*Project Report On*

# Smart Home Energy Consumption

## Big Data Project

*under Dr. Hao Cai*

**Submitted By**

**Hariprasad Sivapatham Anand: 202205341**
**Lakshmi Venkataramu: 202304395**

# Contents

# List of Figures

# List of Tables

# 1 Data Science Salary Analysis

## Abstract

Data science is a rapidly growing field, and salaries in this domain have become a crucial aspect for professionals and organizations alike. This project provides a comprehensive analysis of data science salary trends in 2023, focusing on key factors influencing compensation within the field. By utilizing big data analytics, this study examines variables such as job roles, experience, skills, and geographical regions to reveal patterns and disparities in salary distributions. The findings are visualized through an interactive dashboard, offering insights into salary trends and enabling professionals, employers, and job seekers to make informed decisions about career development and market positioning in data science.

# Data Overview

### Data Source

We have obtained a dataset named *Data Science Salaries from 2023* from the website Kaggle.com. Kaagle Link

NOTE: **All the analysis for the above dataset has been covered in report 1.**

# 2 Reasons for Changing the Dataset from Data Science Salary Analysis to Smart Home Energy Consumption IoT

1. **Relevance to IoT and Smart Technologies:**
   - The Smart Home Energy Consumption dataset aligns closely with current trends in IoT (Internet of Things) and smart technologies, which are more directly related to innovation in data science and technology.
   - This dataset enables the exploration of real-world IoT applications, showcasing how data science can optimize energy efficiency and promote sustainability.
2. **Practical and Societal Impact:**
   - Energy management is a critical global issue, with smart homes playing a significant role in reducing carbon footprints and improving energy efficiency.
   - Working on the Smart Home Energy Consumption dataset provides practical insights into solving real-world problems that have both environmental and economic significance, making it more impactfull compared to salary trend analysis.
3. **Complexity and Analytical Depth:**
   - The smart home dataset involves time-series data, which presents a greater analytical challenge compared to tabular salary data. It allows for the application of advanced methods like anomaly detection, seasonal trend analysis, and forecasting using machine learning models such as ARIMA and LSTM.
   - This shift provided an opportunity to explore diverse analytical techniques and machine learning algorithms beyond basic exploratory data analysis.
4. **Hands-On IoT Data Exploration:**
   - The smart home dataset encompasses diverse variables such as energy consumption, weather conditions, and occupancy, offering a richer dataset for exploration and modeling.
   - By analyzing energy consumption patterns and anomalies, the project delves deeper into IoT data-driven solutions.
5. **Limitations of the Original Dataset:**

- The Data Science Salary Analysis dataset focuses on a relatively straightforward problem of analyzing salary trends, which might not provide enough scope to demonstrate advanced data science methodologies or diverse applications.
- Insights derived from salary analysis are limited to descriptive and predictive analytics, with less focus on real-world operational challenges like those found in IoT-based energy systems.

6. **Interdisciplinary Application:**
    - The smart home dataset bridges multiple disciplines—IoT, energy management, environmental science, and machine learning—making it more interdisciplinary and demonstrating the broader applicability of data science techniques.
7. **Future Research and Career Relevance:**
    - Working on IoT-based smart home data is more aligned with future technological trends and innovations, which adds value to research and professional development in the fields of data science and IoT.
    - It showcases expertise in handling complex datasets and solving sustainability-related challenges, which is highly relevant in academia and industry.

The decision to switch from the Data Science Salary Analysis dataset to the Smart Home Energy Consumption IoT dataset was driven by the need to tackle a more complex, impactfull, and technically challenging problem. This change enhances the project's relevance, depth, and contribution to both personal growth and societal benefit.

# 3 Smart Home Energy Consumption Analysis

## 3.1 Abstract

Smart homes are becoming an integral part of modern living, combining convenience and energy efficiency through the use of IoT and intelligent devices. This project focuses on understanding energy consumption patterns within smart homes using a time-series dataset hosted on Kaggle. The dataset includes over 5 lakh records and captures critical variables such as energy usage, occupancy status, and environmental factors like temperature and humidity.

By leveraging advanced analytical techniques such as seasonal trend analysis, anomaly detection using MA, ARIMA, and LSTM, and correlation studies, we identified patterns and insights that highlight inefficiencies and opportunities for optimization. This report outlines our step-by-step methodology, key findings, and actionable recommendations to make energy usage in smart homes smarter and more sustainable.

# 4 Introduction

## What is a Smart Home?

Let's begin by understanding what a smart home is. A smart home integrates interconnected devices and sensors to automate various functions such as lighting, temperature control, and security. These systems leverage IoT technology, smart thermostats, and intelligent lighting systems to create a more convenient and efficient living environment.

But it's not just about comfort. Smart homes play a critical role in energy efficiency. They allow for real-time monitoring and control of energy usage, helping homeowners reduce costs and environmental impact. These technologies collectively promote sustainability, which is the driving force behind this project.

## Motivation

The motivation for this project stems from the increasing adoption of smart homes and the need to optimize their energy consumption. While smart homes are designed to save energy, inefficiencies can still exist. For instance, appliances may draw energy even when not in use, or heating and cooling systems may operate inefficiently. By analyzing energy consumption patterns, we can uncover these inefficiencies and recommend actionable solutions.

Our project focuses on answering questions like:

- How does energy usage vary across seasons and occupancy status?
- What are the patterns of individual appliances, and how do they contribute to overall energy consumption?
- Can anomalies or inefficiencies be detected and addressed proactively?

By addressing these questions, we aim to make smart homes truly intelligent in managing energy.

## 4.1 Data Overview

Dataset Link: Link

The dataset for this project is sourced from Kaggle and is specifically tailored for analyzing energy consumption in smart homes. With over 5 lakh records and 32 attributes, it provides a comprehensive view of energy usage, occupancy data, and environmental factors.

This is a time-series dataset, which means every record corresponds to a specific timestamp. This structure is ideal for analyzing trends, seasonal variations, and appliance-level usage over time.

Some of the key features include.

1. **Timestamp:** The foundation of the dataset, capturing the exact date and time of each observation.
2. **Energy Consumption:** The primary metric of interest, measuring real-time energy usage in kilowatts.
3. **Occupancy Status:** Indicates whether the home is occupied or unoccupied, helping us distinguish patterns tied to human activity.
4. **Environmental Variables:** Includes temperature, humidity, visibility, and more, providing external context for energy usage trends.

## 4.2 Feature Description

This dataset offers a rich platform for in-depth analysis and the opportunity to apply advanced analytical techniques to derive meaningful insights.

### Software Implementation

- Programming Language: Python
- Platform: Google Colab / Kaggle notebook
- Libraries Used:
  - `sklearn.metrics`: Provides metrics for evaluating machine learning models, such as mean absolute error.
  - `tensorflow`: A library for building and deploying machine learning and deep learning models.
  - `sklearn.preprocessing`: Includes tools for data normalization, scaling, and encoding.
  - `statsmodels.api`: A library for statistical modeling and hypothesis testing.
  - `statsmodels.tsa.arima.model`: Provides tools for time-series analysis, including ARIMA models.
  - `pandas`: A data analysis and manipulation library.
  - `numpy`: A library for numerical computations and array manipulation.
  - `warnings`: Used for managing warnings in Python code.
  - `matplotlib.pyplot`: A library for creating static, animated, and interactive visualizations in Python.

