SUN WATCHER

SAVE THE EARTH FROM ANOTHER CARRINGTON EVENT!

Demo for the 2022 edition of the Space Apps Challenge

SOLAR*CODERS

Romain Garrigou
Armand Garrigou
Maha Belrhazi
Jeanne Bourgeois
Olivier Laurent

CHALLENGE

SAVING THE EARTH FROM ANOTHER CARRINGTON EVENT

Mission : process WIND and DSCOVER satelites datas, find correspondances and to create an AI tha would predict WIND datas and to make them easily readable and accessible to everyone !

SOLAR*CODERS

PROCESSING DATA

TRANSFORMING RAW MEASUREMENTS INTO USABLE DATA

1. Collect data from the NASA website

2. Preprocess magnetic data : track down error values and adjust measurement frequencies through interpolation

3. Use Dynamic Time Warping on magnetic series

```python
import matplotlib.pyplot as plt

def preprocess():

    mag_path = "C:\\Users\\garri\\Desktop\\programming\\spaceapp-challenge\\mag\\mag\\"
    mag_data_files = os.listdir(mag_path)
    mag_data = []
    mag_time = []
    for file_path in mag_data_files:
        file_path = mag_path + file_path
        with open(file_path, 'rb') as f:
            mag  = xarray.Dataset(pickle.load(f))
            time = mag["B1GSE"]["Epoch1"].data
            BGSE = mag["B1GSE"].data
            mag_data.append(BGSE)
            mag_time.append(time)
    mag_data = np.stack(mag_data).reshape(-1, 3)
    mag_data = np.where(mag_data > -1e5, mag_data, np.nan)
    mag_time = np.stack(mag_time).flatten()

    mfi_path = "C:\\Users\\garri\\Desktop\\programming\\spaceapp-challenge\\mfi\\"
    mfi_data_files = os.listdir(mfi_path)
    mfi_data = []
    mfi_time = []
    for file_path in mfi_data_files:
        file_path = mfi_path + file_path
        with open(file_path, 'rb') as f:
            mfi  = xarray.Dataset(pickle.load(f))
            time = xarray.core.utils.Frozen(mfi["BGSE"].indexes.variables).mapping.mapping["Epoch"].data
            BGSE = mfi["BGSE"].data
            mfi_data.append(BGSE)
            mfi_time.append(time)
    mfi_data = np.concatenate(mfi_data, axis=0).reshape(-1, 3)
    mfi_data = np.where(mfi_data > -1e5, mfi_data, np.nan)
    mfi_time = np.concatenate(mfi_time, axis=0).flatten()
    mag_datax = mag_data[:, 0]
```

SOLAR*CODERS

MACHINE LEARNING

TRAINING AN AI TO PREDICT WIND'S DATA FROM DSCOVER'S

A pytorch neural network

- Inputs from DSCOVR, targets from WIND
- Determine mesures quality and uncertainty using expected quantiles
- If necessary, use deep ensembles

```python
import torch.nn as nn
import torch.nn.functional as F


class STD(nn.Module):
    def __init__(self, n_features: int, hidden_units: int, distributional: bool):
        super(STD, self).__init__()
        self.n_features = n_features
        self.hidden_units = hidden_units

        self.input_layer = nn.Linear(n_features, hidden_units)

        # predict value and variance
        self.hidden_layer = nn.Linear(hidden_units, 2 if distributional else 0)

    def forward(self, x):
        out = F.relu(self.input_layer(x))
        return self.hidden_layer(out)


def MLP_STD(n_features: int, hidden_units: int):
    return STD(n_features=n_features, hidden_units=hidden_units)
```

