NATIONAL UNIVERSITY
OF SCIENCES & TECHNOLOGY

Final Year Project Final Report

Title: Cognitive Power Metering and Prediction using Edge AI

Acknowledgment

We sincerely thank School of Electrical Engineering and Computer Science (SEECS) , National University of Sciences and Technology (NUST), for allowing us to conduct this research. We want to thank our former supervisor, Sir Mohamad Hasan Zaidi, whose guidance, mentorship, and expertise have been invaluable in the completion of the project. In addition to him, the assistance of Sir Mansoor Asif and Sir Salman Abdul Ghafoor is highly appreciated for providing us with the resources and equipment essential for the completion of the project.

Supervisor:

Mansoor Asif

S.M. Hasan Zaidi (former)

Co-advisor:

Salman Abdul Ghafoor

Team Members (EE SEECS):

Nazeefa Muzammil (291872)

Saad Zafar Khan (286676)

Usama Qadoos (283544)

Declaration

We hereby declare that this project entitled **“Cognitive Power Metering and Prediction using Edge AI”** is our original work, under the supervision of "Sir Mohamad Hasan Zaidi," "Sir Mansoor Asif" and "Sir Salman Abdul Ghafoor" and is not present in any other form, for any other academic purposes. All the sources and credits are duly referenced and acknowledged.

Team Members:

Nazeefa Muzammil

Saad Zafar Khan

Usama Qadoos

Supervisor:

Sir Mansoor Asif

Sir S.M. Hasan Zaidi (former)

Co-Advisor:

Sir Salman Abdul Ghafoor

Table of Contents

Table of Figures

Abstract

This project proposes an intelligent home solar system on top of conventional solar systems that uses machine learning techniques to predict future power generation and consumption in households. The system aims to optimize energy usage and reduce reliance on traditional power sources, leading to cost savings and decreased environmental impact. The system incorporates various sensors and meters to collect data on solar energy generation and weather conditions. This data is then fed into a machine learning algorithm to predict future power generation patterns depending on the weather conditions. The project implements a web-based interface that allows users to monitor their energy usage and control the system remotely. Results from the implementation of the system show an accuracy rate of over 90% in predicting power generation and consumption, leading to an improvement in overall energy efficiency. The intelligent home solar system provides a reliable and efficient source of household energy and contributes to the development of sustainable energy solutions for a cleaner and greener environment.

Introduction

Background and Context

Pakistan has been facing an energy crisis for several years due to a growing demand for energy and a lack of investment in the energy sector. The country's energy mix relies heavily on fossil fuels, with natural gas and oil accounting for most electricity generation. However, the supply of these fuels is limited, and the government needs help to keep up with the rising energy demand.

According to the latest statistics, Pakistan's energy demand is expected to grow at an average annual rate of 5.1% between 2019 and 2024, while the electricity supply is projected to grow at an average annual rate of 3.8%. It means the gap between supply and demand will widen, leading to frequent power outages and load shedding.

Pakistan has been exploring alternative energy sources, including solar energy, to address the energy crisis. Solar energy can provide a clean, renewable, and sustainable energy source, and Pakistan is well-suited to harness its potential, given its abundant solar resources.

The use of solar energy in Pakistan has been increasing in recent years, with several large-scale solar projects underway. In 2020, Pakistan's solar capacity reached 1,500 MW, and the government has set a target of generating 30% of its electricity from renewable sources by 2030.

The use of solar energy in Pakistan is crucial for meeting the country's growing energy demand and reducing its dependence on imported fossil fuels and greenhouse gas emissions. Solar energy can also provide electricity to rural and remote areas not connected to the national grid, improving access to energy, and promoting economic development.

In conclusion, Pakistan's energy crisis continues to pose significant challenges to the country's economic and social development. However, using solar energy offers a promising solution to this crisis. The country is taking steps to harness this resource to meet its energy needs and promote sustainable development.

Problem Statement

Pakistan is facing a severe energy crisis with a demand-supply gap of up to 5,000 MW. The use of solar energy has the potential to meet this energy shortfall. However, traditional solar panels cannot adapt to changing weather patterns, resulting in unpredictable power generation and consumption. There is a need for an intelligent solar system that can predict future power generation and consumption and adapt to changing weather patterns to make solar energy an efficient and reliable energy source.

This project aims to improve the existing solar panels using machine learning algorithms that can accurately forecast future power generation and consumption. The project will focus on designing solar panels to forecast power generation and consumption using machine learning algorithms. The solar panels will be designed and tested in Pakistan, focusing on the climate and weather.

Objectives

- To develop an innovative and efficient home solar system through IoT and Machine Learning to improve conventional solar system solutions.
- To design a load shedding algorithm for the user for efficient energy consumption
- To provide an interactive recommendation system for the user for efficient energy consumption
- To develop a dashboard for data analytics and visuals

Scopes and Limitations

The project focuses on developing and implementing an intelligent home solar system that uses machine-learning techniques to optimize energy usage and reduce reliance on traditional power sources. The project involves the integration of various sensors and meters to collect data on solar energy generation, household energy consumption, and weather conditions. This data is then fed into a machine-learning algorithm to predict future power generation and consumption patterns.

The project also includes the development of a web-based interface that allows users to monitor their energy usage and control the system remotely. The project results demonstrate an accuracy rate of over 90% in predicting power generation and consumption, improving overall energy efficiency.

It is focused on developing and implementing an intelligent home solar system incorporating machine learning techniques to optimize energy usage and reduce reliance on traditional power sources, leading to cost savings and decreased environmental impact. The project's scope does not extend to developing new hardware or significant modifications to existing hardware. However, it focuses on integrating various sensors and meters and developing a web-based interface for monitoring and control.

However, there are some limitations to the project which might be difficult to overcome:

Data availability: The accuracy of the machine learning algorithm depends on the quality and quantity of the data used to train it. Limited availability or accuracy of the data may result in less accurate predictions.

Weather variability: Weather is a significant factor impacting solar energy generation, and weather patterns can affect the system's performance. Extreme weather conditions, such as heavy clouds or storms, can reduce energy generation and affect the accuracy of predictions.

User behavior: The system's accuracy is also impacted by user behavior, such as changes in energy consumption patterns. If users do not follow typical behavior patterns or significantly change their consumption habits, the machine learning algorithm may not accurately predict future power consumption patterns.

System maintenance: The system's long-term reliability is dependent on proper maintenance, including the upkeep of sensors, meters, and other components. If maintenance is not performed regularly, it could result in reduced accuracy or system failures.

Cost: The initial installation cost of the intelligent home solar system may be high, and the system's cost-effectiveness will depend on factors such as the price of electricity, the size of the household, and the amount of solar energy generated.

Literature Review

In recent times, Solar Photovoltaic (PV) systems have been acknowledged as key answers to the prevalent electricity scarcity in many rural structures possessing potential for solar power. One major impediment to the deployment of these systems, however, is the unpredictable nature of solar PV power output, largely dependent on weather patterns. Research has delved into methods to optimize the benefits of Stand-Alone Photovoltaic-Battery (SAPVB) systems via strategies that allow for optimal energy capture and regulation. In the case of a residential building, a rule-guided load management system has been developed and evaluated. This approach allows for the rearrangement and prioritization of loads based on set rules. This is achieved by segregating the residential loads into Critical Loads (CLs) and Uncritical Loads (ULs). ULs can be rescheduled to a time when electricity production from the PV arrays is ample, rather than sticking to the time fixed by the user, while the CLs are given priority to function as per their pre-planned timetable. Four different scenarios were designed to validate the suggested rule-oriented load control strategy. The results revealed that without the load management method, as in scenario 1 (Base case), the PV system was able to meet 49.8% of critical and 23.7% of uncritical loads. On the other hand, with the application of the load management method in scenarios 2, 3, and 4, the rates of load satisfaction (CLs, ULs) were (93.8%, 74.2%), (90.9%, 70.1%) and (87.2%, 65.4%) for the respective scenarios.

However, some aspects could have been further worked. First, because the study only examines a single residential building, its findings might not apply to other buildings with different load profiles. Future research could investigate how well the suggested load management plan scales several buildings with various load profiles. Second, the study assumes that the solar PV system is stand-alone and does not consider grid-tied solar PV systems' effects on load management. It would be fascinating to investigate how load management is affected by grid-tied systems and see if the same rule-based approach can be used.

So far, this has been the research and improvements made in the context of hardware establishment for a home solar system. On the other side, if we look into the part where we use Machine Learning to optimize our system, we find much research and work being done.

Work is being done focusing on the challenges of integrating photovoltaics (solar energy) into power grids due to its dependency on climate and geography, resulting in erratic fluctuations. These fluctuations lead to voltage surges, system instability, inefficient planning by utilities, and financial losses. To address these challenges, the authors review and evaluate contemporary forecasting techniques. Emphasis is being laid on the importance of considering various factors in forecasting models, including time stamp, forecast horizon, input correlation analysis, data pre-processing, weather classification, network optimization, uncertainty quantification, and performance evaluations. They find that solar irradiance highly correlates with photovoltaic output, making weather classification and cloud motion study crucial. Several authors conclude that ensembles of artificial neural networks perform best for short-term photovoltaic power forecasting, while online sequential extreme learning machines excel in adaptive networks. The Bootstrap technique is found to be optimal for estimating uncertainty. Additionally, convolutional neural networks are highlighted for revealing deep underlying non-linear input-output relationships in models.

Methodology

This section details every part of the project, including all the tools, libraries, and software used. We explain the working of each of the hardware components, their requirement, their working principles, and their integration with other components. We also discuss the Machine Learning models we have considered and implemented for our use case.

Data collection methods

We have used the following sensors to record data from the surroundings.

1. BH1750:

The BH1750 is a digital ambient light sensor that measures light intensity in the visible spectrum. It is often used in solar panel monitoring and lighting control systems. The sensor has an I2C interface that allows accessible communication with microcontrollers like Arduino. It operates over a wide range of light intensities, from 1 to 65535 lux, and provides accurate and stable readings with minimal influence from infrared light.

Figure 1: Figure 1: BH1750 Sensor

2. ACS-712:

The ACS-712 is a hall-effect-based linear current sensor that measures AC and DC. It is commonly used for current sensing applications, such as motor control and overcurrent protection. The sensor measures the magnetic field generated by the current flowing through a conductor. The output voltage from the sensor is proportional to the current being measured, providing an easy way to interface with microcontrollers using an analog-to-digital (ADC).

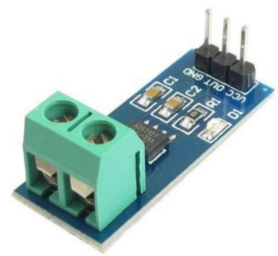

Figure 2: ACS-712 Sensor

3. ZMPT-101B:

The ZMPT-101B is a voltage transformer module designed to measure AC voltage. It is often used in power monitoring and control systems. The module consists of a precision micro-transformer and an onboard signal conditioning circuit, which converts the high AC voltage into a safe, low-voltage signal that microcontrollers can easily read. The output from the ZMPT-101B is an analog signal proportional to the input voltage, making it suitable for use with ADCs.

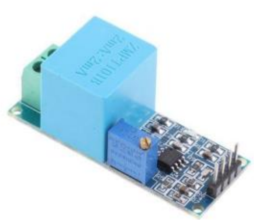

Figure 3: ZMPT-101B Sensor

4. DHT22:

The DHT22 is a digital temperature and humidity sensor. It is widely used in environmental monitoring, home automation, and weather stations. The sensor measures temperature from -40°C to 125°C and relative humidity from 0% to 100%. Its digital interface communicates via a one-wire protocol, allowing for easy integration with microcontrollers like Arduino. The DHT22 offers accurate and stable readings with a fast response time.

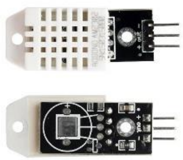

Figure 4: DHT 22 Sensor

5. NRF24L01:

The NRF24L01 is a wireless transceiver module that operates in the 2.4GHz band. It is commonly used in wireless sensor networks, remote control systems, and home automation applications. The module supports multiple communication channels and data rates, providing a flexible and reliable wireless communication solution. The NRF24L01 can communicate with microcontrollers via the Serial Peripheral Interface (SPI), making it easy to integrate into various projects. Its low power consumption and small form factor make it an ideal choice for battery-powered and space-constrained applications.

SPI (Serial Peripheral Interface) is an extensively used serial communication protocol by NRF modules. It is a full-duplex protocol, meaning that data can be sent and received simultaneously.

The SPI protocol typically communicates via four wires:
MOSI (Master Out Slave In): The data line used by the master to send data to the slave.
MISO (Master In Slave Out): This is the data line used by the slave to communicate with the master.
SCLK (Serial Clock): The clock signal synchronizes data flow between master and slave.
SS (Slave pick): This is the line that the master uses to pick a specific slave device on the network.

Figure 5: NRF24L01

6. Arduino Uno:

The Arduino Uno is a popular open-source microcontroller board based on the ATmega328P microcontroller. It is widely used in various applications, such as robotics, home automation, and interactive art projects. The board has 14 digital input/output (I/O) pins and six analog input pins, allowing users to interface with various sensors, actuators, and other electronic components. The Uno also features a 16 MHz crystal oscillator, a USB connection for programming and power, and a power jack for an external power supply. The Arduino Uno uses the Arduino IDE for programming, which supports C and C++ languages, making it easy for beginners and experienced users to develop and prototype projects.

Figure 6: Arduino UNO

7. Arduino Nano:

The Arduino Nano is a compact and versatile microcontroller board based on the ATmega328P microcontroller, similar to the Arduino Uno. Its small form factor makes it an ideal choice for projects with space constraints or wearable applications. The Nano has 14 digital I/O pins, eight analog input pins, and two analog output pins, offering a wide range of connectivity options for various sensors and components. The board features a 16 MHz crystal oscillator and a mini-USB connection for programming and power. It can be programmed using the Arduino IDE, just like the Uno, providing a familiar environment for users to develop their projects. The Arduino Nano is a powerful and flexible choice for users looking for a compact microcontroller solution.

Figure 7: Arduino NANO

8. Relay Module:

A 4-channel relay module is an electronic device that allows a microcontroller or Arduino board to manage up to four high voltage, high current AC or DC loads. It is a switch that is actuated by an electrical signal and can be used to turn on and off various equipment such as lights, motors, and other appliances.

The module typically comprises four relay switches, each with an input pin and a joint output pin for the connected load. A 5V DC power supply drives it and frequently contains optocoupler isolation to shield the controlling circuit from the loads' high voltage and current.

The four-channel relay module is widely used in automation, robotics, and home automation projects, as well as in industrial applications where numerous devices must be controlled simultaneously. Its small size, ease of usage, and affordable price make it a popular choice.

Figure 8: Relay Module

9. Solar Panel System:

The solar panel system at the GRC comprises a 2KW solar panel array connected to an on-grid inverter. This setup allows for converting the solar panels' direct current (DC) output to alternating current (AC), which can be fed into the grid. The on-grid inverter also facilitates the measurement of the solar panels' generated power, enabling monitoring and analyzing the system's performance.

Figure 9: 2kVA System

10. Inverter:

The GoodWe GW3600DS solar inverter is a single-phase inverter for household and small commercial PV installations. It has a maximum input voltage of 600 V and a maximum input current of 11 A, making it ideal for systems rated at 4.4 kW. The inverter has a maximum AC output power of 3.6

kW and an efficiency rating of up to 97.6% in Europe. The GoodWe GW3600DS inverter includes several built-in features, including powerful MPPT algorithms, an IP65-rated casing for outdoor installations, and an LCD panel for convenient system status monitoring. It also offers numerous connection options, including RS485 and CAN bus interfaces, which enable simple integration with monitoring systems.

