Intelligent Sub-metering Devices

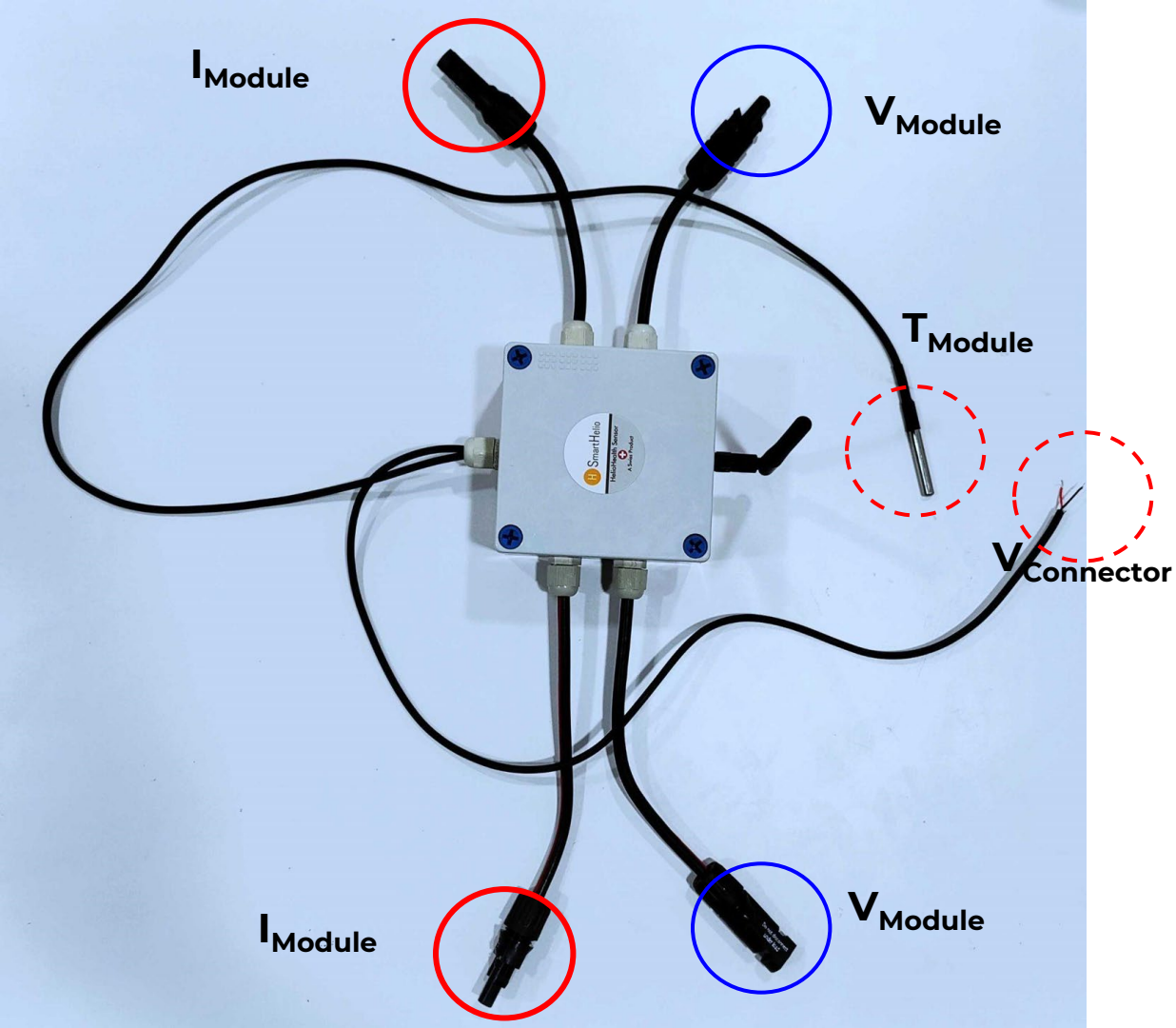
TECHNICAL OVERVIEW

Date: 28th November 2024

Team: SmartHelio/HSLU

SmartHelio
THE DOCTORS OF SOLAR PLANTS

GENTE

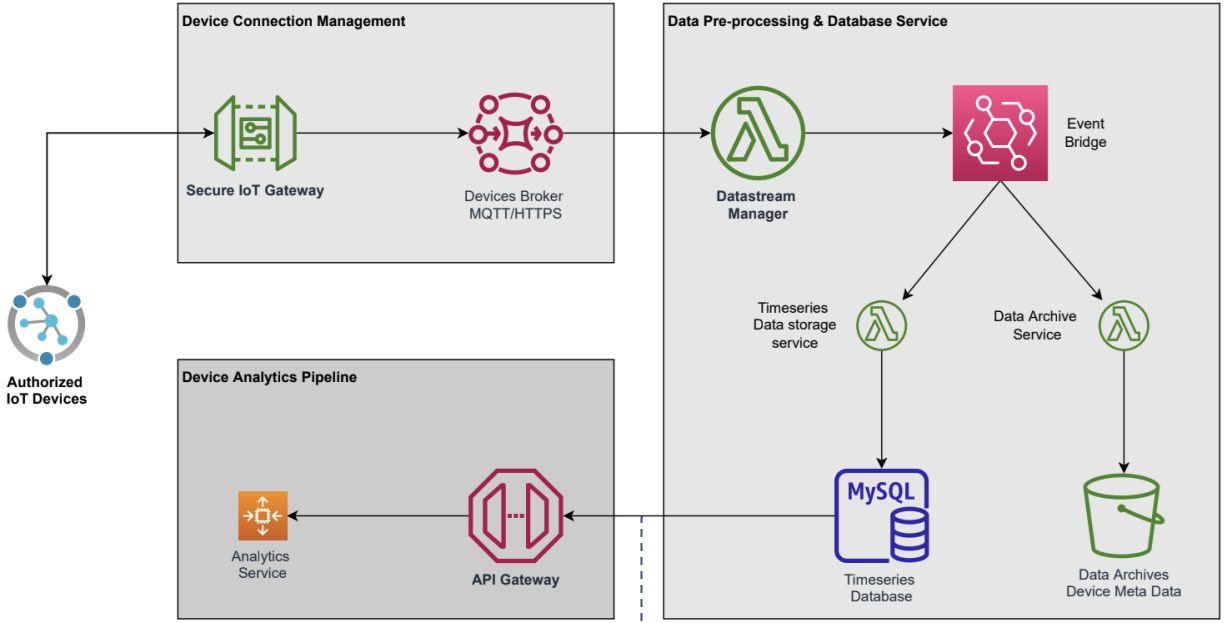


SmartHelio
THE DOCTORS OF SOLAR PLANTS

Upgraded and adapted the device to measure time-series readings (of Voltage, Current, and