


# Building a Python-based Analysis Tool for AWAKE Experiment

## Google Summer of Code 2019

Proposal by Aman Singh Thakur

@singh96aman

# PROPOSAL INFORMATION

## 1.1 PROPOSAL SUMMARY

We are generating 2.5 quintillion bytes of data every day and at the current pace, we need efficient querying and analysis tools to address the issues of big data. AWAKE, one of the experiments at CERN, acquired tens of terabyte of data alone during their experimental run in the year 2017-18. AWAKE scientists need to efficiently process and analyse this data to draw conclusions from their experiments. In this proposal, I'll be describing how I plan to deploy a python based library to solve this issue.

In this project, I aim to create an open sourced analysis tool/library that reads datasets and groups from multiple large HDF files and create an efficient database for searching and loading multiple datasets conveniently. This library, in turn, will identify dataset dependent parameters like height, width, intensity, etc., filter data based on thresholds, visualize the dataset into an image, compare multiple images visually and include as many data analysis routines as possible to address the requirements of scientists. Later, I plan to study the existing analysis of datasets done and port it into a similar analysis with the help of this library. After the library is complete, I plan to write test cases, documentation and example notebooks for each module. This library will later be deployed to open sourced package repositories like conda and PyPI.

## 1.2 DELIVERABLES

### 1.2.1 PHASE 1

- Develop a database by reading through all HDF files.
- Develop an efficient querying mechanism and support for caching of popular groups and datasets. Write searching modules to find relevant datasets.

### 1.2.2 PHASE 2

- If the dataset contains an image, prepare a 2D image out of multiple inter-dependent datasets and filter data as per requirements.

- Visualize the dataset into image or graph plots and make it interactive.
- Visualizing multiple images in one window for better inspection.

### 1.2.3 PHASE 3

- Analyse the image by colour filtering, smoothing, denoising, edge detection, segmentation, etc.
- Learn more about AWAKE experiment and the requirements of scientists.
- Finding correlations in data to determine whether an experiment is successful.

### 1.2.4 PHASE 4

- Port existing analysis into the new library.
- Write test cases, documentation and example notebooks for modules on CERN's SWAN service. Deploying the application.

## 1.3 DETAILED DESCRIPTION

### 1.3.1 PHASE 1

A script module reads through approximately 600,000 HDF files iteratively. If it encounters a dataset in one of the files, it writes into the CSV in the following format:

**Dataset Name, Parent Group Name, HDF File Name, Singleton Value, Size, Shape, Data type**

The advantages of having this structure is:

- CSV can easily be opened in an excel sheet for visualization.
- Using the Pandas library, we can perform column level searching in O(n) time. Pandas library allows for SQL-like search queries on database.
- Once Dataset is found, we can look for its parent group name in HDF File in O (1) time and at the same time find other datasets that contain the metadata of given dataset.
- The entire row of popular datasets used could be cached. Cache size can be defined by scientists.
- CSV format saves the least metadata while storing raw data as compared to JSON or XML. (best-response-data)

- Traversing through this file could be a time-consuming process. As a "bonus", I'll add a progress bar to make the script more transparent (Write Module Code).

```
Indexing 11 files
[=========================                    ]  7/11 ( 63%)  4 to go
```