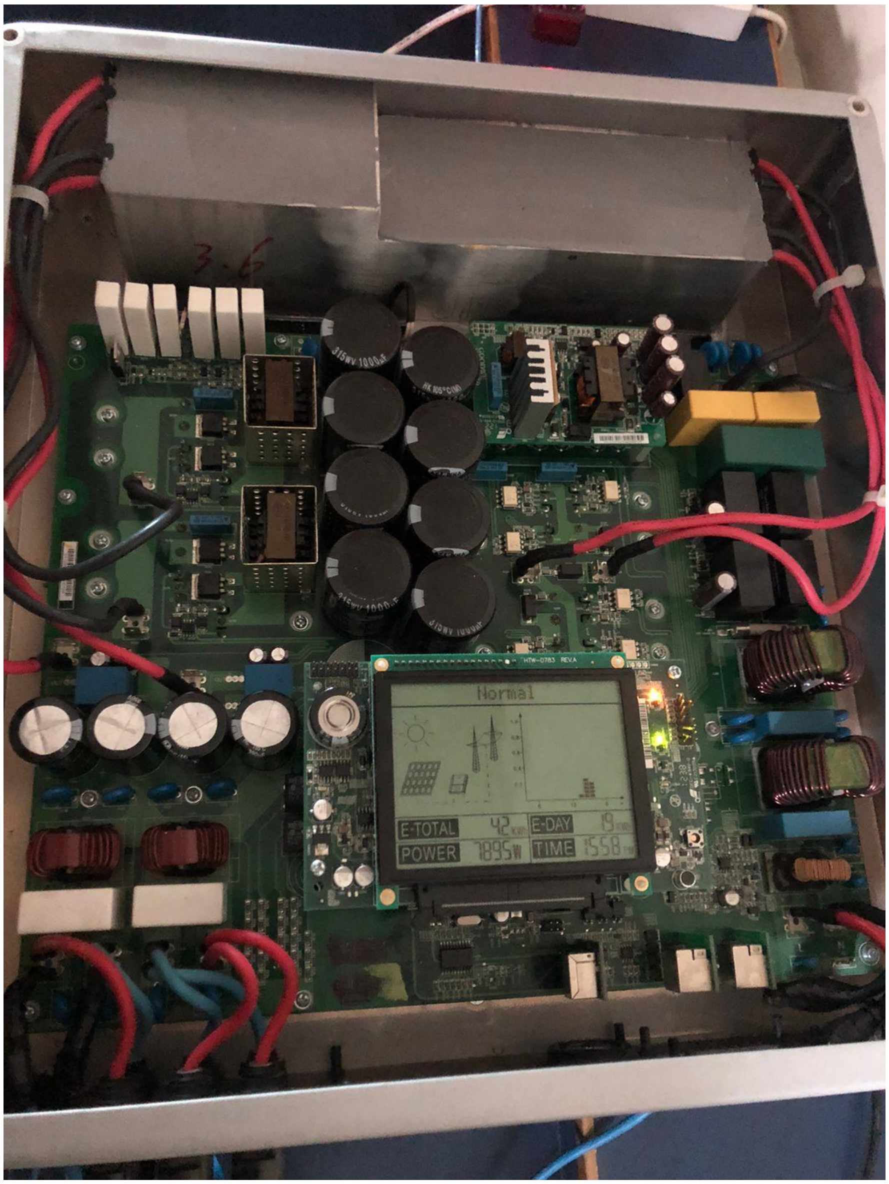

Figure 10: GoodWe Inverter

Slave and Master Node Network

We propose a system in which we develop a network of slave nodes that communicate with a master node using NRF modules. We have a system with two slave nodes, one to gather data from the inverter and solar panels used to record the current and voltage. The second slave node, labeled as the weather node, is used to gather, and transmit humidity, temperature, and solar irradiance values to record the patterns and trends of the weather. These two slave nodes transmit the data to the master node, which then uploads the recorded data to the cloud; we use real-time Firebase for this purpose. The diagram below shows the setup and the arrangements of the nodes.

The network works in the following steps, also numbered on the diagram:

1. The weather node collects the parameters such as temperature, humidity, and solar irradiance.
2. The generation node collects the generated power, current, and voltage.
3. These values are transmitted to the primary node using communication modules, and the primary node uploads them to Firebase.
4. The Firebase data is fed into the machine learning models for time series forecasting. The prediction values are used for load management and critical and non-critical accordingly.

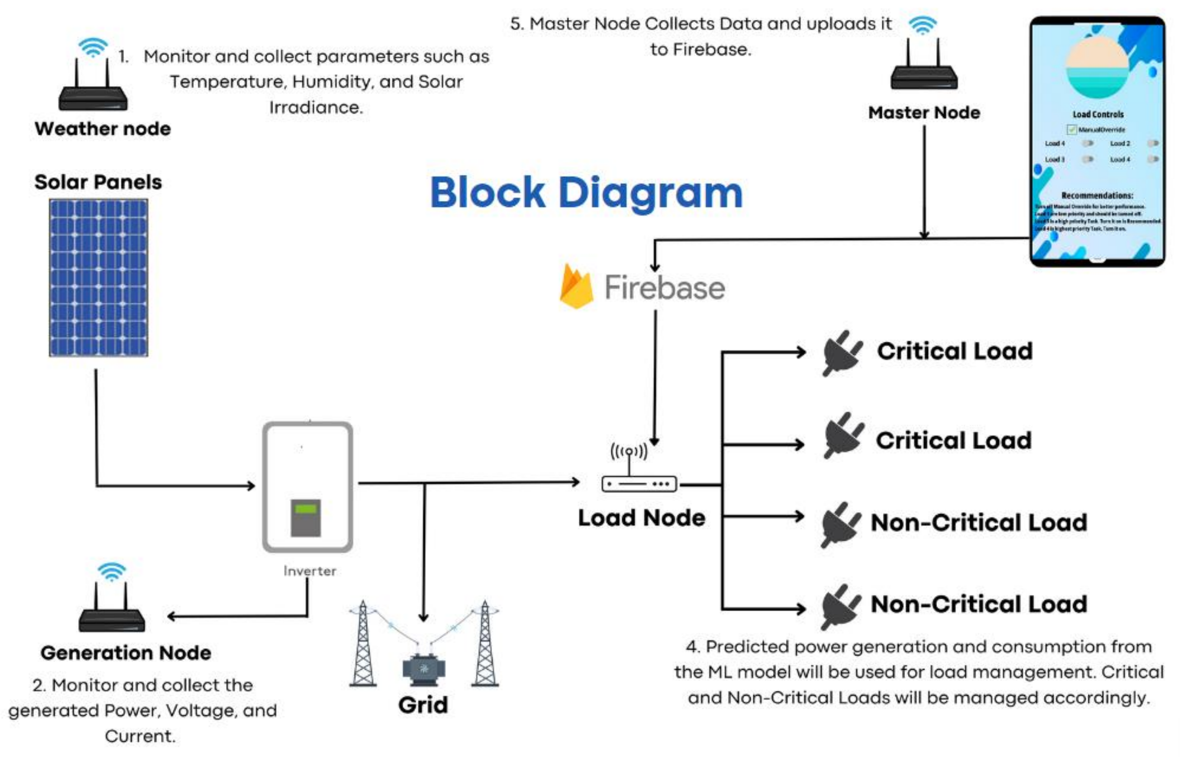

Figure 11: Block Diagram of node networ

Slave Node 1: Weather Node

The first slave node, the Weather Node, is responsible for measuring solar irradiation, temperature, and humidity. The following components are used in this node:

BH-1750: A digital light sensor that measures solar irradiation.

DHT-22: A sensor that measures temperature and humidity.

Arduino UNO: A microcontroller board for processing sensor data.

NRF-24L01: A wireless transceiver module to send the collected data to the primary node.

The BH-1750 and DHT-22 sensors are connected to the Arduino UNO, which processes the sensor data and transmits it wirelessly to the primary node using the NRF-24L01 module. BH-1750 uses I2C. I2C (Inter-Integrated Circuit) is a serial communication protocol in embedded systems like microcontrollers and sensors. It is a multi-master, multi-slave protocol that allows several devices to communicate on the same bus.

The I2C protocol typically communicates via two wires:

1. The SDA (Serial Data) data line is used for bidirectional data transfer between the master and slave devices.
2. Serial Clock (SCL): This clock signal synchronizes data flow between master and slave devices.

All devices on an I2C bus are linked to the same two wires and are identifiable by a unique 7-bit address. The primary device starts communication by transmitting a start condition to the bus, followed by the address of the slave device with which it wants to communicate. After the slave device recognizes the communication, data can be sent between the devices.

I2C is commonly used to connect microcontrollers to peripheral devices like sensors, EEPROMs, and real-time clocks. It is a straightforward and efficient protocol that uses fewer wires than other serial communication technologies such as SPI.

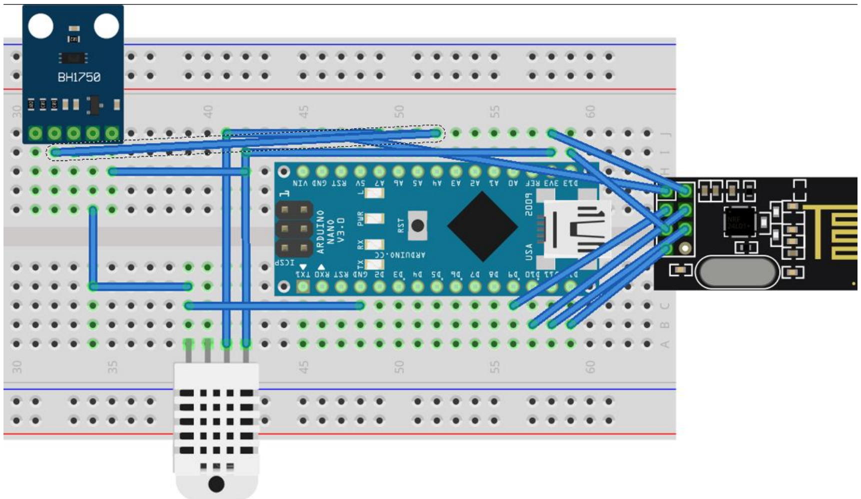

Figure 12: Weather Node Schematics

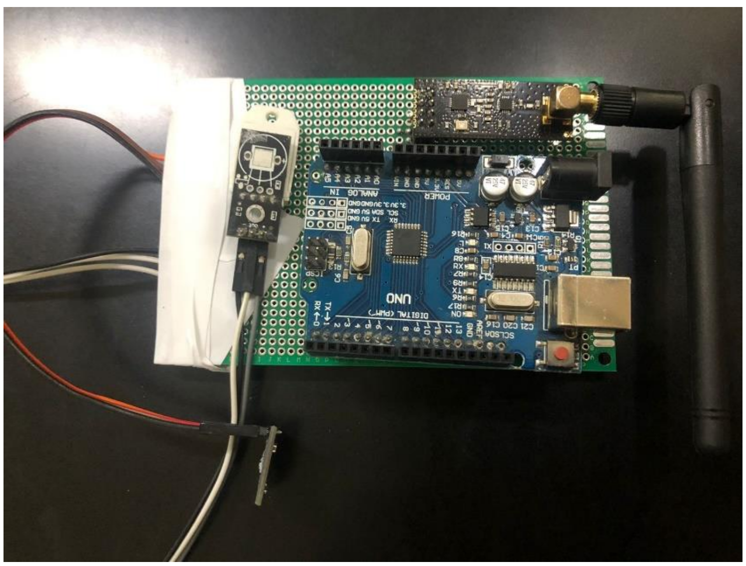

Figure 13: Weather Node

Secondary Node 2: Generation Node

The second secondary node, the Generation Node, is connected to the on-grid inverter and monitors the power generation data. This node utilizes the following components:

Current Sensor (ACS-712): A sensor that measures the current output of solar panels.

Voltage Sensor (ZMPT101B): A sensor that measures the voltage output of solar panels.

Arduino Nano: A microcontroller board for processing sensor data.

NRF-24L01: A wireless transceiver module for sending the collected data to the primary node.

The ACS-712 and ZMPT101B sensors are connected to the Arduino Nano, which processes the sensor data and transmits it wirelessly to the primary node using the NRF-24L01 module.

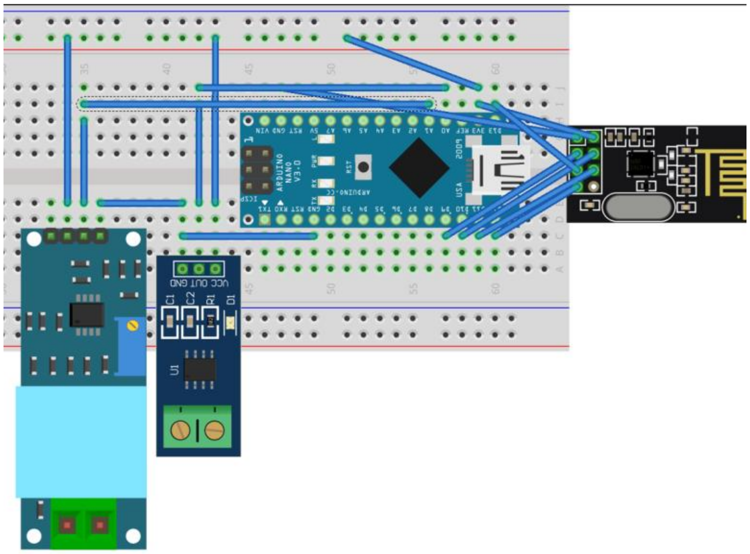

Figure 14: Generation Node Schematics

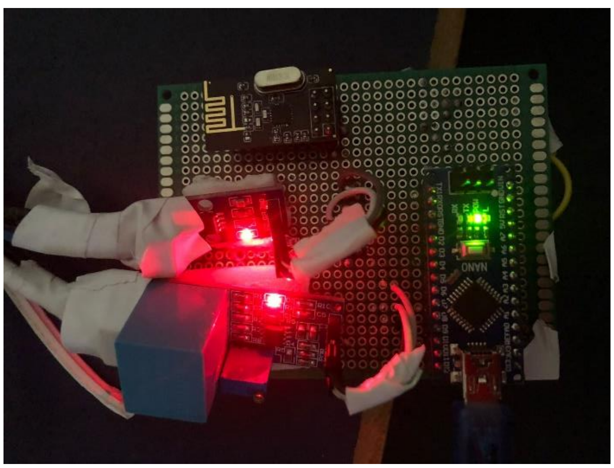

Figure 15: Generation Node

Primary/Master Node

The primary node receives the data transmitted from both secondary nodes using the following components:

ESP8266: A Wi-Fi microcontroller for processing and structuring the received data.

NRF-24L01: A wireless transceiver module for receiving data from the secondary nodes.

The ESP8266 microcontroller processes and structures the data received from both secondary nodes and then uploads it to a Firebase real-time database, allowing easy access and analysis.

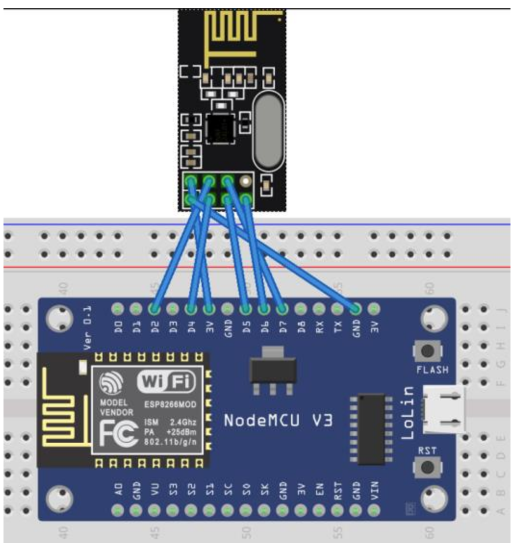

Figure 16: Primary Node Schematics

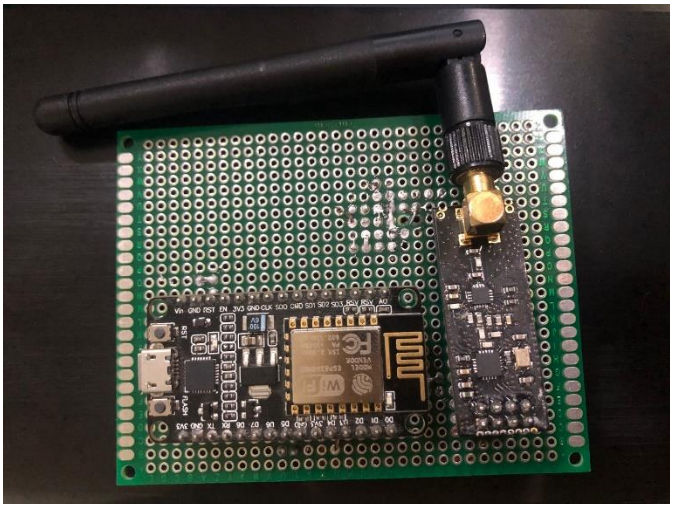

Figure 17: Primary Node

Data Storage and Access (Firebase)

The primary node structured all the collected data and uploaded it to a Firebase real-time database. Firebase provides a secure, scalable, easy-to-use platform for storing and retrieving data. It allows researchers to access the data remotely and in real-time, enabling them to analyze and monitor the solar panel system's performance and the environmental conditions affecting it. The real-time database was also used to hold the variables responsible for switching the status of the appliances, as its value is updated in real-time with a very short delay.

