Summer Internship Programme 2024

PREDICTION OF SWH USING WIND DATA

By

KARTIK DHYANI

Department of Electrical and Electronics

BHARATI VIDYAPEETH’S COLLEGE of ENGINEERING

Under the guidance of

DR.Siva K Srinivas

Scientist C – INCOIS(OMDA)

Indian National Centre for Ocean Information Services (INCOIS)

Ministry of Earth Sciences, Government of India

Pragati Nagar (B.O), Nizampet (S.O)

Hyderabad – 500 090

July 2024

DECLARATION

I hereby declare that the project entitled “FORECASTING OF SWH USING WIND DATA” is an authentic record of my own work completed under the guidance of DR.Siva K Srinivas Scientist C’, Indian National Centre for Ocean Information Services (INCOIS), Ministry of Earth Sciences, Hyderabad, India. I further declare that, wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature.

KARTIK DHYANI

Place : Hyderabad

Date :

ACKNOWLEDGEMENT

I would like to convey my heartfelt gratitude to all the individuals who have played a significant role in the completion of this project. Firstly, I am deeply grateful to my guide, DR.Siva K Srinivas, whose guidance, expertise, and unwavering support have been invaluable throughout this internship period.

I thank the Indian National Centre for Ocean Information Services (INCOIS) for this opportunity.

My sincere thanks to my family for their valuable suggestions and constant support, which helped me to complete this work successfully. To all those mentioned, I extend my sincere gratitude for their invaluable contribution to this report and for being an integral part of my internship journey.

KARTIK DHYANI

TABLE OF CONTENTS

1. Abstract……………………………………………
2. Introduction………………………………………..
3. Data collection…………………………………….
4. Basic terms for ML………………………………...
5. Regression coefficient plots between Wind and Swh before vs after training saterlite data……...
6. Plots of saterlite track on cyclone dates in 2023…...
7. Cyclone Plots of wind and swh data obtained by saterlite data files vs numerical methods, also ploting saterlite path……………………………..
8. Predicted plot of Swh using wind data…………….
9. Results ……………………………
10. Conclusion…………………………………
11. References………………………………
12. Appendix………………………………………

ABSTRACT

This project focuses on forecasting Significant Wave Height (SWH) using wind data, as we observed there is a great regression relationship between them. Regression coefficient, also known simply as a coefficient, is a numerical measure that quantifies the strength and direction of the relationship between two variables in a regression analysis, regression coefficient lies between 0 to 1 . SWH prediction plays a crucial role in maritime operations, coastal management, and offshore industries. By harnessing historical wind data, which has shown a strong correlation with SWH, this study aims to develop a robust forecasting model. The methodology involves data preprocessing, feature engineering, and applying machine learning algorithms such as regression techniques to model the relationship between wind parameters and SWH. Validation and evaluation of the model will be conducted using numerical methods to ensure reliability and accuracy. The outcomes of this research will contribute to enhancing the efficiency and safety of maritime activities through improved SWH forecasting based on readily available wind data.

INTRODUCTION

Significant Wave Height (SWH) is a crucial parameter in oceanography and marine engineering that describes the average height of the highest third of waves in a given sea state. It is measured from trough to crest, typically by buoys, ships, or satellites equipped with wave sensors. Swh holds its importance in various region like :

* **Maritime Safety and Operations**
* **Coastal and Offshore Engineering**
* **Weather Forecasting and Climate Studies**
* **Research and Scientific Exploration**
* **Environmental Monitoring**
* **Offshore Renewable Energy**

DATA COLLECTION

The first and important step is to get wind and swh data , for that we have many sources like :

* National Oceanic and Atmospheric Administration (NOAA)
* Satellite Data (al-l3,J3-l3,c2-l3,etc)
* Open Data Platforms (Copernicus Marine Environment Monitoring Service (CMEMS))

After collecting data we have to check and make sure that there is both wind and swh is present if they are not present we have to download 2 different files and then comapre it and plot regression coffiecient between them ,but make sure that both file are of same year , in my project I have taken year of 2023 for my research , then perform our task like plotting and finding regression coffiecient between wind and swh to show that they actually have a good regressin cofiecient between them not only in cyclone days but also normal days **.**

BASIC TERMS FOR ML

For ml we have to know basic terms like:

MSE,RMSE,R-squared ,Types of Regression

* **Mean Squared Error (MSE)**:

MSE measures the average of the squared differences between predicted values and actual values.

*MSE = Σ(*yi *–* pi)*2/*n

* **Root Mean Squared Error (RMSE)**:

RMSE is the square root of the MSE and provides a measure of the average magnitude of the errors in the predicted values.

RMSE = sqrt [(Σ(yi – pi)²) / n]

* **R-squared (Coefficient of Determination)**:

R-squared is a statistical measure that indicates the proportion of the variance in the dependent variable that is predictable from the independent variables.

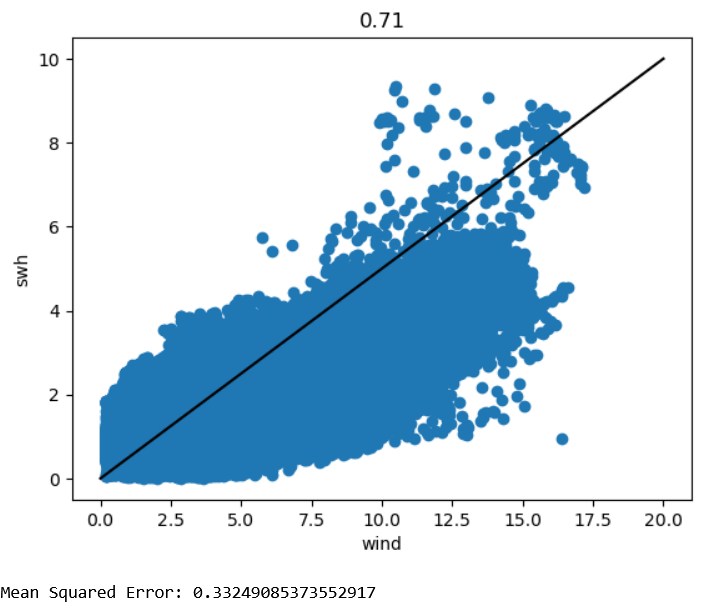
R-squared = **1 – (∑( yii − y ) ^2 / ∑ ( yi − y ) ^2)**

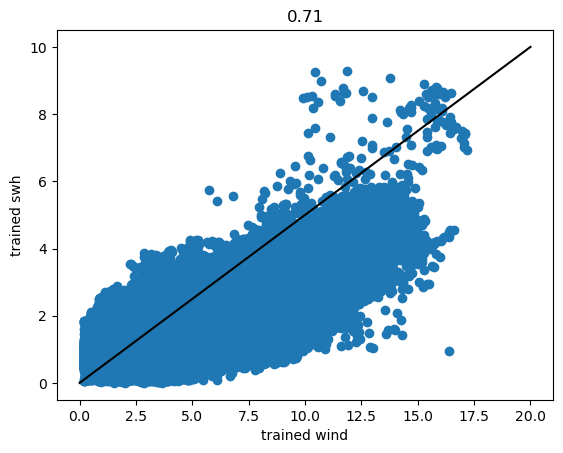
* **Types of regression:**

1. LinearRegression: Linear regression models the relationship between a dependent variable y and one or more independent variables x by fitting a linear equation.
2. Polynomial Regression**:** Polynomial regression extends linear regression by adding higher-degree polynomial terms to the model equation.
3. Support Vector Regression (SVR)**:** SVR is a type of regression analysis where the goal is to minimize the error, but in the context of a margin, meaning that SVR tries to fit as many instances as possible within a boundary that has a margin of tolerance.

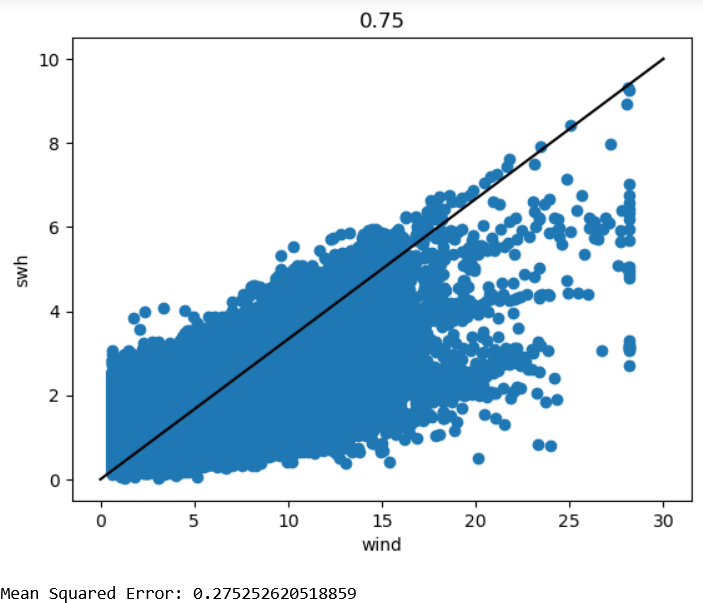
Regression coefficient plots between Wind and Swh before vs after training saterlite data

