TNSDC IBN NAAN MUDHALVAN

Artificial Intelligence

MEASURING ENERGY CONSUMPTION

Submitted By,

K.FAHMI HANAN

A.HARI AKSHANA

J.HARIPRIYA

S.R.DHANUSHIYA

S.P.DEVISREE

**Introduction:**

Energy consumption measurement involves quantifying the amount of energy used by a system, device, building, or process over a specific period. Energy can take various forms, such as electricity, natural gas, oil, or other fuels. Accurate measurement is essential for several reasons, including Energy Efficiency, Environmental Impact, Resource Management, Billing and Cost Control.

**Problem Statement:**

The problem statement for measuring energy consumption is to develop accurate and reliable methods for measuring the amount of energy used by a person, device, or system over a period of time. Energy consumption can be difficult to measure, especially for large buildings or industrial facilities. Additionally, energy consumption patterns can vary over time, making it difficult to get an accurate picture of energy.

**Objectives:**

The primary objective of this project is to design and implement an energy consumption monitoring system using Python. This system will enable real-time monitoring and analysis of energy usage, contributing to energy efficiency and cost savings.

**Scope:**

The scope of this project encompasses the following key aspects:

* + Measurement and monitoring of electricity consumption.
  + Data collection, storage, and analysis.
  + Real-time visualization and reporting.
  + Implementation of energy efficiency measures.

**Design Thinking:**

Empathize:

The team would start by interviewing homeowners to understand their needs and wants when it comes to measuring energy consumption. They would also ask homeowners about their current experiences with measuring energy consumption, and what they would like to see improved.

Define:

Based on the interviews, the team would define the problem that they are trying to solve. For example, they might define the problem as: "How can we develop a new method for measuring energy consumption in homes that is more accurate, reliable, and user-friendly?"

Ideate:

The team would then brainstorm possible solutions to the problem. They might come up with ideas such as a new type of energy meter that is easier to install and use, a new software application that helps users to understand and analyze their energy consumption data, or a combination of both.

Make informed decisions about our energy use:

For example, if we know that our home is using a lot of energy, we may decide to switch to a renewable energy provider or invest in solar panels.

Measuring energy consumption is also important for businesses and governments. Businesses can use energy consumption data to identify opportunities to reduce their energy costs and improve their environmental performance. Governments can use energy consumption data to develop energy policies and programs that promote energy efficiency and renewable energy.

**Goal:**

The goal of measuring energy consumption is to accurately and reliably determine the amount of energy used by a person, device, or system over a period of time. This information can be used to:

Identify areas for energy conservation:

By understanding how much energy is being used and where it is being used, we can identify areas where we can reduce our energy use.

Set energy efficiency goals:

Once we have identified areas for energy conservation, we can set goals to reduce our energy use. Measuring energy consumption over time can help us to track our progress and ensure that we are meeting our goals.

Verify the performance of energy efficiency measures:

When we implement energy efficiency measures, such as installing new insulation or upgrading to more efficient appliances, we can measure our energy consumption to verify that the measures are having the desired effect.

Make informed decisions about energy use:

 By having a good understanding of our energy consumption, we can make informed decisions about how to use energy more efficiently. For example, we may decide to switch to a renewable energy provider or invest in solar panels.

**System Architecture:**

1. Sensor Selection and Installation:

1.1 Sensor Types:

* + Electricity meters
  + Current clamps

1.2 Sensor Calibration:

Sensors will be calibrated according to manufacturer guidelines to ensure data accuracy.

2. Data Retrieval:

Python scripts will be developed to collect data from sensors via Modbus communication protocols.

3. Data Management:

3.1 Database Selection:

The system will utilize the MySQL database to store collected energy consumption data.

3.2 Database Schema:

The database schema will include tables for timestamped energy consumption data, sensor IDs, and other relevant metadata.

4. Data Logging:

A data logging mechanism will be implemented to record energy consumption data at regular intervals. Timestamps will be included for each data entry.

5. Data Visualization:

5.1 Visualization Tools:

* + Matplotlib
  + Seaborn
  + Plotly

5.2 Dashboard Development:

A web-based dashboard will be created using the Dash framework to display energy consumption trends, real-time data, and key metrics.

6. Data Analysis:

6.1 Data Analysis Tools:

* + Pandas
  + NumPy

6.2 Analysis Objectives:

* + Identify energy consumption trends.
  + Detect anomalies and irregularities.
  + Provide insights for energy efficiency improvements.

7. Alerts and Notifications:

An alerting system will be implemented to send email notifications when energy consumption exceeds predefined thresholds or when anomalies are detected.

