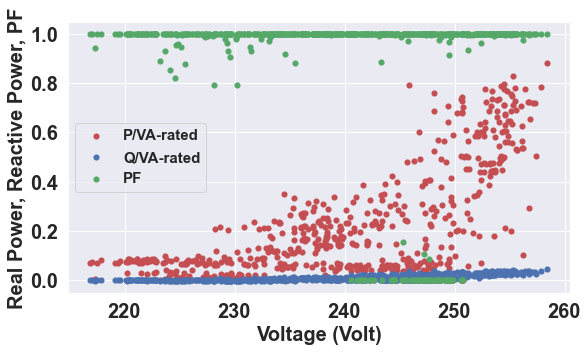
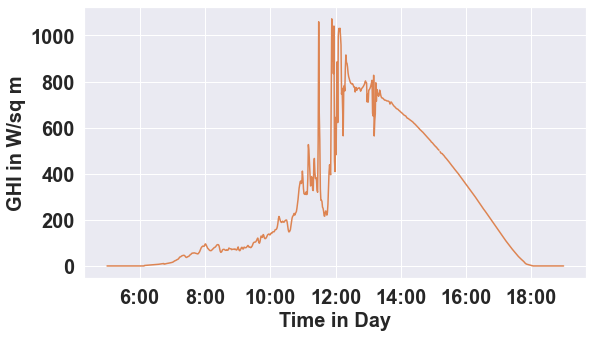
Solar Curtailment Project Progress Journal

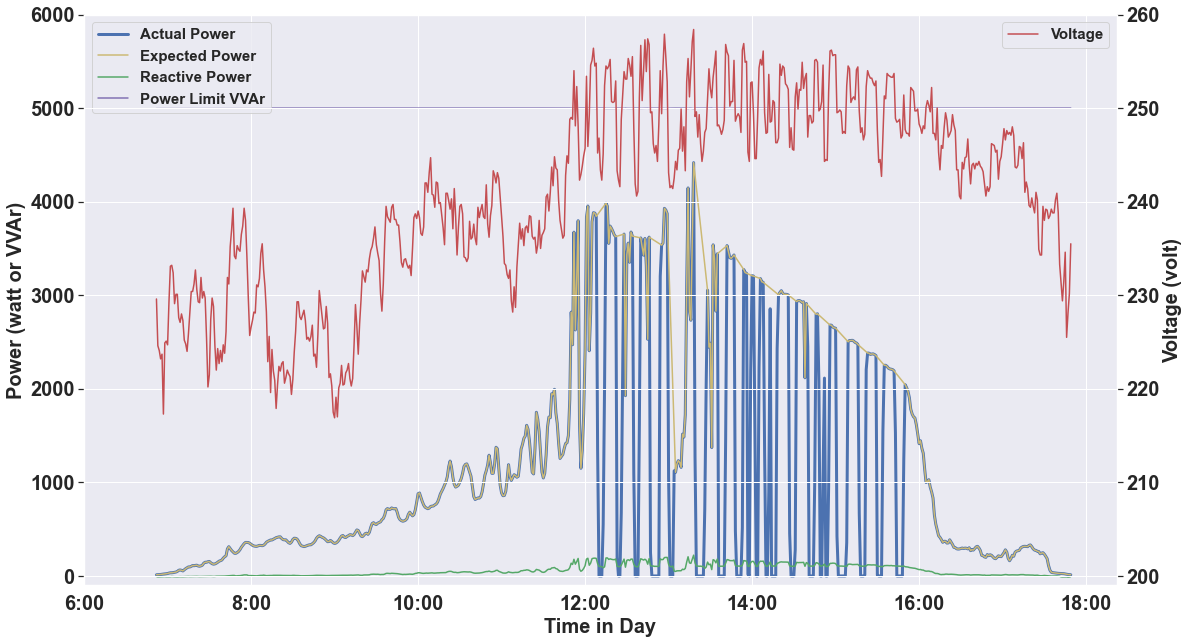
# 13/09/2022

# Result on Tripping – non clear sky day (sample 1)

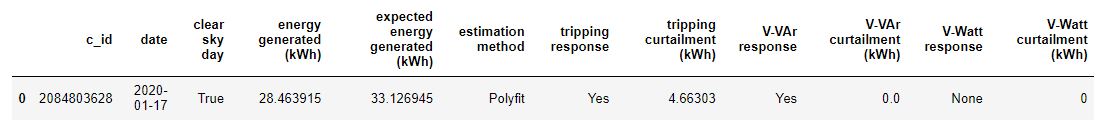
Graphical user interface, application, website