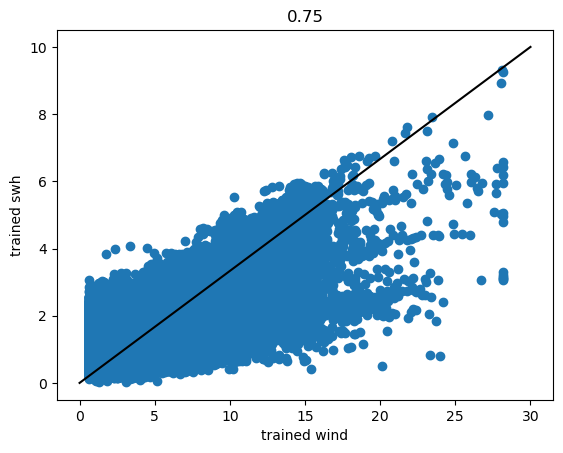
1. al-l3 2023:



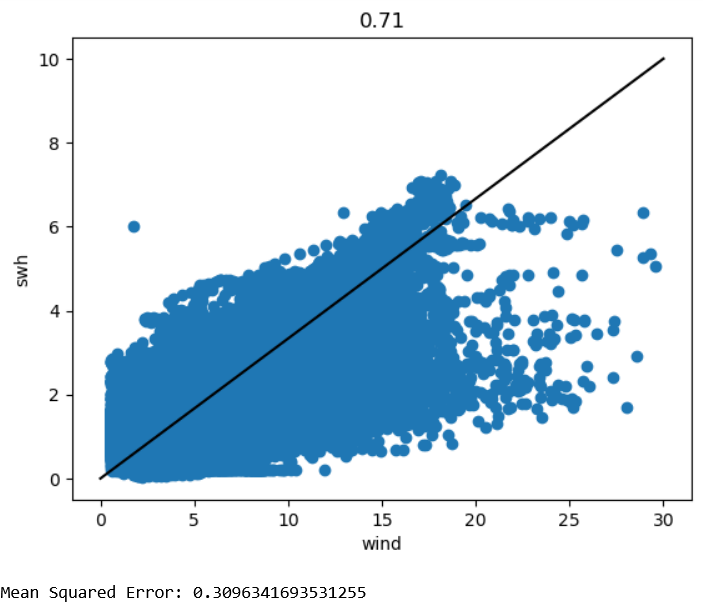


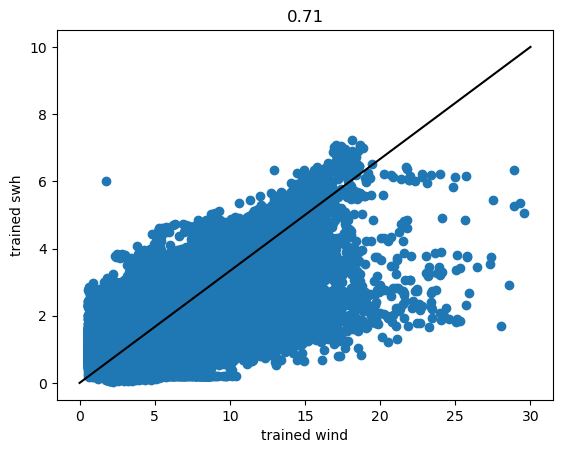
1. C2-l3 2023:



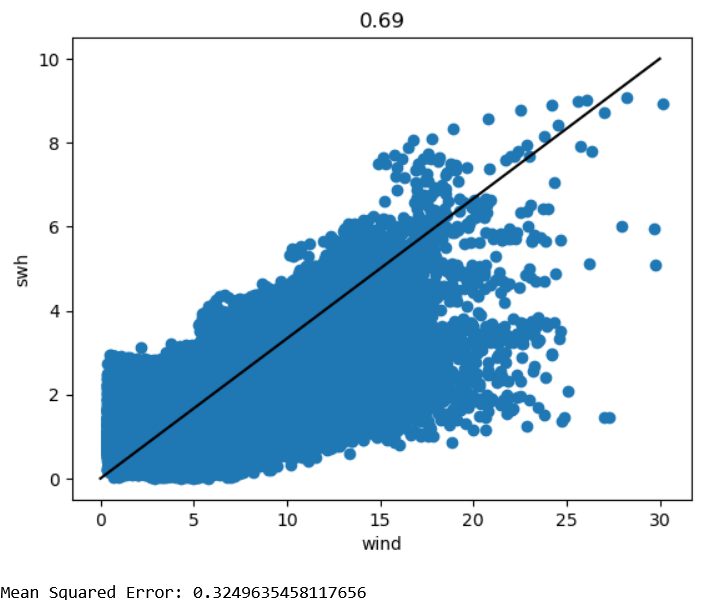


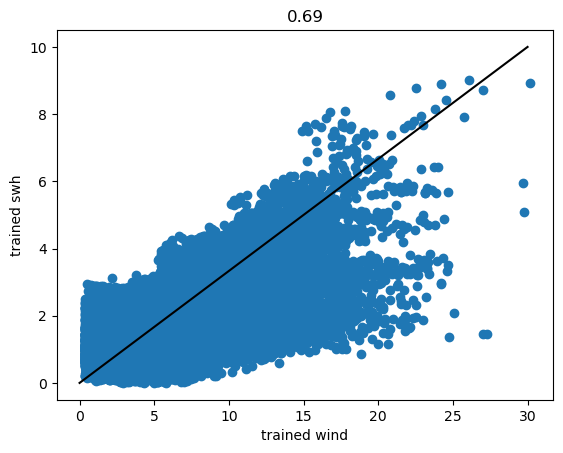
1. H2b-l3 2023:



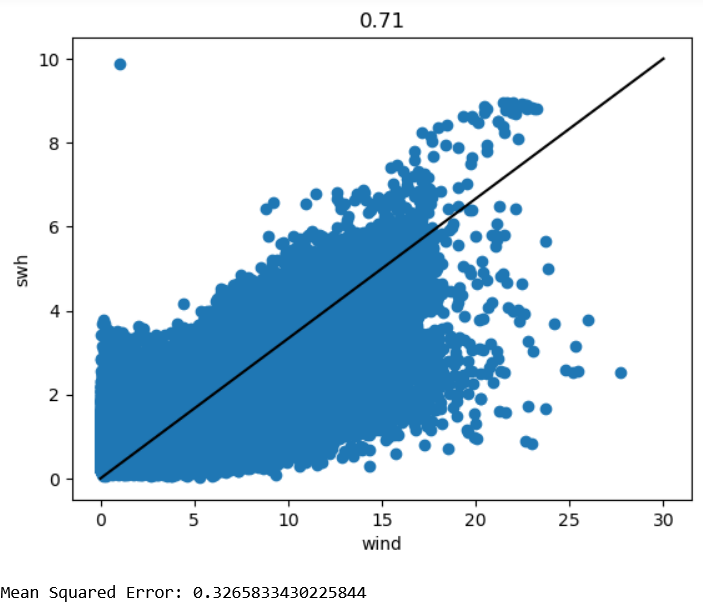


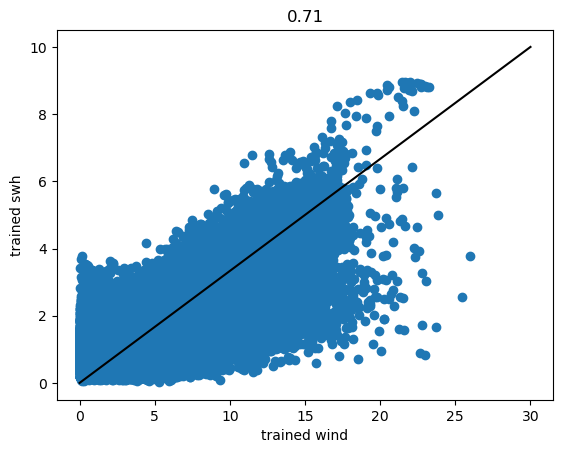
1. H2c-l3 2023:



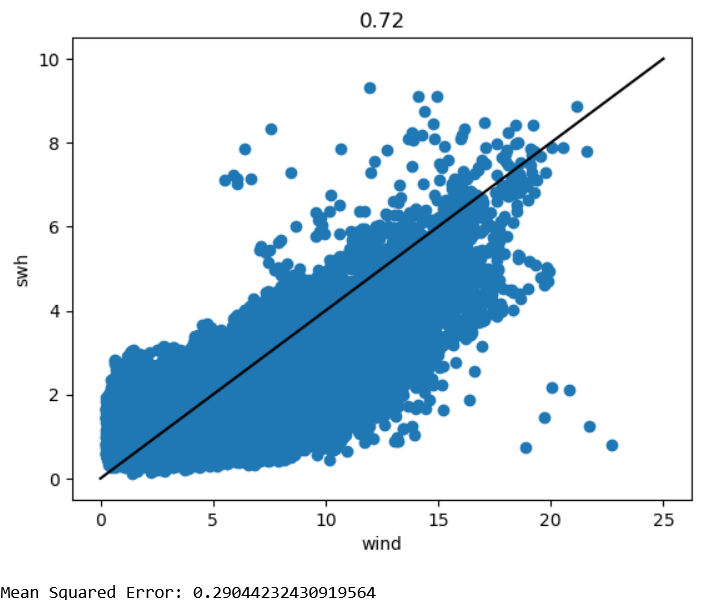


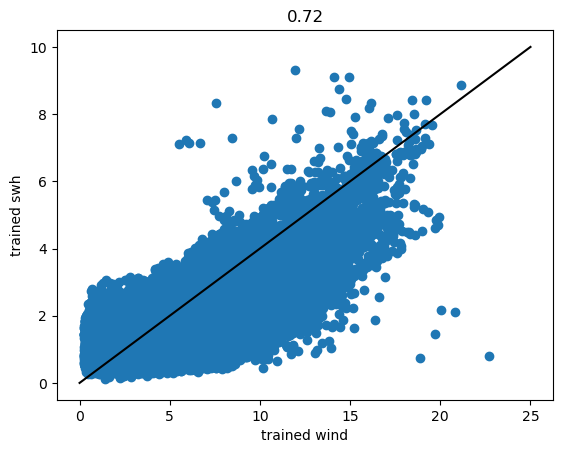
1. J3-l3 2023:



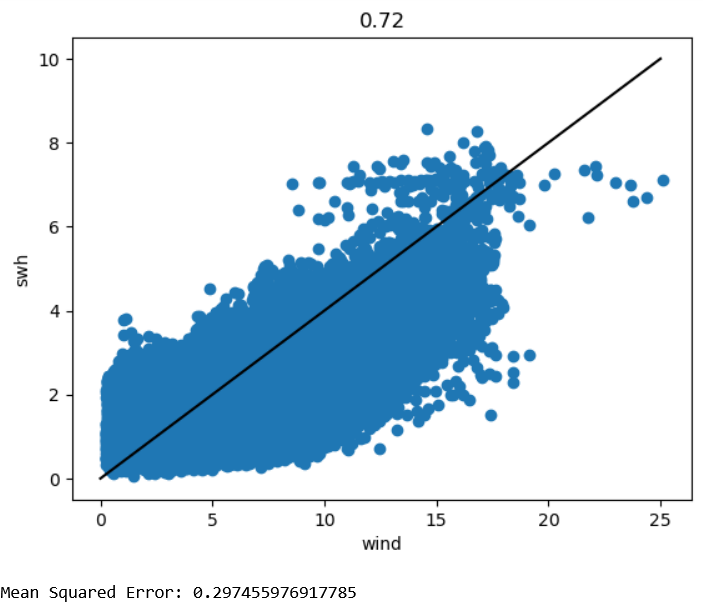


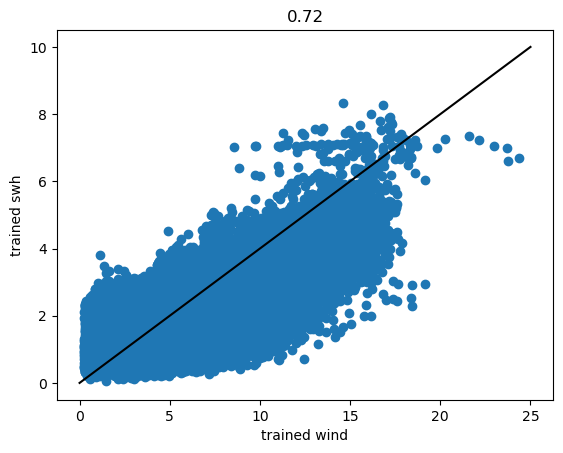
1. S3a-l3 2023:



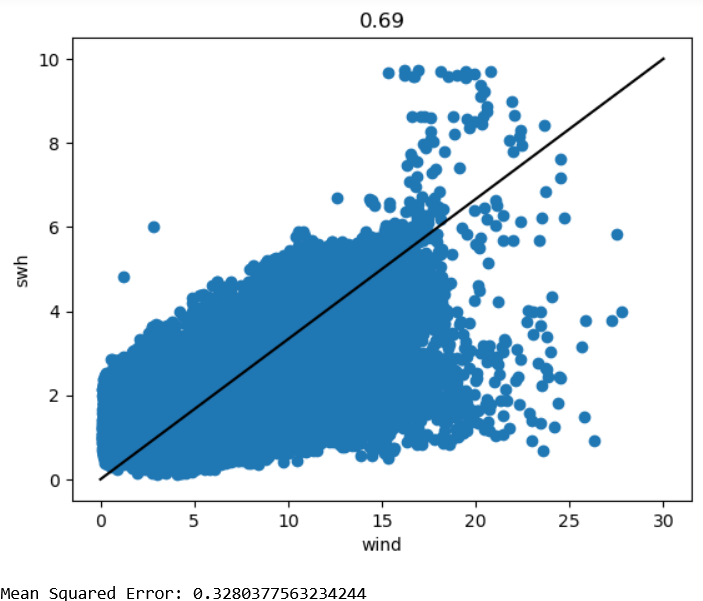


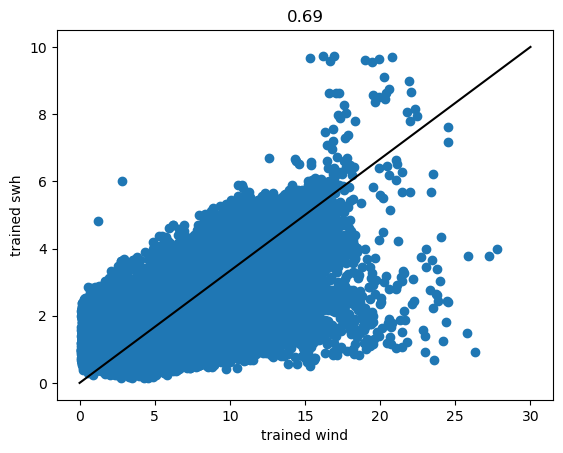
1. S3b-l3 2023:



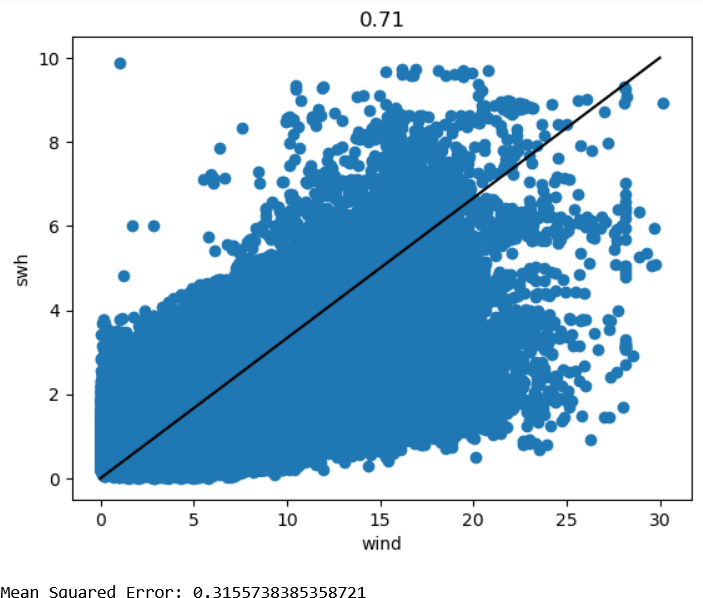


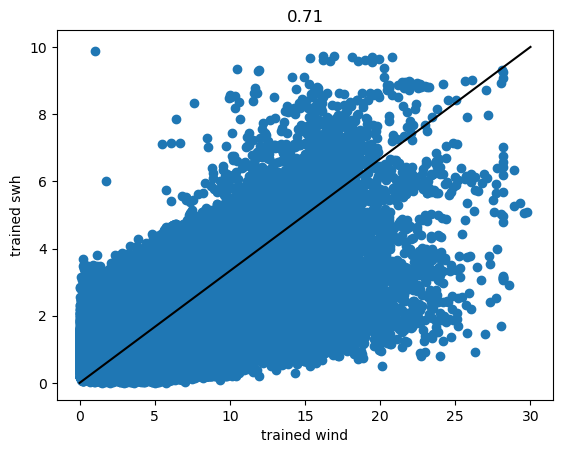
1. S6a-l3 2023:





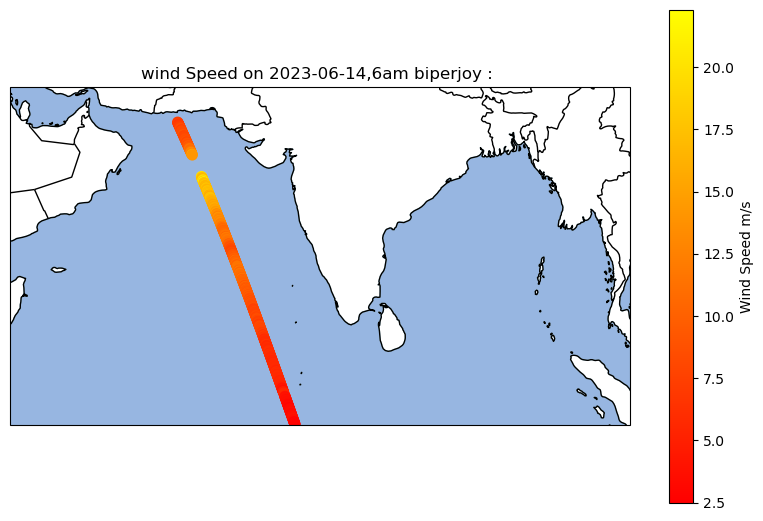
1. All satellite:

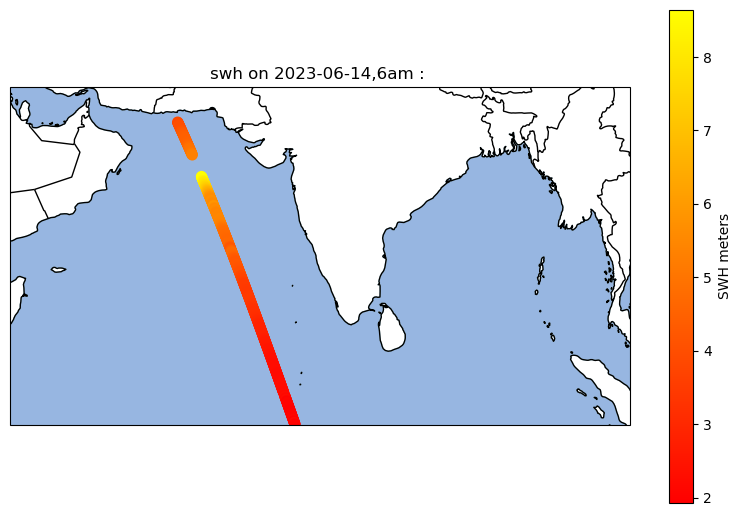




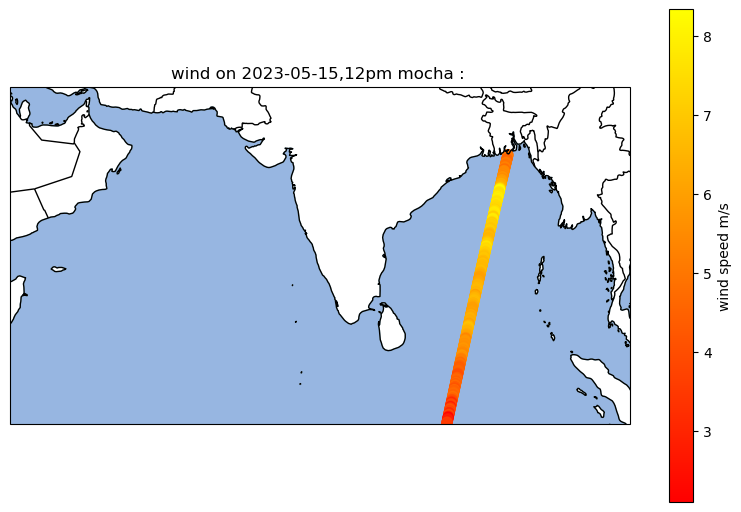
Plots of saterlite track on cyclones in 2023:

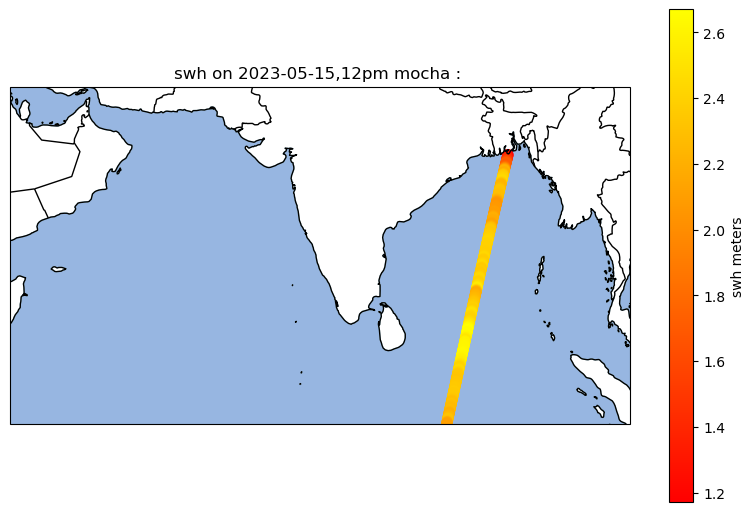
1. Biparjoy



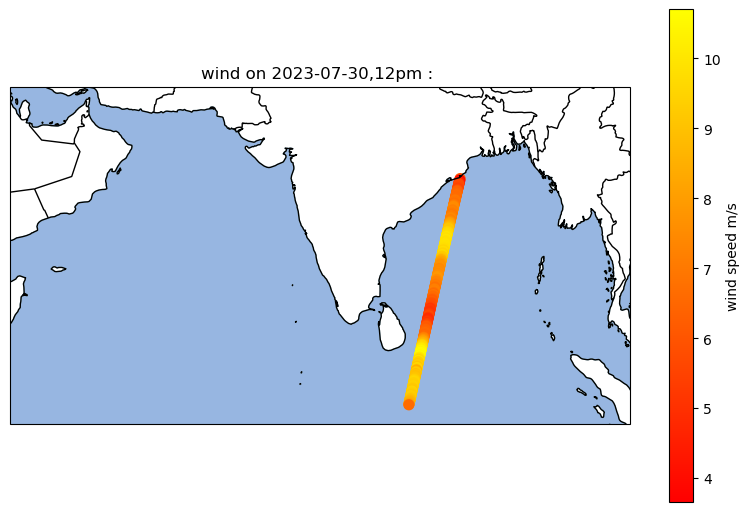


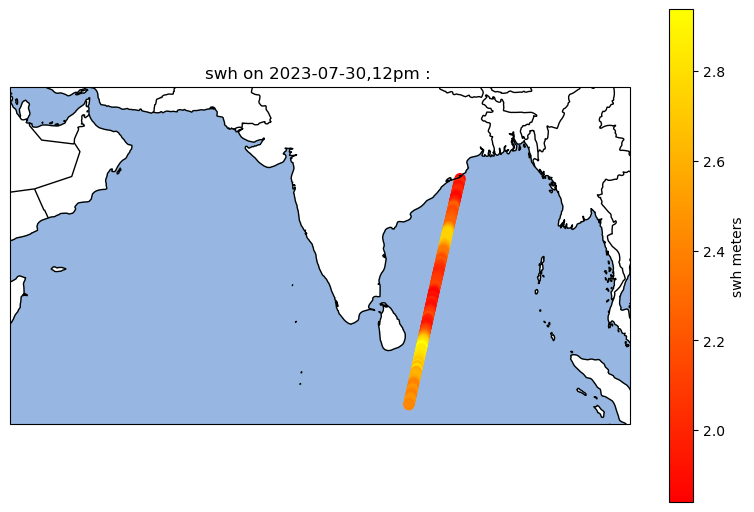
1. Mocha



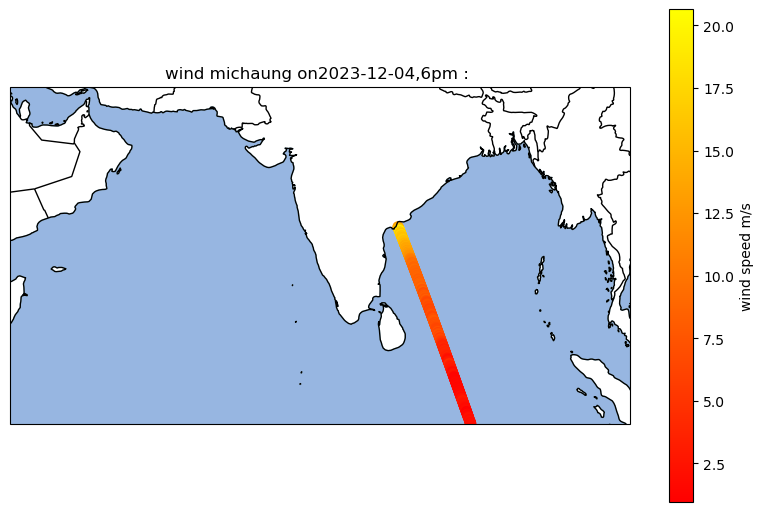


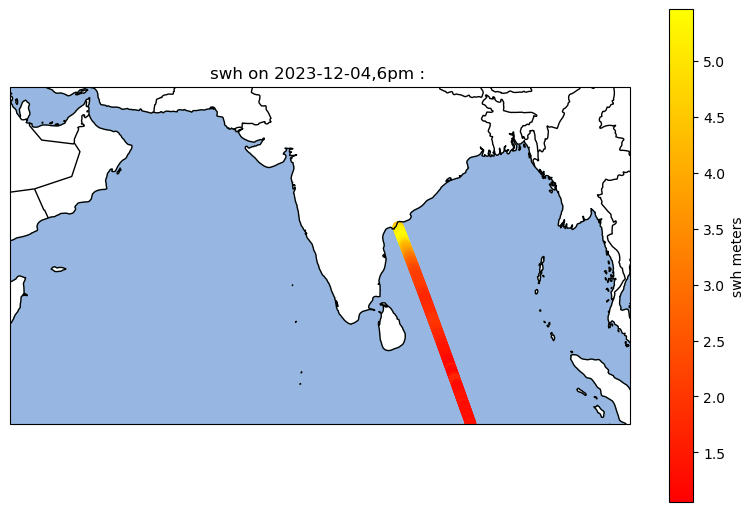
1. BOB 3





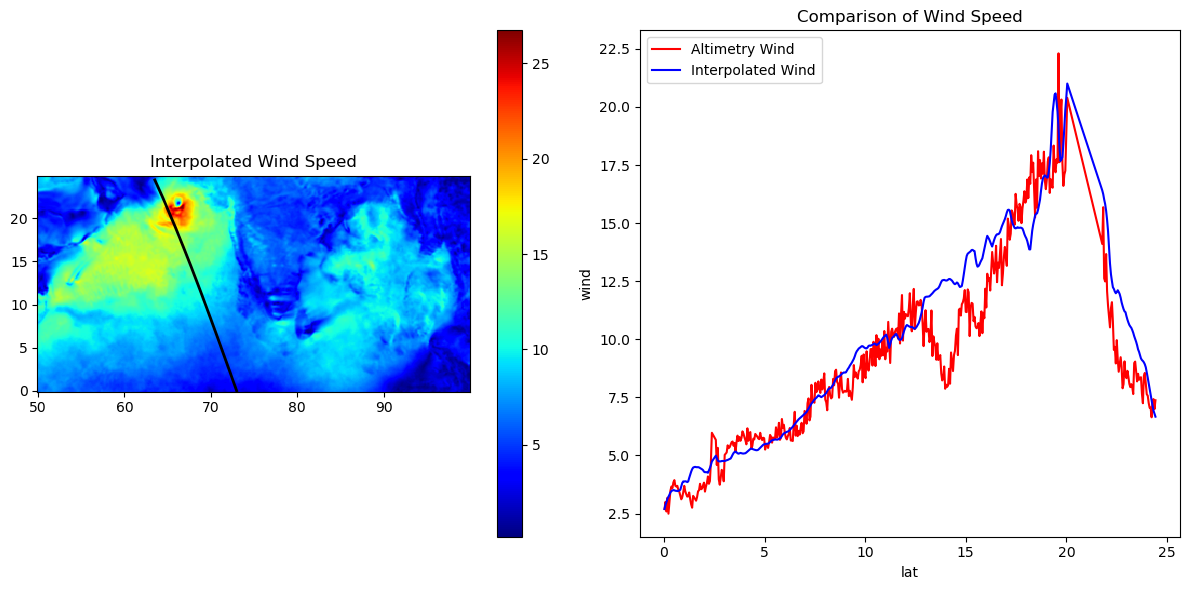
1. Michaung

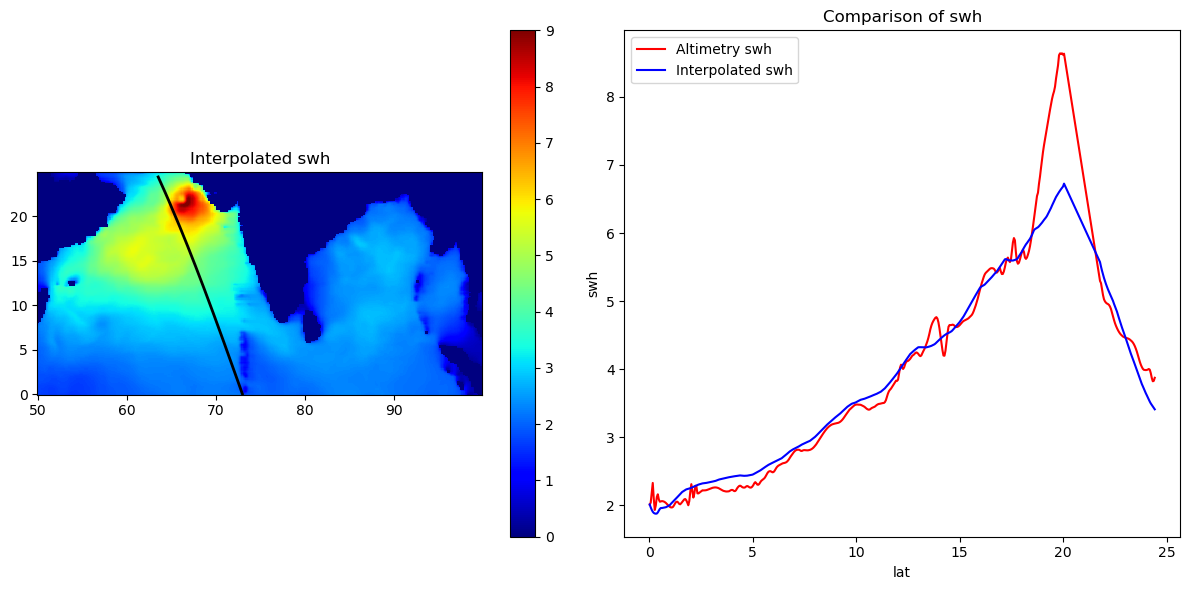




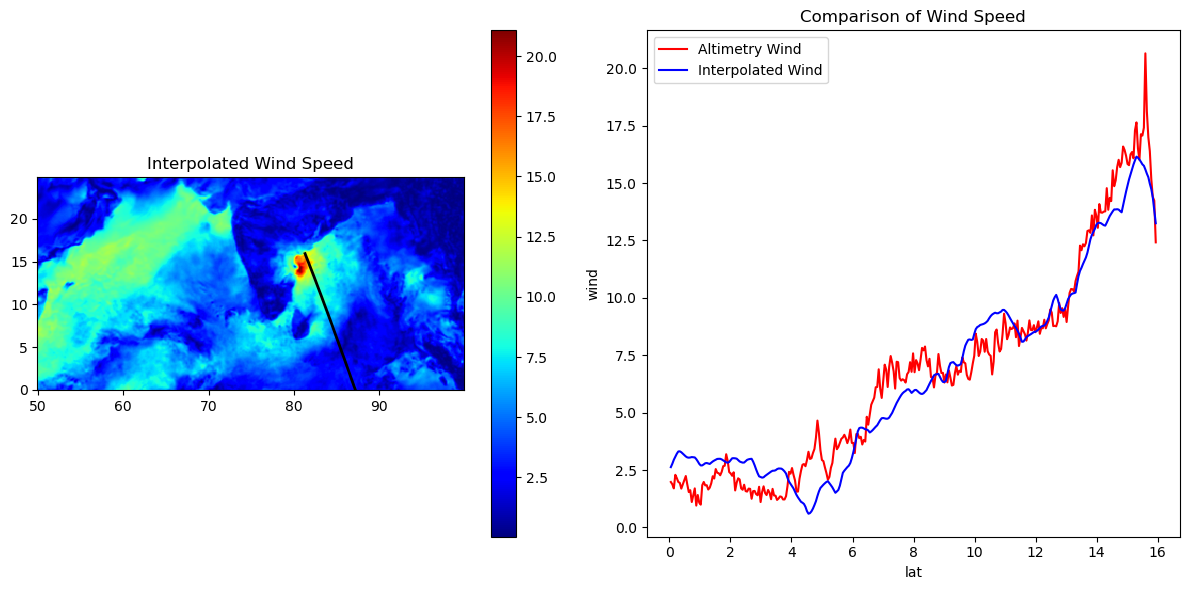
Cyclone Plots of wind and swh data obtained by saterlite data files vs numerical methods, also ploting saterlite path