| Variable Name | Data Type | Non-Null Count |
|---|---|---|
| time | object | 503910 |
| use [kW] | float64 | 503910 |
| gen [kW] | float64 | 503910 |
| House overall [kW] | float64 | 503910 |
| Dishwasher [kW] | float64 | 503910 |
| Furnace 1 [kW] | float64 | 503910 |
| Furnace 2 [kW] | float64 | 503910 |
| Home office [kW] | float64 | 503910 |
| Fridge [kW] | float64 | 503910 |
| Wine cellar [kW] | float64 | 503910 |
| Garage door [kW] | float64 | 503910 |
| Kitchen 12 [kW] | float64 | 503910 |
| Kitchen 14 [kW] | float64 | 503910 |
| Kitchen 38 [kW] | float64 | 503910 |
| Barn [kW] | float64 | 503910 |
| Well [kW] | float64 | 503910 |
| Microwave [kW] | float64 | 503910 |
| Living room [kW] | float64 | 503910 |
| Solar [kW] | float64 | 503910 |
| temperature | float64 | 503910 |
| icon | object | 503910 |
| humidity | float64 | 503910 |
| visibility | float64 | 503910 |
| summary | object | 503910 |
| apparentTemperature | float64 | 503910 |
| pressure | float64 | 503910 |
| windSpeed | float64 | 503910 |
| cloudCover | object | 503910 |
| windBearing | float64 | 503910 |
| precipIntensity | float64 | 503910 |
| dewPoint | float64 | 503910 |
| precipProbability | float64 | 503910 |

Table 1: Dataset Features

- `matplotlib.dates`: Provides tools for working with date-based data in Matplotlib.
- `utils`: A custom module for utility functions such as saving plots and metrics.
- `sklearn.decomposition`: Tools for dimensionality reduction, such as PCA.
- `sklearn.cluster`: Implements clustering algorithms like Agglomerative Clustering.
- `fcmeans`: A library for fuzzy c-means clustering.
- `collections.Counter`: A tool for counting hashable objects.
- `scipy.cluster.hierarchy`: Provides tools for hierarchical clustering.
- `scipy.spatial.distance`: Tools for distance computations in clustering.
- `sklearn.mixture`: Includes Gaussian Mixture models for clustering.
- `io.StringIO`: Handles string-based file-like objects.
- `seaborn`: A statistical data visualization library built on Matplotlib.
- `os`: A standard library for interacting with the operating system.

# 5 Task 1: Data Preprocessing and Correlation Analysis

## 5.1 Data Prepossessing

To prepare the dataset for analysis, we followed these prepossessing steps:

1. **Datetime Conversion:**
   - We converted Unix timestamps into human-readable datetime formats for easier interpretation and time-series analysis.
   - For example, 1700224800 corresponds to 2016-01-01 05:00:00.
2. **Resampling:**
   - The data was resampled to hourly intervals to standardize irregular entries. Resampling helps smooth variations and aligns the dataset for time-series analysis.
3. **Feature Engineering:**
   - From the datetime column, we extracted new features such as hour, day, and month. This added depth to our analysis by capturing cyclical trends in energy usage.
4. **Combining Features:**
   - Redundant columns, like Kitchen 12, Kitchen 14 and Kitchen 38, were aggregated into a single Kitchen column to reduce noise and complexity.
5. **Encoding Categorical Variables:**
   - Categorical features like "occupancy" and "weather conditions" were transformed into numerical formats using one-hot encoding. This step ensures compatibility with machine learning algorithms.

These preprocessing steps ensured that the dataset was clean, structured, and ready for meaningful analysis.

## 5.2 Correlation Analysis

Correlation analysis is a fundamental step in understanding the relationships between variables in our dataset. The heatmaps visually represent the strength and direction of the correlations between different features, with values ranging from -1 to 1:

- 1: Strong positive correlation (as one variable increases, the other also increases).
- -1: Strong negative correlation (as one variable increases, the other decreases).
- 0: No correlation between variables.

### 5.2.1 Energy Correlation Analysis

This heatmap focuses on the correlations between energy consumption variables like appliances, occupancy, and overall energy usage.

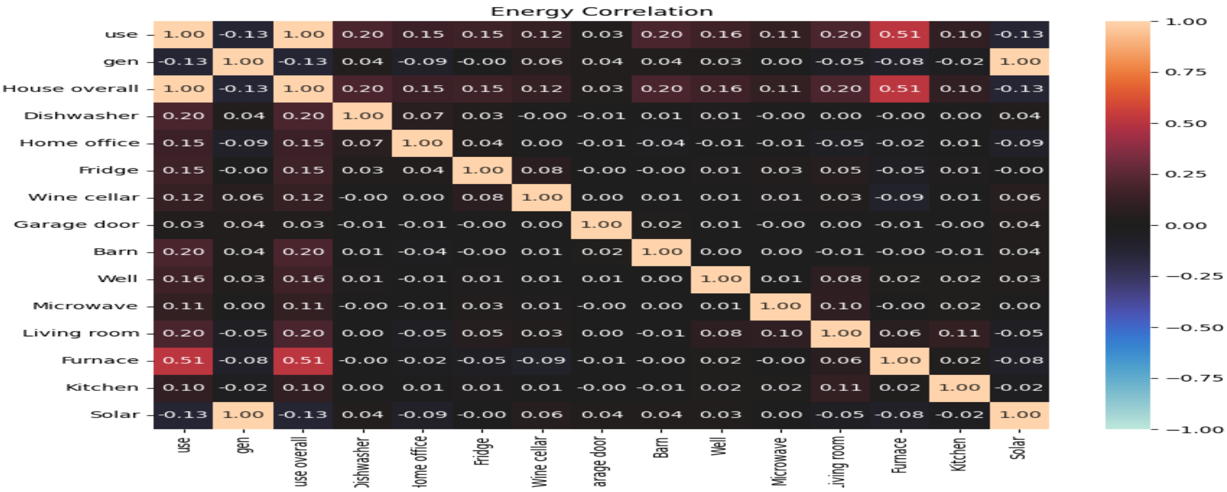


Figure 1: Energy Corelation Analysis

### Key Observations

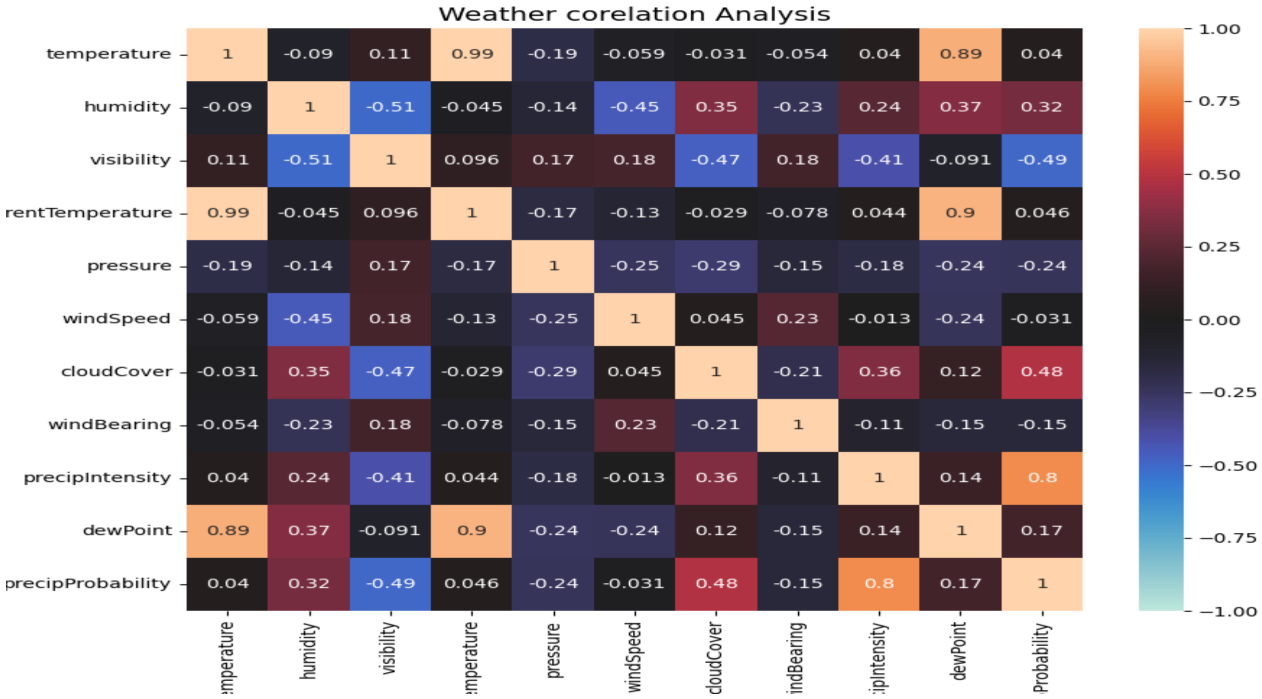
1. **Strong Correlations:**
   - "Furnace" and "House overall" have a correlation of 0.51, showing that the furnace contributes significantly to the total energy usage of the house.
   - Appliance-level variables like Kitchen, Living room, and Fridge show low correlation values, indicating independent energy usage patterns for most devices.
2. **Weak or No Correlation:**
   - Many appliances like Garage Door and Microwave show weak correlations with other variables, suggesting their usage patterns are more random or unrelated to other energy behaviors.

