energy is incident onto the other.

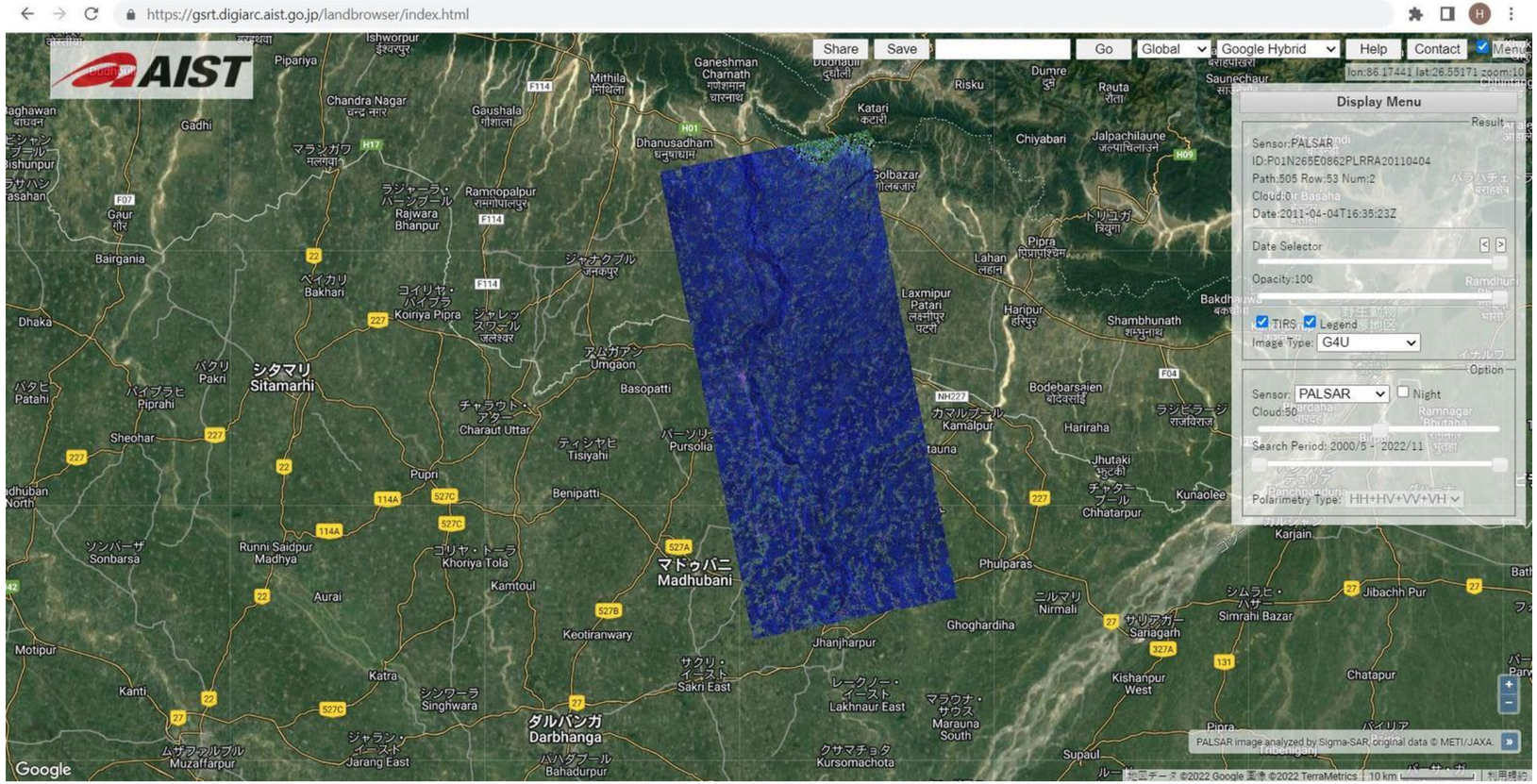
- Volume Scattering: This scattering occurs inhomogeneous medium. Radar signals can penetrate into the medium and will give volume information.

- Double Bounce Scattering: This reflection occurs at two surfaces at right angle and due to double bounce, the image appears very bright