*Figure 1 : Progress Bar in Action for 11 HDF Files given for testing*

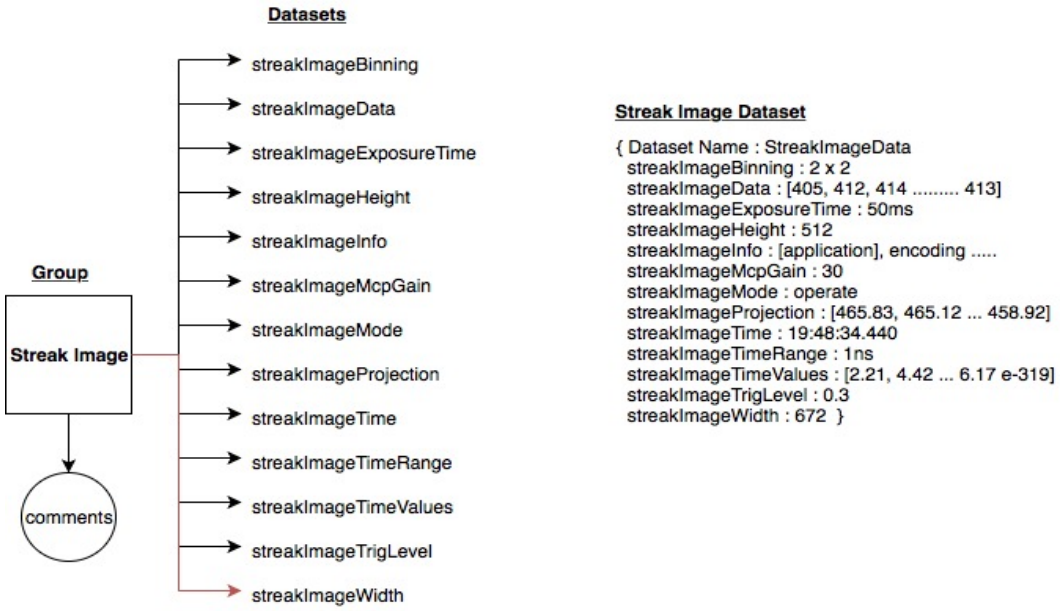


*Figure 2 : An Example of how a dataset will look after processing.*

```
'''
Method to show all data in dataset dict
'''
def show_dataset(dataset):
    print ("Showing the dataset "+dataset['DatasetName'])
    for row in dataset.keys():
        print(row+" || "+str(dataset[row]))

'''
Method to search from parentGroup and create a dataset dict.
Currently, It automatically detects height and width for StreakImageDataset.
Later, It'll be able to detect height, width, comments, thresholds for any dataset if it displays an image.
These automatically detect features will pave way for automized data analysis.
'''
def get_dataset(h5File, parentGroup):
    dataset = {'DatasetName' : h5File.name}
    for file in parentGroup.keys() :
        values = parentGroup[file].name.split("/")
        val = list(parentGroup[file])
        if len(val) == 1 :
            if str(values[len(values)-1]).find("Height") :
                dataset.update({"Height" : val[0]})
            elif str(values[len(values)-1]).find("Width") :
                dataset.update({"Width" : val[0]})
            else :
                dataset.update({values[len(values)-1] : val[0]})
        else :
            dataset.update({parentGroup.name+"/"+str(values[len(values)-1]) : list(val)})
    return dataset
```

*Figure 3 : Crude Implementation of a searching module*

The CSV generated after testing for 11 files is here. Later, I'll write searching modules that will find and create the dataset and its parameter and return the dataset in the form of a key-value hash structure called **dict**. I'll also perform query based searching such as **"/AwakeEventData/TT41.BCTF.412340/ Acquisition/totalCurrentPreferred > 0.06"** and return datasets or subsets of data based on Boolean values as shown in the link provided (Search Module). As a "bonus", I'll be writing searching modules that will search datasets group-wise, data type-wise and size-wise. These will help scientists identify similar datasets.

### 1.3.2 PHASE 2

Modules provided in this phase will use the dataset value as provided by the previous phase and decide whether to plot image or types of graph plots. Using libraries like numpy, scipy and matplotlib, the image will be rearranged, filtered by X, Y coordinates or plotted as a graph. I'll code the modules in such a way that these parameters are automatically identified or manually entered by the scientists. If the dataset is not an image then modules will identify X, Y as categorical or continuous data and plot graphs (such as bar graph, box plot, frequency histogram, scatter graphs, etc.) for a better understanding of data.

