Imperial College London
Department of Earth Science and Engineering
MSc in Applied Computational Science and Engineering

Independent Research Project
Project Plan

ARGOWorks:
Developing of a new software platform to analyse and visualize ARGO floats data

by

Vagif R. Aliyev

Va719@imperial.ac.uk
GitHub login: acse-va719

Supervisors:
Dr. Yves Plancherel
Ophelie Meuriot

June 2020

GitHub repository: https://github.com/acse-2019/irp-acse-va719

Introduction

Studying ocean is very essential for Earth as it cover 70% of its surface, absorbs 1/3 of carbon dioxide emissions and plays the major role in protecting us from the effects of greenhouse. Thus, obtaining data and understanding it in this field has become more essential as these factors have become more threating. One of the leading providers of ocean data is ARGO floats with over 3200 floats (growing) reporting with 10,000 profiles per month, providing a range of properties such as water mass, salinity, temperature etc (Argo Research use, n.d.).

The goal of the project is to take advantage of this data revolution to apply modern computing techniques to provide new user-friendly software to democratize access to this new ARGO data stream. The target is to aid and facilitate analysis of these data for researchers in this field.

Problem Description

Argo floats has been first deployed in 1999 to combat the lack of data in the field and efforts are still being put to increase the number of floats (Roemmich, n.d.). Thus, there is an abundance of data available to study the ocean better. The primary goal of the project is to develop a machine learning algorithm that can be trained by the users to detect similar patterns in salinity (initially) across other current and historical data. This will be beneficial to scientist to observe how one part of the ocean can affect other parts across different or same timeline.

Through research I have found the Argo.py python library that eases the Argo data access, manipulation and visualisation (Maze, 2020). For the initial stages of the project, I will be making use of this library to access the data however the plan is to create my own data retriever functions at the end to make my programme less dependent on others work.

As of 2 December 2019, the ARGO Google Earth layer provided by ARGO itself has stopped functioning (Argo and Google Earth, n.d.). There are few alternatives available however it does not provide the same quality and ability to manipulate the

data freely by the user. Thus, one of the goals of the project is to fill the void left behind and build up a new platform using the Google. There are two options available. Google Earth which enable user to travel and explore the world interactively. On the other hand, Earth Engine is a tool specially created for analysing geospatial information. Using this, properties such as water coverage and etc can be analysed as well.

Objectives (& Progress)

First stage is to enable the user to create training/test data from the Argo sensor data. The user will indicate the pattern they are interested in model learning to identify, in our simplest case this is the range where the local minima lies. The model then after gone through training will have to identify the local minima on its own.

The training images will be created randomly using the sensor data and will be converted to images. These images will then be processed suing DataTurks where the ML data can be annotated (Data Turks, 2020). This programme is widely used by large companies and also allows for teams to collaborate easily which will be required in our case to obtain a larger dataset.

After collecting a minimal satisfactory amount of data, variety of training models will have to be tested to determine which one suits this project most. Second way to increase the efficiency of the model would be to play around with the hyperparameters till the most ideal ones are determined.
Once a satisfactory result is achieved from this model, work on adding extra features can begin. Such as allowing variety of users to add training data to the model with descriptions of their name, affiliation and the region they are providing training for.

Second Stage is the visualization of sensor location on google earth engine mapping data. The locations of the sensors have been successfully retrieved and simple graph has been created on Google Maps API for early testing. Next, the created code needs to be transferred using the Earth Engine and its features not available in google maps will be explored. If there is time remaining at the end, a more global