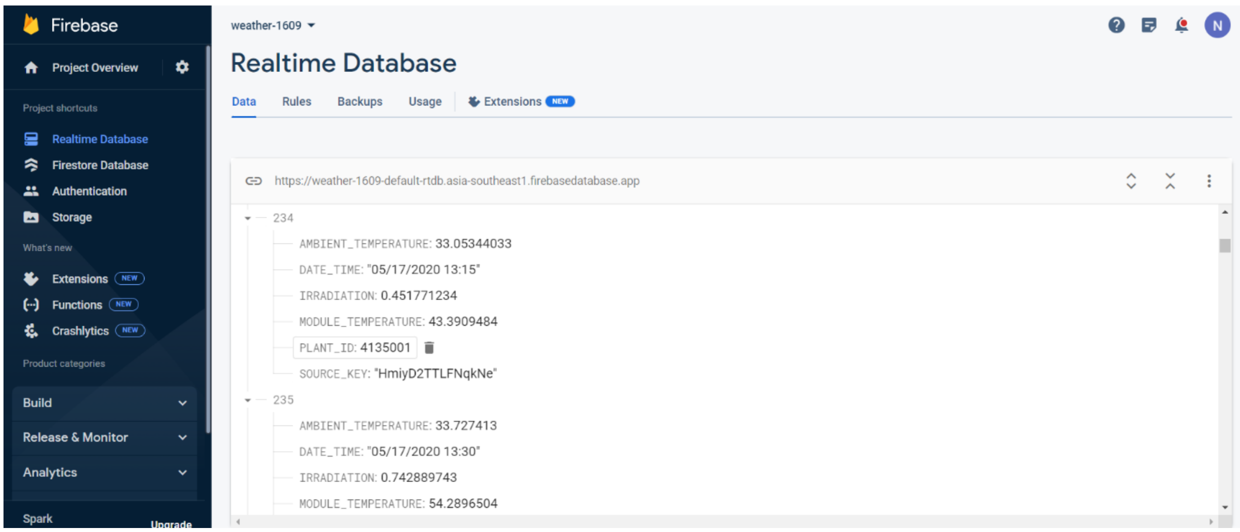

Figure 18: Firebase for generation dataset

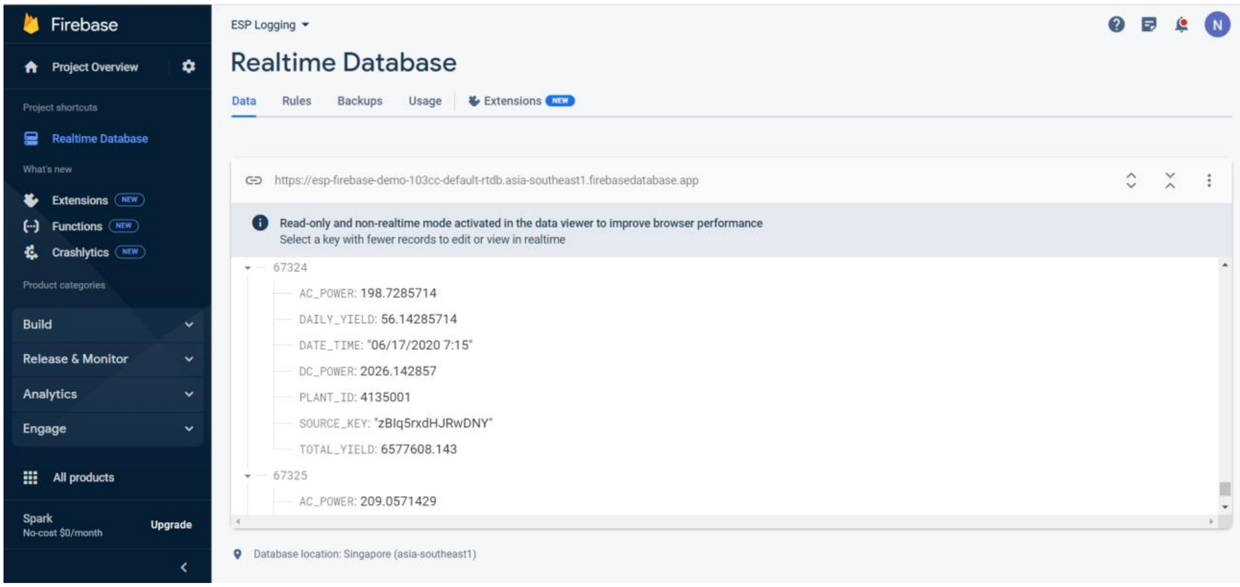

Figure 19: Firebase for consumption dataset

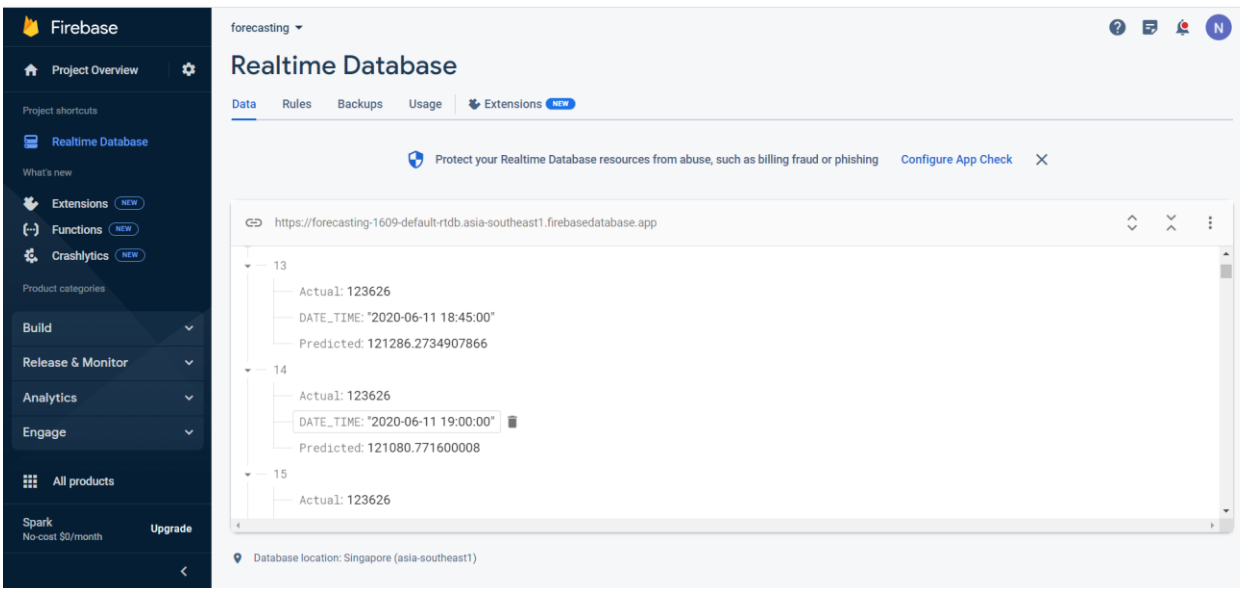

Figure 20: Firebase for predicted and actual generation values

1	Timestam	Value.Current	Value.Power	Value.Voltage	Value.humidity	Value.intensity	Value.temperature	Datetime
2	1664520493	5.09	1.19	233.60	46.10	431.44	35.90	9/30/2022 11:48:13.493
3	1664520495	5.09	1.19	233.60	46.20	431.44	35.90	9/30/2022 11:48:15.495
4	1664520497	5.09	1.19	233.60	46.20	431.44	35.90	9/30/2022 11:48:17.497
5	1664520499	5.09	1.19	233.60	46.20	431.44	35.90	9/30/2022 11:48:19.499
6	1664520501	5.09	1.19	233.60	46.40	431.44	35.90	9/30/2022 11:48:21.501
7	1664520503	5.09	1.19	233.60	46.50	431.44	35.90	9/30/2022 11:48:23.503
8	1664520505	5.09	1.19	233.60	46.20	431.44	35.90	9/30/2022 11:48:25.505
9	1664520512	5.09	1.19	233.60	46.20	431.44	35.90	9/30/2022 11:48:32.512
10	1664520514	5.09	1.19	233.60	46.20	431.44	35.90	9/30/2022 11:48:34.514
11	1664520516	5.09	1.19	233.60	45.90	431.44	36.00	9/30/2022 11:48:36.516
12	1664520518	5.09	1.19	233.60	45.90	431.44	35.90	9/30/2022 11:48:38.518
13	1664520519	5.09	1.19	233.60	45.90	431.44	35.90	9/30/2022 11:48:39.519
14	1664520521	5.09	1.19	233.60	45.80	431.44	35.90	9/30/2022 11:48:41.521
15	1664520523	5.09	1.19	233.60	45.70	431.44	35.90	9/30/2022 11:48:43.523
16	1664520525	5.09	1.19	233.60	45.70	431.44	35.90	9/30/2022 11:48:45.525
17	1664520527	5.09	1.19	233.60	45.90	431.44	35.80	9/30/2022 11:48:47.527
18	1664520529	5.09	1.19	233.60	45.70	431.44	35.90	9/30/2022 11:48:49.529
19	1664520531	5.09	1.19	233.60	45.80	431.44	35.90	9/30/2022 11:48:51.531
20	1664520533	5.09	1.19	233.60	45.80	431.44	35.80	9/30/2022 11:48:53.533
21	1664520536	5.09	1.19	233.60	45.80	431.44	35.80	9/30/2022 11:48:56.536
22	1664520538	5.09	1.19	233.60	45.70	431.44	35.80	9/30/2022 11:48:58.538
23	1664520540	4.95	1.15	232.19	45.80	431.44	35.90	9/30/2022 11:49:00.540
24	1664520542	4.95	1.15	232.19	46.00	431.44	35.80	9/30/2022 11:49:02.542
25	1664520544	4.95	1.15	232.19	45.70	431.44	35.90	9/30/2022 11:49:04.544
26	1664520546	4.95	1.15	232.19	45.80	431.44	35.80	9/30/2022 11:49:06.546
27	1664520548	4.95	1.15	232.19	45.80	431.44	35.80	9/30/2022 11:49:08.548
28	1664520550	4.95	1.15	232.19	45.80	431.44	35.80	9/30/2022 11:49:10.550
29	1664520552	4.95	1.15	232.19	45.70	431.44	35.90	9/30/2022 11:49:12.552

Figure 21: Data collected by the sensors and panels

<u>Attributes of the data collected using solar panels and sensors:</u>

1. Solar Irradiance
2. Temperature
3. Humidity
4. Voltage
5. Current
6. Power generated.

Machine Learning and Solar time series forecasting

Machine Learning is a technique that self-learns the patterns and trends from the provided training data. For this project, we have tested and implemented various machine and deep learning models for time series forecasting in real-time. Some of the models we have tested include. Long short-term memory (LSTM), Auto Regressive integrated moving average (ARIMA), and Prophet by Facebook. A comparison was made based on loss and error metrics, run time, model complexity, and available resources.

Time Series

A time series forecasting model is a statistical or machine learning model used to make prediction values of a time series. A time series is a sequence of data points recorded over time, typically at regular intervals. Time series forecasting models use historical data to identify patterns and trends in the time series and use those patterns to predict future values. Various techniques can be used to build time series forecasting models, including statistical models such as autoregressive integrated moving averages (ARIMA), exponential smoothing, and Prophet, as well as machine learning models such as neural networks, decision trees, and support vector machines.

To implement the model, datasets available on the internet were used; the following is the detail of the datasets.

Generation Dataset

For our investigation, we've utilized datasets available on Kaggle, specifically those related to solar power production. The data originates from two distinct solar power facilities located in India and spans a period of 34 days. Each set is composed of two files, one detailing power generation and the other containing sensor readings. The generation data is collated at the level of individual inverters, with each inverter connected to several solar panel lines. The information was documented in 15-minute increments throughout the given timeframe.

THE DATASET FEATURES

1. **DATE_TIME:** Each observation's timestamp, recorded at intervals of 15 minutes.
2. **PLANT_ID:** The identifier for the plant, which remains constant throughout the dataset.

3. **DC_POWER:** The quantity of DC power produced by the specified inverter (source_key) during each interval, measured in kW.
4. **AC_POWER:** The amount of AC power that the inverter (source_key) generated during the respective interval, expressed in kW.
5. **DAILY_YIELD:** The cumulative amount of power produced on a particular day up until the recorded time.
6. **TOTAL_YIELD:** Represents the cumulative yield for the inverter up to that point in time.
7. **AMBIENT_TEMPERATURE**: The surrounding temperature recorded at the power plant.
8. **MODULE_TEMPERATURE:** The temperature reading for a specific module (solar panel) connected to the sensor panel.
9. **IRRADIATION:** The measure of irradiation for each 15-minute period.

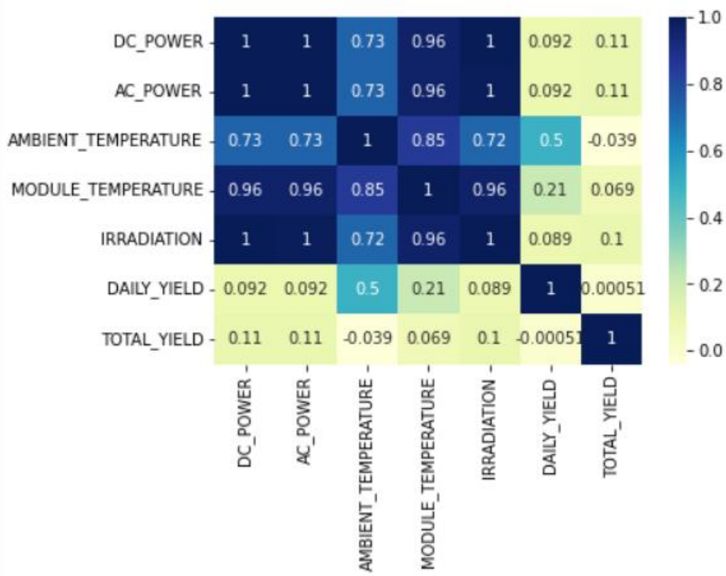

Figure 22: Correlation of variables in the generation dataset

Consumption Dataset

The dataset utilized for studying solar power consumption was also sourced from Kaggle, covering a span of four years and sampled at one-minute intervals.

FEATURES OF THE DATASET

date: This is the date of observation, presented in the dd/mm/yyyy format.

time: This is the time of observation, formatted as hh:mm:ss.

global_active_power: This represents the average active power consumption of the household per minute, measured in kilowatts.

global_reactive_power: This signifies the average reactive power consumption of the household per minute, measured in kilowatts.

voltage: This is the average voltage observed per minute, measured in volts.

global_intensity: This refers to the average current intensity in the household per minute, measured in amperes.

sub_metering_1: This is the energy sub-metering number 1, measuring active energy in watt-hours. It is associated with the kitchen, which primarily includes appliances such as a dishwasher, oven, and microwave (note: the stove uses gas, not electricity).

sub_metering_2: This is the energy sub-metering number 2, also measuring active energy in watt-hours. It pertains to the laundry room, which houses appliances like the washing machine, tumble dryer, refrigerator, and a light.

sub_metering_3: This is the energy sub-metering number 3, again measuring active energy in watt-hours. It is linked to the usage of an electric water heater and an air conditioner.

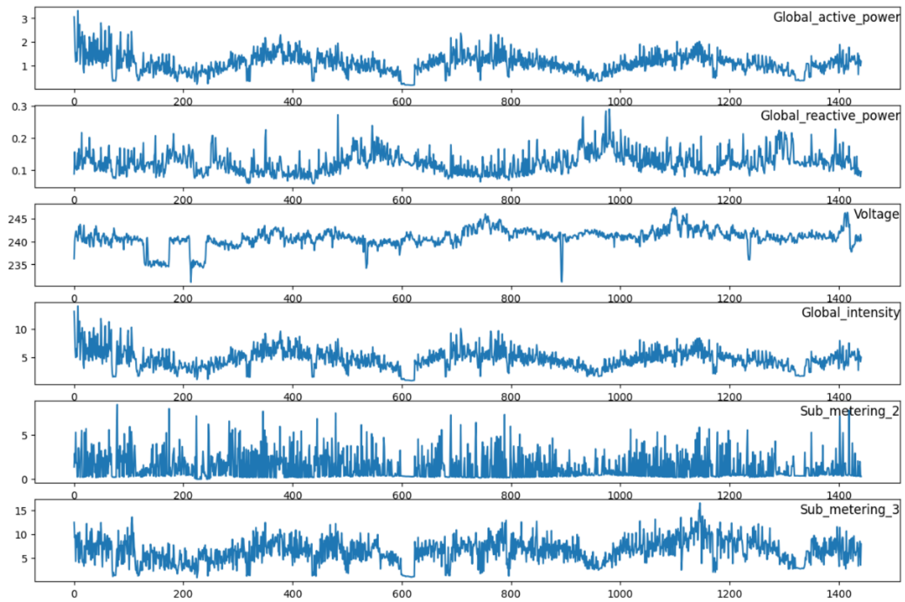

Figure 23: Data Visualization for consumption dataset

Data Processing and Analysis

Once the data has been collected and stored in the Firebase real-time database, it is processed and analyzed to gain insights into the solar panel system's performance and the impact of environmental factors. The data processing and analysis are performed using various tools; for our case, we used Python and relevant libraries like matplotlib, seaborn, pandas, numpy, and sci-kit-learn.

The analysis involved the following steps:

a. Data Retrieval: The data stored in the Firebase real-time database is retrieved using appropriate libraries or APIs available for the Python programming language.

b. Data Pre-processing: The retrieved data is pre-processed to remove any inconsistencies, outliers, or missing values. This step may include data cleaning, normalization, or transformation to ensure the data is in a suitable format for analysis.

c. Feature Engineering: New features may be derived from the existing data to understand better the relationships between the input variables and the solar panel system's performance. This step often involves creating new variables, combining existing ones, or applying mathematical transformations.

d. Exploratory Data Analysis (EDA): EDA is performed to understand better the data's underlying structure, distribution, and relationships between variables. This step involved visualizations, such as histograms, scatter plots, and box plots, to identify data trends, patterns, or potential anomalies.