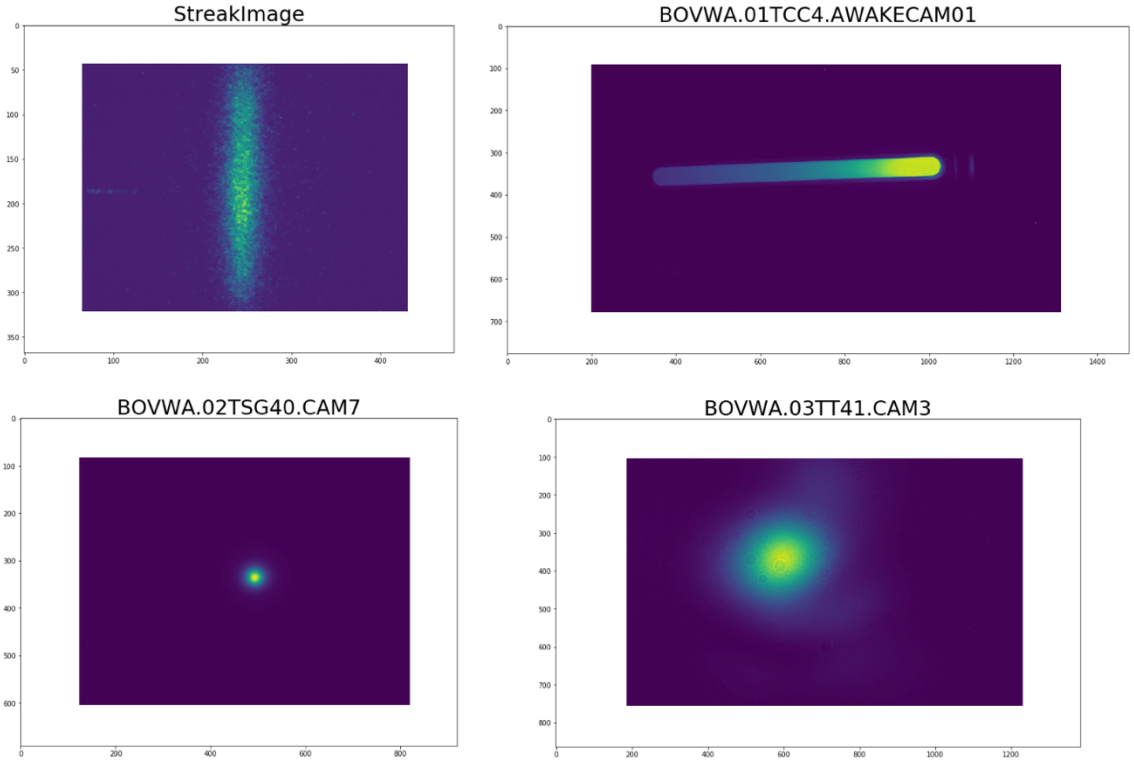


*Figure 4 : Multiple Images displayed for Visual Analysis*

After completing these tasks, I'll make these images and graphs **interactive** with features of screenshot, zoom, coordinate values on hover over the graph, etc. I'll

also provide modules to plot and display multiple graphs/images on one screen for better visual analysis.

### 1.3.3 PHASE 3

For datasets containing images, it's important to analyse images to draw conclusions from experiments. Most of the data analysis routines are already provided in scipy package and data is pre-processed as per requirements in previous phases. However, there is a need to run some generic data analysis routines like applying colour filters, applying rotations and transformations, segmentation, fitting shapes to images or comparing to Gaussian curves, calculating mean, variance and RMSE, and many others.