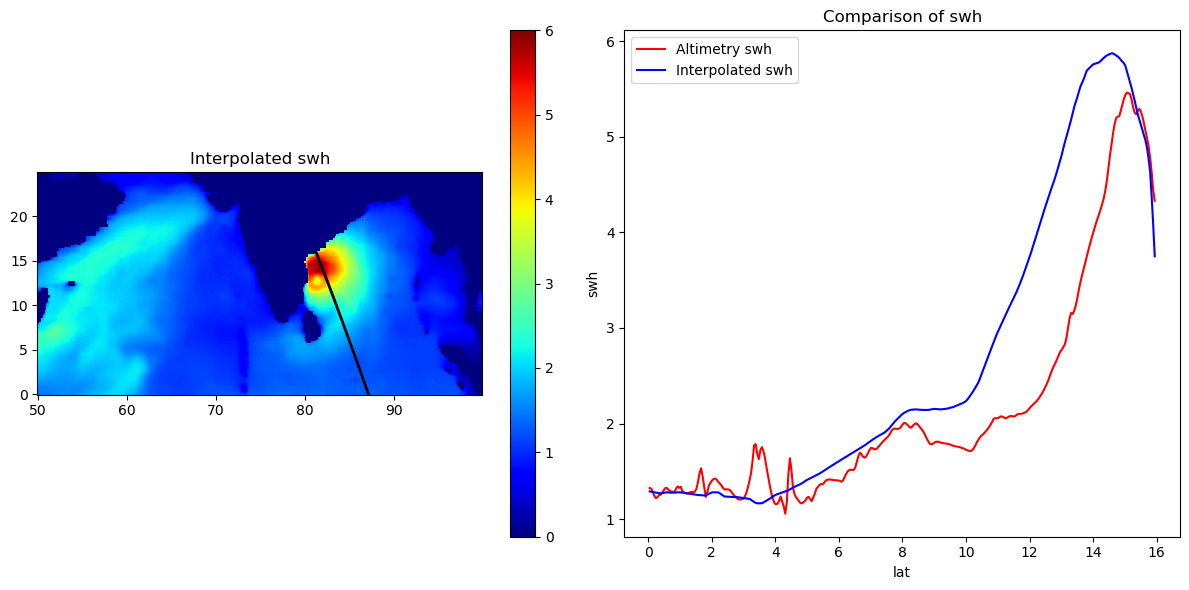
1. Biparjoy





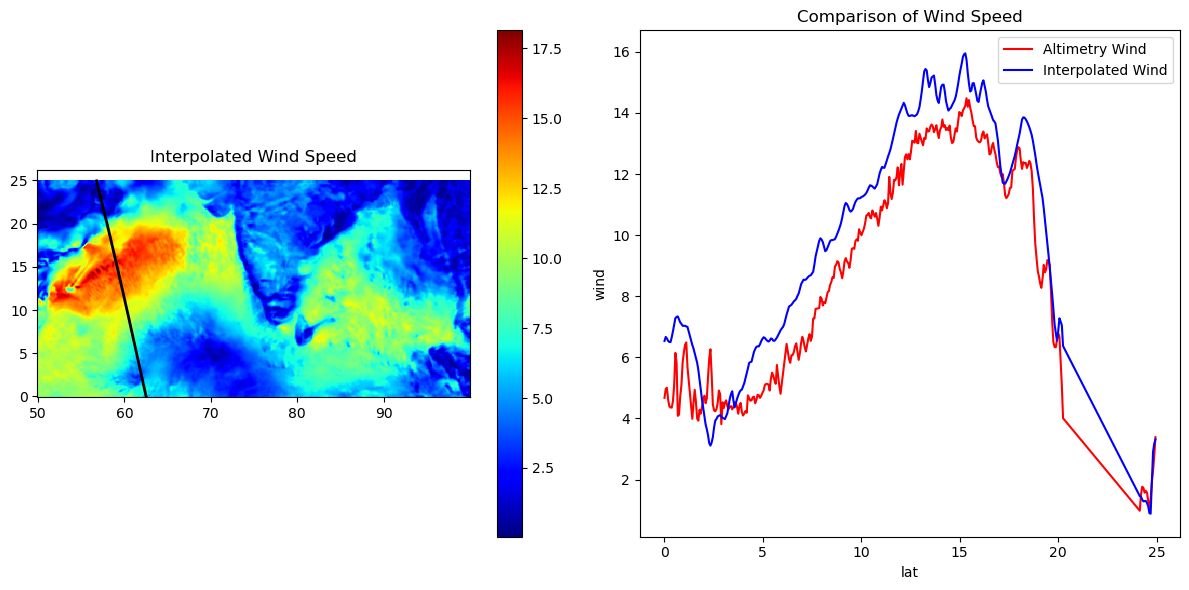
1. Michaung

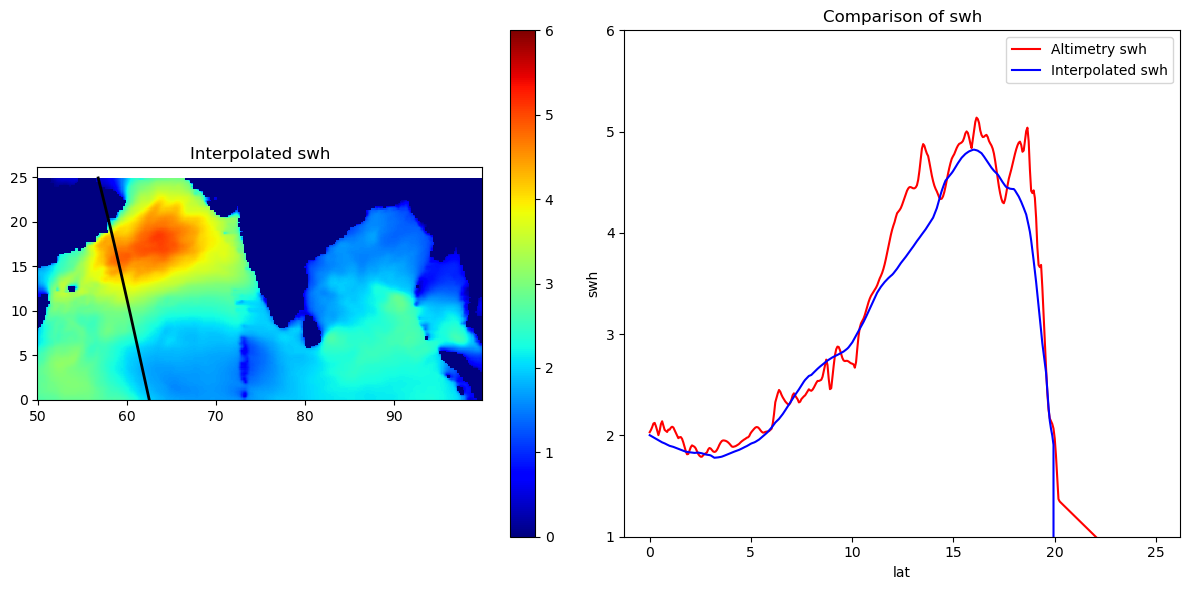




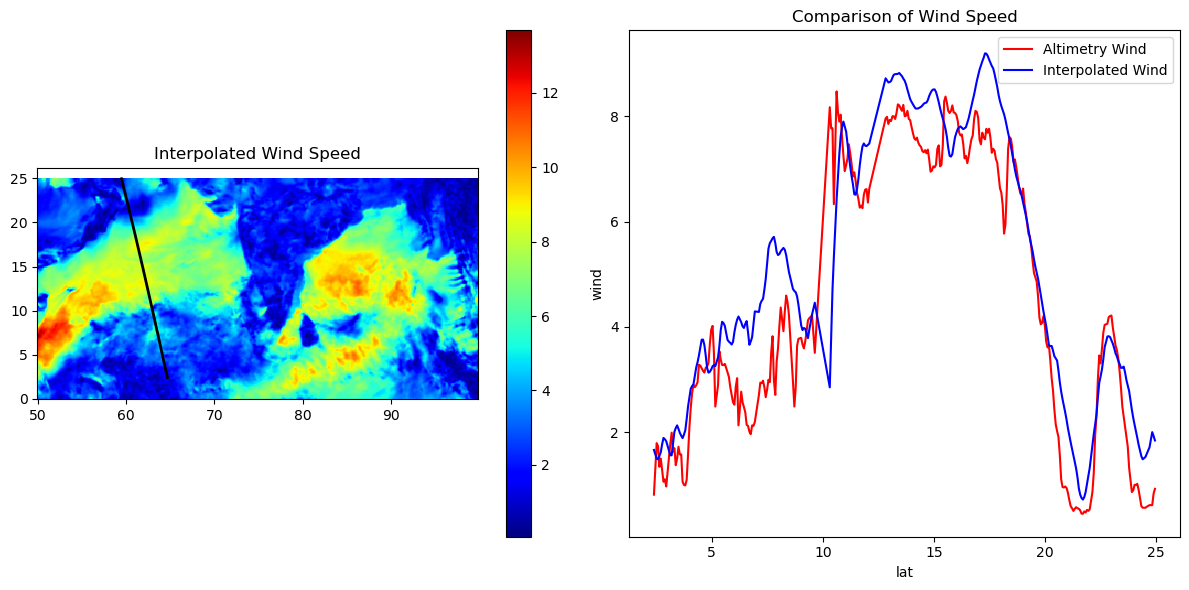
Normal day Plots of wind and swh data obtained by saterlite data files vs numerical methods, also ploting saterlite path