Temperature) across the PV modules and MC4 connectors across the PV sites. *This opened a new dimension for advance analysis and system modeling*

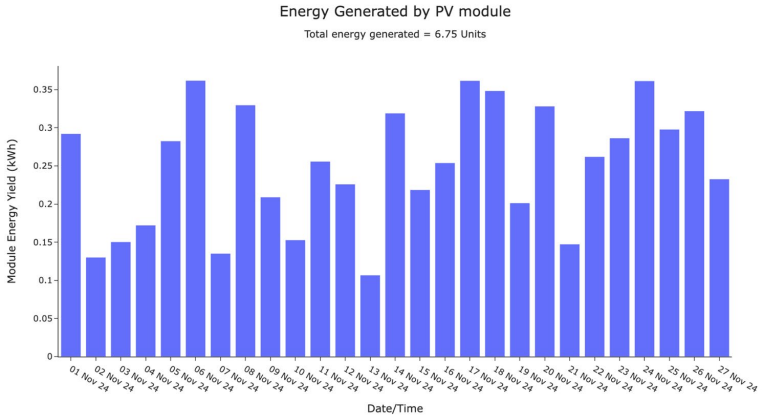
Data Architecture: IoT Platform for Seamless Data-flows

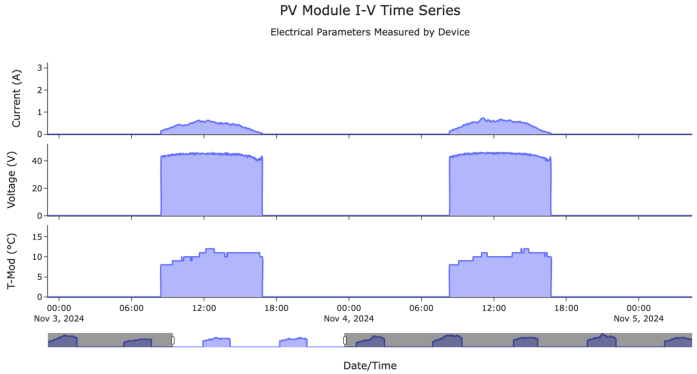
Intelligent sub-metering devices are deployed across the PV modules.



