BE Project

Group 4

Introduction :

This project is aimed at building a Hybrid DC Micro-grid using Machine Learning for load demand forecasting.

Technology Stack :

1. Python3, JupyterLab, Colab.
2. Dash library for Dashboard.
3. Tensorflow for DL based model training.
4. Apache Kafka for simulated pub-sub producer-consumer model for streaming data points in real time.

Installation (VM 1 : Master Node) :

1. Setup an Ubuntu-based virtual machine on AWS EC2, having 1 GB RAM.
2. Download the Kafka installer.

```
ubuntu@ip-172-31-37-97:~$ sudo apt update
ubuntu@ip-172-31-37-97:~$ sudo apt install python3-pip -y
ubuntu@ip-172-31-37-97:~$ pip3 install gdown
ubuntu@ip-172-31-37-97:~$ export PATH=$PATH:/home/<username>/.local/bin
ubuntu@ip-172-31-37-97:~$ gdown 1gq2ZL8koe-v654Q8Epn4iz_obOsLcRQ3
```

3. Run the installer *(requires 500 MB data to download all files)* :

```
ubuntu@ip-172-31-37-97:~$ sh ./BE.sh
```

4. Change the IP addresses in ***$KAFKA_HOME/config/server.properties*** as the public IP of this VM. using :

```
ubuntu@ip-172-31-37-97:~$ nano $KAFKA_HOME/config/server.properties
```

5. Starting the Background Processes (Kafka) in one terminal :

```
ubuntu@ip-172-31-37-97:~$ sh ./kafka_start_1G.sh
```

Allow for the startup processing (2-3 mins). After running the project, for stopping these processes use sh ./stop.sh in another terminal.

6. Install the required Python libraries using :

```
ubuntu@ip-172-31-37-97:~$ pip3 install confluent_kafka dash plotly dash_bootstrap_components
```

7. Copy and paste the contents of the Master VM folder inside the home folder of this virtual machine. It contains code for the Dashboard and the Kafka Consumer.
8. Run the code using (using 2 separate terminals via Tilix) :

```
ubuntu@ip-172-31-37-97:~$ python3 dashboard.py
ubuntu@ip-172-31-37-97:~$ python3 kafka_consumer.py
```

Installation (VM 2 : Inference Node) :

1. Setup another Ubuntu-based virtual machine on AWS EC2, having 2 GB RAM, to host the trained prediction models to estimate solar generation and load demand.
2. Copy and paste the contents of the Inference VM folder inside the home folder of this virtual machine. It contains code for rendering the models as endpoints using a Flask server.
3. Install the required libraries :

```
ubuntu@ip-172-31-37-98:~$ pip3 install flask h5py numpy tensorflow
```

4. Run the code for the Flask server using (using 1 separate terminals via Tilix) :

```
ubuntu@ip-172-31-37-98:~$ python3 inference.py
```

Installation (Raspberry Pi) :

1. Setup a Raspberry Pi 3 B with Wifi connection (use either SSH or monitor and keyboard to control the Pi).
2. Copy and paste the contents of the Rpi folder inside the home folder of Raspberry Pi. It contains code for the Kafka Producer, Switching Algorithm and Fan Speed Control.
3. Change the IPs mentioned in the Kafka Producer and Switching Algorithm code to the public IP of the Inference Node.
4. Setup the Hardware as per the pin numbers mentioned in the code.
5. Install the required libraries :

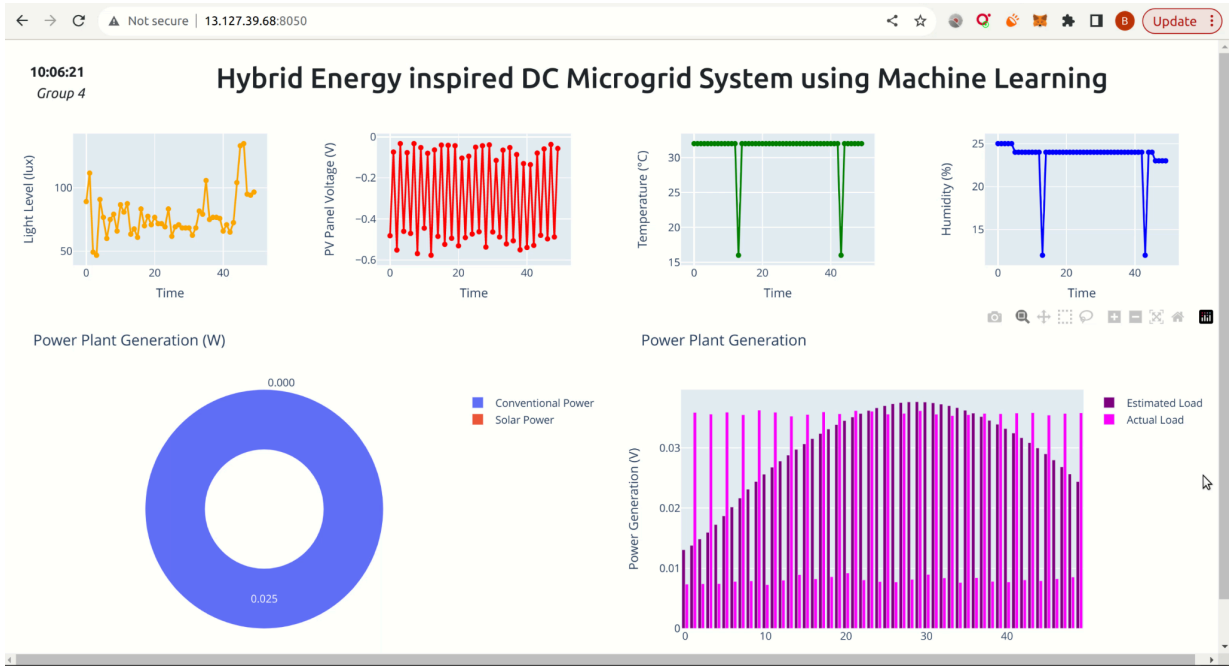
```
pi@192.168.43.112:~$ pip3 install RPi.GPIO numpy requests confluent_kafka adafruit-circuitpython-ads1x15 Adafruit_DHT

pi@192.168.43.112:~$ sudo apt-get install -y python3-smbus
```

6. Run the code for the Kafka Producer, Switching Algorithm and Fan Speed Control (using 3 separate terminals via Tilix) :

```
ubuntu@ip-172-31-37-97:~$ python3 send_data_using_kafka.py
ubuntu@ip-172-31-37-97:~$ python3 fan_speed_control.py
ubuntu@ip-172-31-37-97:~$ python3 switching_algorithm.py
```

View the Dashboard using a Browser :

<Public IP of Master Node>:8050/



Terminating the Background Processes (on Master Node):

Stop Kafka :

```
tweety@tweety-HP-ProBook-450-G3:~$ sh ./stop.sh
```