SOLAR*CODERS

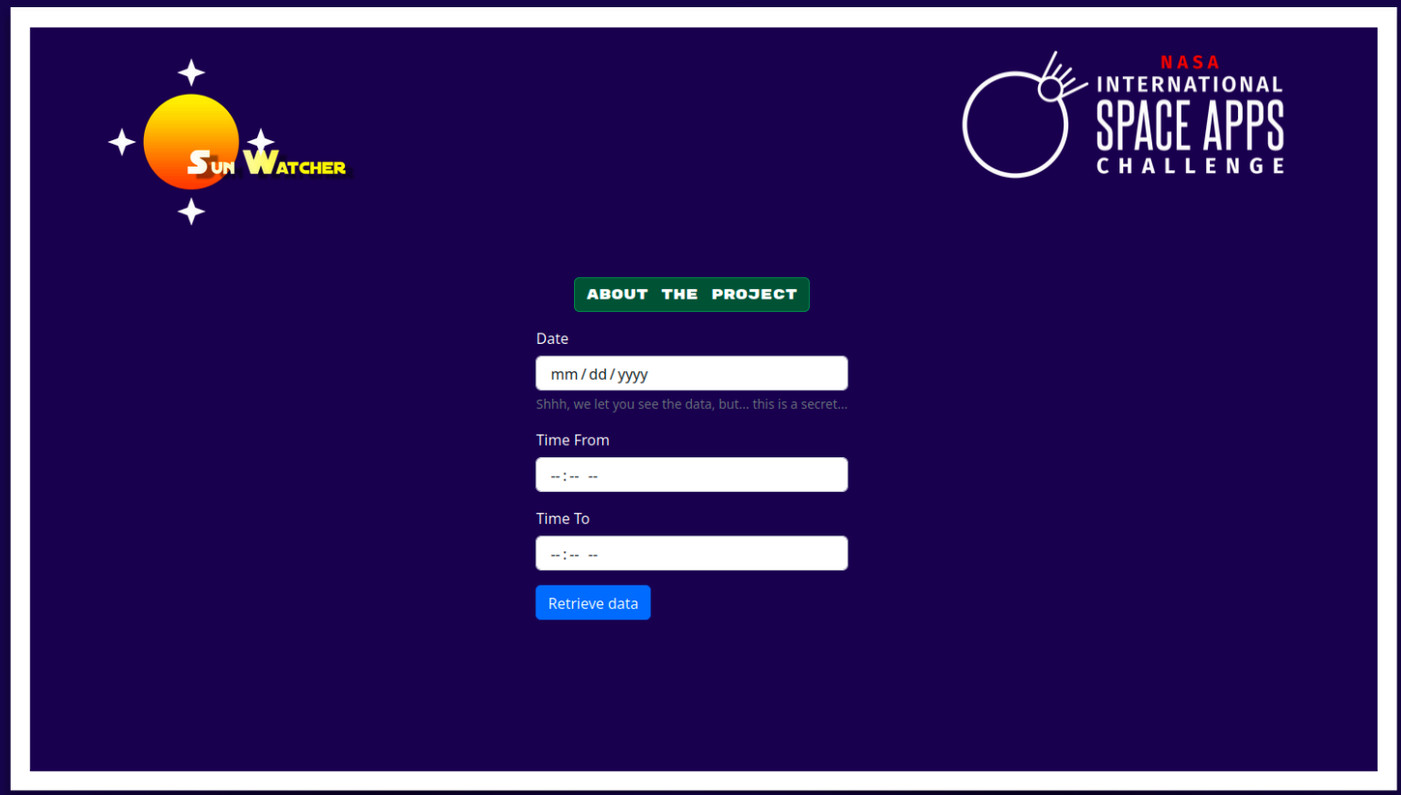
WEBSITE

MAKING DATA AVAILABLE TO THE PUBLIC

Front page

- A description of our tool

- An interface to get computed data easily



SOLAR*CODERS

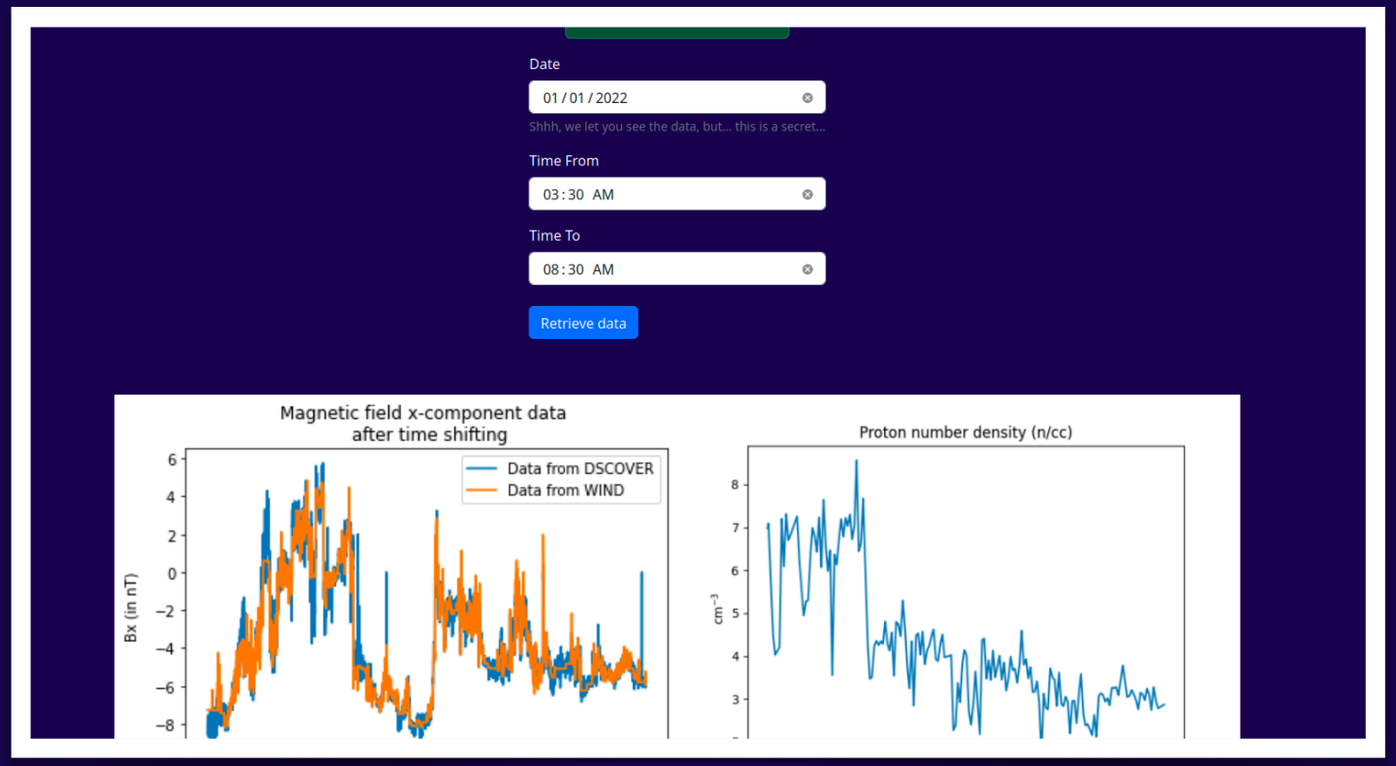
WEBSITE

MAKING DATA AVAILABLE TO THE PUBLIC

A request example:

- See easily what's computed

- An attractive way to show our results to the public



THANK YOU

FOR LETTING US HAVING THAT AMAZING EXPERIENCE AS NASA SCIENTISTS!!

SOLAR*CODERS