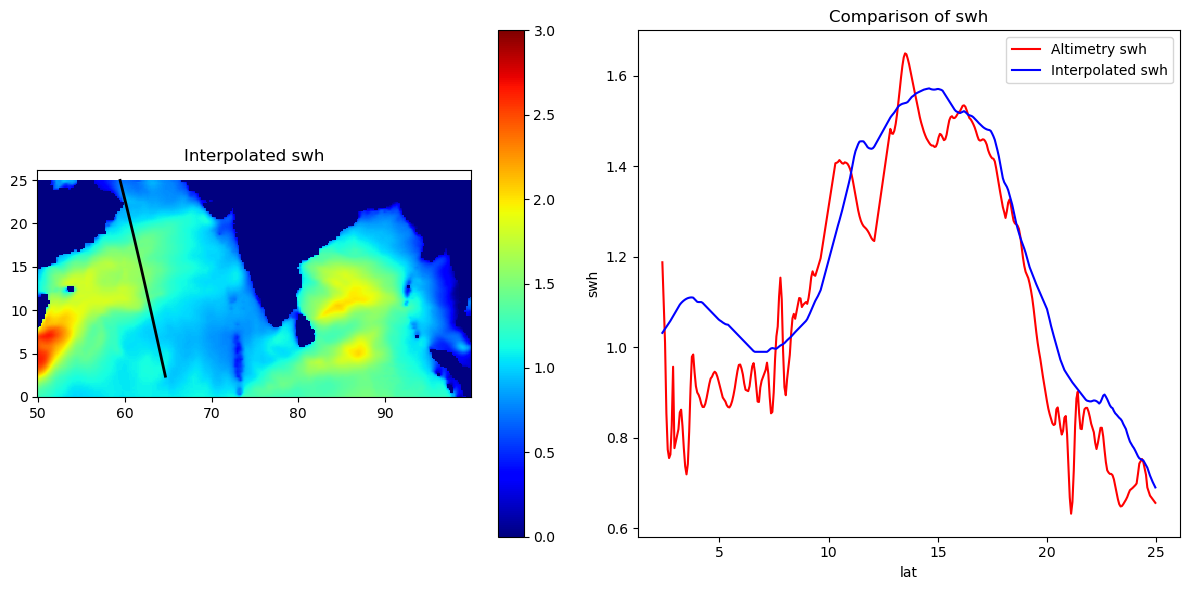
1. 7 july 2023,2am





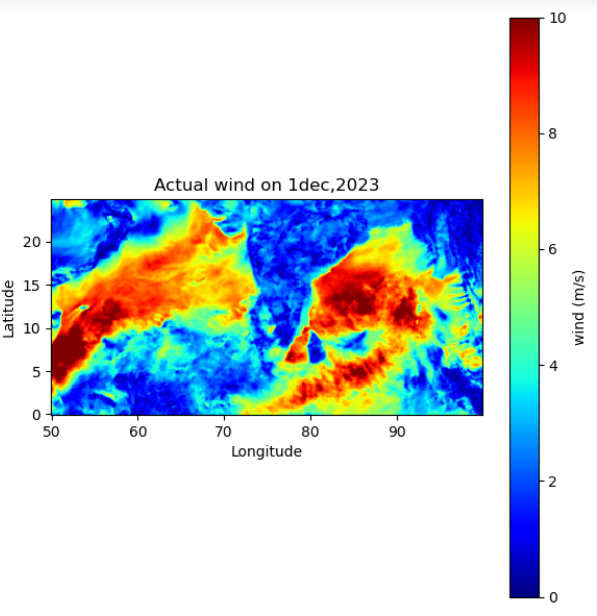
1. 1 December 2023,1am



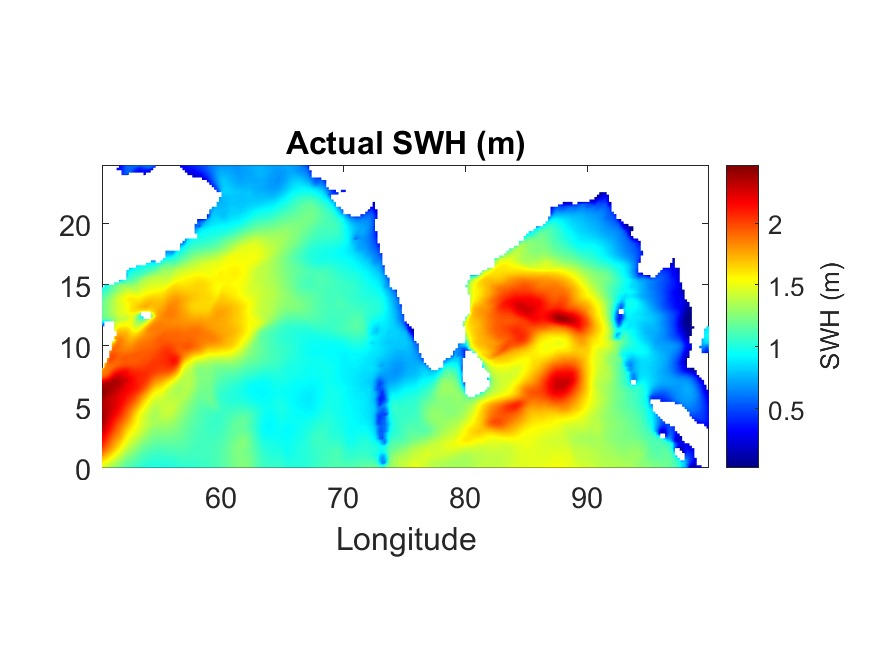


Predicted plot of Swh using wind data

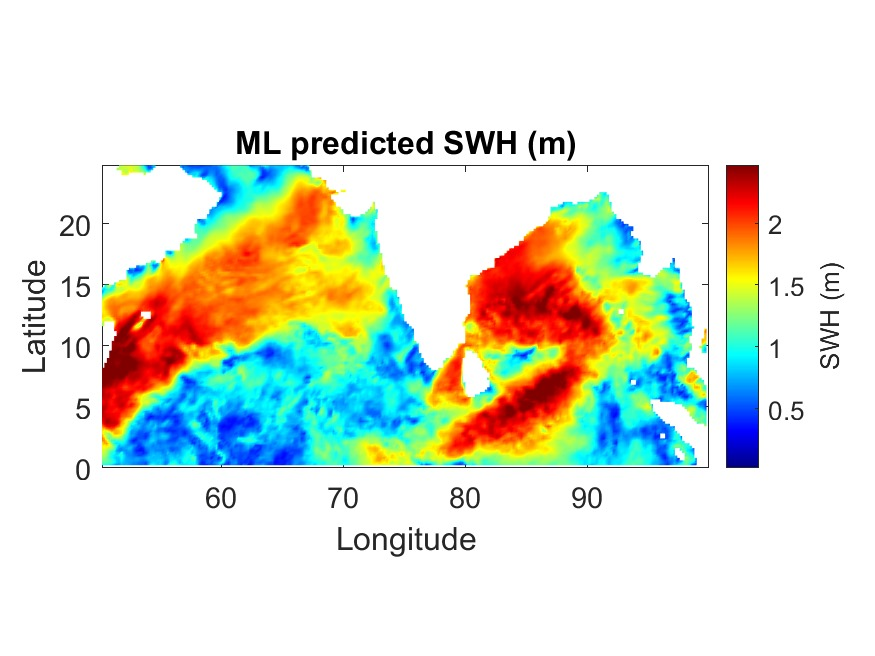
1. Actual Wind plot on 1 dec,2023:



1. Actual Swh plot :



1. Predicted swh using LR model:



RESULTS

I have plotted and found the value of R^2 of wind vs swh before training the data and after training the data for 8 satellites for year 2023. Then my report includes the plots of satellite tracks during the cyclone dates in 2023 , plots also show the intensity of wind and swh on the path of the satellite. Then there are cyclone Plots of wind and swh data obtained by saterlite data files vs numerical methods, also ploting saterlite path and comparing the altimetry data and interpolated data to show how much they overlap , I have done this for 2 cyclone days and 2 normal days..At last I have plotted actual wind and swh then I have plotted ml predicted Swh.

CONCLUSION

For this project I have gone through a lot of research and new learning like learning new algorithms , studied about type of regressions also knowing the mathematics behind regression , and what are errors in machine learning ,along with this I got to know that how to train data and compare data and see plots.

As we can see and compare the plots of actual and predicted swh during non cyclone day , we can clearly see and say that our model is giving good results but its is over predicting and not giving perfect results , for perfect results we have to do more research and we have to opt more complex models of machine learning.

REFRENCES

* For learning about ML I refered : GreekforGreek,Javatpoint,Machiene learning Mastery.
* For downloading data : Copernicus.eu,NOAA Coastwatch,AVISO.

APPENDIX

1. **To plot wind vs swh and find regression between them before vs after training the mode,also to find MSE**

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

import statsmodels.api as sm

from sklearn.metrics import r2\_score

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import math

file = pd.read\_excel(r'C:\Users\HP\AppData\Local\Temp\5ec31193-ba89-4bb5-9c09-30a00f4381b6\_WAVE\_GLO\_PHY\_SWH\_L3\_NRT\_014\_001.7z.1b6\cmems\_obs-wave\_glo\_phy-swh\_nrt\_al-l3\_PT1S\_202211\cmems\_obs-wave\_glo\_phy-swh\_nrt\_al-l3\_PT1S\_2023.xlsx')

wind = file['wind'].values.reshape(-1, 1)

swh = file['swh']

model = LinearRegression()

# Fit the model

model.fit(wind, swh)

# Make predictions

swh\_pred = model.predict(wind)

#r\_squared = math.sqrt(r2\_score(swh, swh\_pred))

r\_squared = "{:.2f}".format(math.sqrt(r2\_score(swh, swh\_pred)))

#print('R-squared:', r\_squared)

plt.title( r\_squared)

plt.plot([0, 20], [0, 10], color='k')

plt.scatter(wind,swh)

plt.xlabel("wind")

plt.ylabel("swh")

plt.show()

y = file[['swh']]

X = file[['wind']].values.reshape(-1, 1)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

print(f"Mean Squared Error: {mse}")

y\_train\_pred = model.predict(X\_train)

r2\_train ="{:.2f}".format(math.sqrt(r2\_score(y\_train, y\_train\_pred)))

#print('Training R-squared:', r2\_train)

plt.plot([0, 20], [0, 10], color='k')

plt.scatter(X\_train, y\_train)

plt.title( r2\_train)

plt.xlabel("trained wind")

plt.ylabel("trained swh")

1. **To plot satellite path showing wind speed on path during cyclones**

import pandas as pd

import matplotlib.pyplot as plt

import cartopy.crs as ccrs

import cartopy.feature as cfeature

import numpy as np

data = pd.read\_excel(r'C:\Users\HP\Downloads\BOB 3, 30july 12 pm.xlsx')

fig = plt.figure(figsize=(10, 8))

# Create a GeoAxes in the PlateCarree projection

ax = plt.axes(projection=ccrs.PlateCarree())

# Set the extent to indaia

ax.set\_extent([50, 100, 0, 25])

# Add coastlines and borders

ax.coastlines()

ax.add\_feature(cfeature.BORDERS)

ax.add\_feature(cfeature.OCEAN, edgecolor='lightblue')

sc = ax.scatter(data['lon'], data['lat'], c=data['wind'], cmap='autumn', s=50)

ax.set\_title('wind on 2023-07-30,12pm : ')

cbar = plt.colorbar(sc, ax=ax, orientation='vertical', pad=0.05 , shrink=0.8)

cbar.set\_label('wind speed m/s')

ax.grid()

plt.show()

1. **To plot satellite path showing Swh on path during cyclones**

import pandas as pd

import matplotlib.pyplot as plt

import cartopy.crs as ccrs

import cartopy.feature as cfeature

import numpy as np

data = pd.read\_excel(r'C:\Users\HP\Downloads\BOB 3, 30july 12 pm.xlsx')

fig = plt.figure(figsize=(10, 8))

# Create a GeoAxes in the PlateCarree projection

ax = plt.axes(projection=ccrs.PlateCarree())

# Set the extent to indaia

ax.set\_extent([50, 100, 0, 25])

# Add coastlines and borders

ax.coastlines()

ax.add\_feature(cfeature.BORDERS)

ax.add\_feature(cfeature.OCEAN, edgecolor='lightblue')

sc = ax.scatter(data['lon'], data['lat'], c=data['swh'], cmap='autumn', s=50)

ax.set\_title('swh on 2023-07-30,12pm : ')

cbar = plt.colorbar(sc, ax=ax, orientation='vertical', pad=0.05 , shrink=0.8)

cbar.set\_label('swh meters')

ax.grid()

plt.show()