For finding correlations in data, multiple correlation techniques can be used such as the Pearson Correlation Coefficient, Spearman's Correlation and Kendall's Tau technique to name a few.

$$\rho_{X,\,Y} = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}}$$

*Figure 5 : Basic formula for calculating correlation*

Using formula in fig 5, we can figure out whether there is negative, no or positive dependence on associated variables depending value of correlation [-1,1].
For example, when testing correlation on efficiency dataset from University of California, Irvine, I discovered that mpg has negative correlation with weight and horsepower (non-linear in nature as is evident in fig 6) but has no relationship with acceleration which is consistent. (About Dataset and Pair-wise Correlations)

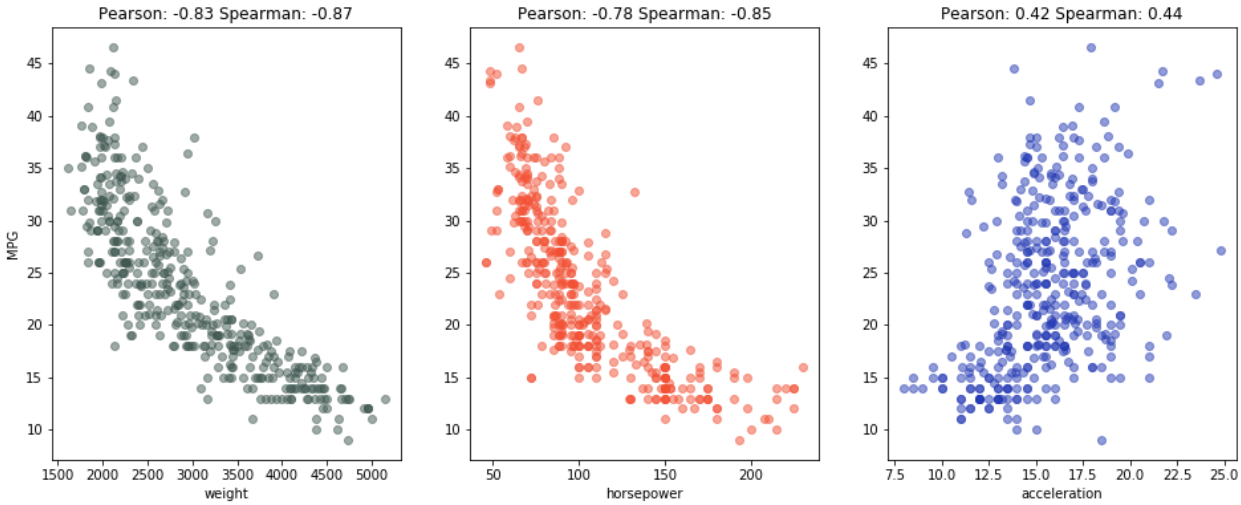


*Figure 6 : Testing mgh vs weight, horsepower and acceleration in Fuel efficiency dataset*

With in-depth discussion with the team at CERN and enhanced understanding of AWAKE experiment, I'll try to apply similar concepts to experiments to yield whether they were successful or not. Below are some generic data analysis routines (Data Analysis Module) :