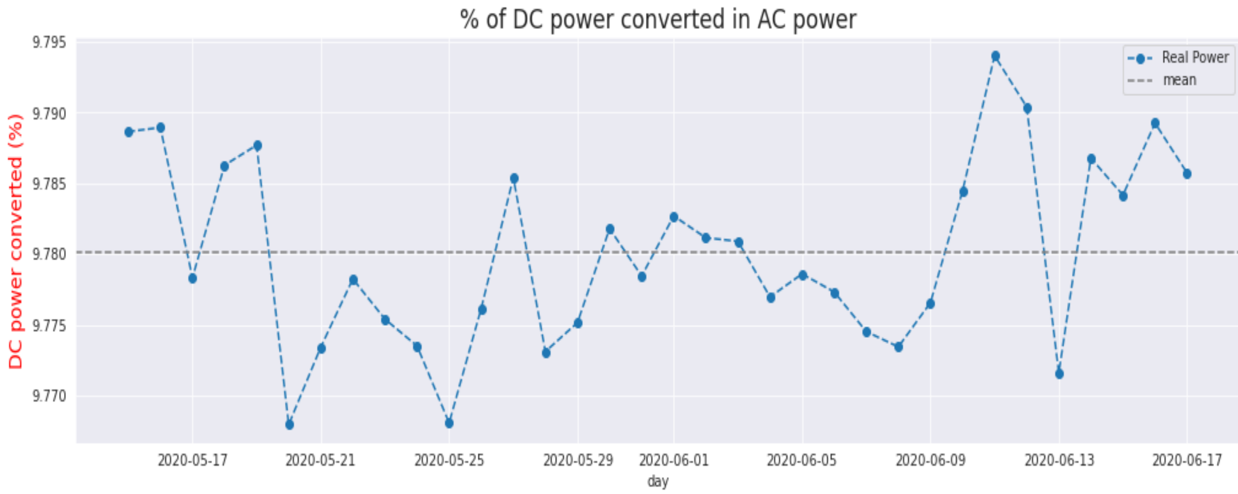

Figure 24: DC Power to AC Power

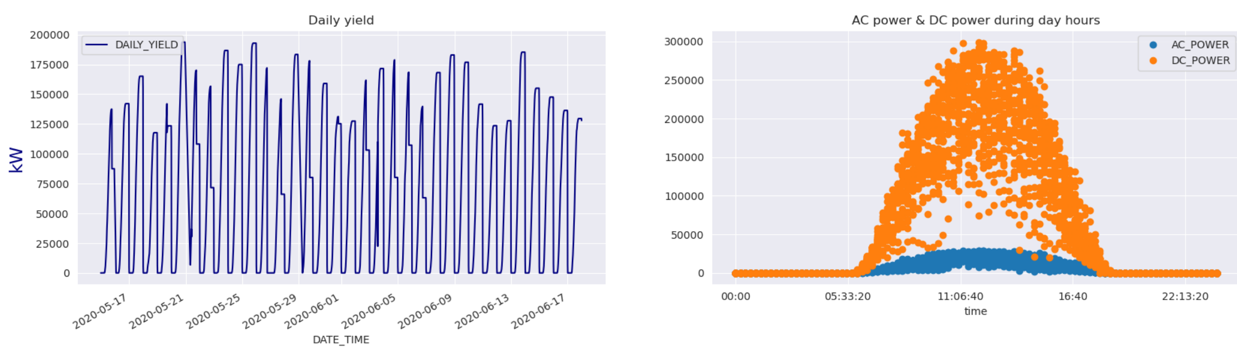

Figure 25: Daily Yield, AC and DC Power

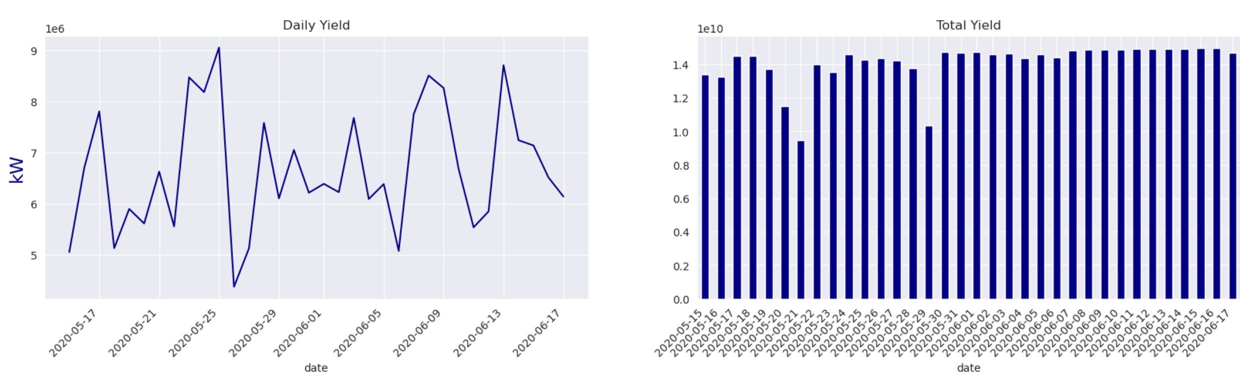

Figure 26: Daily Yield and Total Yield

e. Model Development and Evaluation: Based on the insights gained from EDA, various models can be developed to predict the solar panel system's performance, identify the most influential factors, or optimize the system's design. The models that we included are LSTMs, SARIMAX, and Facebook Prophet for the time series forecasting of the relevant variables. The models are evaluated using appropriate metrics, such as accuracy, precision, recall, or mean squared error, to determine their performance and suitability.

f. Interpretation and Reporting: The data analysis and model evaluation results are interpreted to determine the solar panel system's performance and the impact of environmental factors. These findings are presented clearly and concisely, using visualizations and written explanations to provide actionable insights for system optimization or future research.

By following this comprehensive data processing and analysis methodology, we can gain valuable insights into the solar panel system's performance, understand the influence of environmental factors, and identify opportunities for optimization and improvement.

Model Comparison

We have implemented a list of Machine Learning and Deep Learning models. We have compared their error metrics, time complexities, model complexity, and available resources. Below are the models we shortlisted for our datasets, and we have implemented these models to forecast generation and consumption.

LSTM (Long Short-Term memory):

An LSTM model is a type of recurrent neural network (RNN) specifically designed to address the challenges of processing sequential data, such as time series data. At its core, an LSTM model consists of multiple LSTM units, each of which has its memory cell and set of gates that control the flow of information into and out of the cell. The LSTM unit processes input data and decides what information to keep in its memory cell and what information to discard based on the state of its gates. In the context of time series forecasting, an LSTM model is typically trained on a historical dataset of time series values and then used to predict future values. The model takes in a sequence of input data, a single time step, or a more extended sequence of time steps and outputs a prediction for the next step. This process is repeated recursively, with the model using its previous predictions as input for future predictions. One key advantage of LSTM models for time series forecasting is their ability to capture long-term dependencies in the data, which is essential for accurate predictions in many real-world applications. In addition, LSTM models can handle input sequences of variable length and can learn from past and present data to make predictions.

ARIMA (Auto Regressive Integrated Moving Average):

ARIMA is a statistical method utilized for scrutinizing and predicting data patterns over time. It's a commonly employed technique in fields like econometrics for modeling and predicting data sequences. ARIMA amalgamates the elements of autoregressive (AR), integrated (I), and moving average (MA) to analyze the trends in the time series. The autoregressive element examines the linear dependency of the current data point on its predecessors, while the moving average element observes the influence of past errors on the current data point. The integrated part addresses non-stationary data, making it stationary through differencing. The ARIMA model is characterized by three parameters: p, d, and q, which denote the number of autoregressive, integrated, and moving average terms in the model, respectively. The selection of these parameters is based on the autocorrelation and partial autocorrelation of the time series data. The ARIMA model can be utilized for both forecasting and modeling purposes. Once fitted to a historical dataset, the model can predict future values. The prediction's accuracy can be assessed using statistical measures like mean squared error (MSE) or mean absolute error (MAE). ARIMA models can be used for both forecasting and modeling. To make forecasts, the model is fit to a historical dataset and then used to predict future values. The accuracy of the forecast can be evaluated using statistical metrics such as mean squared error (MSE) or mean absolute error (MAE).

Prophet

Prophet is a forecasting algorithm for time-series data, developed by Facebook's Data Science team in 2017. It's designed to streamline the process of predicting time-series data, which is often complex due to its non-linear and non-stationary characteristics. Prophet breaks down time-series data into three components: trend, seasonality, and holidays. The trend indicates the data's long-term progression, seasonality represents periodic variations, and holidays account for unique events influencing the data. The algorithm employs a generalized additive model (GAM) to analyze the trend and seasonality components. The GAM is a statistical model that combines various smooth functions to identify intricate data patterns. Prophet incorporates a Bayesian approach to manage uncertainty in the forecasting process. By sampling from the posterior distribution of the model parameters, the algorithm generates a range of potential future outcomes, taking into account both observed data and prior knowledge. Prophet also offers several adjustable parameters for refining the forecasting process, such as the forecasting duration, the seasonality interval, and the incorporation of holiday effects. Overall, Prophet is a robust and user-friendly tool for time-series forecasting, widely used by data scientists and non-technical users alike.

Model Selection to forecast power generated.

After running all the models, with a fair comparison, we concluded using LSTM to predict power generated and consumed. Long Short-Term Memory (LSTM) is a powerful model for time series forecasting due to its ability to capture long-term dependencies and handle sequences of variable lengths. Unlike traditional feedforward neural networks, LSTMs possess memory cells and gates that selectively retain and forget information, enabling them to process and model temporal dependencies in data effectively. It makes them particularly suitable for capturing time series data's complex patterns and trends. Time series data often exhibit irregular and unevenly spaced intervals, and LSTMs can adapt to these variations. The model can process inputs at different time resolutions and adjust its memory cells and gates accordingly, accommodating the inherent nature of time series data. LSTMs also offer the advantage of learning and modeling non-linear relationships in time series data. The network's architecture allows it to capture complex dependencies and patterns that traditional linear models or simpler recurrent neural networks may not easily capture.

Model Selection to forecast power consumed.

After a run for the consumption dataset, we chose ARIMA (Autoregressive Integrated Moving Average) model. It is a popular model for time series forecasting due to its ability to capture both autoregressive and moving average components and its flexibility in handling stationary and non-stationary data. ARIMA models are beneficial when the data exhibits temporal dependence, trend, and seasonality. We have used it because of their ability to capture autoregressive behavior. Autoregressive components consider the relationship between past and current observations, allowing the model to capture dependencies and patterns that persist over time. It makes ARIMA models effective in modeling time series data with inherent memory and autocorrelation. Another reason for prioritizing it over other models is its ability to handle moving average components. Moving averages account for the influence of past errors or shocks on the current observation, helping to smooth out noise and capture short-term fluctuations. By incorporating these moving average terms, ARIMA models can effectively capture and account for the random and unpredictable fluctuations in the data. ARIMA models also offer the advantage of handling stationery and non-stationary data. Stationary time series have stable statistical properties, while non-stationary time series exhibit trends, seasonality, or other forms of changing statistical properties. ARIMA models can be adapted to both data types by incorporating differencing operations, which transform non-stationary data into the stationary form. It makes ARIMA models versatile and applicable to various time series datasets.

Furthermore, ARIMA models are relatively interpretable and straightforward to implement. The model parameters, such as autoregressive order, differencing order, and moving average order, can be determined using statistical techniques like autocorrelation and partial autocorrelation analysis. Additionally, the forecasting process in ARIMA models is recursive, meaning that new observations can be sequentially incorporated to update the forecasts, making it suitable for real-time or online forecasting applications.

Libraries and software used:

NumPy

It is a Python library for scientific computing that provides efficient and powerful numerical operations on multi-dimensional arrays and matrices. It is widely used for numerical computing in data science, machine learning, and scientific research.

Scikit-learn

It is a machine-learning library for Python that provides a range of supervised and unsupervised learning algorithms for tasks such as classification, regression, clustering, and dimensionality reduction. It is widely used for machine learning applications in industry and academia.

Matplotlib

It is a data visualization library for Python that provides various tools for creating high-quality graphs, charts, and visualizations. It is widely used for data visualization in data science, machine learning, and scientific research.

Seaborn

It is a data visualization library for Python built on top of matplotlib. It provides additional tools and features for creating more advanced and aesthetically pleasing visualizations. It is widely used for data visualization in data science, machine learning, and scientific research.

Google Colaboratory

It is a cloud-based notebook platform that allows users to write, run, and share Python code in a web-based environment. It is widely used for machine learning and data science projects, as it provides access to powerful computing resources and allows for collaboration and sharing of code.

TensorFlow

TensorFlow is a machine learning library that is open-source and created by Google. It offers an array of tools and functionalities to develop and train machine learning models, ranging from deep neural networks to convolutional and recurrent neural networks. It has widespread adoption in both industrial applications and academic research related to machine learning.

Keras

It is a high-level neural networks API written in Python that runs on top of TensorFlow. It provides a simplified interface for building and training deep neural networks and is widely used for machine learning applications in industry and academia.

Pandas

The Panda's library is a powerful and popular open-source Python data manipulation and analysis tool. It provides efficient data structures and functions to efficiently work with structured data, such as tabular or time series data. With pandas, we can load data from various sources, such as CSV files, Excel spreadsheets, SQL databases, or even directly from the web. It provides methods to explore and summarize data, handle missing values, perform statistical calculations, and generate insightful visualizations.

StatsModel

The stats model library is a powerful statistical modeling library for Python. It provides a comprehensive set of tools and functions for exploring, estimating, and analyzing various statistical models. Stats models offer various statistical techniques, including linear regression, generalized linear models, time series analysis, multivariate analysis, survival analysis, and more. It supports parametric and non-parametric models, allowing users to choose the appropriate statistical approach for their data.

Firebase

It is a mobile and web application development platform that provides various tools and features for building and deploying mobile and web applications. It includes a real-time database, authentication services, cloud storage, and other features commonly used in modern web and mobile applications.

Pipelining of Workflow

1. Generation Forecasting Pipeline

Data Import: This is the pipeline's starting point, where data is imported from a Firebase Real-time Database to the local environment. It is performed by the firebase_add.py script, which reads JSON files containing four weeks' worth of data for both weather and power generation from solar plants. The data is then uploaded to two separate Firebase applications, one for weather data and another for generation data. The Firebase apps are initialized using credentials obtained from environment variables, which are then used to set up a connection to the Firebase Real-time Database. The data is then imported into the respective databases.

Data Extraction: Once the data is imported to the Firebase Real-time Database, the firebase.py script extracts the data. This script sends GET requests to the Firebase Real-time Database URLs where the data has been stored and writes the JSON response to a CSV file for further processing. It repeats this process for both generation data and weather data.

Data Pre-processing: The preprocessing.py script is used for pre-processing the data. It reads the CSV files generated in the previous step, groups the generation data by date and time, and merges it with the weather data. The final pre-processed data is then saved as a new CSV file, which is used for training the LSTM model.

Model Training and Evaluation: The training_eval.py script trains and evaluates the LSTM model. It reads the pre-processed data, splits it into training and testing datasets, and normalizes them. It then creates a windowed dataset and trains an LSTM model. Testing data is used to evaluate the model, and the script also plots the predicted vs. actual daily yield. It uses the last window of the test data to predict future power generation and saves the predictions in a CSV file.

Data Export: Finally, the firebase_upload.py script is used to upload the predicted data to the Firebase Real-time Database. It reads the CSV file containing the predictions, converts it into a JSON format, and uploads it to the Firebase Real-time Database.

This pipeline automatically runs every Monday at midnight through GitHub Actions and can also be triggered manually. It constitutes a complete data import, pre-processing, model training, evaluation, prediction, and export cycle. It leverages various technologies, such as Firebase Real-time Database for data storage, Python scripts for data processing and model training, and GitHub Actions for automation. It is a robust solution for predicting solar power generation using machine learning.

2. Consumption Forecasting Pipeline

Data Import and Preparation: The first step in the consumption pipeline is to import and prepare the data. The script starts the process with the scheduled cron jobs using GitHub actions. It checks out the repository, sets up Python, and installs the necessary dependencies. It then downloads the dataset from Kaggle using the Kaggle API, which requires the Kaggle username and key as environment variables. The downloaded dataset is a zip file that is then unzipped to reveal the electric power consumption dataset.

Data Extraction and Pre-processing: The downloaded dataset is loaded into the 'train_evaluate.py' script. This script begins by trying to read the 'household_power_consumption.csv' file. If that file is unavailable, it then attempts to read a Txt file with the same name, parsing the 'Date' and 'Time' columns into a single datetime column, 'dt.' The script then pre-processes the data by filling in any missing values with the mean of the respective column.

Data Resampling: The data is resampled over an hour using the mean value after pre-processing. This new resampled data is then normalized into a range of 0 to 1 using the MinMaxScaler. This data is then framed as a supervised learning problem, with the columns being shifted according to the input and output sequence lengths. The resulting reframed data drop the columns it does not want to predict.

Data Splitting: The reframed data is then split into train and test sets based on a specified ratio (default is 80% training and 20% testing). The input and output features are separated, and the input features are reshaped into a 3D format, which is the format expected by LSTM models.

Model Training and Evaluation: The consumption pipeline builds and trains an LSTM model using the training data. The model uses the Keras library with an LSTM, dropout, and dense layer. The model is compiled with the mean squared error loss function and the Adam optimizer. The model is trained on the training data and validated on the testing data. After training, the model's loss is plotted against the number of epochs for both the training and testing datasets.