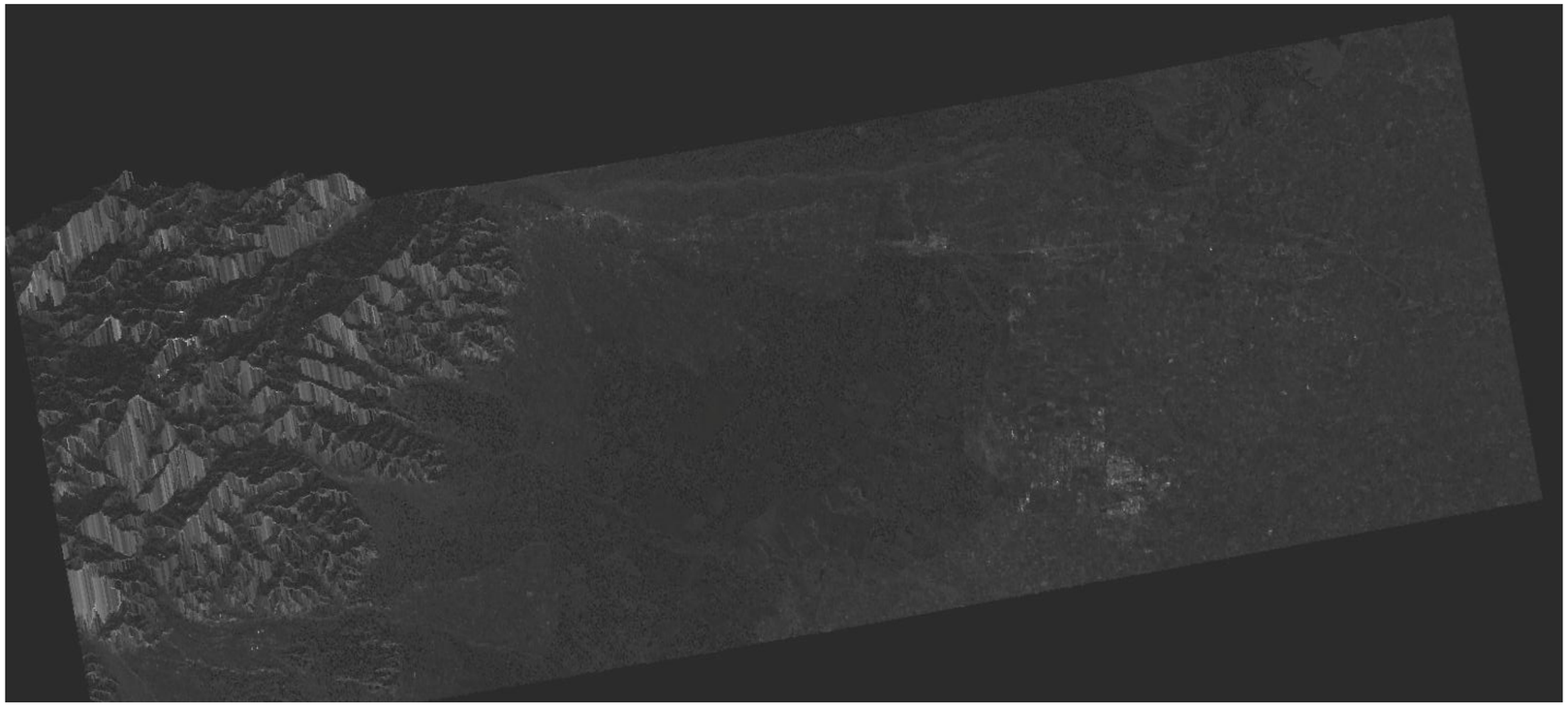
# Procedure

Phase 1:

- To get PolSAR data from ALOS, we used LandBrowser, a system which can search, browse and download satellite data.



2. We extracted scattering information for 255 points from the tif into csv format



Phase 2:

1 . From the CSV data, we extracted Surface scattering (Ps), Double Bounce Scattering (Pd), Volume scattering (Pv), and Helix (Pc) and height values.

2. Then, we normalized the scattering values by dividing them by the sum of respective values. Then we extracted height values using gis tool.
https://drive.google.com/drive/folders/16QE2MeZclMpC98gbLHztKtTQtfgXak-V

3. Then, we used ht from ht map. So we now made a model to extrapolate ht to the entire area, keeping pv,pc, and pd values as dependent variables and studying the effect of scattering altering the tree height.

4. Then we compared our height values with standard height values and got a consistency of 80%

# Phase 3:

We ran the following python code to convert the previous height data to get the map and graph :

```
df=pd.read_csv("f.csv", na_values= '#DIV/0!' )                # importing datasets
df.replace([np.inf, -np.inf], np.nan, inplace=True)   # df = pd.read_csv('f.csv')
print(df)
x = df[['ps', 'pv', 'pc', 'pv']]
y = df['Height_Map']
x=df.iloc[:,0:4]
y=df.iloc[:,4]
print(x)
print(y)


from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.22)
```

```
# x_train = pandas.get_dummies(x_test[0, 1, 2, 3])
# y_train = pandas.get_dummies(y_test[cols])
# from sklearn import preprocessing
# le = preprocessing.LabelEncoder()
# from sklearn.ensemble import RandomForestClassifier
# le= RandomForestClassifier(n_estimators= 10, criterion="entropy")
# le.fit(x_train, y_train)
# classifier.predict(x_test)
from sklearn.preprocessing import StandardScaler
st_x= StandardScaler()
x_train= st_x.fit_transform(x_train)
x_test= st_x.transform(x_test)

from sklearn.linear_model import LinearRegression
clf=LinearRegression()
clf.fit(x_train,y_train)
from sklearn.ensemble import RandomForestClassifier
clf= RandomForestClassifier(n_estimators= 10, criterion="entropy")
clf.fit(x_train, y_train)
# y_pred= clf.predict(x_test)
# from sklearn.metrics import confusion_matrix
# cm= confusion_matrix(y_test, y_pred)
print(clf.score(x_test,y_test))
ytest_pred= clf.predict(x_test)
from sklearn.metrics import confusion_matrix
cm= confusion_matrix(y_test, ytest_pred)
# matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sn
plt.rcParams['figure.figsize'] = (10,6)
```