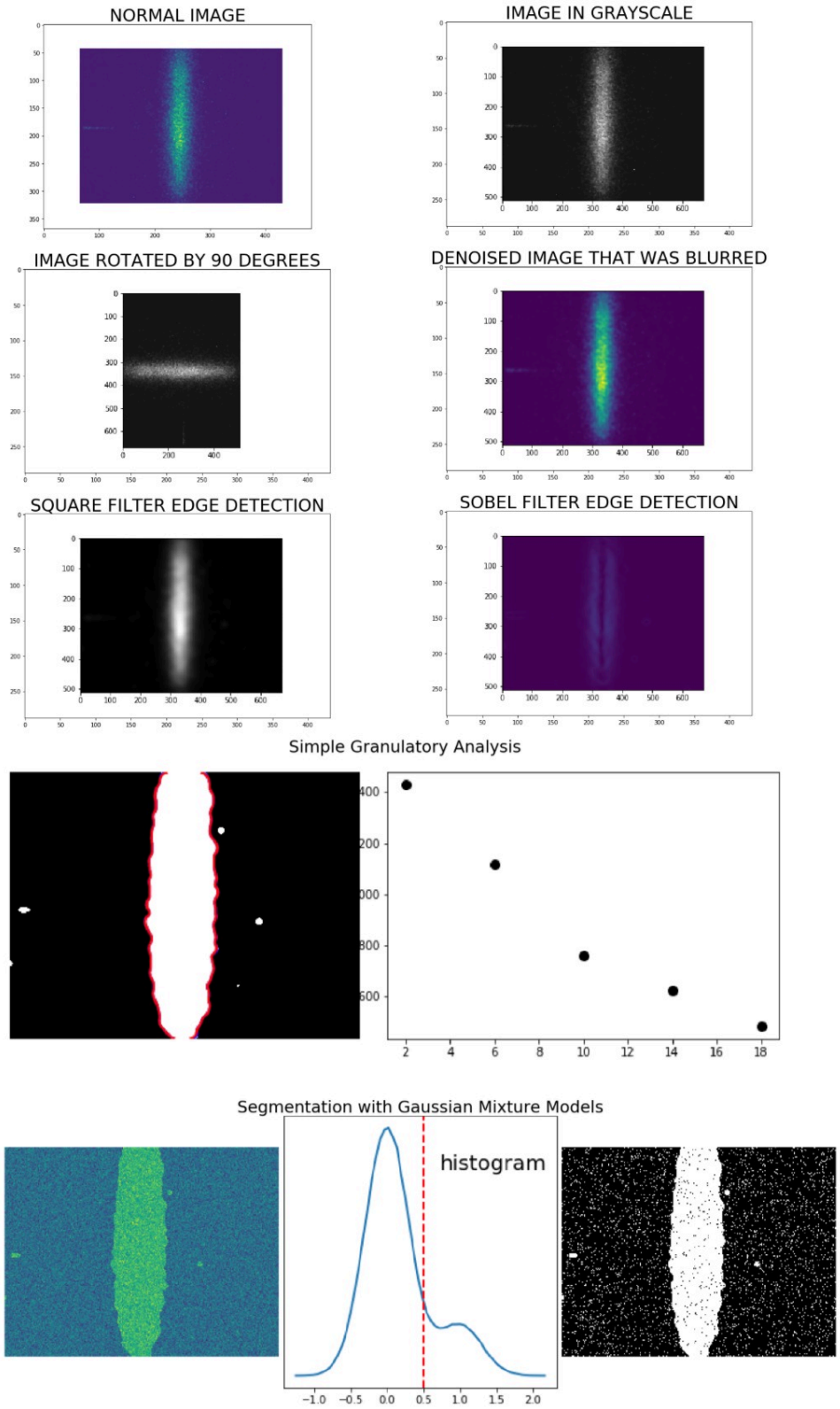


*Figure 7 : Few Data Analysis Routines Running for Streak Image Dataset*

```
256 def optimize_curve():
257     img = mpimg.imread('PNGFiles/Dataset1.png')
258     hist,bins,_ = plt.hist(img.ravel(), bins=256, range=(0.0, 1.0), fc='k', ec='k')
259     bin_centers = np.mean([bins[:-1],bins[1:]], axis=0)
260     p_opt,_ = curve_fit(gaussian, bin_centers, hist, maxfev=1000000)
261     mu, sigma, amp = p_opt
262     fit = gaussian(bin_centers, mu, sigma, amp)
263 #   plt.plot(bin_centers, hist)
264 #   plt.plot(bin_centers, fit)
265 #   plt.imshow(img)
266     '''For StreakImage Dataset'''
267     '''Optional values for the parameters so that the sum of squared residuals of function is minimized.'''
268     print (p_opt)
269     '''Estimated Covariance Matrix. The diagonals provide the variance of the parameter estimate."'''
270     print (_)
271 
272 
273 optimize_curve()
```

```
[9.98045573e-01 8.54891981e-04 4.05709073e+05]
[[5.05096368e+11 1.10482141e+11 4.52160337e+17]
 [1.10482141e+11 2.41662862e+10 9.89031897e+16]
 [4.52160337e+17 9.89031897e+16 4.04772204e+23]]
```