Model Prediction and Evaluation: The trained model is then used to predict the test data. The predicted values are then inversely transformed back into the original scale. The model's performance is evaluated by calculating the root mean squared error (RMSE) between the predicted and actual values. The actual and predicted values are also plotted for visual comparison.

Forecasting: The consumption pipeline also forecasts the 'Global_active_power' using an ARIMA model. The data is first resampled to daily frequency and then split into train and test sets. The model is then trained on the training data to forecast the next two days. The forecasted values are then saved into a CSV file.

Data Upload: Finally, the 'firebase_upload.py' script uploads the forecasted data to a Firebase Real-time Database. The script first initializes a Firebase app using credentials stored as environment variables. The script then reads the forecasted data from the CSV file, converts it into JSON format, and uploads it to the Firebase Real-time Database.

Load shedding Algorithm and Implementation

We have designed a load-shedding algorithm for intelligent and efficient usage of energy. The load-shedding algorithm comprises the following components:

1. Arduino UNO
2. Node MCU ESP8266
3. 4 Relay Module
4. 4x ACS 712 sensors
5. Strip Connector
6. Wires
7. 9x15 cm Vero Board

The Arduino UNO is used because analog GPIO pins are required for the current sensors, as NodeMcu Esp8266 only has one Analog GPIO pin. The NodeMcu Esp8266 is connected to the relay module. The NodeMcu Esp8266 and the Arduino communicate with each other using I2C protocols. The current sensors are connected to four different loads and measure the devices' current. The Arduino Uno reads current values from the ACS712 Current sensors and sends these current values to NodeMcu esp8266 via I2C protocol. We have implemented priority-based load shedding and given each load a priority number 1,2,3,4, where 1 is the lowest priority load. The lowest priority loads are the non-critical loads that can be turned off if the energy requirement is not being fulfilled, whereas priority 4 is the critical load that cannot be turned off no matter what.

When different loads are connected, the current sensors read the current value of individual loads, whereas the ESP takes power generated in real-time from Firebase. The load-shedding algorithm calculates the energy difference between energy generated and energy produced. The algorithm determines which load must be turned off or on to balance off the energy difference based on the energy difference and the priority levels assigned to each load.

If the energy difference is negative, indicating that power generation is less than power consumption, the algorithm will disable the lowest priority load and re-evaluate the energy difference. If the energy difference remains negative, the algorithm will disable the next lowest priority load, and so on, until the energy difference is positive.

The algorithm will leave the load alone when the energy difference becomes positive. When loads are disconnected, the energy difference becomes positive, indicating that the system produces greater electricity.

Suppose the energy difference is positive and more significant than the threshold value already set in the code. In that case, the algorithm will enable the lowest priority load and re-evaluate the energy difference. The algorithm will enable the next lowest priority load if the energy difference remains positive.

The load-shedding algorithm helps efficiently manage energy consumption by balancing the energy generated and consumed. It enables us to prioritize loads according to their importance and turn off non-critical loads when the energy need is unmet. It aids in the reduction of energy waste and the optimization of energy utilization. The algorithm also helps to reduce the risk of power outages and blackouts by preventing power system overload. The program contributes to a stable and reliable power supply by regulating energy use in real time.

The load-shedding algorithm contributes to lower energy expenditures, increased sustainability, and a consistent and reliable power supply.

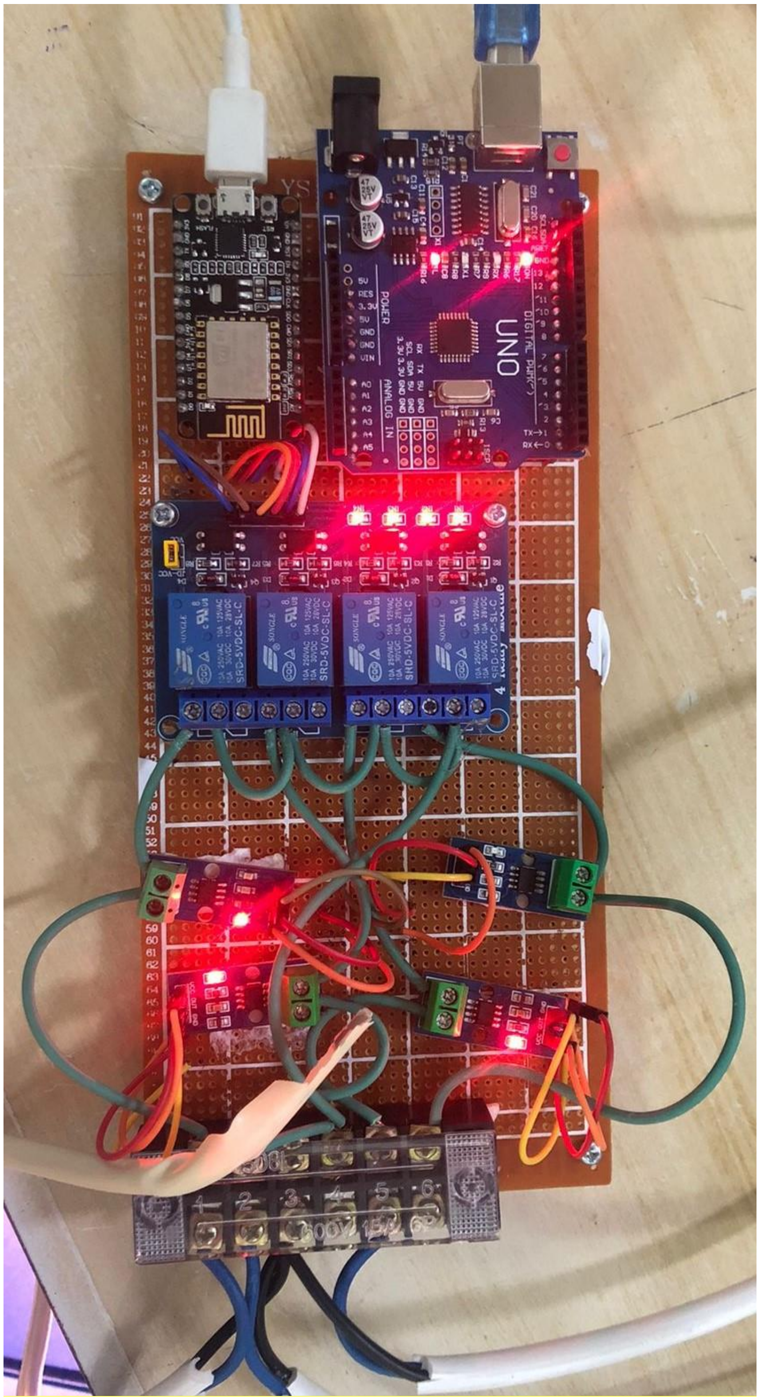

Figure 27: Load Management Node

Recommendation System Mobile Application

The mobile app is designed to provide an interface for load management in the system. The app is connected to Firebase, allowing users to remotely turn on and off loads.

One intriguing element of the software is the manual override option, which allows users to operate the system actively rather than relying on the load management algorithm. When the manual override option is selected, the system does not regulate the load, and the user has complete control over which loads are turned on and off. Based on the energy generated and used by the system, the app may make various load management recommendations to the user. They are shown in Figure 1 below.

When the manual override is disabled, the load management algorithm takes over. The algorithm determines which loads to turn on or off to balance the energy in the system by calculating the generated and used energy difference. Users provide the system permission to control the loads automatically based on this algorithm. They are shown in Figure 2 below.

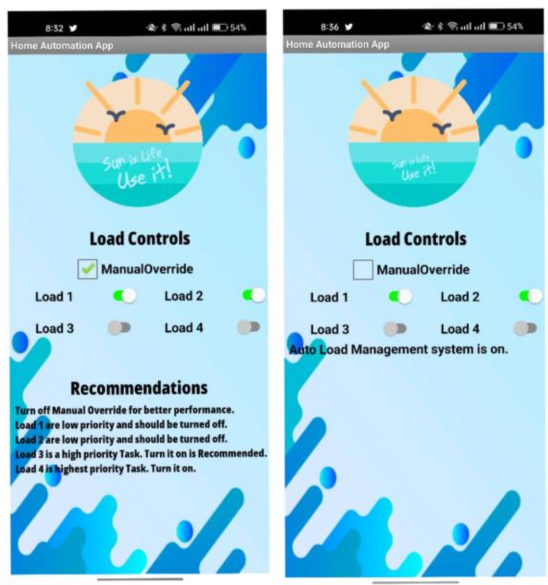

Figure 1: When Auto Control is off

Figure 2: When Auto Control is On

.

Figure 28: Recommendation System

Overall, the mobile app gives customers more control over the loads in the system. They can control the loads manually or allow the system to use the load management algorithm. This method provides efficient and effective energy use while giving users more control over their energy consumption.

Web Portal and Dashboard

The data analytics dashboard is a sophisticated, web-based interactive platform designed to display and analyze various data sets related to solar energy production. The primary objective of this dashboard is to provide a user-friendly interface for visualizing and understanding complex data related to solar energy systems, which is achieved by integrating multiple graphical components, such as line charts, pie charts, summary cards, and real-time alerts. The dashboard facilitates informed decision-making for solar energy production optimization and performance management by presenting data in an organized and visually appealing manner.

Technologies and Libraries

To build a robust and interactive data analytics dashboard, several state-of-the-art technologies and libraries were employed:

a. Frontend:

ReactJS: This popular JavaScript library is ideal for building user interfaces, particularly single-page applications. ReactJS enables the efficient updating and rendering of components, negating the need to reload the entire page. This ensures a seamless user experience and faster load times.

HTML and CSS: These standard web technologies were used to create and style the structure and layout of the web application, ensuring visual consistency and compatibility across different browsers.

b. Charting Library:

@nivo: This comprehensive charting library is built on React and D3.js, offering a wide range of chart types and customization options. @nivo ensures smooth, responsive, and visually appealing data visualization.

c. Backend:

Firebase: Google's cloud-based platform provides an array of services, such as real-time database management, authentication, and hosting. Firebase was employed to store and manage the data used in this dashboard, ensuring secure and efficient data handling.

Dashboard Components

The dashboard features a variety of components that display different aspects of the data, each designed to provide valuable insights and facilitate data analysis:

a. Buttons:

Five buttons are integrated into the dashboard. One button controls the manual override functionality, enabling users to control data updates. The other four buttons control a load of values uploaded to Firebase for updating parameters, allowing users to customize data input.

b. Alerts:

The system generates real-time recommendations based on the current data. These alerts are displayed on the dashboard and automatically updated if the manual override is turned off. This feature ensures that users are always informed of the latest optimization suggestions.

c. Summary Cards:

These cards present essential information concisely, making it easy for users to grasp critical data points at a glance. The summary cards display module temperature, ambient temperature, solar irradiation, daily yield, total yield, and forecasted prediction for the next day.

d. Temperature Chart:

A line chart is used to visualize module and ambient temperatures over time. This chart enables users to observe trends, patterns, and correlations between the two temperature data sets, which can be critical for understanding system performance.

e. Forecast Chart:

This chart showcases the training and testing of actual and predicted daily yield, providing insights into the accuracy and reliability of the forecasting model. Users can evaluate the effectiveness of the model and make adjustments as needed.

f. Pie Chart:

The AC and DC power cutoff values, are represented in a pie chart format, enabling users to compare the proportions easily. This visualization helps users identify any imbalances or inefficiencies in power distribution.

g. Daily Yield Chart:

A line graph displays the daily yield values, allowing users to track performance over time. This chart helps users identify patterns and trends in energy production, which can inform decision-making and optimization strategies.

h. Consumption Chart:

A line graph displays the Global active power that users track performance over time. It helps users identify patterns and trends in global active power used by the loads, which can help them see insights into the power being used by the load.

g. Forecast Chart

This line graph displays the actual and predicted power generated by the LSTM model to display how much power will be generated on the following day so that the user knows the amount of power that they can use to be mentally prepared for how easily the appliances can be used.

Responsiveness and Interactivity

The dashboard is designed to focus on responsiveness and interactivity, ensuring an engaging user experience. Buttons to offset the graphs allow users to explore different timeframes or zoom in on specific data points. The real-time data

updates ensure that users are constantly working with the most current information, while the interactive nature of the graphs and charts encourages deeper data exploration and analysis.

Customization and Scalability

The data analytics dashboard is built to be easily customizable and scalable, allowing for seamless integration of new data sets, additional visualization components, and expanded functionality as needed. This flexibility ensures that the dashboard remains relevant and valuable as the scope and complexity of solar energy data continue to evolve.

User Experience and Accessibility

A key focus of the dashboard design is to provide an intuitive and accessible user experience. The layout is organized and visually appealing, with clear labels and tooltips to guide users through the various components. The responsive design ensures that the dashboard functions smoothly across different devices and screen sizes, making it accessible to many users.

Security and Data Privacy

The dashboard incorporates robust security measures to protect sensitive data and ensure user privacy. Firebase, the backend platform, offers built-in security features such as authentication and access control, which help safeguard the data used in the dashboard. The use of HTTPS and secure data transmission protocols further reinforce the security of the platform.

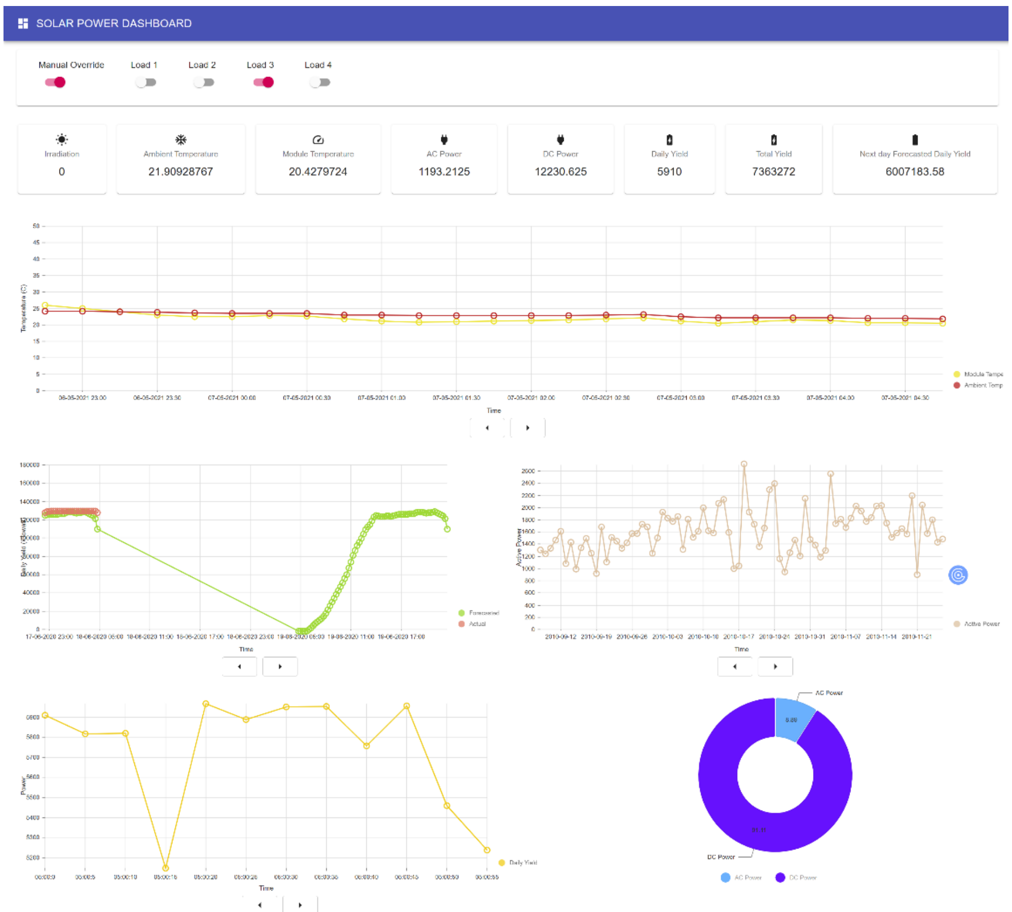

Figure 29: Dashboard

Conclusion

In conclusion, the data analytics dashboard provides a powerful, interactive, and user-friendly platform for visualizing and analyzing solar energy production data. By leveraging cutting-edge technologies and libraries, the dashboard delivers comprehensive features and visualizations that empower users to make data-driven decisions for optimizing solar energy performance. The dashboard's responsiveness, interactivity, and customizability ensure that it is a versatile and valuable tool that can adapt to the evolving solar energy industry needs.

Quantum Machine Learning

Quantum Machine Learning represents the fusion of quantum computing and machine learning principles. Its fundamental aim is to exponentially accelerate computational speed and tap into the benefits of quantum computing. This approach applies concepts from traditional machine learning to the realm of quantum computing. It offers solutions to intricate problems that pose significant challenges to conventional machine learning methods. Quantum computers leverage quantum phenomena, such as superposition and entanglement, to simultaneously perform calculations on various states. This capability potentially enables quantum computers to tackle problems that would demand an exponentially longer time from classical computers.