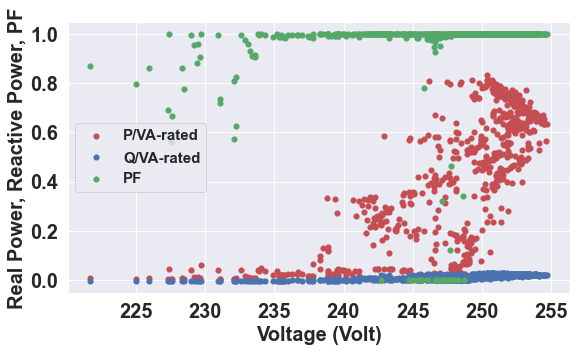
Description automatically generated

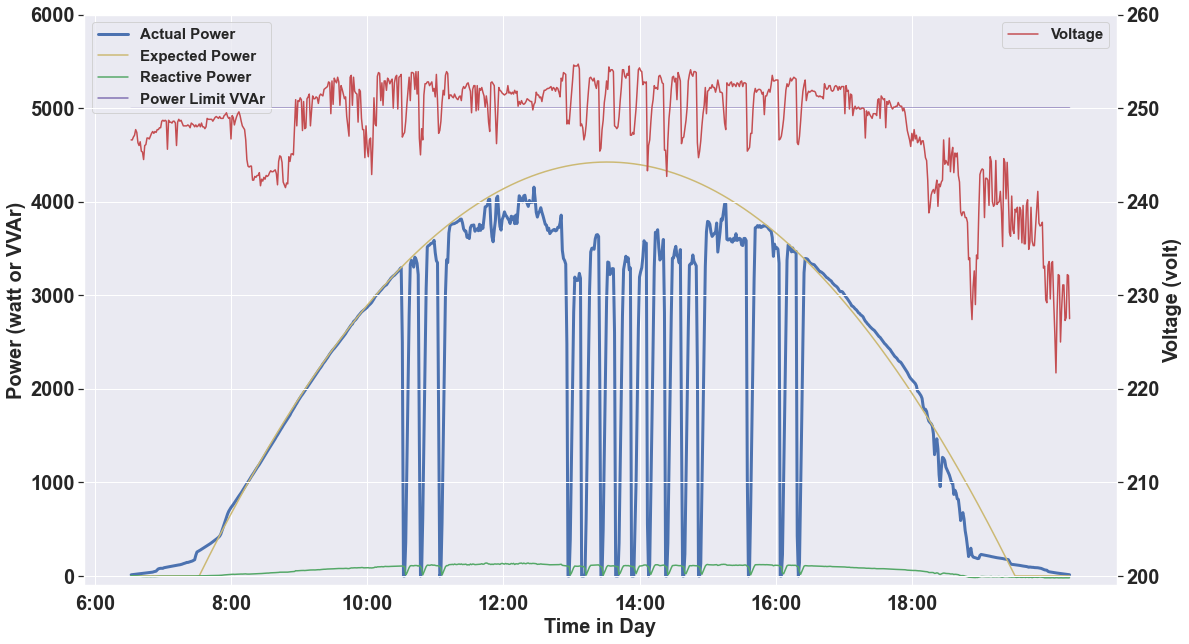




# Result on Tripping – clear sky day (sample 11)



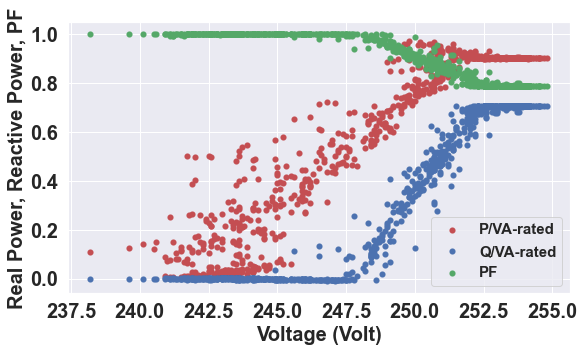


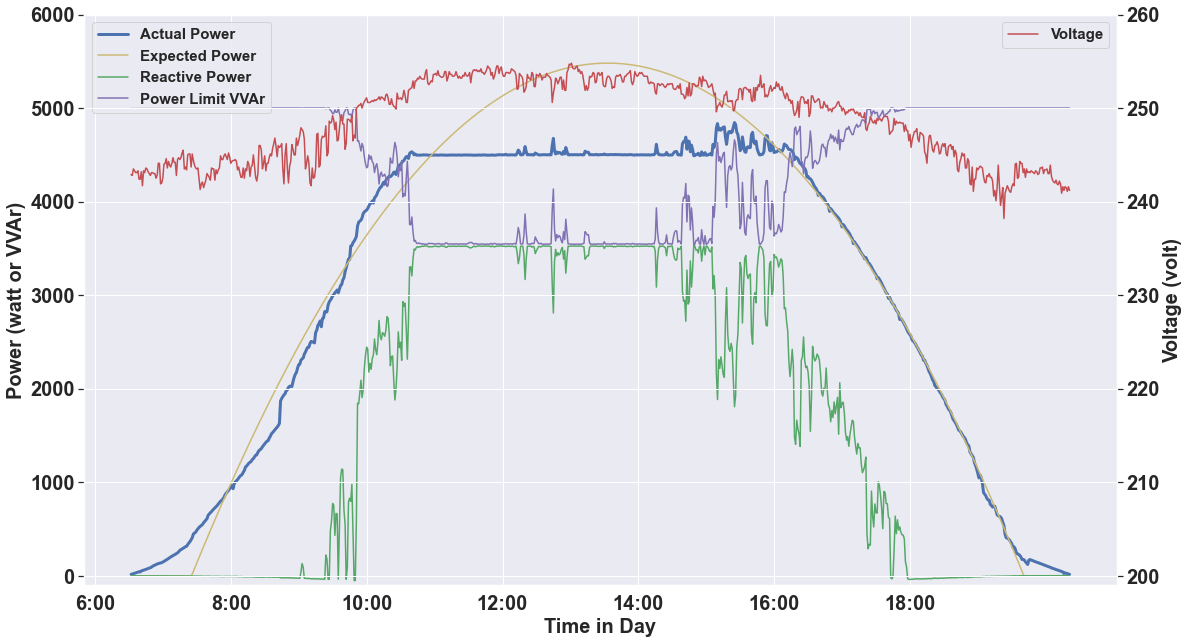