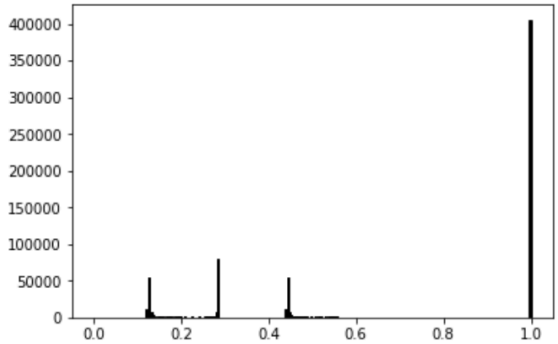


*Figure 8 : Using Scipy.optimize.curve_fit to fit a non-linear function f and get estimated values*

To re-iterate, I'll be providing with few modules that'll automatically run these routines on the image and give results in one go. These modules will help scientists understand the basic structure, features and behaviour of the dataset with ease and later move to advance routines such as hypothesis testing, if needed.

### 1.3.4 PHASE 4

In this phase, I'll inspect the previous analysis done on AWAKE experiment and try to get similar results using the modules created in previous phases. Later, I'll be working on modifying previously built modules to bring them to industry level standards by adding documentation and test cases to each one of them and I'll write example notebooks that demonstrate the basic features of the library on SWAN. After mentors have reviewed my code, I'll be moving to deploy the application on conda and PyPI package manager.

## 1.4 SCHEDULE

I'll be starting my work from April 10th after submission deadline with tasks given in "proposed timeline". The coding phase would start from May 13th after all preliminary tasks are done. Till May 31st, I'll be able to devote 2 hours daily on weekdays and 6-7 hours on weekends as I'm currently interning as a software developer at Accolite Software India Pvt. Ltd (Bangalore). From June 1st to July 1st, I'll be working daily about 5-7 hours due to holidays. In this time, my first GSoC evaluation and my 8th semester examination will have concluded. I plan to be finished with two and a half phases of the library by this time. From July 1st to July 26th, I'll be joining the same company again and I will be following the above-mentioned routine again. Since this project requires a lot of feedback and testing, I have provided with a buffer week for 1 month. In conclusion, I'll be devoting around 100 days or roughly 350+ hours to finish this project.

## 1.5 PROPOSED TIMELINE

### I. PRE-SUMMER: APRIL 10TH – MAY 5TH

- Will explore more data analysis routines.
- Explore libraries like NumPy, SciPy, Matplotlib and other such data analysis and visualization libraries.
- Prepare elaborated diagrams to explain the working of modules for all the four phases.
- Prepare crude implementations of modules of all four phases, if possible.

### II. COMMUNITY BONDING: MAY 6TH – MAY 12TH

- Get in touch with mentors Spencer and Marlene.
- Share my doubts occurred during pre-summer phase with mentors to get a clear picture on implementation and expectations.
- Learn about coding and design standards expected from the project.
- Discuss phase 1 expectations and diagrams
- Learn about SWAN service.
- Learn more about AWAKE experiment and get requirements for data analysis routines.

## III. MILESTONE 1: MAY 13TH – MAY 19TH

- Discuss AWAKE doubts with mentors to have clearer picture for analysis tools.
- Start coding the base module to index HDF files and create a database.
- Upgrade this module to handle all types of corner cases, error handling and return dataset metadata if required.
- Begin writing a basic searching module to find dataset and intelligently detect its parameter using group names, group comments and dataset names.