8. Energy Reports:

Periodic reports will be generated in PDF format using the ReportLab library. These reports will summarize energy consumption data and provide insights for optimization.

9. Energy Efficiency Measures:

Based on data analysis, energy-saving strategies will be developed and implemented to reduce consumption and enhance efficiency.

11. Testing and Calibration:

Regular testing and calibration of sensors and data collection components will be conducted to ensure accurate data collection.

12. Maintenance and Sustainability:

A maintenance plan will be established to monitor the health of the system, including sensor maintenance and software updates. Sustainability considerations include code efficiency and resource optimization.

13. Feedback and Iteration:

Feedback from users and stakeholders will be collected regularly to drive continuous improvement in the system's performance and features.

**Code:**

import os

import pandas as pd

import plotly.express as px

from plotly.offline import init\_notebook\_mode, plot

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_squared\_error, r2\_score

init\_notebook\_mode(connected=True)

def load\_and\_process\_data\_in\_folder(folder\_path, energy\_column\_index=1):

csv\_files = [f for f in os.listdir(folder\_path) if f.endswith('.csv')]

dataframes = []

for file\_name in csv\_files:

file\_path = os.path.join(folder\_path, file\_name)

df = pd.read\_csv(file\_path)

df = df.drop\_duplicates()

df = df.dropna()

dataframes.append(df)

return dataframes

def calculate\_energy\_consumption(dataframes, energy\_column\_index=1):

mse\_values = []

r2\_values = []

dataset\_names = [] # Collect dataset names

for df in dataframes:

if df is not None and len(df.columns) > energy\_column\_index:

if len(df) <= 1:

print(f"Skipping dataset with too few data points for splitting: {len(df)}")

continue

dataset\_name = df.columns[1]

df['Datetime'] = pd.to\_datetime(df['Datetime'])

df['Year'] = df['Datetime'].dt.year

df['Month'] = df['Datetime'].dt.month

df['Day'] = df['Datetime'].dt.day

df['Hour'] = df['Datetime'].dt.hour

df['Minute'] = df['Datetime'].dt.minute

df.drop(['Datetime'], axis=1, inplace=True)

energy\_column = df.columns[energy\_column\_index]

X = df.drop([energy\_column], axis=1)

y = df[energy\_column]

# Traing the Model

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

if len(X\_train) == 0 or len(X\_test) == 0:

print(f"Skipping dataset with no data points for training/testing.")

continue

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

model = RandomForestRegressor(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

mse\_values.append(mse)

r2\_values.append(r2)

dataset\_names.append(dataset\_name)

print(f"Dataset Name: {dataset\_name}")

print(f"Mean Squared Error (MSE): {mse}")

print(f"R-squared (R2): {r2}")

# Calculate the energy consumption for the given dataset

total\_energy\_consumption = dataframe[energy\_column\_name].sum()

average\_energy\_consumption = ataframe[energy\_column\_name].mean()

peak\_energy\_consumption = dataframe[energy\_column\_name].max()

return {

'Total Energy Consumption': total\_energy\_consumption,

'Average Energy Consumption': average\_energy\_consumption,

'Peak Energy Consumption': peak\_energy\_consumption

}

return dataset\_names, mse\_values, r2\_values

def visualize\_energy\_consumption(dataframes,dataset\_names, mse\_values, r2\_values):

# Extract the energy statistics

labels = ['Total', 'Average', 'Peak']

categories = [f'Dataset {i+1}' for i in range(len(dataframes))]

total\_consumption = [calculate\_energy\_consumption(df)['Total Energy Consumption'] for df in dataframes]

average\_consumption = [calculate\_energy\_consumption(df)['Average Energy Consumption'] for df in dataframes]

peak\_consumption = [calculate\_energy\_consumption(df)['Peak Energy Consumption'] for df in dataframes]

# Create a Plotly grouped bar chart for energy consumption statistics

fig = px.bar(x=categories \* len(labels), # Duplicate categories for each set of bars

y=total\_consumption + average\_consumption + peak\_consumption,

color=labels \* len(dataframes), # Add color for each category

labels={'x': 'Datasets', 'y': 'Energy Consumption'},

title='Energy Consumption Statistics')

fig.update\_layout(barmode='group') # To group the bars by category

plot(fig)

# Create bar chart for MSE using Plotly

mse\_fig = px.bar(x=dataset\_names, y=mse\_values, labels={'x': 'Datasets', 'y': 'Mean Squared Error (MSE)'})

mse\_fig.update\_layout(title='MSE for Different Datasets')

plot(mse\_fig)