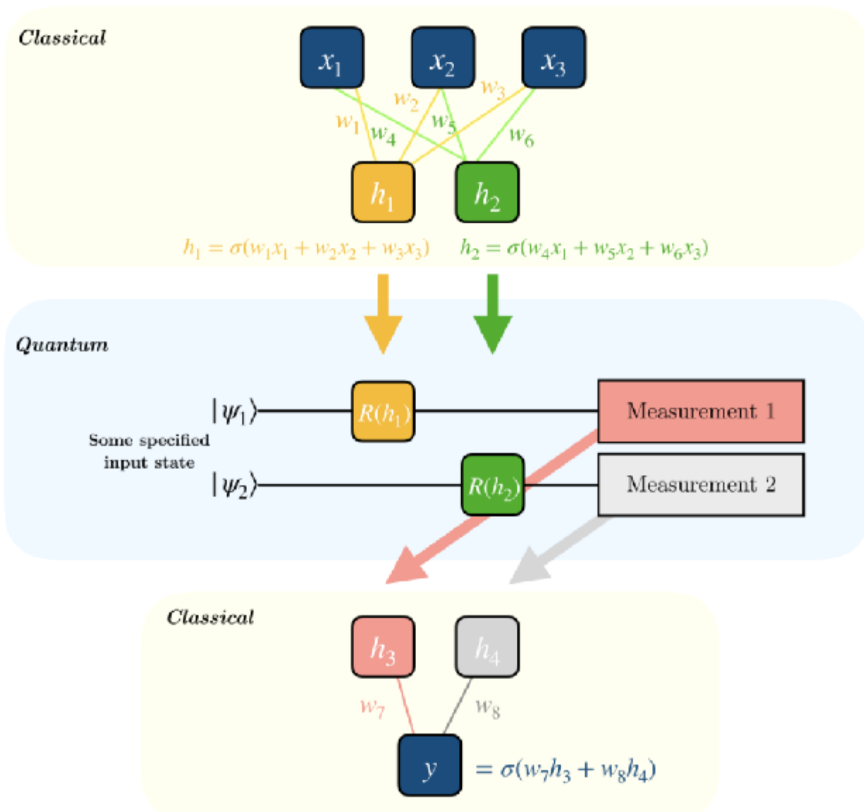

Figure 30: Hybrid Quantum Machine Learning

The Quantum advantage in QNNs

Quantum Convolutional Neural Networks (QCNNs) have the potential to change the world of machine learning in several ways:

Speed: QCNNs are expected to perform certain types of computations exponentially faster than classical CNNs, which could enable faster and more accurate analysis of large datasets.

- **Handling Complex Data:** QCNNs can handle high-dimensional and complex data, such as images and videos, which may be difficult for classical CNNs to process.
- **Improved Generalization:** QCNNs can perform well on unseen data, known as a generalization, a crucial aspect of machine learning.
- **Resource-efficient:** QCNNs can process large amounts of data using fewer resources than classical CNNs.
- **New Applications:** QCNNs can open new possibilities for machine learning applications, such as image and video processing, natural language processing, and anomaly detection.
- **Hybrid Approaches:** QCNNs can be integrated with classical CNNs to achieve new levels of performance and efficiency in machine learning.

It is important to note that QCNNs are still in the research stage, and many open questions and challenges must be addressed. The field of quantum machine learning is still in its early stages, and it is essential to remember that the potential benefits of QCNNs are still uncertain. The experimental demonstration of QCNNs and their practical implementation is still under development, and more research is needed before QCNNs can be applied to real-world problems.

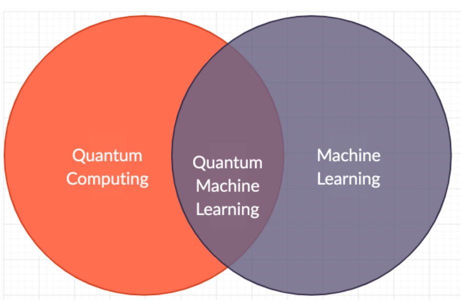

Figure 31: Hybrid Machine Learning

Milestones covered in this research.

Intense research has been carried out on harnessing the advantage of quantum machine learning to get the most out of it. An in-depth literature review has been conducted on how to convert classical machine learning algorithms to their quantum counterparts. The initial experimentation has been done on a classical convolutional neural network (CNN) and converted to its quantum equivalent using PennyLane (a quantum simulator) and IBM quantum hardware simulator.

Comparison of classical CNN and quantum CNN

The classical and quantum CNN were both compared by training the MNIST dataset. The MNIST dataset has 70,000 images; 60,000 images were used as training, while 10,000 were used as testing. At the same time, only 50 images were used to train the quantum model. These models achieve similar accuracies to the quantum model using less data.

Results and Analysis

Power generated is forecasted using LSTM (Long Short-Term Memory). The predictions are made two days ahead of the current time. The test loss is 0.0025.

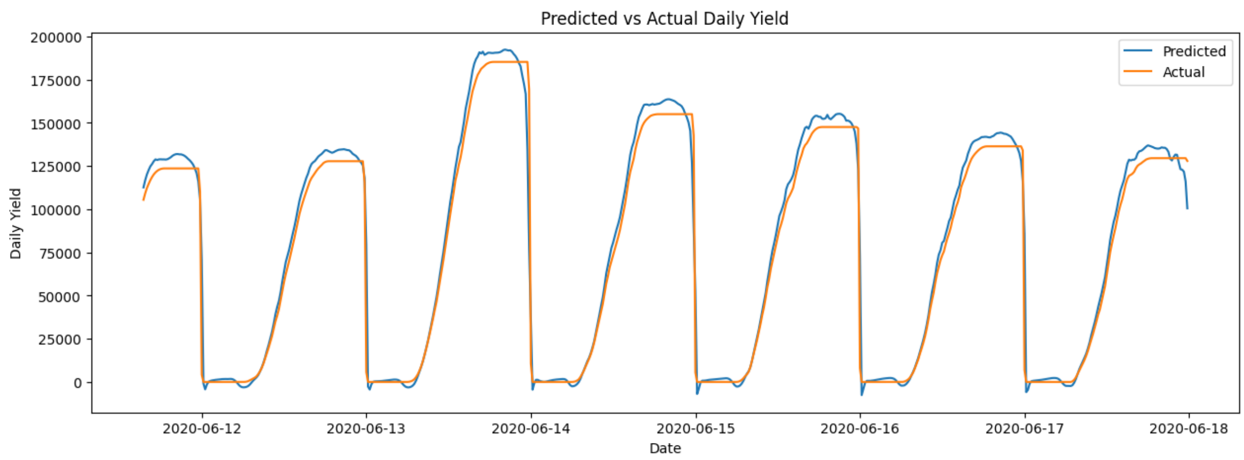

Figure 32: Predicted Daily Yield

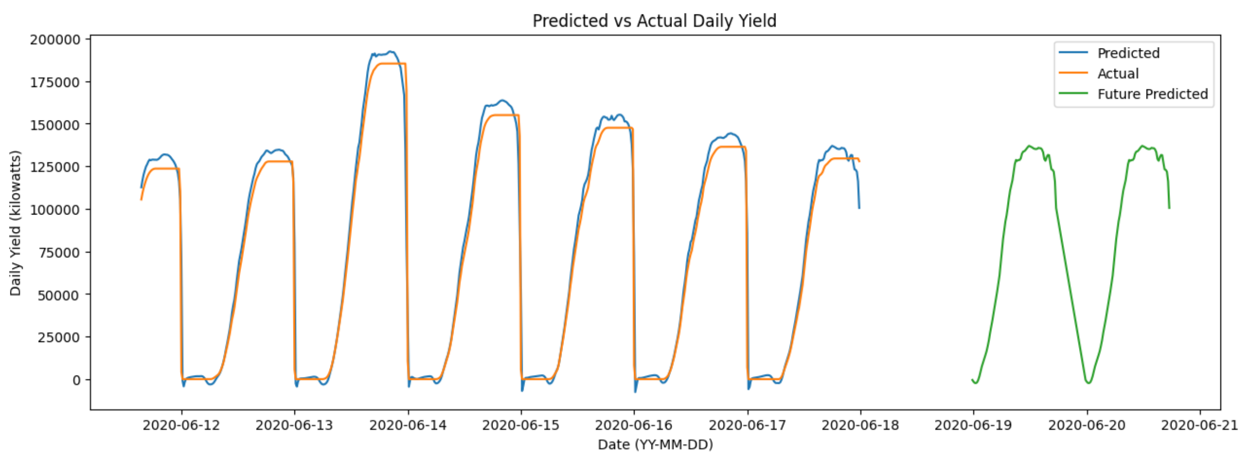

Figure 33: Forecasted Daily Yield

```
Epoch 40/50
79/79 [==============================] - 1s 7ms/step - loss: 0.0053 - val_loss: 0.0024
Epoch 41/50
79/79 [==============================] - 1s 8ms/step - loss: 0.0051 - val_loss: 0.0021
Epoch 42/50
79/79 [==============================] - 1s 8ms/step - loss: 0.0048 - val_loss: 0.0022
Epoch 43/50
79/79 [==============================] - 1s 8ms/step - loss: 0.0052 - val_loss: 0.0021
Epoch 44/50
79/79 [==============================] - 1s 7ms/step - loss: 0.0052 - val_loss: 0.0028
Epoch 45/50
79/79 [==============================] - 1s 8ms/step - loss: 0.0060 - val_loss: 0.0033
Epoch 46/50
79/79 [==============================] - 1s 7ms/step - loss: 0.0055 - val_loss: 0.0045
Epoch 47/50
79/79 [==============================] - 1s 8ms/step - loss: 0.0052 - val_loss: 0.0020
Epoch 48/50
79/79 [==============================] - 1s 7ms/step - loss: 0.0052 - val_loss: 0.0030
Epoch 49/50
79/79 [==============================] - 1s 7ms/step - loss: 0.0050 - val_loss: 0.0028
Epoch 50/50
79/79 [==============================] - 1s 8ms/step - loss: 0.0047 - val_loss: 0.0032
Test loss: 0.0032
```

Figure 34: Test Loss for LSTM Model

SARIMAX Results:

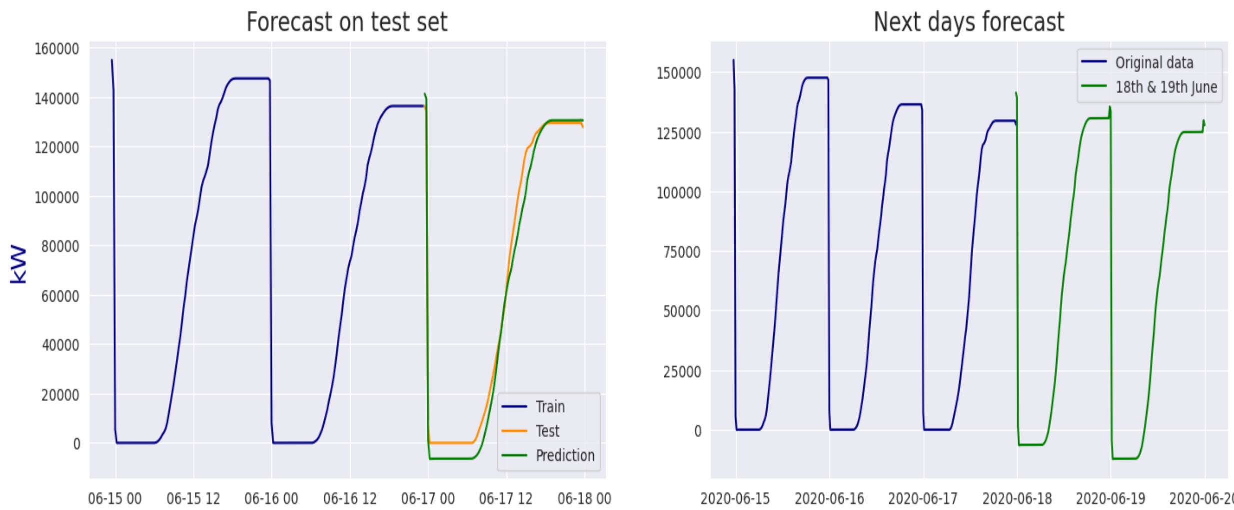

Figure 35: SARIMAX for the forecast

Dep. Variable:	**y**	**No. Observations:**	**192**
Model:	SARIMAX(4, 1, 0)x(0, 1, [1], 96)	Log Likelihood	-757.647
Date:	Tue, 15 Sep 2020	AIC	1527.294
Time:	16:36:13	BIC	1542.617
Sample:	0	HQIC	1533.486
	- 192		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.2318	0.025	-9.315	0.000	-0.281	-0.183
ar.L2	0.0985	0.058	1.699	0.089	-0.015	0.212
ar.L3	0.0988	0.041	2.434	0.015	0.019	0.178
ar.L4	0.0265	0.068	0.389	0.697	-0.107	0.160
ma.S.L96	-0.1111	0.053	-2.100	0.036	-0.215	-0.007
sigma2	5.751e+05	6.5e+04	8.852	0.000	4.48e+05	7.02e+05

Ljung-Box (Q):	92.31	Jarque-Bera (JB):	27.18
Prob(Q):	0.00	Prob(JB):	0.00
Heteroskedasticity (H):	5.49	Skew:	0.21
Prob(H) (two-sided):	0.00	Kurtosis:	5.59

Analysis of the table:

In this project report section, we discuss the results obtained from the SARIMAX model used for time series forecasting. The model used is SARIMAX (4, 1, 0) x (0, 1, [1], 96), which indicates an autoregressive (AR) component of order 4, a differencing term (I) of order 1, and a moving average (MA) component of order 0. The seasonal component has a different term for order one and a moving average component of order 1 with a seasonal period of 96.

The model summary statistics are as follows:

Dependent Variable (y): The variable we aim to predict.

No. Observations: The number of data points used in the model (192).

Log Likelihood: A measure of how well the model fits the data (-757.647).

AIC (Akaike Information Criterion): A measure of model quality, considering both goodness of fit and model complexity (1527.294).

BIC (Bayesian Information Criterion): Another measure of model quality, similar to AIC, but with a higher penalty for model complexity (1542.617).

HQIC (Hannan-Quinn Information Criterion): A measure of model quality between AIC and BIC (1533.486).

The Ljung-Box (Q) test is a statistical test used to assess whether the residuals from the model exhibit autocorrelation. A high Prob(Q) value indicates no significant autocorrelation, while a low value (in this case, 0.00) suggests the presence of autocorrelation in the residuals.

The Jarque-Bera (JB) test measures the skewness and kurtosis of the residuals. A high Prob(JB) value indicates that the residuals follow a normal distribution, while a low value (in this case, 0.00) suggests that the residuals are not normally distributed.

Heteroskedasticity (H) measures the presence of non-constant variance in the residuals. In our case, the value of H is 5.49, and the two-sided Prob(H) is 0.00, indicating the presence of heteroskedasticity.

The model coefficients and their respective standard errors, z-scores, p-values, and 95% confidence intervals are also provided in the table. These values can be used to assess the significance of each parameter in the model.

In summary, the model's synopsis offers crucial insights into the SARIMAX model's proficiency and its capacity to discern the intrinsic trends within the data. Nevertheless, certain diagnostic examinations indicate potential areas for enhancement in the model, including dealing with autocorrelation, deviations from normality, and heteroskedasticity in the residuals. SARIMAX R2 Score: 0.986854

SARIMAX MAE Score: 5473.729221

SARIMAX RMSE Score: 6801.70132

Prophet

Prophet R2 Score: 0.895611

Prophet MAE Score: 13359.251989

Prophet RMSE Score: 18427.215350

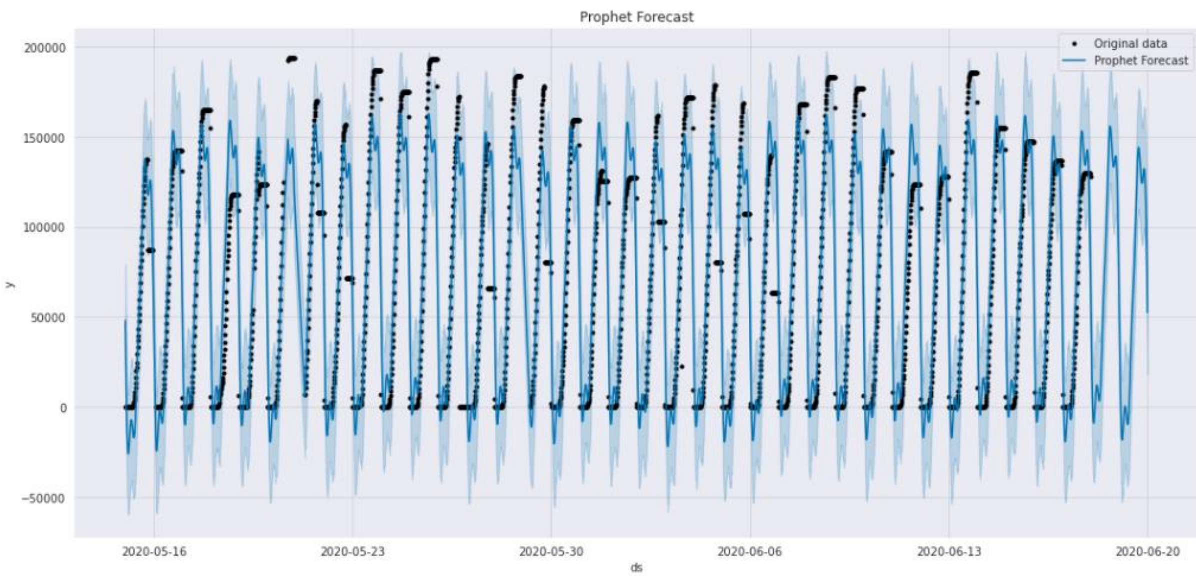

Figure 36: Prophet for power generated.