### Insights

- **Redundancy Check:** Variables with high correlations, such as "Use" and "House Overall, overlaps in functionality or data representation. The duplicate column "Use" was removed. Similarly "Gen" and "Solar" had high correlations with overlapping functionality, so the variable "Gen" was removed.
- **Independence:** Weak correlations among appliances suggest they contribute uniquely to energy consumption and can be analyzed individually.

### 5.2.2 Weather Correlation Analysis

This heatmap explores the relationship between weather-related variables (temperature, humidity, visibility, wind speed, etc.) and their influence on energy consumption patterns.



| | emperature | humidity | visibility | emperature | pressure | windSpeed | cloudCover | indBearing | ipIntensity | dewPoint | Probability |
|---|---|---|---|---|---|---|---|---|---|---|---|
| temperature | 1 | -0.09 | 0.11 | 0.99 | -0.19 | -0.059 | -0.031 | -0.054 | 0.04 | 0.89 | 0.04 |
| humidity | -0.09 | 1 | -0.51 | -0.045 | -0.14 | -0.45 | 0.35 | -0.23 | 0.24 | 0.37 | 0.32 |
| visibility | 0.11 | -0.51 | 1 | 0.096 | 0.17 | 0.18 | -0.47 | 0.18 | -0.41 | -0.091 | -0.49 |
| rentTemperature | 0.99 | -0.045 | 0.096 | 1 | -0.17 | -0.13 | -0.029 | -0.078 | 0.044 | 0.9 | 0.046 |
| pressure | -0.19 | -0.14 | 0.17 | -0.17 | 1 | -0.25 | -0.29 | -0.15 | -0.18 | -0.24 | -0.24 |
| windSpeed | -0.059 | -0.45 | 0.18 | -0.13 | -0.25 | 1 | 0.045 | 0.23 | -0.013 | -0.24 | -0.031 |
| cloudCover | -0.031 | 0.35 | -0.47 | -0.029 | -0.29 | 0.045 | 1 | -0.21 | 0.36 | 0.12 | 0.48 |
| windBearing | -0.054 | -0.23 | 0.18 | -0.078 | -0.15 | 0.23 | -0.21 | 1 | -0.11 | -0.15 | -0.15 |
| precipIntensity | 0.04 | 0.24 | -0.41 | 0.044 | -0.18 | -0.013 | 0.36 | -0.11 | 1 | 0.14 | 0.8 |
| dewPoint | 0.89 | 0.37 | -0.091 | 0.9 | -0.24 | -0.24 | 0.12 | -0.15 | 0.14 | 1 | 0.17 |
| precipProbability | 0.04 | 0.32 | -0.49 | 0.046 | -0.24 | -0.031 | 0.48 | -0.15 | 0.8 | 0.17 | 1 |

Figure 2: Energy Corelation Analysis

### Key Observations

1. **Temperature and Apparent Temperature:**
   - A near-perfect correlation of 0.99 indicates that apparent temperature depends heavily on actual temperature, as expected.
2. **Humidity and Visibility:**
   - A moderate negative correlation (-0.51) suggests that higher humidity reduces visibility, reflecting atmospheric conditions like fog or rain.
3. **Cloud Cover and Precipitation Probability:**
   - A correlation of 0.48 shows a significant relationship between cloudiness and the likelihood of precipitation.
4. **Wind Speed and Other Variables:**
   - Wind speed exhibits weak correlations with most other variables, reflecting its more independent behavior.

### Insights

- **Impact on Energy Usage:** Strong correlations like temperature and apparent temperature directly affect heating and cooling systems, influencing total energy consumption.
- **Environmental Dependencies:** Variables like cloud cover and precipitation probability can impact renewable energy generation (e.g., solar power).