## IV. MILESTONE 2: MAY 20TH – MAY 26TH

- Finish writing basic searching module.
- Create a file for caching as defined by the user and add support to the searching module.
- Add support for searching datasets group-wise, data type-wise and size-wise.
- Test phase 1 modules on a large number of datasets and report back with results.

## V. MILESTONE 3: MAY 27TH – JUNE 9ND

- Rectify bugs in phase 1 code base, if any.
- Write a module to plot a 2D image out of dataset raw data using its metadata. Add feature to filter image if X and/or Y thresholds are given.
- Write a module to plot as many as graph plots out of dataset as possible in one window.

## VI. MILESTONE 4: JUNE 10TH – JUNE 23RD

- Add features to make the plots more interactive.
- Code a module that can plot multiple datasets in one window for visual analysis.
- Testing phase 2 modules on different datasets and rectifying bugs.
- Finalize the general set of analysis routines required by the scientists.
- Write a module that can run these routines and visualize the output and save that file, if necessary.

## VII. EVALUATION 1: JUNE 24TH – JUNE 28TH

- Get feedback from mentors about completing phase 1 & 2 and starting phase 3.
- Rectify logical fallacies and bugs.
- Refactor and Refine the code base.

## VIII. MILESTONE 5: JUNE 29TH – JULY 7TH

- Test few routines for correlations and analysis and get feedback.
- Finish phase 3 module and test it against the diversity of dataset.
- Rectify the bugs, if any.
- Inspect previous analysis done on AWAKE experiments with the help of mentors.
- Begin porting existing analysis to new analysis done using previous phases modules.

## IX. MILESTONE 6: JULY 8TH – JULY 21ST

- Write example notebooks on SWAN to demonstrate each module and its ease of use.
- Add test suite to each module and perform unit and integration testing. Rectify bugs, if any.
- Perform full-scope refactoring, restricting and refining to meet industry standards.

## X. EVALUATION 2: JULY 22ND – JULY 26TH

- Get feedback from mentors from the completion of the project completely.
- Refactor code and make modifications as per requirements.
- Submit the evaluation of the mentor.

## XI. BUFFER PERIOD: JULY 27TH – AUG 19TH

- Buffer time to finish any previous pending tasks.
- Add documentation to each module and deploying the application
- Test the package and report bugs, if any.
- Update the deployed version of the application with fixes and updates.
- Prepare a document of all my tasks and experience at CERN.

# ABOUT ME

## 2.1 BASIC INFORMATION

### I. NAME AND CONTACT INFORMATION

- **Name:** Aman Singh Thakur
- **Location:** Bangalore, India
- **Time zone:** UTC +4:30 (Indian Standard Time)
- **Email:** singh96aman@gmail.com
- **Github:** github.com/singh96aman
- **Mobile:** +91-7760709950
- **Skype:** singh96aman
- **Facebook:** facebook.com/singh96aman
- **University:** Manipal University, Manipal
- **Field of Study:** Bachelor's of Technology in Computer Science and Technology with specialization in Intelligent Systems. (Batch of 2019)
- **Year of Study:** Final Year

### II. MEETING WITH MENTORS

- I'll be shifting my working hours to evenings (6-8PM IST) or night (9-11PM IST). I would have to go to the office for internship in Accolite Software at regular working hours. I'll be available for communication in afternoons or evenings (Central European Time).
- For the days I am not busy with the other internship, I'll be available for the entire day for work purposes.
- I'm reachable anytime through E-Mail, Slack, Gitter, WhatsApp, Skype.