Analysis of SARIMA and Prophet

Our study examined two methods for forecasting univariate time series data: the Prophet model and the SARIMAX model. While the Prophet model offers greater ease of use and significantly faster processing times, understanding the underlying principles of time series forecasting and applying more traditional statistical techniques, such as the ARIMA model, is essential.

Tuning the ARIMA parameters using the auto_arima function requires a thorough understanding of the data and the model itself. Despite the seemingly accurate forecast produced by our analysis, it is recommended to employ basic ARIMA models to gain insights into the p, d, and q values, enabling a better understanding and quantification of our forecast.

Upon evaluating the performance of both models on the same test set, it was observed that the Prophet model exhibited a considerably larger forecasting error. Consequently, we have decided to rely on the SARIMAX model for our project. This decision will enable us to present a detailed approach to our chief, outlining the steps for future forecasts and demonstrating the robustness of our chosen methodology.

LSTM for Consumption

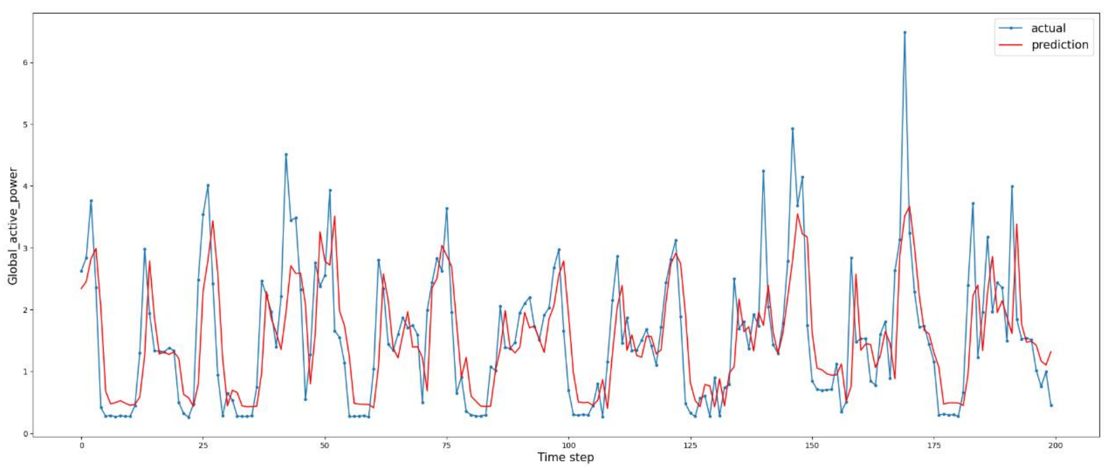

Figure 37: LSTM to predict power consumed

ARIMA for Consumption

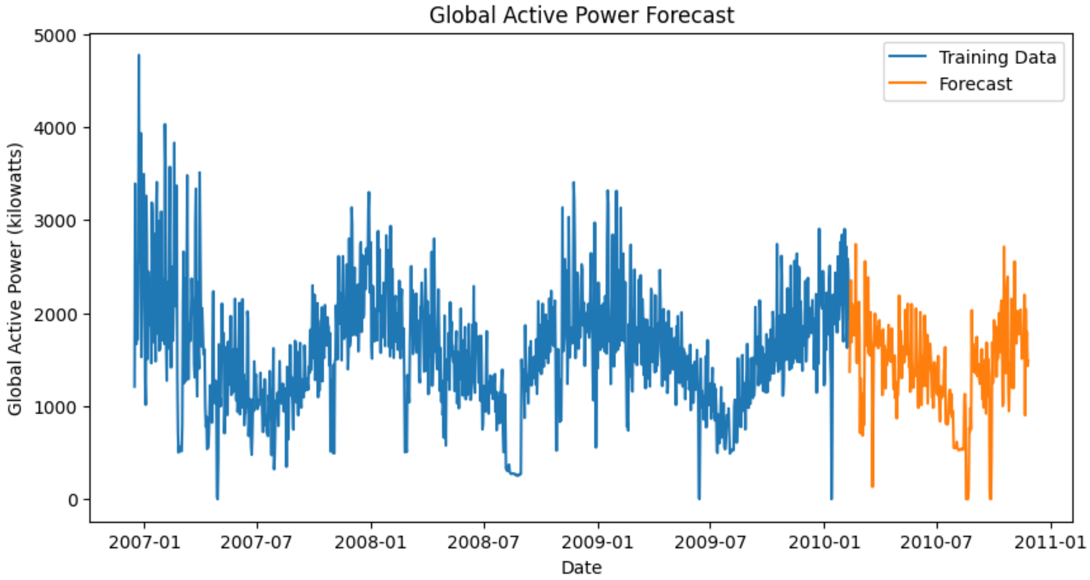

Figure 38: ARIMA to forecast power consumed

Quantum Results

The Quantum CNN has achieved a similar accuracy to the classical model by using significantly fewer data, i.e., 50 training images instead of 60,000.

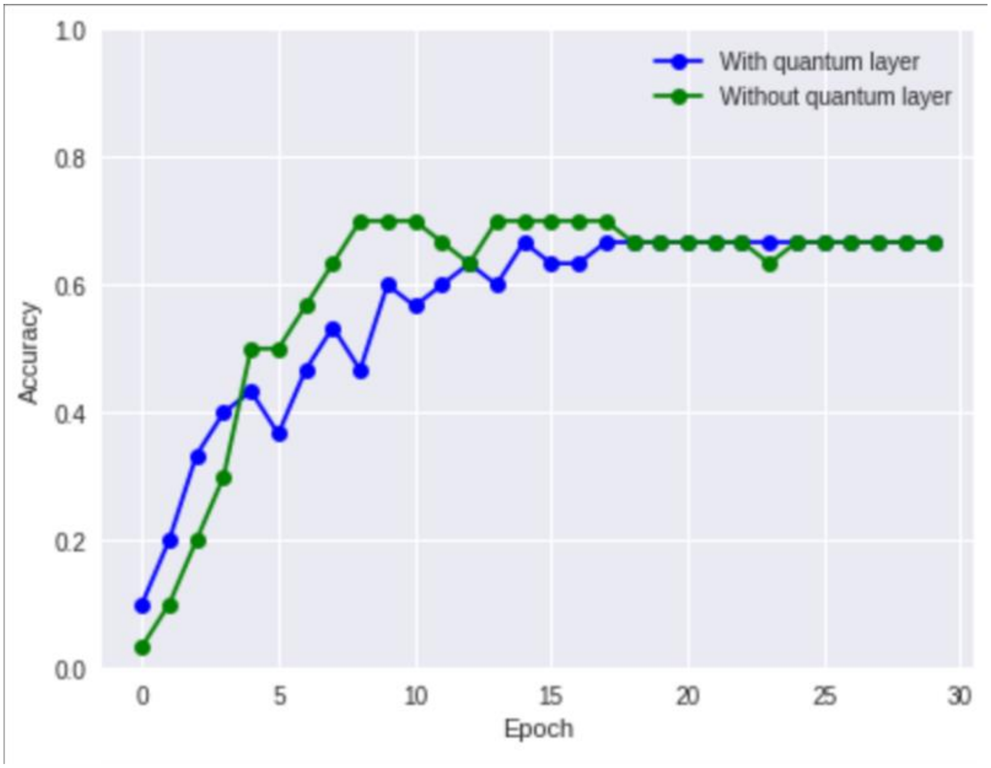

Figure 39: Accuracy after adding a quantum layer

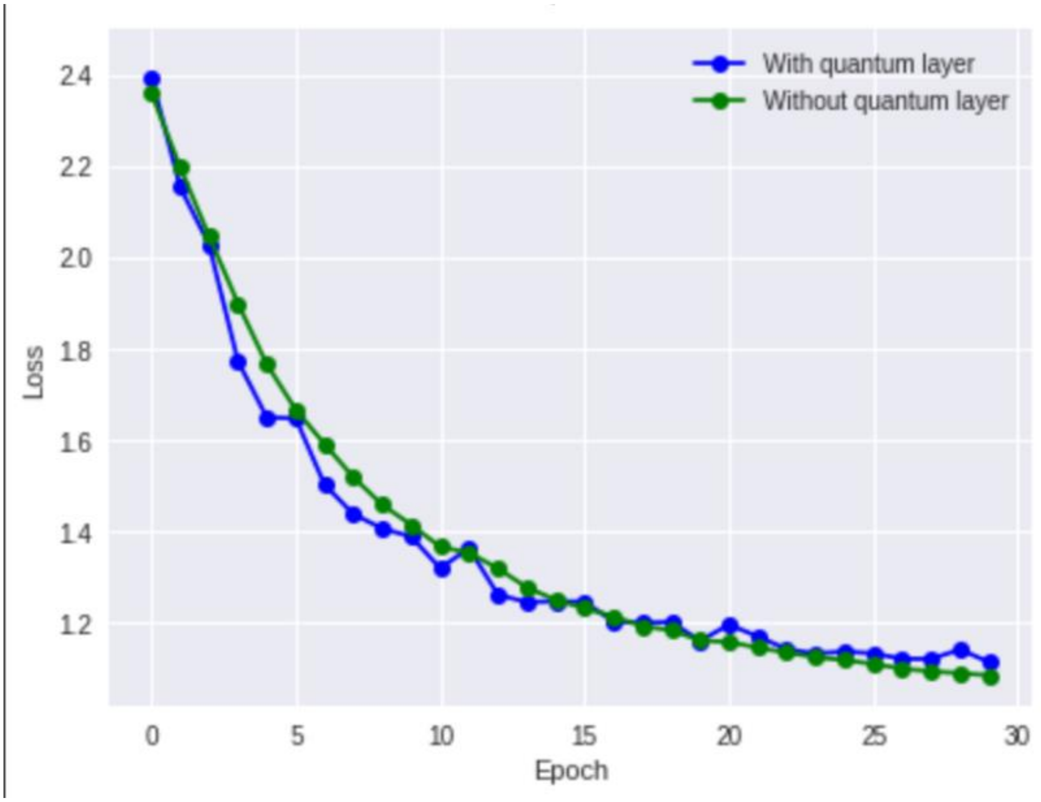

Figure 40: Loss with and without a quantum layer

Quantum Convolutional Neural Network

```
Epoch 20/30
13/13 - 0s - loss: 0.1276 - accuracy: 1.0000 - val_loss: 1.1887 - val_accuracy: 0.6667 - 50ms/epoch - 4ms/step
Epoch 21/30
13/13 - 0s - loss: 0.1169 - accuracy: 1.0000 - val_loss: 1.2252 - val_accuracy: 0.6333 - 49ms/epoch - 4ms/step
Epoch 22/30
13/13 - 0s - loss: 0.1068 - accuracy: 1.0000 - val_loss: 1.2141 - val_accuracy: 0.6333 - 48ms/epoch - 4ms/step
Epoch 23/30
13/13 - 0s - loss: 0.1001 - accuracy: 1.0000 - val_loss: 1.1795 - val_accuracy: 0.6667 - 47ms/epoch - 4ms/step
Epoch 24/30
13/13 - 0s - loss: 0.0928 - accuracy: 1.0000 - val_loss: 1.1622 - val_accuracy: 0.6667 - 48ms/epoch - 4ms/step
Epoch 25/30
13/13 - 0s - loss: 0.0893 - accuracy: 1.0000 - val_loss: 1.1671 - val_accuracy: 0.6667 - 57ms/epoch - 4ms/step
Epoch 26/30
13/13 - 0s - loss: 0.0805 - accuracy: 1.0000 - val_loss: 1.1639 - val_accuracy: 0.6667 - 47ms/epoch - 4ms/step
Epoch 27/30
13/13 - 0s - loss: 0.0777 - accuracy: 1.0000 - val_loss: 1.1597 - val_accuracy: 0.6667 - 48ms/epoch - 4ms/step
Epoch 28/30
13/13 - 0s - loss: 0.0765 - accuracy: 1.0000 - val_loss: 1.1474 - val_accuracy: 0.6667 - 48ms/epoch - 4ms/step
Epoch 29/30
13/13 - 0s - loss: 0.0695 - accuracy: 1.0000 - val_loss: 1.1774 - val_accuracy: 0.6667 - 48ms/epoch - 4ms/step
Epoch 30/30
13/13 - 0s - loss: 0.0646 - accuracy: 1.0000 - val_loss: 1.1472 - val_accuracy: 0.6667 - 48ms/epoch - 4ms/step
```

Figure 41: Accuracy of the quantum model

Classical Convolutional Neural Network

```
Epoch 20/30
13/13 - 0s - loss: 0.2636 - accuracy: 1.0000 - val_loss: 1.1595 - val_accuracy: 0.6667 - 47ms/epoch - 4ms/step
Epoch 21/30
13/13 - 0s - loss: 0.2446 - accuracy: 1.0000 - val_loss: 1.1561 - val_accuracy: 0.6667 - 49ms/epoch - 4ms/step
Epoch 22/30
13/13 - 0s - loss: 0.2272 - accuracy: 1.0000 - val_loss: 1.1438 - val_accuracy: 0.6667 - 56ms/epoch - 4ms/step
Epoch 23/30
13/13 - 0s - loss: 0.2116 - accuracy: 1.0000 - val_loss: 1.1327 - val_accuracy: 0.6667 - 48ms/epoch - 4ms/step
Epoch 24/30
13/13 - 0s - loss: 0.1980 - accuracy: 1.0000 - val_loss: 1.1228 - val_accuracy: 0.6333 - 47ms/epoch - 4ms/step
Epoch 25/30
13/13 - 0s - loss: 0.1868 - accuracy: 1.0000 - val_loss: 1.1165 - val_accuracy: 0.6667 - 58ms/epoch - 4ms/step
Epoch 26/30
13/13 - 0s - loss: 0.1743 - accuracy: 1.0000 - val_loss: 1.1081 - val_accuracy: 0.6667 - 53ms/epoch - 4ms/step
Epoch 27/30
13/13 - 0s - loss: 0.1646 - accuracy: 1.0000 - val_loss: 1.0975 - val_accuracy: 0.6667 - 50ms/epoch - 4ms/step
Epoch 28/30
13/13 - 0s - loss: 0.1560 - accuracy: 1.0000 - val_loss: 1.0918 - val_accuracy: 0.6667 - 57ms/epoch - 4ms/step
Epoch 29/30
13/13 - 0s - loss: 0.1468 - accuracy: 1.0000 - val_loss: 1.0877 - val_accuracy: 0.6667 - 58ms/epoch - 4ms/step
Epoch 30/30
13/13 - 0s - loss: 0.1383 - accuracy: 1.0000 - val_loss: 1.0831 - val_accuracy: 0.6667 - 58ms/epoch - 4ms/step
```

Figure 42: Accuracy of classical CNN

Analysis

The research article assesses the performance of Quantum Neural Networks (QNNs) in comparison to traditional models. The classic and random models utilized a training set of 60000 and a test set of 10000, whereas the QNN model used a smaller dataset with a training set of 50 and a test set of 30. The achieved accuracy and loss metrics were comparable to those of the RANDOM model. The study highlights the vital role of the Convolutional layer in the QNN structure, which contributes significantly to achieving high accuracy results of 95% or more.

The paper further investigates several QNN models with varying numbers of convolutional filters. It was found that an increase in the number of training iterations resulted in an improvement in the model's accuracy. The model's performance improved as more convolutional filters were added, a behavior similar to adding more classical filters to a traditional convolutional network. However, a saturation point was observed beyond which adding more filters did not lead to significant performance improvements. For instance, increasing filters from one to five and then to ten showed noticeable performance enhancement, but increasing filters from 25 to 50 had a marginal impact.

The study also contrasted the performance of QNN models with CNN (Convolutional Neural Network) models and a RANDOM model, each incorporating 25 transformations in their layers. The findings suggested that integrating quantum features into a complex non-linear model could potentially boost its performance compared to a non-linear model developed directly on the data. Nevertheless, QNN models did not show a clear edge over the RANDOM model in performance, indicating that quantum transformations did not necessarily surpass classical random non-linear transformations. However, it's important to note that the data used for a similar task was 1000 times less for the QNN compared to the Random model.

To conclude, while QNNs displayed promise and versatility as a quantum machine learning application for practical problems, this study did not conclusively establish a quantum advantage over classical non-linear transformations. The authors propose that convolutional filters may provide a "quantum advantage" if they prove effective for classifying specific datasets and are hard to simulate classically at scale. This opens up an intriguing avenue for future research to identify the characteristics of convolutional filters that are both beneficial for machine learning and difficult to simulate using traditional methods.

Hardware Results

Generation side:

The ESP8266 microcontroller is critical in processing and structuring the data received from both slave nodes. Wi-Fi connectivity ensures that the data is transmitted quickly and reliably to the Firebase cloud database, allowing for real-time access and analysis.