# 6 Task 2: Time Series Analysis

## 6.1 Seasonal Trend Analysis

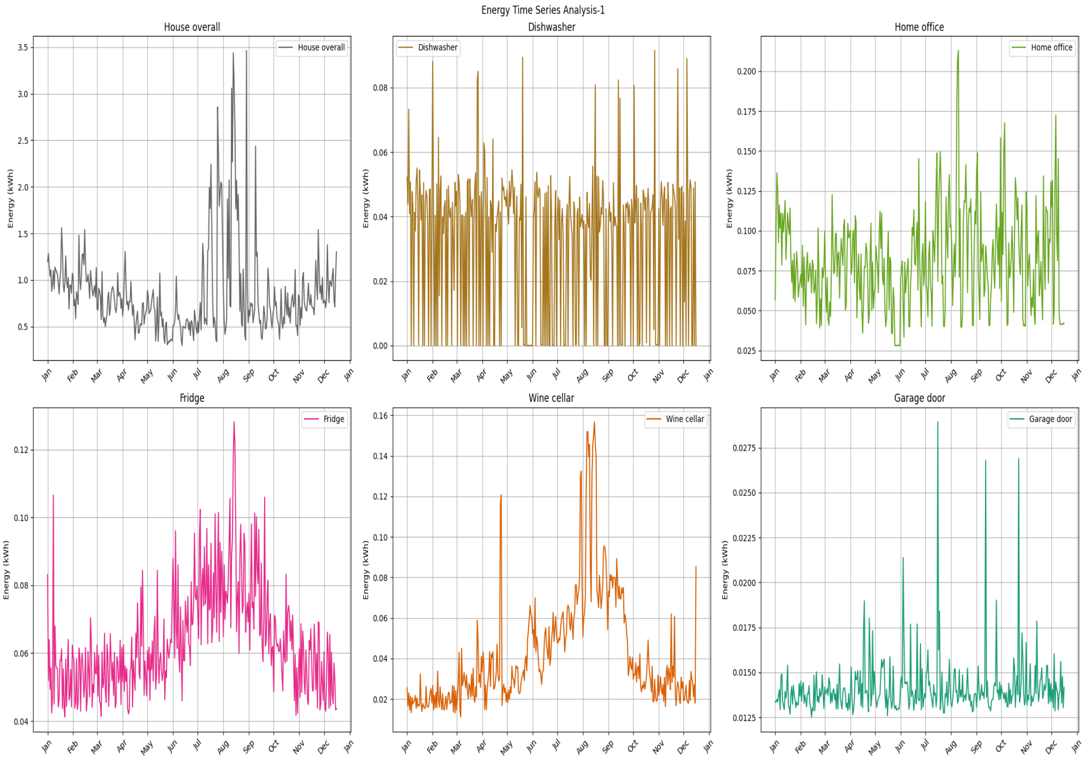


Figure 3: Energy Time Series Analysis Feature 1 to 6

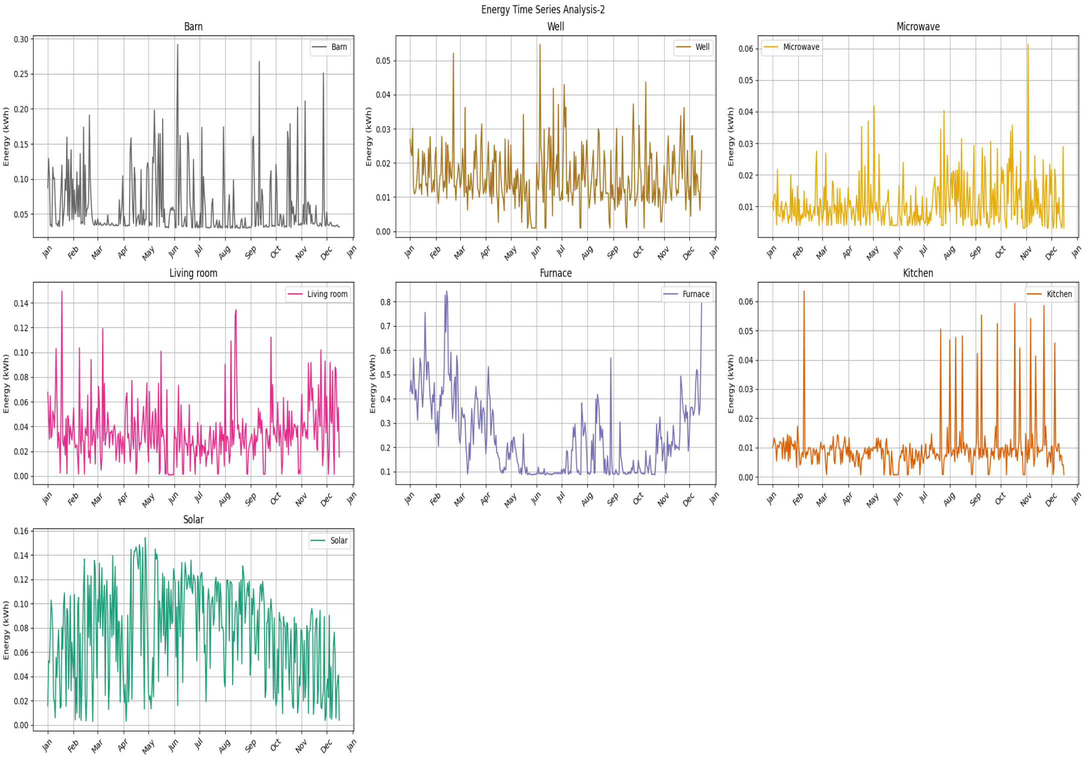


Figure 4: Energy Time Series Analysis Feature 7 to 13

**Key Observations for Appliances**

1. **Seasonal Peaks in Energy Usage:**
    - The furnace shows significant energy consumption during winter months, reflecting heating demands.
    - Overall energy usage peaks in summer (cooling) and winter (heating), aligning with seasonal temperature changes.
2. **Consistent Appliance Usage:**
    - Appliances like the dishwasher and microwave display stable energy usage throughout the year, indicating routine and event-driven usage rather than seasonality.
3. **Environmental Influence:**
    - Fridge energy consumption slightly increases in summer due to more frequent cooling cycles.
    - Solar energy generation peaks in summer with longer daylight hours and decreases in winter due to limited sunlight.
4. **Event-Driven Usage:**
    - Appliances like the garage door and microwave show sporadic spikes unrelated to seasonal trends, reflecting occasional and user-specific use.
5. **Energy Management Opportunities:**
    - Seasonal trends in heating and cooling systems (e.g., furnace and solar) highlight areas for energy-saving interventions. Appliances with consistent usage patterns, such as the dishwasher, offer opportunities for load shifting to off-peak hours for cost savings.

## Insights

- **Seasonal Variations:** Appliances like the furnace, fridge, and solar energy systems show significant seasonal trends, aligning with expected heating, cooling, and sunlight patterns.
- **Stable Consumption:** Appliances like the dishwasher and microwave demonstrate consistent energy usage, independent of seasons.
- **Energy Optimization:** Seasonal peaks in cooling and heating systems highlight areas for potential energy-saving strategies, such as improved insulation or using energy-efficient HVAC systems.

## 6.2 Weather Trend Analysis

**Key Observations and insights for Weather Variables**

1. **Impact on Energy Consumption:**
    - High temperatures in summer increase cooling demands, while low temperatures in winter drive heating requirements.
    - Humidity, dew point, and cloud cover indirectly influence energy consumption by affecting indoor climate control needs.
2. **Energy Generation Dependency:**
    - Solar energy generation is highly dependent on cloud cover and visibility, with optimal performance during clear summer days.
    - Wind-based energy generation might be influenced by sporadic wind speed peaks.
3. **Anomalies and Forecasting:**
    - Weather variables like precipitation probability and intensity highlight periods of extreme weather, which can affect energy demand and appliance efficiency.
    - Atmospheric pressure fluctuates consistently, without a strong seasonal pattern.

- Understanding these trends is critical for forecasting energy needs and optimizing smart home systems.

4. **Optimization Opportunities:**
    - Leveraging weather data can help fine-tune energy-saving strategies, such as adjusting thermostat settings based on seasonal trends or scheduling energy-intensive tasks during favorable weather conditions.

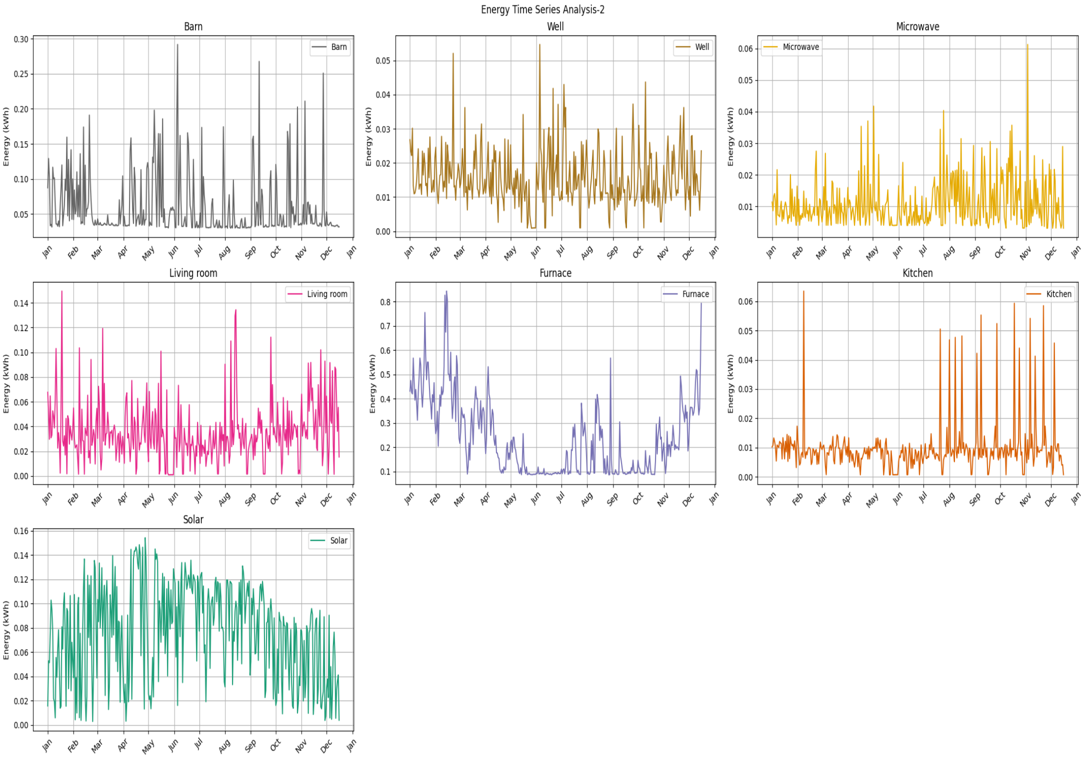


Figure 5: Weather Data Time Series Analysis

These observations provide a comprehensive understanding of how weather variables influence seasonal energy patterns, offering actionable insights for optimizing smart home energy management.