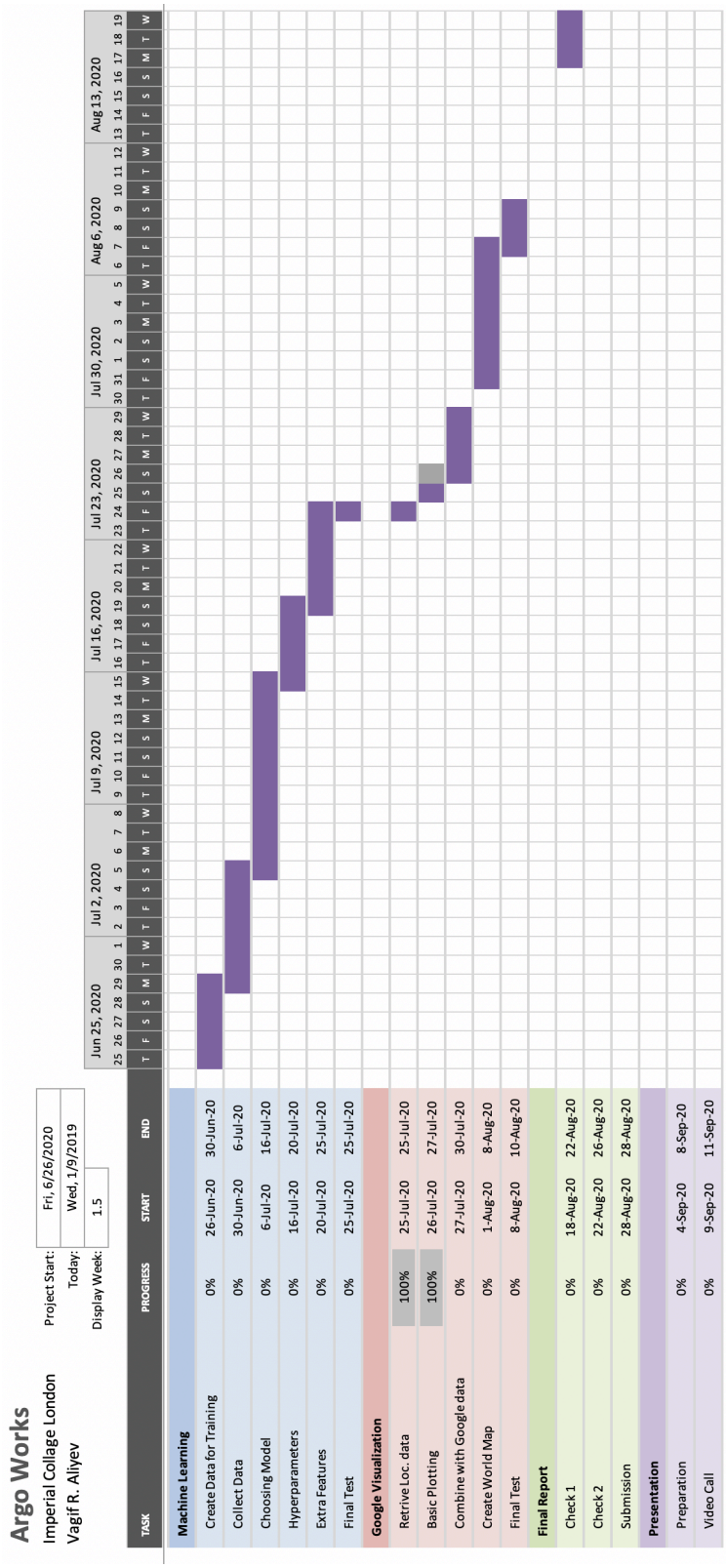
app can be created that indicated the location of all the sensors available in the Argo network on an interactive world map.

Once the project has been developed successfully, the goal would be to create a tutorial and demonstration for others and contribute to their research for example the EURO ARGO (Euro-Argo ERIC, 2020).

GitHub repository: https://github.com/acse-2019/irp-acse-va719

Schedule

Represented in the form of a Gant chart. Covers the basic outline of the schedule to be followed.



The main of the project is the machine learning part and the other section is only an extra feature that can be implemented if extra time remains. This also perhaps can be linked with the research of my other colleagues.

As mentioned in the objectives, creating a proper dataset comes first and afterwards a variety of common models will be implemented. After choosing the best model, the ideal hyperparameters will be searched to give us the best results. Thus, for achieving high score extensive testing will have to be done.

GitHub repository: https://github.com/acse-2019/irp-acse-va719

Possible Risks

Meeting the deadline of the project is obviously important but in the case of project involved with machine learning this carries more risks. First of all, a large sum of training data needs to be created for the programme to learn the patterns with more efficiency. Second problem even with access to large computational power such as Google Labs the models take a long time to run and they can only be run maximum of 2 at a time. However, to find the most appropriate model and hyperparameters thus many tests will have to conducted.

Bias is always a large possible risk, however since the data has gone through quality control (QC) by experts at Brest, France and Monterey, California this issue is not something that can be controlled by me (About Argo, n.d.). Also, the training dataset will be created by experts in the field thus this issue has low risk of being faced. The major problem that could be faced, is either not having enough data or containing homogeneous data (lacking variability in data). This risk has to be accounted for, so the model is not over-optimized.

Another issue is that the patterns shows by the user might be quite hard to detect to begin with since the data is retrieved from real life and contains noise.

GitHub repository: https://github.com/acse-2019/irp-acse-va719

Reference

Argo.ucsd.edu. n.d. *About Argo*. [online] Available at: <http://www.argo.ucsd.edu/About_Argo.html> [Accessed 26 June 2020].

Argo.ucsd.edu. n.d. *Argo And Google Earth*. [online] Available at: <http://www.argo.ucsd.edu/Argo_GE.html> [Accessed 26 June 2020].

Argo.ucsd.edu. n.d. *Argo Research Use*. [online] Available at: <http://www.argo.ucsd.edu/Research_use.html> [Accessed 26 June 2020].

DataTurks. 2020. *Data Turks*. [online] Available at: <https://dataturks.com/> [Accessed 26 June 2020].

Euro-Argo ERIC. 2020. *Euro-Argo ERIC*. [online] Available at: <https://www.euro-argo.eu/> [Accessed 26 June 2020].

Maze, G., 2020. *Argopy*. [online] GitHub. Available at: <https://github.com/euroargodev/argopy> [Accessed 26 June 2020].

Roemmich, D., n.d. *On The Beginnings Of Argo: Ingredients Of An Ocean Observing System*. [online] Available at: <http://www.argo.ucsd.edu/Argo_Beginnings.pdf> [Accessed 26 June 2020].