# Create bar chart for R2 using Plotly

r2\_fig = px.bar(x=dataset\_names, y=r2\_values, labels={'x': 'Datasets', 'y': 'R-squared (R2)'})

r2\_fig.update\_layout(title='R2 for Different Datasets')

plot(r2\_fig)

#Main Program

# Load and process data from a folder

folder\_path = "D:\\NM"

dataframes = load\_and\_process\_data\_in\_folder(folder\_path)

# Calculate MSE and R2 values for all datasets

dataset\_names, mse\_values, r2\_values = calculate\_energy\_consumption(dataframes)

#Virtualize the result

print(“Energy Consumption Report”)

visualize\_energy\_consumption(dataframes,dataset\_names, mse\_values, r2\_values)

**Description:** The interpreter used here is Jupyter Notebook.

1. Importing Libraries:

The code begins by importing necessary Python libraries for data processing, visualization, and machine learning. These include `os` for file operations, `pandas` for data manipulation, `plotly.express` for creating interactive visualizations, and various modules from the `sklearn` library for machine learning tasks like data preprocessing and regression.

2. Initializing Plotly:

`**init\_notebook\_mode(connected=True)`** initializes Plotly for use within a Jupyter Notebook, allowing you to create interactive plots.

3. Data Loading and Preprocessing:

**`load\_and\_process\_data\_in\_folder(folder\_path, energy\_column\_index=1)`**

This function loads multiple CSV files from a specified folder and preprocesses the data. It iterates through each CSV file in the folder, reads the data into a Pandas DataFrame, removes duplicates and missing values, and stores the resulting DataFrames in a list.

4. Energy Consumption Calculation:

**`calculate\_energy\_consumption(dataframes, energy\_column\_index=1)`**

This function calculates Mean Squared Error (MSE) and R-squared (R2) values for energy consumption predictions using a Random Forest Regressor model. It processes each DataFrame from the list of dataframes. For each DataFrame, it:

* Converts the 'Datetime' column to a datetime format and extracts features like year, month, day, hour, and minute.
* Splits the data into features (X) and the energy consumption column (y).
* Splits the data into training and testing sets (80% training, 20% testing).
* Standardizes the features using StandardScaler.
* Fits a Random Forest Regressor model to the training data.
* Predicts energy consumption on the test data.
* Computes and stores the MSE and R2 values for evaluation.
* The results are printed to the console.

5. Visualization of Results:

**`visualize\_energy\_consumption(dataset\_names,mse\_values, r2\_values)`**

This function creates bar charts using Plotly to visualize the MSE and R2 values for different datasets. It takes the names of datasets, the MSE values, and the R2 values as input and plots them using Plotly. Two separate bar charts are created - one for MSE and another for R2.

6. Main Code Execution:

The main code execution consists of:

* Specifying the folder path from which to load the data.
* Calling `load\_and\_process\_data\_in\_folder` to load and preprocess the data, resulting in a list of DataFrames.
* Calling `calculate\_energy\_consumption` to calculate MSE and R2 values for the loaded datasets.
* Calling `visualize\_energy\_consumption` to visualize the results with bar charts.

**Dataset:**

The hourly energy consumption dataset encompasses detailed records of energy consumption across 12 different companies. This dataset offers a comprehensive view of how energy usage fluctuates over time, with hourly granularity. It includes data points that track the amount of energy consumed by each company on an hourly basis, allowing for in-depth analysis of consumption patterns, trends, and potential insights into efficiency and demand management. By examining this dataset, researchers and analysts can gain a deep understanding of how energy consumption varies over time, potentially revealing opportunities for optimizing energy usage and reducing costs while ensuring a more sustainable and environmentally friendly approach to resource management within these 12 organizations.  
**Link: https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption**

**Challenges Due to Limited Expertise:**

It's unfortunate that we are currently unable to implement an alert system due to our limited knowledge in this area. We understand the importance of having alerts and appreciate the need for them, but our current skillset and expertise are not sufficient to create an effective alert system. We recognize that this may sound frustrating or disappointing, but it's an area we're eager to learn more about in the future. Until then, we would need assistance or guidance from someone with more experience in this specific domain in order to ensure that any alert system we develop is both effective and reliable.

**Summary:**

The project focuses on analyzing hourly energy consumption data from 12 different companies. The dataset provides a granular view of how energy consumption fluctuates over time, offering insights into usage patterns and opportunities for efficiency improvements. By thoroughly examining this data, the project aims to identify trends and potential strategies for optimizing energy consumption within these organizations. This analysis could have far-reaching implications, potentially helping these companies reduce costs and promote more sustainable energy practices, ultimately contributing to a more efficient and environmentally friendly approach to resource management.