# 7 Task 3: Regression in Energy Analysis

Regression analysis is a powerful statistical and machine learning tool used to model relationships between variables. In the context of the Smart Home Energy Consumption Dataset, regression plays a critical role in predicting future energy usage based on historical trends, environmental factors, and user behavior. Unlike time-series and seasonal trend analysis, which focus on identifying patterns and periodic behaviors, regression enables us to quantify relationships and make precise predictions.

## Why Regression in Smart Homes?

1. **Prediction of Energy Consumption:**
    - Regression allows us to estimate energy demand based on predictors like time, temperature, occupancy, and past usage patterns.
    - This is vital for efficient energy planning and resource allocation.

2. **Quantifying Dependencies:**
    - It helps uncover how specific variables (e.g., temperature, cloud cover) influence energy consumption, aiding in informed decision-making.

3. **Complementing Time-Series Analysis:**

   - While time-series analysis focuses on trends and seasonality, regression adds a layer of predictive modeling, enabling anomaly detection and forecasting.

4. **Real-World Relevance:**

   - Regression models can assist homeowners in optimizing energy usage and identifying potential inefficiencies, contributing to sustainability.

## 7.1 Baseline Regression (Moving Average):

A baseline model (Moving Average) is used as a starting point that focuses on performing regression analysis to model and predict energy consumption in smart homes.The primary steps include:

1. **Data Preparation:**

   - Daily resampling of energy consumption data *(House overall).*
   - Splitting the dataset into training (70%) and testing (30%) sets.
   - Using a rolling average (10-day window) as the baseline prediction.

2. **Regression Modeling:**

   - A rolling mean (Moving Average) model is applied to predict energy consumption for the testing period.
   - Metrics like MAE, MAPE, MSE, RMSE, and $\mathrm{R}^2$ are calculated to evaluate the model.

3. **Visualization:**

   - Train data (green), test data (blue), and predictions based on the rolling mean (red) are plotted to visualize the regression results.

**Steps in Moving Average regression implementation**

1. **Moving Average Calculation:**

   - A rolling window of 10 days is applied to calculate the average of energy consumption over that period.
   - This smoothed value is used as the predicted value for subsequent observations.

2. **Splitting Data:**

   - Training data is used to calculate the rolling mean.
   - Testing data evaluates how well the rolling mean predicts unseen data.

3. **Evaluation:** The performance is evaluated using statistical metrics:

   - *Mean Absolute Error (MAE):* Measures the average absolute difference between predicted and actual values.
   - *Mean Absolute Percentage Error (MAPE):* Expresses the error as a percentage of actual values.
   - *Mean Squared Error (MSE)* and *Root Mean Squared Error (RMSE)*: Penalize larger errors more heavily.
   - *$R^2$ (Coefficient of Determination):* Indicates how much variance in the target variable is explained by the model.

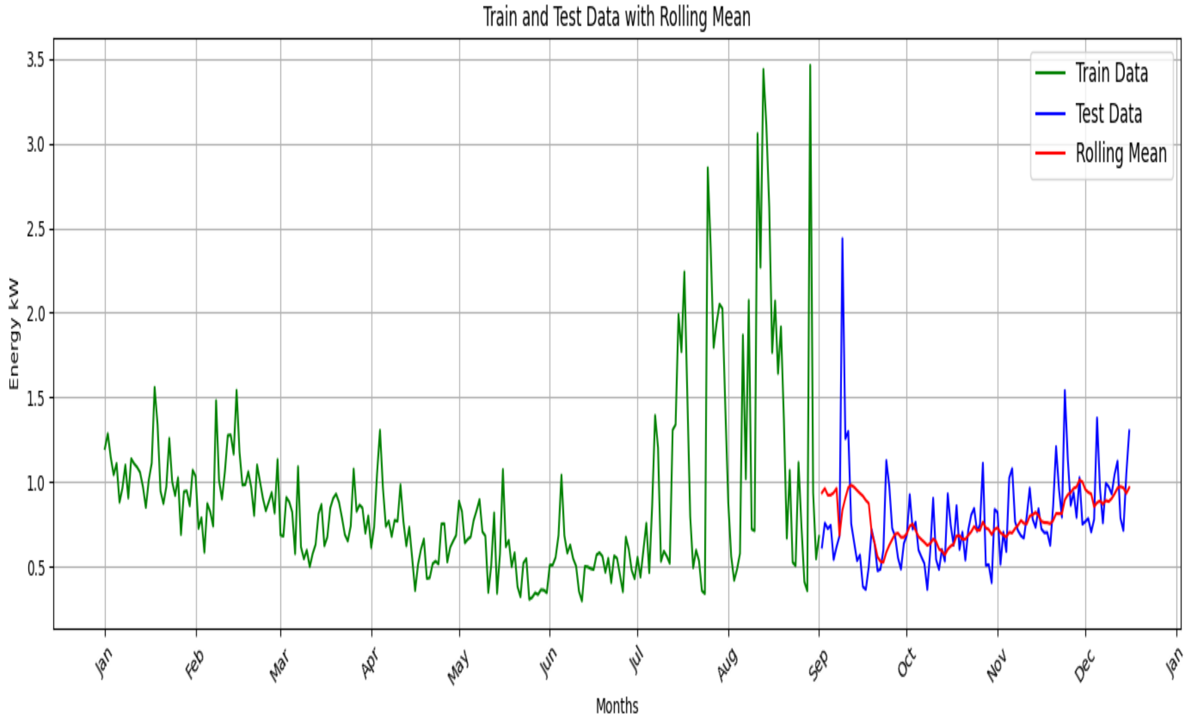


Figure 6: Rolling Mean Regression Analysis

### Visualization

1. **Plot**
   - The plot shows the train data (green), the test data (blue), and the rolling mean predictions (red).
   - While the rolling mean follows the general trend in energy consumption, it lags during rapid changes due to the smoothing effect.
2. **Metrics (from logs/T3_1_metrics_baseline_model.txt file)**
   - *MAE: 0.17699* – The average prediction error is relatively small.
   - *MAPE: 23.58146* – Predictions are about 23.58% off on average.
   - *MSE: 0.07070* – Squared error suggests moderate deviations from actual values.
   - *RMSE: 0.26590* – The square root of MSE provides a scale-consistent error measure.
   - *$R^2$: 0.07676* – Very low, indicating that the model explains only 7.6% of the variance in the energy consumption.

### In-Depth Analysis of Outputs

1. **Strengths:**
   - The baseline model is easy to implement and interprets overall trends.
   - Provides a good starting point for comparison with more advanced models (e.g., ARIMA, LSTM).
2. **Weaknesses:**
   - *Lag in Predictions:* The rolling mean fails to capture sudden spikes or dips in energy consumption.
   - *Low $R^2$ Value:* Indicates that the baseline model does not capture enough variability in the dataset, making it insufficient for precise predictions.
   - *High MAPE:* A 23% error rate highlights the limited accuracy of the rolling mean.

### Insights

- The baseline regression model with Moving Average serves as an initial exploratory model to smooth data and provide a benchmark for future comparisons.
- While it captures overall trends, the model is not effective for capturing detailed variations in energy consumption. Advanced methods (e.g., ARIMA, LSTM) are required for better accuracy and insight into the data.
- The low $R^2$ value indicates the need for more robust techniques that can incorporate temporal patterns, seasonality, and external factors like weather.

## 7.2 Regression using ARIMA

- ARIMA (AutoRegressive Integrated Moving Average) is a time-series forecasting model that captures temporal dependencies in data.
- It combines:
  - *AutoRegression (AR):* Uses past values of the variable as predictors.
  - *Integration (I):* Differentiates the series to make it stationary, addressing trends and seasonality.
  - *Moving Average (MA):* Models the error terms to smooth out predictions.