### III. DEVELOPMENT ENVIRONMENT

- MacBook (mid-2015 model) with dual boot: macOS Sierra and Windows 10
- 2.2 GHz base speed (i7) with 16GB DDR3 RAM and Intel Iris Pro 1536MP in-built Graphics card.
- Python version: Python 3.6.4 :: Anaconda, Inc.
- IDEs: Anaconda Navigator, Jupyter Notebooks, VS Code, Sublime Text

## 2.2 TECHNICAL PROWESS

I started my journey into computer science from 6th grade when I was introduced to C++ programming language. Since then, I have learned languages such as C, C++, C#, Java, Python, JavaScript and I have worked on Android, .NET, Flask, Django, Angular 6, React, React Native, Web Development and Full stack development. I have always been fascinated by python programming language and I put my skills to create a python-based flask web application which was based on the premise on live stock trading simulation for my college fest in 2017. Later, I also created an SVM model using NumPy and SciPy to predict the direction of movement of stock and the probability of its movement based on historical data. In 2018, I was selected as an intern in the Indian Institute of Technology, Kharagpur, India and I was working on the diagnosis of Acute Lymphoblastic Leukemia using decision trees coded in python programming language. Also, I have been heading Linux Users Group, Manipal and orchestrated install fests, GSoC workshops as well as Git workshops. In 2016, I went onto do an internship in data analysis in IIM Lucknow on Data Analysis and Visualization in R.

## 2.3 WHY AWAKE?

One of the main reasons this project appealed to me was because of AWAKE. Currently, Advanced Plasma Wakefield Acceleration Experiment (AWAKE) is accelerator R&D based at CERN working on proof of concept experiments but it has the potential to replace and reduce the form factor of LHC in the near future. The Large hadron collider causes beams to collide at high speeds with each there and then record the resulting events caused by the collision. These conditions are similar to the big bang and study of these events could help us how the universe is made and what is it composed of. To collide particles at high speed, we need to charge particles to high energies. To charge particles to high energies, current particle accelerators are using alternating current to develop high voltage difference and accelerate the particles in turn but for charging particles to higher energies, we need larger colliders than LHC which is not feasible.

Plasma Wakefield draws an analogy from a boat passing by a surfer and the surfer using that wave to accelerate. Using an intense proton beam (boat), we create waves and inject particles using an electron beam to accelerate them to even high energies (about 1000 times stronger) than previously possible. This can reduce the size and cost of the experiment. I believe this is the future of

particle accelerators and by working on this project I'll be exposed to technologies, scientists and data that will carve the future of this field.

I'll start learning about AWAKE experiment in depth immediately after the community bonding starts and will be meticulous to how visual analysis is done. Using this platform and opportunity, I can write scripts to automize some generic processes for scientists and thus contribute to the community.

## 2.4 WHY SHOULD YOU CHOOSE ME?

The internships and work I have done in my prior college years have laid foundation for concepts that I feel I can apply in this project. I have been promoting open source community and git since I have been part of Linux users group, Manipal. I am part of FOSSASIA community and I have submitted PR's to this and other organizations. I have extreme passion and discipline towards this field and my previous mentors have spoken fondly about me. I feel my candidature is well suited for this project because of my expertise in python, git as well as my zest to learn more about data analysis routines in python. Gathering experience in data analysis and learning more about the open source community could really open big doors in the future. The days I spend interning at CERN would really give me a lot of exposure and would allow me to interact with some of the best minds in this world, big data and cutting edge technologies.

## 2.5 OPEN SOURCE CONTRIBUTION AND REFERENCES

### I. PULL REQUESTS

- Pre-GSoC PR: #10 Executing all tasks for evaluation task.
- Non-GSoC PR: #2149 Fix Height of form input fields in EvalAI frontend.

### II. RELEVANT WORK

- Deployed a python library as a personal exercise to this project The-ML-lib
- Deployed Flask based web application for live stock trading MarketWatch
- Deployed Flask based web application that runs SVM on historical stock data MarketWatchBeta

- Current repository for all my work for AWAKE proposal Github Repository.
- Part of **"GitHub Pro"** Program

## III. REFERENCES


- Plasma Wakefield Experiment
- HDF5 for Python
- Cern HSF Proposal
- Forbes article
- Image Processing
- MongoDB uses JSON
- Best Response Data
- Introduction to Correlations