# Result on VVAr (sample 14)

Graphical user interface, application, Word

Description automatically generated

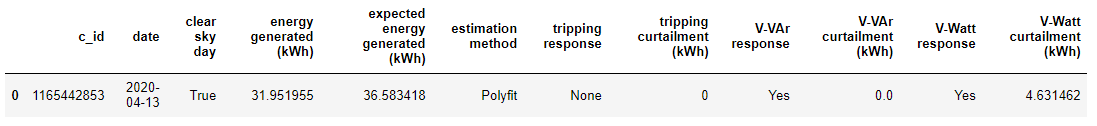
­­­­ ­­­­

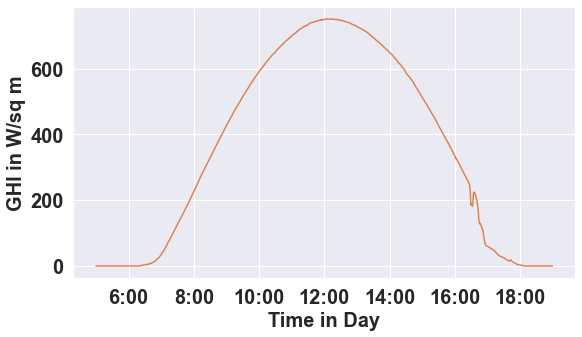
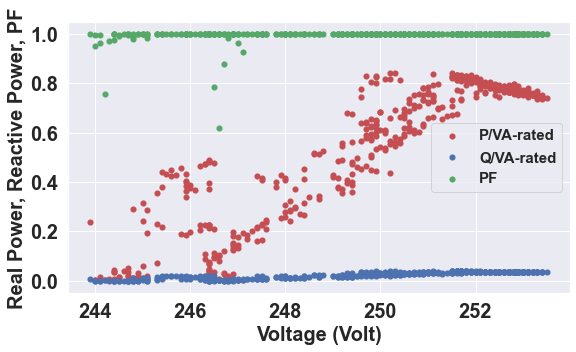