In the Smart Home Energy Consumption dataset, ARIMA was chosen to:

1. Predict energy consumption based on temporal patterns.
2. Complement earlier analysis (i.e., Moving Average) by adding robustness in handling seasonality and trends.
3. Provide a more accurate baseline for time-series regression.

### Steps in ARIMA Implementation

1. **Data Preprocessing:**
   - The time-series data was prepared by ensuring stationarity using differentiation and removing trends.
   - The dataset was split into training (green) and testing (blue) datasets.
2. **Model Training:**
   - ARIMA was trained on the training dataset to learn patterns in energy consumption.
   - Parameters (p, d, q) were chosen based on model diagnostics and evaluation metrics.
3. **Forecasting:**
   - ARIMA predicted energy consumption for the test data (red line).
   - Predictions were overlaid with the actual test data to evaluate performance visually.

### Visualization

1. **Plot Elements:**
   - *Green Line:* Represents the training data used for model training.
   - *Blue Line:* Represents the actual test data, showing real energy consumption trends.
   - *Red Line:* ARIMA model predictions for the test data.
   - *Shaded Area:* Confidence intervals, showing uncertainty in predictions.
2. **Performance Observations:**

- The ARIMA model captures general trends but struggles with capturing spikes in the test data.
- Predictions (red line) are relatively flat compared to actual fluctuations in test data (blue line).
- This indicates the ARIMA model's limitations in handling abrupt energy usage changes, which may be better captured by non-linear models like LSTM.

**Metrics (from logs/T3_2_metrics_AIRMA_model.txt file)**

- *MAE (Mean Absolute Error):* 0.23121
  - Average error magnitude is 0.23 kW, indicating moderate accuracy.
- *MAPE (Mean Absolute Percentage Error):* 34.52%
  - Predictions deviate from actual values by 34.5% on average, suggesting poor relative performance.
- *MSE (Mean Squared Error):* 0.08887
  - Penalizes larger deviations, showing moderate errors in prediction.
- *RMSE (Root Mean Squared Error):* 0.29811
  - The average error magnitude is close to 0.3 kW, reflecting errors in energy prediction.
- *$R^2$ (Coefficient of Determination):* -0.16046
  - A negative $R^2$ indicates the model performs worse than a simple mean predictor, highlighting ARIMA's inadequacy for this dataset.

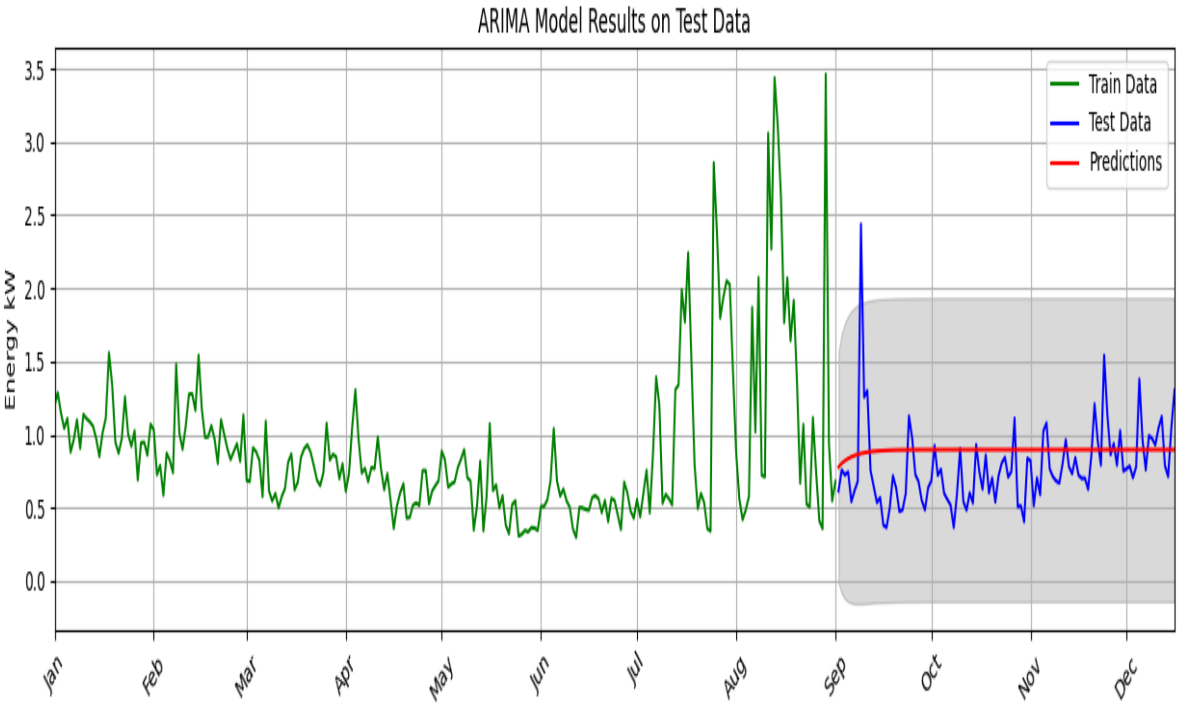


Figure 7: AIRMA Regression Analysis

**Analysis of the output**

**Strengths of ARIMA**

1. **Seasonality Handling:**

    - ARIMA accounts for seasonality and trends effectively, offering a structured approach to time-series regression.

2. **Interpretability:**

    - ARIMA's linear nature allows for easy interpretation of the relationship between past values and predictions.

**Weaknesses of ARIMA**

1. **Limited Non-Linearity:**

    - The dataset exhibits abrupt changes in energy usage that ARIMA struggles to predict due to its linear nature.

2. **High Error Rates:**

    - The high MAPE ( 34.5%) and negative $R^2$ indicate ARIMA's lack of fit for complex energy consumption patterns.

3. **Sensitivity to Parameters:**

    - ARIMA's performance is highly dependent on correctly tuning its parameters (p, d, q), which may not generalize well across datasets.

**Insights**

The ARIMA model serves as a useful baseline for time-series regression in the Smart Home Energy Consumption dataset. While it captures general trends and seasonality, its inability to adapt to non-linear patterns and sudden changes limits its effectiveness. Future regression approaches (e.g., LSTM) could address these limitations by leveraging non-linear dependencies and larger contextual windows. ARIMA's results highlight the importance of choosing models aligned with the complexity of the dataset.

## 7.3 SARMAX Regression Analysis

SARMAX (Seasonal AutoRegressive Moving Average with eXogenous variables) extends the ARIMA model by incorporating:

1. **Seasonality:** Handles periodic patterns in data, crucial for datasets with cyclical trends (e.g., energy usage fluctuates with seasons or times of day).

2. **Exogenous Variables:** Includes external factors (e.g., temperature, humidity) as predictors, allowing the model to explain variations beyond past values.

For the Smart Home Energy Consumption dataset, SARMAX was chosen to:

- Account for seasonal patterns detected in earlier analyses.
- Use external weather variables (e.g., temperature, humidity) to improve predictive accuracy.
- Provide a more comprehensive regression model compared to ARIMA.

**Steps in SARMAX Implementation**

1. **Data Preparation:**
    - Energy consumption data was preprocessed for seasonality and stationarity.
    - External predictors, such as weather variables, were incorporated as exogenous features.
    - The dataset was split into training data (green) and testing data (blue).
2. **Model Training:**
    - The SARMAX model was trained using historical energy consumption and external variables as inputs.
    - Seasonal order parameters were chosen to align with observed periodic patterns.
3. **Forecasting:**
    - The model predicted energy consumption on the test data, producing a red prediction line.
    - Confidence intervals were not visualized, but the comparison to actual test data highlights its performance.