1. **To compare Altimetry wind and Interpolated wind and showing wind on indian domain on cyclone days**

import numpy as np

import pandas as pd

import netCDF4 as nc

import matplotlib.pyplot as plt

from scipy.interpolate import RegularGridInterpolator

# Load netCDF dataset

dataset = nc.Dataset(r'C:\Users\HP\Downloads\ForPlots\cmems\_obs-wind\_glo\_phy\_nrt\_l4\_0.125deg\_PT1H\_2023061406\_R20230614T00\_06.nc')

# Extract data from netCDF variables

lon = dataset.variables['lon'][:]

lat = dataset.variables['lat'][:]

u = dataset.variables['eastward\_wind'][:]

v = dataset.variables['northward\_wind'][:]

w = np.sqrt(u\*\*2 + v\*\*2)  # Calculate wind speed magnitude

# Read Excel data

T = pd.read\_excel(r'C:\Users\HP\Downloads\ForPlots\Biporjoy.xlsx')

# Resize T['lat'] to shape (400,)

# Define grid for interpolation

Xi, Yi = np.meshgrid(np.arange(50, 100, 0.125), np.arange(0, 25, 0.125))

# Interpolate onto the regular grid

f = RegularGridInterpolator((lon, lat), w[0,:,:].transpose())

Z = f((Xi, Yi))

# Plotting

plt.figure(figsize=(12, 6))

fi = RegularGridInterpolator((lon, lat), w[0,:,:].transpose())

Zi = fi((T['lon'], T['lat']))

#fi = RegularGridInterpolator((T['lon'], T['lat']),w[0,:,:].transpose());

#zi = fi((Xi,Yi))

# Plot interpolated data

plt.subplot(121)

plt.pcolor(Xi, Yi, Z, shading='interp', cmap='jet')

plt.colorbar()

plt.gca().set\_aspect('equal', adjustable='box')

plt.plot(T['lon'], T['lat'], 'k', linewidth=2)

plt.title('Interpolated Wind Speed')

# Plotting Zi against T\_lat\_resized

plt.subplot(122)

plt.plot(T['lat'], T['wind'], 'r', label='Altimetry Wind')

plt.plot(T['lat'], Zi, 'b', label='Interpolated Wind')  # Assuming Zi is interpolated along Xi (first dimension)

plt.legend()

plt.title('Comparison of Wind Speed')

plt.xlabel("lat")

plt.ylabel("wind")

plt.tight\_layout()

plt.show()

1. **To compare Altimetry swh and Interpolated swh and showing wind on indian domain on cyclone days**
2. import numpy as np
3. import pandas as pd
4. import netCDF4 as nc
5. import matplotlib.pyplot as plt
6. from scipy.interpolate import RegularGridInterpolator
7. # Load netCDF dataset
8. dataset = nc.Dataset(r"C:\Users\HP\Downloads\ForPlots\WAVERYS\_20230614\_R20230614.nc")
9. # Extract data from netCDF variables
10. lon = dataset.variables['longitude'][:]
11. lat = dataset.variables['latitude'][:]
12. swh = dataset.variables['VHM0'][:]
13. # Read Excel data
14. T = pd.read\_excel(r'C:\Users\HP\Downloads\ForPlots\Biporjoy.xlsx')
15. # Resize T['lat'] to shape (400,)
16. # Define grid for interpolation
17. Xi, Yi = np.meshgrid(np.arange(50, 100, 0.125), np.arange(0, 25, 0.125))
18. # Interpolate onto the regular grid
19. f = RegularGridInterpolator((lon, lat), swh[0,:,:].transpose())
20. Z = f((Xi, Yi))
21. # Plotting
22. plt.figure(figsize=(12, 6))
23. fi = RegularGridInterpolator((lon, lat), swh[0,:,:].transpose())
24. Zi = fi((T['lon'], T['lat']))
25. # Plot interpolated data
26. plt.subplot(121)
27. plt.pcolor(Xi, Yi, Z, shading='interp', cmap='jet')
28. plt.clim(0, 9)
29. plt.colorbar()
30. plt.gca().set\_aspect('equal', adjustable='box')
31. plt.plot(T['lon'], T['lat'], 'k', linewidth=2)
32. plt.title('Interpolated swh')
33. # Plotting Zi against T\_lat\_resized
34. plt.subplot(122)
35. plt.plot(T['lat'], T['swh'], 'r', label='Altimetry swh')
36. plt.plot(T['lat'], Zi, 'b', label='Interpolated swh')  # Assuming Zi is interpolated along Xi (first dimension)
37. plt.legend()
38. plt.title('Comparison of swh')
39. plt.xlabel("lat")
40. plt.ylabel("swh")
41. plt.tight\_layout()
42. plt.show()
43. **To predict swh using wind data, through LR model**

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

import statsmodels.api as sm

from sklearn.metrics import r2\_score

# for all saterlite 2023

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import math

f\_al = pd.read\_excel(r'C:\Users\HP\AppData\Local\Temp\5ec31193-ba89-4bb5-9c09-30a00f4381b6\_WAVE\_GLO\_PHY\_SWH\_L3\_NRT\_014\_001.7z.1b6\cmems\_obs-wave\_glo\_phy-swh\_nrt\_al-l3\_PT1S\_202211\cmems\_obs-wave\_glo\_phy-swh\_nrt\_al-l3\_PT1S\_2023.xlsx')

f\_c2 = pd.read\_excel(r'C:\Users\HP\AppData\Local\Temp\4ba83712-7480-409d-8ecf-bfc1fc4a33d1\_WAVE\_GLO\_PHY\_SWH\_L3\_NRT\_014\_001.7z.3d1\cmems\_obs-wave\_glo\_phy-swh\_nrt\_c2-l3\_PT1S\_202211\cmems\_obs-wave\_glo\_phy-swh\_nrt\_c2-l3\_PT1S\_2023.xlsx')

f\_h2b = pd.read\_excel(r'C:\Users\HP\AppData\Local\Temp\7ebf34e0-5fd8-42c4-a3a7-7811ce1eadf2\_WAVE\_GLO\_PHY\_SWH\_L3\_NRT\_014\_001.7z.df2\cmems\_obs-wave\_glo\_phy-swh\_nrt\_h2b-l3\_PT1S\_202211\cmems\_obs-wave\_glo\_phy-swh\_nrt\_h2b-l3\_PT1S\_2023.xlsx')

f\_h2c = pd.read\_excel(r'C:\Users\HP\AppData\Local\Temp\3d26f131-7fba-4f53-b446-e3d3c478f10f\_WAVE\_GLO\_PHY\_SWH\_L3\_NRT\_014\_001.7z.10f\cmems\_obs-wave\_glo\_phy-swh\_nrt\_h2c-l3\_PT1S\_202211\cmems\_obs-wave\_glo\_phy-swh\_nrt\_h2c-l3\_PT1S\_2023.xlsx')

f\_j3 = pd.read\_excel(r'C:\Users\HP\AppData\Local\Temp\d2d0e52f-9b1c-4613-9203-297b6878cf8c\_WAVE\_GLO\_PHY\_SWH\_L3\_NRT\_014\_001.7z.f8c\cmems\_obs-wave\_glo\_phy-swh\_nrt\_j3-l3\_PT1S\_202211\cmems\_obs-wave\_glo\_phy-swh\_nrt\_j3-l3\_PT1S\_2023.xlsx')

f\_s3a = pd.read\_excel(r'C:\Users\HP\Downloads\Copy of cmems\_obs-wave\_glo\_phy-swh\_nrt\_s3a-l3\_PT1S\_2023.xlsx')

f\_s3b = pd.read\_excel(r'C:\Users\HP\Downloads\Copy of cmems\_obs-wave\_glo\_phy-swh\_nrt\_s3b-l3\_PT1S\_2023.xlsx')

f\_s6a = pd.read\_excel(r'C:\Users\HP\AppData\Local\Temp\58eaf504-a13a-4179-8fee-ca19e90d08bf\_WAVE\_GLO\_PHY\_SWH\_L3\_NRT\_014\_001.7z.8bf\cmems\_obs-wave\_glo\_phy-swh\_nrt\_s6a-l3\_PT1S\_202211\cmems\_obs-wave\_glo\_phy-swh\_nrt\_s6a-l3\_PT1S\_2023.xlsx')

ft = pd.concat([f\_al,f\_c2,f\_h2b,f\_h2c,f\_j3,f\_s3a,f\_s3b,f\_s6a])

X = ft['wind'].values.reshape(-1, 1)

y = ft['swh']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

y\_train\_pred = model.predict(X\_train)

# Calculate R-squared value

r2\_train = "{:.2f}".format(math.sqrt(r2\_score(y\_train, y\_train\_pred)))

print(r2\_train)

dataset= nc.Dataset(r"C:\Users\HP\Downloads\WAVERYS\_20230701\_R20230701.nc")

lon1 = dataset.variables['longitude'][:]

lat1 = dataset.variables['latitude'][:]

swh = dataset.variables['VHM0'][:].reshape(1,-1)

swh\_p = model.predict(X\_train)

Xi, Yi = np.meshgrid(np.arange(50, 100, 0.125), np.arange(0, 25, 0.125))

# Interpolate onto the regular grid

f = RegularGridInterpolator((lon1, lat1), swh\_p[0,:,:].transpose())

Z = f((Xi, Yi))

# Plotting

plt.figure(figsize=(12, 6))

plt.subplot(121)

plt.pcolor(Xi, Yi, Z, shading='interp', cmap='jet')

plt.clim(0, 2)

plt.colorbar()

plt.gca().set\_aspect('equal', adjustable='box')

plt.title('Predicted swh')