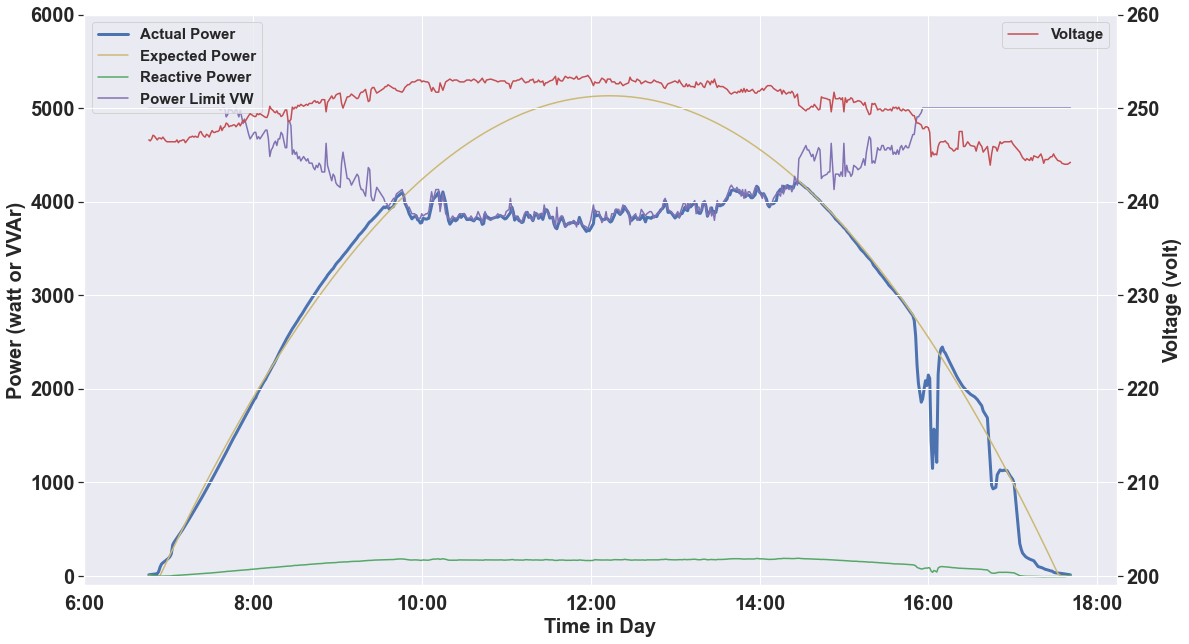
## There are two issues:

1. In the reactive power scatter plot, the Q/VA should have been negative. Problem with polarity correction?
2. In the power and voltage plot, the power limit vvar is below the actual power value. Probably the actual VA limit of the inverter is higher than the ac capacity of the inverter?

# Result on VWatt (sample 4)



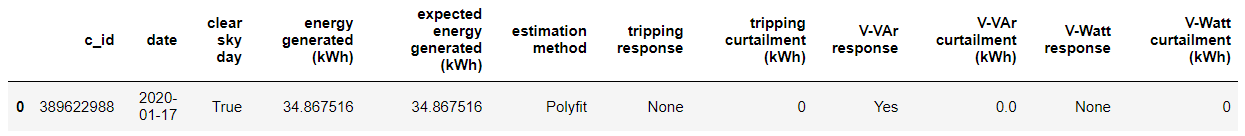


# Result on Incomplete Dataset (sample 5)



The tool will judge the dataset is incomplete only if there are less than 1000 rows in the data. The data should be more than that because the data resolution is either 60 or 5 seconds in SolA dataset.

# Result on No Curtailment Site in Clear Sky Day



Chart, line chart

Description automatically generatedChart, scatter chart

Description automatically generated

Chart, line chart

Description automatically generated

# Polyfit with Constrain Idea Testing

When we implement the polyfit with constrain into a clear sky day without curtailment, it seems we get overestimating:

With constrain:

Chart, line chart

Description automatically generated

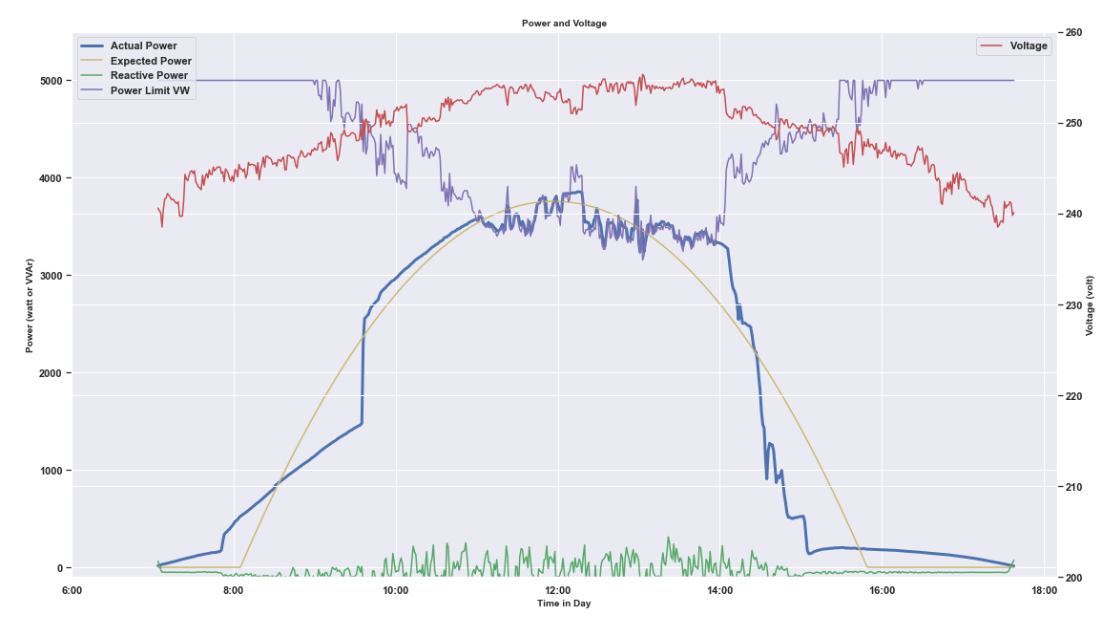
Without:

Chart, line chart

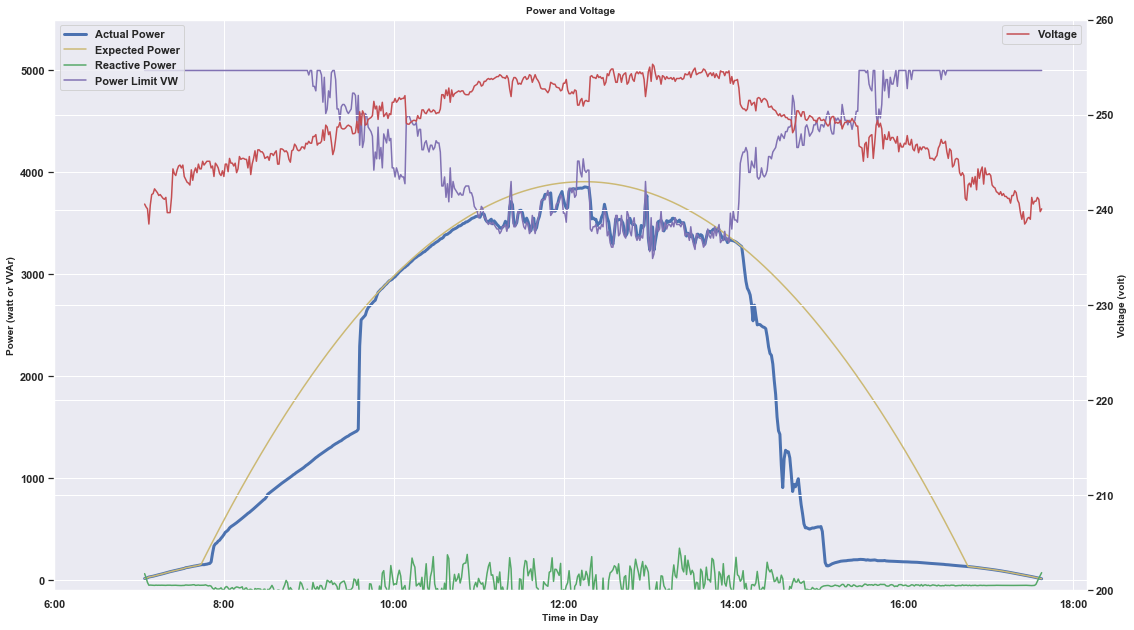
Description automatically generated

Should we give up on this? Or should we still think to optimize it since VWatt calculation is inaccurate and sometime it is underestimating, eg in sample 3:

Without constrain:



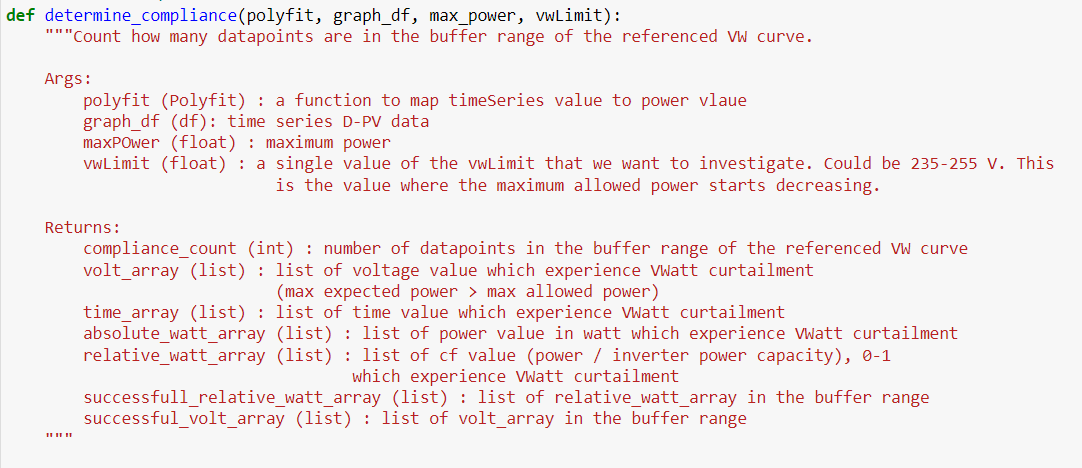
With constrain:



23/9 2022

Documentation

* Function and class docstring are done



* Naming convention edit is done:
  + Function, variable: lower\_case
  + Constant: UPPER\_CASE
  + Class: CamelCase
* Readme is done : https://github.com/mssamhan31/Solar-Curtailment
  + About
  + Getting started
  + Tool use demonstration: Screen capture of input, & output
  + High Level Explanation of How The Algorithm Works
  + Tool Limination & notes
  + Some Related Articles and Papers
  + Contributing
  + Project Partners
  + Authors
  + License
  + Contact
* Dataset information is done : <https://github.com/mssamhan31/Solar-Curtailment/blob/main/documentations/solar%20curtailment%20dataset%20information.docx>

# AC Capacity Curtailment

Chart

Description automatically generated with low confidenceA picture containing chart

Description automatically generated

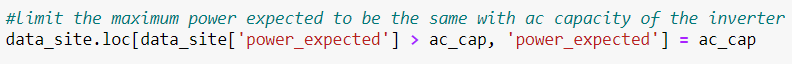
Chart

Description automatically generatedA picture containing chart

Description automatically generated

For c\_id 198317149, it is clear that the power is curtailed not because of tripping, VVAr, or VWatt response. Rather, it is because the poor sizing of the inverter. The ac capacity of the inverter is too low to accommodate the DC power generated by the PV array. Hence, the power is curtailed by the ac capacity of the inverter. In this case, the ac capacity is 5000 watt.

To accommodate this, we add one condition for the polyfit algorithm: make sure that the expected power generated is never more than the ac capacity of the inverter. Screenshot below:



Result for sample 4:

Chart, line chart

Description automatically generated

# Publish the script as a package

The main reference we use is <https://realpython.com/pypi-publish-python-package/> .

## Package Name

As discussed with Baran, we should try to name it with something general without CANVAS word. So, we name it ‘solar-curtailment’, which is still available in PyPI (python package index).

So later, we can install the package using this command.

pip install solar-curtailment

We will, however, try to publish it using the name trialsamhan2 for trial.