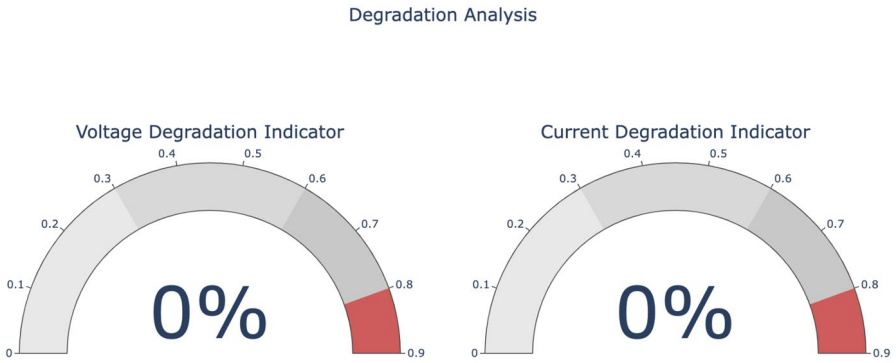
Secure Data Access to 3rd Party

Data Visualization: *Visualize and analyse the monitored devices*





Disclaimer : *Zero values indicate that data was not logged. Probable reasons can be ongoing maintenance, low solar irradiance due to shading, cloud cover or rain. Zero values can also occur due to inverter shutdown.*



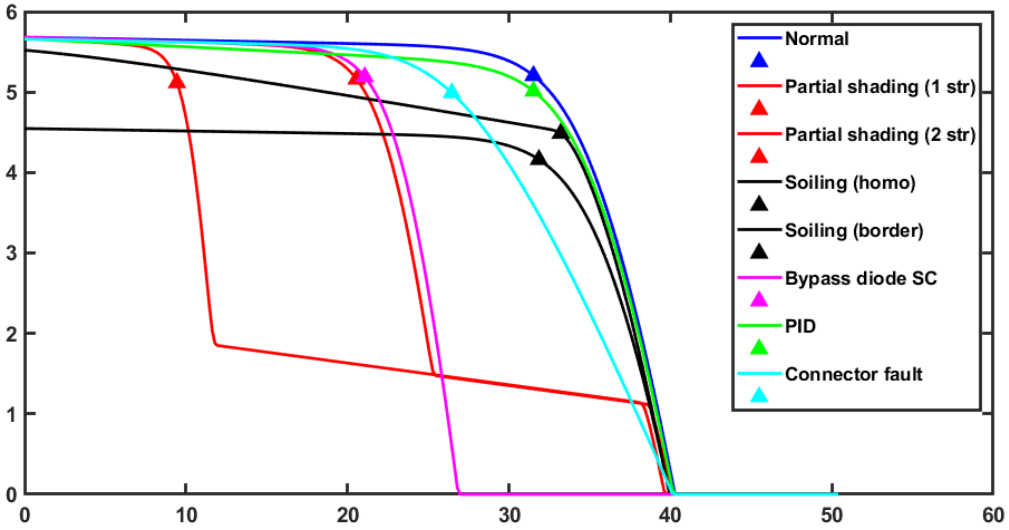
Disclaimer : *This analysis is based on all the data points collected to date, results will be better if available data is more than six months.*

Note: *The model requires at least 6 months of module performance data to accurately estimate the degradation numbers.*

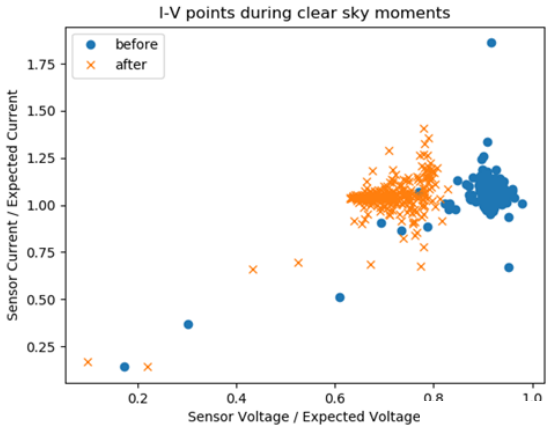
METHODOLOGY:

Faults affect Voltage and Current of the PV system in a unique manner. Which changes the I-V characteristic of the PV module.