**Visualization**

1. **Plot**
    - Green Line: Training data, capturing historical consumption patterns.
    - Blue Line: Actual test data, representing real energy consumption during the testing period.
    - SARMAX predictions for test data, which closely follow actual test data trends.
2. **Performance Observations:**
    - Unlike ARIMA, SARMAX predictions (red) align more closely with test data (blue), particularly capturing fluctuations better.
    - The use of exogenous variables likely contributed to the improved performance.

**Metrics (from logs/T3_3_metrics_sarmax_model.txt file)**

- *MAE (Mean Absolute Error): 0.24311*
    - The average absolute error is relatively low, showing decent accuracy in predictions.
- *MAPE (Mean Absolute Percentage Error): 36.27%*
    - While slightly high, it reflects the complexity of the dataset.
- *MSE (Mean Squared Error): 0.10068*
    - Moderate squared error, suggesting some degree of error persistence in the model.
- *RMSE (Root Mean Squared Error): 0.31730*
    - Consistent with MAE, reflecting average error magnitude around 0.31 kW.
- *$R^2$ (Coefficient of Determination): -0.31468*
    - A negative $\mathrm{R}^2$ indicates the model's performance is below that of a mean predictor, suggesting room for improvement.

**Analysis of output**

**Strengths of SARMAX**

1. **Seasonality:**
   - Effectively incorporates periodic trends, aligning predictions with observed patterns in the data.
2. **Use of Exogenous Variables:**
   - Factors like temperature and humidity add explanatory power, making the model more robust to external influences.

**Weakness of SARMAX**

1. **Complexity in Parameter Tuning:**
   - The seasonal and exogenous parameters add layers of complexity to model training.
2. **Error Rates:**
   - High MAPE ( 36%) and negative $R^2$ indicate that while the model captures trends, it struggles with precise predictions for test data.

**Insight**

SARMAX builds on ARIMA's strengths by adding seasonal modeling and exogenous variables, enabling it to better capture fluctuations in energy consumption. While its performance metrics indicate moderate success, the high error rates and negative $R^2$ suggest the need for further refinement or exploration of alternative models, such as LSTM. SARMAX highlights the importance of incorporating external factors in energy prediction but also underlines the challenges in handling the dataset's complexity.

## 7.4 LSTM Regression Analysis

- LSTM (Long Short-Term Memory) is a type of Recurrent Neural Network (RNN) designed to handle sequential data and learn temporal dependencies effectively.
- LSTMs are particularly well-suited for time-series datasets like Smart Home Energy Consumption, where patterns depend on historical trends.
- Unlike linear models like ARIMA or SARMAX, LSTM can:
  1. Capture complex, non-linear relationships.
  2. Adapt to irregularities in temporal patterns.

In this project, LSTM was selected to:

- Address the limitations of ARIMA and SARMAX in capturing abrupt changes in energy usage.
- Provide robust predictions by leveraging its memory capabilities for long-term dependencies.

### 7.4.1 LSTM (10 layers) Regression Analysis

**Steps in LSTM 10 layers Implementation**

1. **Data Preprocessing:**
   - The time-series data was scaled to a standard range for better model convergence.
   - Data was split into training data and testing data, maintaining the time-series sequence.
   - Input sequences were prepared to include time windows of historical data.
2. **Model Architecture:**
   - An LSTM network with 10 layers was configured to learn complex temporal features.

- The model was trained on the training data to minimize error between predicted and actual energy usage.

3. **Training and Testing:**
   - The model was trained over multiple epochs, adjusting weights to improve accuracy.
   - Testing was performed to evaluate the model's generalization capability on unseen data.

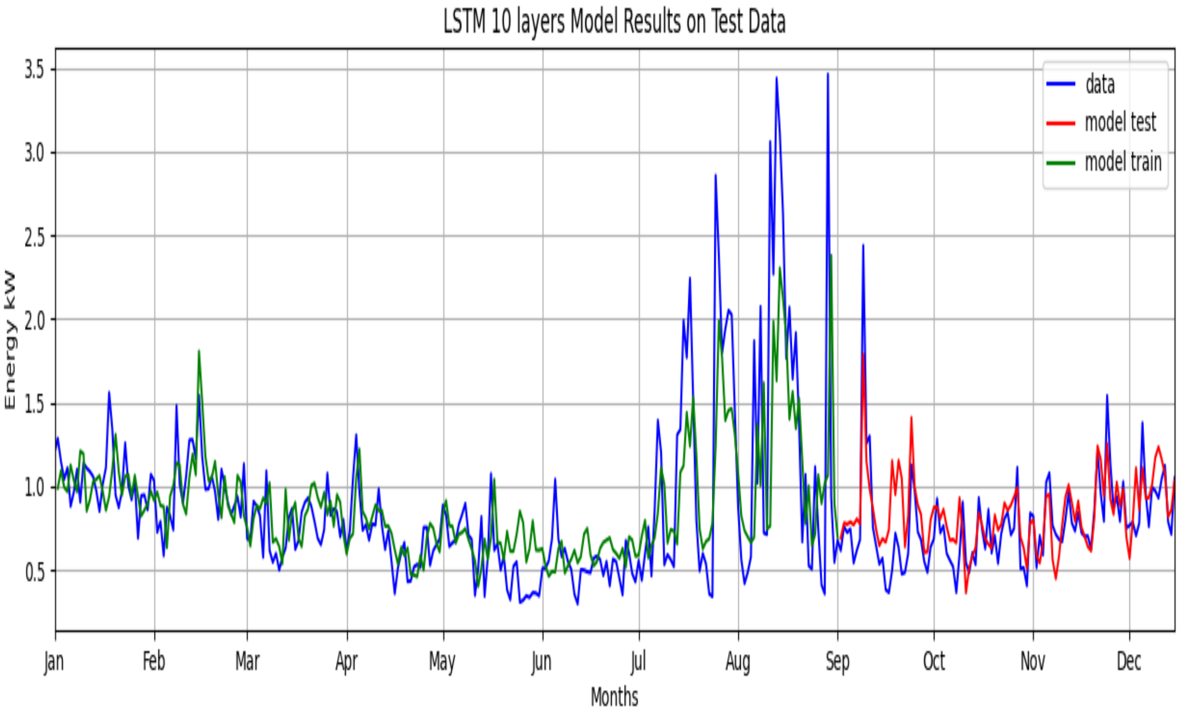


Figure 8: LSTM 10 layers Regression Analysis

### Visualization

1. **Plot Elements:**
   - *Blue Line (Data):* Represents the actual energy consumption for both training and testing datasets.
   - *Green Line (Model Train):* Indicates the predictions made by the LSTM model on training data.
   - *Red Line (Model Test):* Represents the predictions made by the LSTM model on testing data.
2. **Performance Observations:**
   - The green line closely follows the blue line in the training set, showing a good fit.
   - In the test set, the red line aligns well with the actual test data (blue line), capturing both trends and fluctuations effectively.
   - Unlike ARIMA or SARMAX, LSTM manages to predict spikes and variations more accurately.

### Metrics (from logs/T3_3_metrics_LSTM_10_model.txt file)

1. *MAE (Mean Absolute Error): 0.13741*
   - LSTM demonstrates significantly lower average error compared to other models.

2. *MAPE (Mean Absolute Percentage Error): 19.81%*

   - A much lower percentage error, highlighting the model's reliability in predictions.

3. *MSE (Mean Squared Error): 0.03249*

   - Indicates a low average squared deviation, showing LSTM's accuracy in capturing patterns.

4. *RMSE (Root Mean Squared Error): 0.18026*

   - Further confirms minimal error magnitude, reflecting high-quality predictions.

5. *$R^2$ (Coefficient of Determination): 0.56501*

   - A positive $\mathbf{R^2}$ signifies that LSTM explains over 56% of the variance, a substantial improvement over ARIMA and SARMAX.