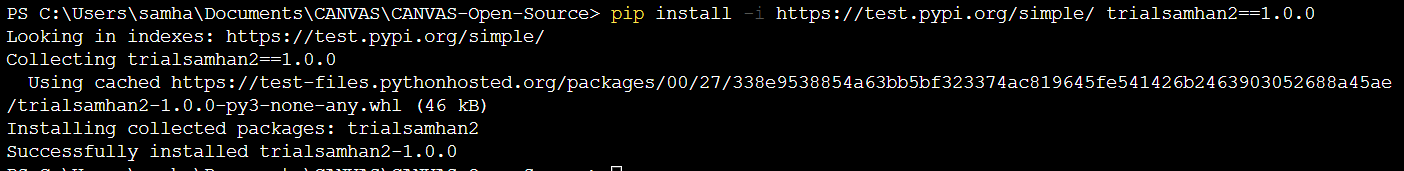
# Implementation to a Package

## Transformed the module into package

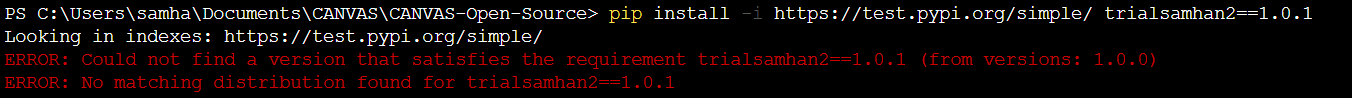
Graphical user interface, text

Description automatically generated with medium confidence

## Trial in TestPyPI



## Issue when changing the version in TestPyPI



Solved. This issue does not happen in pypi. Seems like only happen in TestPyPI.

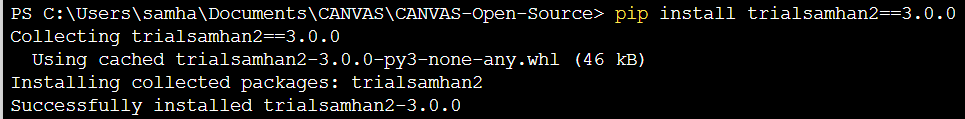
# Trial in PyPI

Install version 1.0.1

A screenshot of a computer

Description automatically generated

Install version 3.0.0



## Implementation

Text

Description automatically generated

# 6 Oct 2022

# Readme Baran Review

Followed up

# VVAr according to AS NZS 4777.2 2015 and 2020

Chart, line chart

Description automatically generated

## Allowed Range

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | V1 (volt) | V2 (volt) | V3 (volt) | V4 (volt) |
| 2015 | 207 | 216-230 | 235-255 | 244-265 |
| 2020 | 180-230 | 180-230 | 230-265 | 230-265 |

However, according to Baran, this is not applicable because the site is indeed Australia B, as informed by the industry partner. Australia C site, for instance, is a remote area.

# VVAr detection algorithm

We say that it is VVAr if V3 in its range, V4 in its range, and the percentage of power scatter between V3 and V4 in the buffer range is higher than certain threshold, which is 80%.

Chart, scatter chart

Description automatically generated

# Script file separation

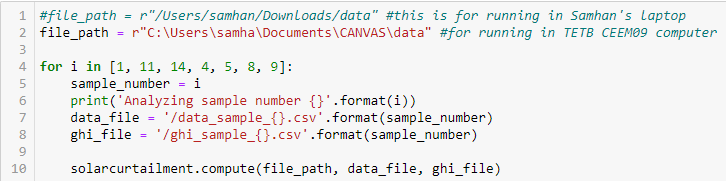
Graphical user interface, application

Description automatically generated

Merge to main?

# Current Implementation





# VVAr according to four different Standards:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| V-Var standards | VAr\_injection/  VA\_rated (%) | V1 (volt) | V2 (volt) | V3 (volt) | V4 (volt) | Var\_absorbtion/  VA\_rated(%) |
| SAPN TS-129 | 31 | 207 | 220 | 248 | 253 | 44 |
| AS/NZS 4777-2015 | 30 | 207 | 220 | 250 | 265 | 30 |
| ENA recommendation – 2019 | 41 | 207 | 220 | 240 | 258 | 60 |
| AS/NZS 4777 – 2020 (Australia B – small systems) | 30 | 205 | 220 | 235 | 255 | 40 |

Taken from CANVAS Final Report

It means, the possible range for V3 and V4 are:

V3: 235 – 250 V

V4: 253 – 268 V

However, we put 1.5 V margin to take some measurement random error into account, so the possible range for V3 and V4 are:

V3: 233.5 – 251.5 V

V4: 251.5 – 269.5 V

# Progress 11 October 2022

* Removed all functions that are not required
* Finished transforming the script to OOP using class. All functions are changed into class method
* Tested implementing from PyPI: pip install trialsamhan2==22.0.0

Implementation:



Text

Description automatically generated

Inside solarcurtailment.compute:

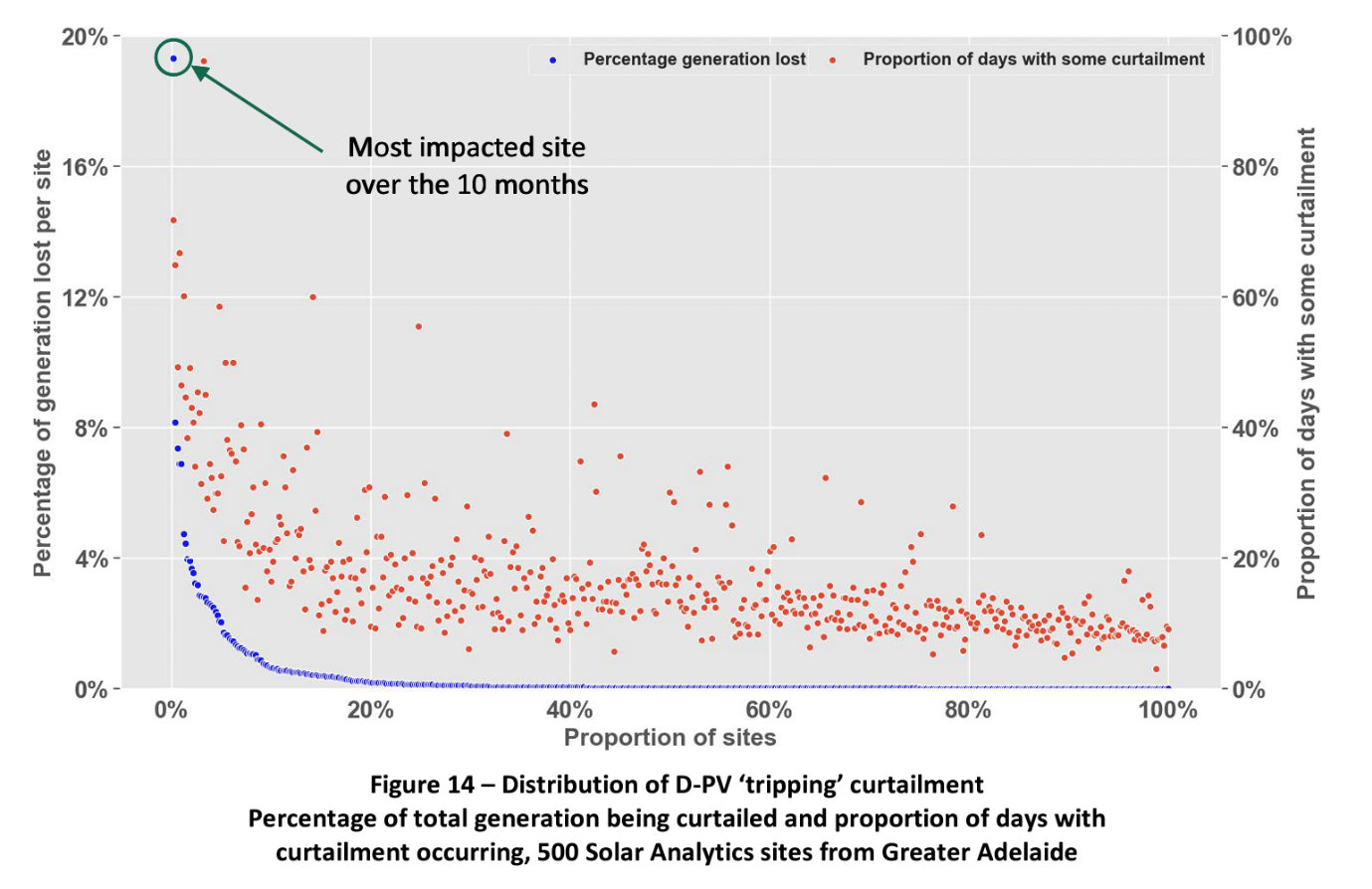
Text

Description automatically generated

# 20 October 2022

## 1. Fig. 14 plot for only clear sky days

This is figure 14 plot from the final report, showing the percentage of total generation being curtailed due to tripping and the proportion of days with curtailment.



Next, this is similar plot but analyzing only the clear sky days.

Chart, scatter chart

Description automatically generated

Analysis:

The biggest percentage generation lost is increased from less than 20% to nearly 25%. We can also see that if we analyze only clear sky days, some sites do not experience curtailment at all for all days (last 20% of the data).

## 2. Updated summary spreadsheet with only clear sky days data

The summary spreadsheet has been updated with the file name SolA Curtailment Summary\_Final\_tripping.xlsx

The script used to produce those 2 results are available in /other/Tripping in Clear Sky Days Analysis.ipynb

# Reference

Packaging project https://packaging.python.org/en/latest/tutorials/packaging-projects/