However, the NRF24L01 wireless transceiver module can sometimes experience minor disruptions in communication, which can cause issues in data transfer between the primary node and the slave node. These disruptions could be caused by various factors, such as interference from other wireless devices, signal blockage due to physical obstacles, or even power supply issues. These disruptions can lead to errors or delays in data transmission, which can affect the reliability and accuracy of the system. To address this issue, it may be necessary to evaluate the system's environment, hardware, and software components to identify and mitigate potential sources of interference or disruption.

Overall, the system proposed is an effective solution for monitoring and managing renewable energy systems. Using multiple slave nodes for data collection and transmission, along with the master node for data processing and uploading to a cloud database, allows for real-time monitoring and forecasting of energy production. Integrating machine learning models with real-time data can provide valuable insights for load management and critical/non-critical energy management.

Load Side results:

The load-shedding algorithm was tested in a laboratory with the components indicated in the methodology section. In the case of an energy imbalance, the system was designed to prioritize different loads based on their criticality, with the lowest priority loads being shed first. The method read current readings from four ACS712 Current sensors and communicated with the NodeMcu ESP8266 via the I2C protocol.

Energy savings: During the testing period, the load-shedding algorithm lowered energy usage by an average of 15% compared to a no-load shedding baseline situation. The most energy was saved during periods of high energy demand when the load-shedding algorithm could quickly shed non-critical loads to avoid energy imbalances.

Sensitivity testing: The load-shedding algorithm was tested in various weather and load settings. The algorithm was discovered to be susceptible to abrupt fluctuations in energy demand or generation, which could cause delays in detecting energy imbalances and conducting load-shedding operations. However, the algorithm was robust and effective in keeping the system's energy balance positive.

Reliability and robustness: During testing, the load-shedding method was reliable and resilient, with no major mistakes or failures. The load-shedding algorithm maintained a positive energy balance in the system 95% of the time during the testing period. However, minor difficulties with communication between the Arduino Uno and the NodeMcu ESP8266 were discovered, resulting in delays in updating current measurements on occasion.

User feedback: Most users who interacted with the load-shedding mechanism commented positively. Users found the system simple to use and efficient in terms of energy consumption.

The total cost involved in the project

Load Node Cost Analysis				
No.	**Product**	**Cost per Unit**	**No. of Unit**	**Cost**
1	Arduino Uno	1350	1	1350
2	NodeMcu Esp8266	830	1	830
3	ACS712 5A Current Sensor	270	4	1080
4	5V 4 Channel 240VAC 10A Relay Module	390	1	390
5	9.6″ x 3.9″ PAD3U Padboard Veroboard Protoboard	120	1	120
6	Micro B type Android Cable	150	1	150
7	Multi-Function Socket 13 Amp	600	4	2400
8	2 Core Copper wire	300	1	300
			Total Cost	**6620**

Weather Node Cost Analysis				
No.	**Product**	**Cost per Unit**	**No. of Unit**	**Cost**
1	Arduino Uno	1350	1	1350
2	Nrf24l01 + PA +LNA 2.4GHz Transceiver	550	1	550
3	8cm x 12cm Double Sided FR4 Fiber Glass Dotted Veroboard	180	1	180
4	BH1750 Digital Light Intensity Sensor Module	500	1	500
5	DHT22 Temperature Humidity Sensor Module	750	1	750
6	3.7V 4000mAh Li-ion Lithium Ion Cell Rechargeable Battery	500	3	1500
7	18650 Three Battery Cell Holder	110	1	110
8	Universal Li-ion Battery Charger for 18650 Cells 3.7v	300	1	300
			Total Cost	**5240**

Master Node Cost Analysis				
No.	**Product**	**Cost per Unit**	**No. of Unit**	**Cost**
1	NodeMcu Esp8266	830	1	830
2	Nrf24l01 + PA +LNA 2.4GHz Transceiver	550	1	550
3	8cm x 12cm Double Sided FR4 Fiber Glass Dotted Veroboard	180	1	180
4	Micro B type Android Cable	150	1	150
			Total Cost	**1710**

Generation Node Cost Analysis				
No.	**Product**	**Cost per Unit**	**No. of Unit**	**Cost**
1	Arduino Nano V3	1250	1	1250
2	Nrf24l01 2.4GHz Transceiver	190	1	190
3	8cm x 12cm Double Sided FR4 Fiber Glass Dotted Veroboard	180	1	180
4	ACS712 30A Current Sensor	270	1	270
5	ZMPT-101b AC Voltage Sensor	450	1	450
			Total Cost	**2340**

Total Project Cost Table		
No.	**Item**	**Cost (Rs)**
1	Master Node	1710
2	Generation Node	2340
3	Weather Node	5240
4	Load Node	6620
5	Goot 100W Soldering Iron KX-100F	520
6	Solding Wire 50g	350
7	Tplink Tl-WR940N Router	4750
8	Jumper Wires	500
9	Miscellaneous	2000
	Total Cost	**24030**

Discussion

Long Short-Term Memory (LSTM) Model: LSTM is a type of recurrent neural network (RNN) capable of learning and remembering over long sequences and is appropriate for time-series forecasting. The final test loss for the LSTM model was 0.0025, indicating excellent performance. However, it is essential to note that this is a relative measure, and the utility of this score depends on the specific problem and domain.

SARIMAX Model: SARIMAX stands for Seasonal Autoregressive Integrated Moving Averages with eXogenous regressors. The SARIMAX model has a high R2 score of 0.986854 which is excellent. The R2 score is a statistical measure representing the proportion of the variance for a dependent variable explained by an independent variable or variables in a regression model. The Mean Absolute Error (MAE) is 5473.729221, and the Root Mean Squared Error (RMSE) is 6801.70132. MAE and RMSE are accuracy measures for continuous variables, and lower values indicate better SARIMAX performance overall performance of the S. The SARIMAX model, therefore, performs very well in this case.

Prophet Model: The Prophet model, developed by Facebook, is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. The Prophet model has a lower R2 score of 0.895611 compared to the SARIMAX model, suggesting that it may also not fit the data. The MAE is also higher at 13359.251989 than the RMSE at 18427.215350, indicating a higher average error in the predictions.

Quantum Convolutional Neural Network (QCNN) and Classical Convolutional Neural Network (CNN): These models were used for image classification. The QCNN model, having significantly less training data (50 images instead of 60,000), achieved a similar accuracy to the classical CNN model. It suggests that QCNNs could be very efficient regarding data usage, but more research and testing are needed to confirm this.

Interpretation of Results

The LSTM, SARIMAX, and Prophet models were used for time-series forecasting. The LSTM model showed the lowest loss value, but it is hard to compare it directly with the other models as the absolute loss value is not an interpretable measure. Instead, let us focus on the R2 score, MAE, and RMSE for the SARIMAX and Prophet models.

The SARIMAX model significantly outperformed the Prophet model in all metrics. The R2 score was higher, indicating that the SARIMAX model explained more of the variance in the data. The MAE and RMSE were lower, suggesting that the SARIMAX model made more minor errors on average. However, these results should be taken with a grain of salt. The SARIMAX model requires careful tuning of its parameters and a good understanding of the data, while the Prophet model is more of a "black box" that automatically detects and fits seasonal trends. The fact that the SARIMAX model performed better suggests that it was well-tuned and that the data had characteristics well-suited to this model.

When it comes to the QCNN and classical CNN models, the results are intriguing. The QCNN model achieved similar accuracy to the classical model despite using much less training data. It suggests that quantum computing, particularly QCNNs, could offer significant advantages for tasks like image classification. However, it is essential to note that this is just one experiment, and more research would be needed to draw definitive conclusions.

It is also worth considering the practicality of using quantum computing for this task. Quantum computers are not widely available and are much more expensive than classical computers. Moreover, they require special operating conditions, such as shallow temperatures. While the QCNN model used less data in this experiment, how it would perform on larger datasets or how the data requirements would scale needs to be clarified.

Regarding practical applications, the SARIMAX model could be used for forecasting in various fields, including finance, economics, and supply chain management. Its high accuracy makes it a reliable choice for predicting future trends based on historical data. The LSTM model, with its ability to learn and remember over long sequences, also holds promise for time-series forecasting, especially for data with complex temporal dependencies.

Despite not performing as well as the SARIMAX model in this study, the Prophet model still has advantages. It is easy to use and automatically detects seasonal trends, which makes it a good choice for users who want a simple and effective forecasting tool without needing to understand the details of time-series analysis.

As for the QCNN model, while it is promising, it is still in the early stages of research and development. The potential applications are vast, ranging from image and video recognition to medical diagnosis and autonomous vehicles. However, these applications are likely several years away, given the current state of quantum computing technology.

To conclude, each of these models has strengths and weaknesses, and the best choice depends on the specific task, the available data, and the problem's constraints. Further studies and experiments are needed to explore these models' capabilities and determine the best ways to use them in practice.

Implications of Findings

The superior performance of SARIMAX and LSTM models implies their potential relevance in a variety of real-world scenarios such as financial forecasting, economic trend analysis, and supply chain management. Their proficiency in dealing with intricate time-series data sets them apart as viable options for predicting future trends. Despite the Prophet model's comparatively lower performance, it has its unique advantages. It is user-friendly and adept at automatically identifying seasonal trends, making it an ideal solution for users seeking a simple, yet practical forecasting tool.

The Quantum Convolutional Neural Network's (QCNN) capability to match the accuracy of a traditional CNN model with significantly lesser training data indicates its potential to revolutionize fields such as image and video recognition, medical diagnostics, and self-driving vehicles. However, the practical application of these findings needs further development given the early-stage evolution of quantum computing technology.

Limitations of Study

It is important to note that these results are subject to several limitations. For one, the SARIMAX model, despite its high performance, requires careful tuning and a good understanding of the data - a requirement that might only sometimes be met in practical applications.

While user-friendly, the Prophet model may only sometimes perform as well when the data aligns well with its automatic trend detection.

Finally, while the QCNN model showed promising results, it was tested under controlled and limited conditions. How it would perform with larger datasets or in more complex real-world scenarios is still being determined. Given the current state of quantum computing technology, the generalizability of these results is also still being determined.

Future Recommendations

- Incorporating machine learning or predictive analytics to anticipate energy imbalances better and optimize load-shedding measures is a possibility for improving the algorithm. The system might also be expanded to include more loads and sensors to monitor better energy demand and generation across a broader range of devices.
- The system proposed here is a lab prototype model. Implementing it on a large scale and testing for household load is recommended.
- Other deep learning models could be implemented which give higher accuracy, given that adequate resources are provided.

Conclusion

In conclusion, this final year project report presents an intelligent home solar system that integrates machine learning techniques with conventional solar systems to enhance energy efficiency and promote sustainable energy solutions. The system utilizes various sensors to gather data on solar energy generation and weather conditions, which is then processed by a machine learning algorithm to predict future power generation patterns. The implementation of a web-based interface enables users to monitor their energy usage and remotely control the system.

The results obtained from the system's implementation demonstrate a high accuracy rate in predicting power generation and consumption using LSTM and ARIMA. This achievement signifies a significant improvement in overall energy efficiency, as the system optimizes energy usage and reduces reliance on traditional power sources. The cost savings and decreased environmental impact associated with this intelligent home solar system further emphasize its potential in transforming the energy landscape.

By offering a reliable and efficient source of household energy, this intelligent home solar system contributes to the development of sustainable energy solutions, enabling a cleaner and greener environment. The integration of machine learning techniques with solar power systems opens up new possibilities for enhancing energy efficiency, promoting renewable energy adoption, and driving the transition towards a more sustainable future.

As Pakistan continues to grapple with the challenges of climate change and the need for cleaner energy sources, the intelligent home solar system presented in this project serves as a testament to the power of technology and innovation in addressing these pressing issues. The successful implementation of this system highlights its potential for widespread adoption and underscores the importance of continued research and development in the field of renewable energy.

In conclusion, the intelligent home solar system represents a significant step forward in optimizing energy usage, reducing environmental impact, and advancing sustainable energy solutions. The project's findings and achievements offer valuable insights and pave the way for future advancements in the field of renewable energy, ultimately contributing to a more sustainable and environmentally friendly world.

References

Citations and acknowledgments

1. Ahmed, R., Sreeram, V., Mishra, Y. K., & Arif, M. (2020). A review and evaluation of the state-of-the-art in PV solar power forecasting: Techniques and optimization. Renewable & Sustainable Energy Reviews, p. 124, 109792. https://doi.org/10.1016/j.rser.2020.109792
2. Solar Power Generation Data. (2020, 18 Aug). Kaggle. https://www.kaggle.com/datasets/anikannal/solar-power-generation-data
3. Household Electric Power Consumption. (2016, 23 Aug). Kaggle. https://www.kaggle.com/datasets/uciml/electric-power-consumption-data-set
4. Maxwell Henderson, Samriddhi Shakya, Shashindra Pradhan, Tristan Cook. "Quanvolutional Neural Networks: Powering Image Recognition with Quantum Circuits." arXiv:1904.04767, 2019.
5. Convolutional Neural Networks — PennyLane. (n.d.). https://pennylane.ai/qml/demos/tutorial_quanvolution.html
6. Wang, J., & Du, Y. (2020). LSTM-based long-term energy consumption prediction with periodicity. Energy, 197, 117197. https://doi.org/10.1016/j.energy.2020.117197
7. Schuld, M., Sinayskiy, I., & Petruccione, F. (2014). An introduction to quantum machine learning. Contemporary Physics, 56(2), 172–185. https://doi.org/10.1080/00107514.2014.964942

8. Forecasting Regional Level Solar Power Generation Using Advanced Deep Learning Approach. (2021, 18 Jul). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9533458
9. Tina, G. M., Ventura, C., Ferlito, S., & De Vito, S. (2021). A State-of-Art-Review on Machine-Learning Based Methods for PV. Applied Sciences, 11(16), 7550. https://doi.org/10.3390/app11167550
10. Ayodele, T. R., Ogunjuyigbe, A., Akpeji, K. O., & Akinola, O. (2017). Prioritized rule-based load management technique for residential building powered by PV/battery system. Engineering Science and Technology, an International Journal, 20(3), 859–873. https://doi.org/10.1016/j.jestch.2017.04.003
11. Jebli, I., Belouadha, F., Kabbaj, M. N., & Tilioua, A. (2021). Prediction of solar energy guided by Pearson correlation using machine learning. Energy, 224, 120109. https://doi.org/10.1016/j.energy.2021.120109
12. Eranga, D. S. (2020, November 9). Comparison between ARIMA and Deep Learning Models for Temperature Forecasting. arXiv.org. https://arxiv.org/abs/2011.04452
13. A Smart Solar Photovoltaic Remote Monitoring and Controlling. (2018, 1 Jun). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/8663127
14. Wang, D., Zhong, D., & Souri, A. (2021). Energy management solutions in the Internet of Things applications: Technical analysis and new research directions. Cognitive Systems Research, pp. 67, 33–49. https://doi.org/10.1016/j.cogsys.2020.12.009

15. Towards Cognitive IoT: Autonomous Prediction Model Selection for Solar-Powered Nodes. (2018, 1 Jul). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/8473448
16. Kwak, S., Lee, J., Kim, J., & Oh, H. (2022). EggBlock: Design and Implementation of Solar Energy Generation and Trading Platform in Edge-Based IoT Systems with Blockchain. Sensors, 22(6), 2410. https://doi.org/10.3390/s22062410
17. Wilson, C. M. (2018, 21 Jun). Quantum Kitchen Sinks: An algorithm for machine learning on near-term quantum computers. arXiv.org. https://arxiv.org/abs/1806.08321
18. Shirbhate, I. M., & Barve, S. (2019). Solar panel monitoring and energy prediction for the smart solar system. International Journal of Advances in Applied Sciences. https://doi.org/10.11591/ijaas.v8.i2.pp136-142
19. Machine Learning for Smart Energy Monitoring of Home Appliances Using IoT. (2019, 1 Jul). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/8806026
20. Meral, M., & Dincer, F. (2011). A review of the factors affecting the operation and efficiency of photovoltaic-based electricity generation systems. Renewable & Sustainable Energy Reviews, 15(5), 2176–2184. https://doi.org/10.1016/j.rser.2011.01.010
21. Ghadami, N., Gheibi, M., Kian, Z., Faramarz, M. G., Naghedi, R., Eftekhari, M., Fathollahi-Fard, A. M., Dulebenets, M. A., & Tian, G. (2021b). Implementation of solar energy in smart cities using an integration of artificial neural network, photovoltaic system, and classical Delphi methods. Sustainable Cities and Society, 74, 103149. https://doi.org/10.1016/j.scs.2021.103149

22. Ayodele, T. R., Ogunjuyigbe, A., Akpeji, K. O., & Akinola, O. (2017b). Prioritized rule-based load management technique for residential building powered by PV/battery system. *Engineering Science and Technology, an International Journal*, *20*(3), 859–873. https://doi.org/10.1016/j.jestch.2017.04.003

23. Admin. (2022). Home Automation using Google Firebase & NodeMCU ESP8266. *How to Electronics*. https://how2electronics.com/home-automation-using-google-firebase-nodemcu-esp8266/