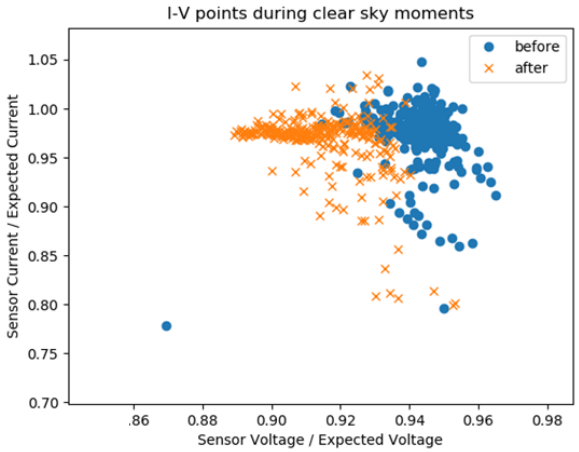
Information of change in signature can be used for detecting/classifying the faults and estimating the state of health of the PV modules.



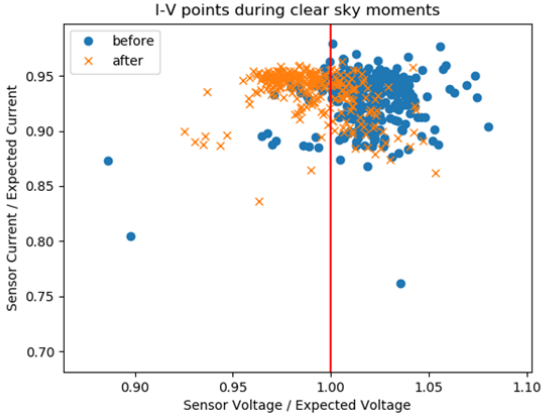
Methodology: *PV Panel Health Algorithms*



Damaged PV Module



Vegetation covering the string/module



Soil deposition on modules

Thank You!

SmartHelio
THE DOCTORS OF SOLAR PLANTS

GENTE

DOCUMENTATION

IoT Device API

Description: The API will help you to filter out the Sensor/Device data based on start and end date if you are an authorised user. The data/output is in JSON format. A sample example of the output will be attached below.

Endpoint: A GET request

- This endpoint(a GET Request) shows you the data for the particular/selected device/sensor-id
- It filters out the data with the help of start date and end date
- The date format should be written in DD-MM-YYYY format inside the request body
- If the date format is not followed OR if the inputs of start/end date are logically incorrect then you will get respective error message as an output
- Sensor IDs must be written as an Array of strings, for ex: ["51257rffs6", "uhvwdd6268",....]
- Username and password are required in the request body in string format to check if you are an authorised user to access the data
- Content Type: application/json (header)

The request body should be 'raw', for example:

Body raw (json)

```
{
    "username": "<your_username>",
    "password": "<your_password",
    "sensor_ids": ["sensor_ID_1","sensor_ID_2", ......],
    "startDate": "DD-MM-YYYY",
    "endDate": "DD-MM-YYYY"
}
```

Output format will be an array of JSON objects:

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify


```
{
    "username": "demo@smarthelio.com",
    "password": "demo@user",
    "sensor_ids": ["8991102105531120141f"],
    "startDate": "01-09-2023",
    "endDate": "02-09-2023"
}
```


Body Cookies Headers (7) Test Results Status: 200 OK Time: 446 ms Size: 71.77 KB Save as Example

Pretty Raw Preview Visualize JSON


```
[
    {
        "ccid": "8991102105531120141f",
        "timeStamp": 1693548840,
        "voltage": 37.59,
        "current": 0.09,
        "moduleTemp": 27
    },
    {
        "ccid": "8991102105531120141f",
        "timeStamp": 1693548960,
        "voltage": 11.52,
        "current": 0.19,
        "moduleTemp": 27
    },
    {
        "ccid": "8991102105531120141f",
        "timeStamp": 1693549080,
        "voltage": 35.44,
        "current": 0.21,
        "moduleTemp": 27
```


Postbot Runner Start Proxy Cookies Trash

- *ccid* is the one of the sensor/device id which you have insert as an input to the request body
- *Timestamp* is in unix/epoch format(type: Integer)
- *Voltage* is in volts(type: float)
- *Current* is in ampere(type: float)
- *moduleTemp* is in degree celsius(type: float)