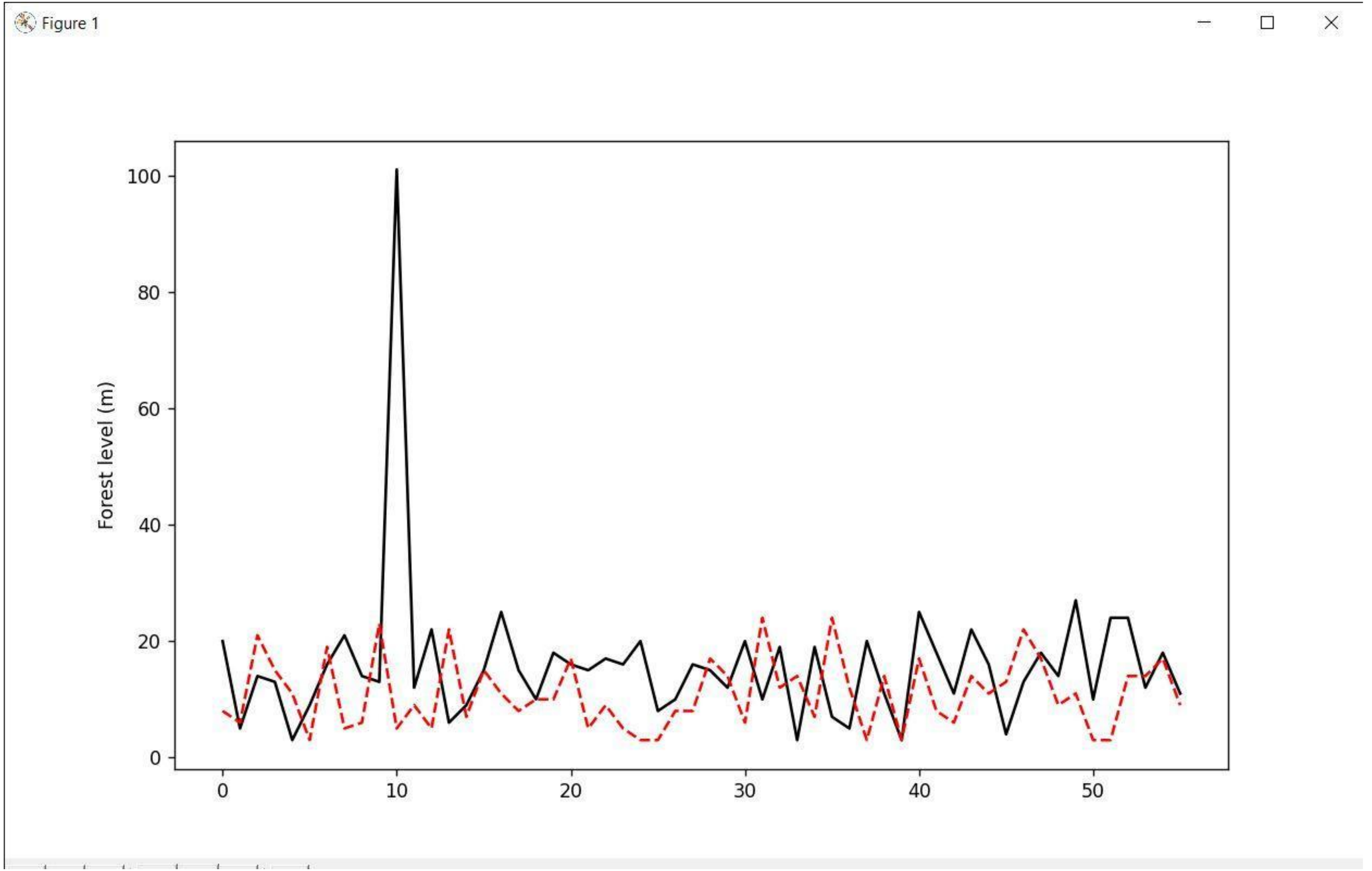


Figure 1
Forest level (m)
0
20
40
60
80
100
0
10
20
30
40
50