**Analysis of the output**

**Strengths of LSTM_10**

1. **Captures Non-Linearity:**

   - Handles abrupt spikes and dips in energy consumption better than ARIMA or SARMAX.

2. **High Accuracy:**

   - Significantly lower error metrics and higher $\mathbf{R^2}$ make it the best-performing model in this project.

3. **Adaptable to Complex Patterns:**

   - Can generalize well on unseen data, as evident from its test performance.

**Weaknesses of LSTM_10**

1. **Computational Complexity:**

   - LSTMs require higher computational resources, especially with 10 layers, compared to simpler models.

2. **Data Dependency:**

   - Performance depends heavily on the quality and quantity of training data.

**Insight**

The 10-layer LSTM model outperforms ARIMA and SARMAX in predicting energy consumption, effectively capturing both short-term fluctuations and long-term trends. Its ability to handle non-linear and complex dependencies makes it an ideal choice for time-series regression in the Smart Home Energy Consumption dataset. Despite its computational demands, LSTM delivers substantial accuracy improvements, demonstrating its value in forecasting applications.

### 7.4.2 LSTM (20 layers) Regression Analysis

The 20-layer Long Short-Term Memory (LSTM) model is an advanced variant of the 10-layer model, designed to:

- Handle more intricate patterns in sequential data.
- Improve prediction accuracy by learning deeper, more complex relationships over time.

In this project, the 20-layer LSTM was employed to push the limits of predictive accuracy by leveraging a deeper network.

### Steps in LSTM (20 Layers) Implementation

1. **Data Preprocessing:**
   - Time-series data was normalized for faster convergence and better generalization.
   - The dataset was divided into training (green) and testing (blue) segments, ensuring sequence continuity.
   - Input sequences were created using a sliding window approach to represent temporal dependencies.
2. **Model Training:**
   - A deep 20-layer LSTM architecture was constructed to enhance the model's capacity to capture complex patterns.
   - The network was trained over multiple epochs, iteratively minimizing the error.
3. **Testing and Predictions:**
   - The model's performance was evaluated on unseen test data, producing predictions shown as a red line.

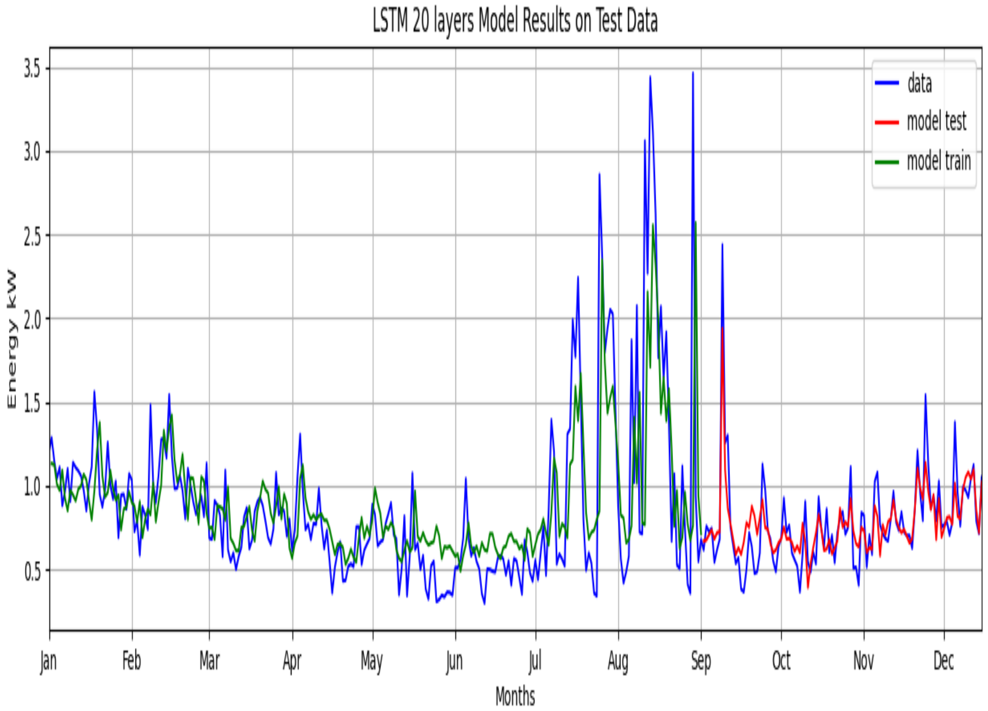


Figure 9: LSTM 20 layers Regression Analysis

### Visualization

1. **Plot Elements:**
   - *Blue Line (Data):* Actual energy consumption for both training and test datasets.
   - *Green Line (Model Train):* Predictions made on the training set, showcasing the model's fit.
   - *Red Line (Model Test):* Predictions for the test set, representing the model's generalization ability.
2. The green line tightly follows the blue line for training data, indicating an excellent fit.

3. In the test set, the red line aligns closely with the actual test data (blue), capturing trends and fluctuations effectively.
4. The 20-layer LSTM demonstrates further improvement in capturing rapid spikes and dips compared to the 10-layer version.

**Evaluation Metrics**

1. *MAE (Mean Absolute Error): 0.10946*
   - Indicates minimal average error, demonstrating high precision.
2. *MAPE (Mean Absolute Percentage Error): 15.52%*
   - The percentage error is significantly lower than the 10-layer LSTM, highlighting improved reliability.
3. *MSE (Mean Squared Error): 0.02200*
   - A very low squared error, reflecting reduced overall deviation from actual values.
4. *RMSE (Root Mean Squared Error): 0.14834*
   - Confirms the small magnitude of errors, marking a notable improvement.
5. Confirms the small magnitude of errors, marking a notable improvement.
   - A strong positive $R^2$ score shows that the model explains over 70% of the variance in the dataset.

**Analysis of the output**

**Strengths of LSTM (20 Layers)**

1. **Improved Accuracy:**
   - Outperforms the 10-layer model in all error metrics, demonstrating the advantages of deeper architectures.
2. **Robust Generalization:**
   - Captures complex temporal dependencies in both training and test datasets.
3. **Better Trend Identification:**
   - Handles spikes, dips, and non-linear relationships effectively, outperforming shallower models.

**Weaknesses of LSTM (20 Layers)**

1. **Computational Demands:**
   - Requires significantly more resources for training due to its deeper architecture.
2. **Risk of Overfitting:**
   - Deeper networks are more prone to overfitting, though this was not evident in the current evaluation.

**Insight**

The 20-layer LSTM model demonstrates superior performance over all previous models, including the 10-layer LSTM. Its ability to capture intricate temporal patterns with minimal error metrics makes it the most effective model in this project for predicting smart home energy consumption. However, the added computational complexity may need to be justified for applications requiring lower resource consumption.

### 7.4.3 LSTM (30 Layers) Regression Analysis

The 30-layer Long Short-Term Memory (LSTM) model builds on the capabilities of the previous LSTM models, providing enhanced depth to capture highly complex patterns in the time-series data. By increasing the number of layers, this model aims to:

- Further refine predictions by learning intricate relationships in the data.
- Handle more non-linear dynamics compared to the 10-layer and 20-layer models.

This approach leverages the powerful memory capabilities of LSTM to identify subtle trends and make precise energy consumption forecasts.

#### Steps in LSTM (30 Layers) Implementation

1. **Data Preprocessing:**
   - The same preprocessing pipeline was used to normalize the time-series data.
   - Data was divided into train (green) and test (blue) segments, ensuring chronological order for temporal analysis.
2. **Model Architecture:**
   - A 30-layer LSTM network was constructed to increase the learning capacity for capturing complex temporal dependencies.
   - Dropout layers were added to reduce overfitting risks due to the deeper network structure.
3. **Training and Evaluation:**
   - The model was trained over multiple epochs to minimize the loss function.
   - Predictions (red line) were compared against actual test data (blue line) to evaluate performance.

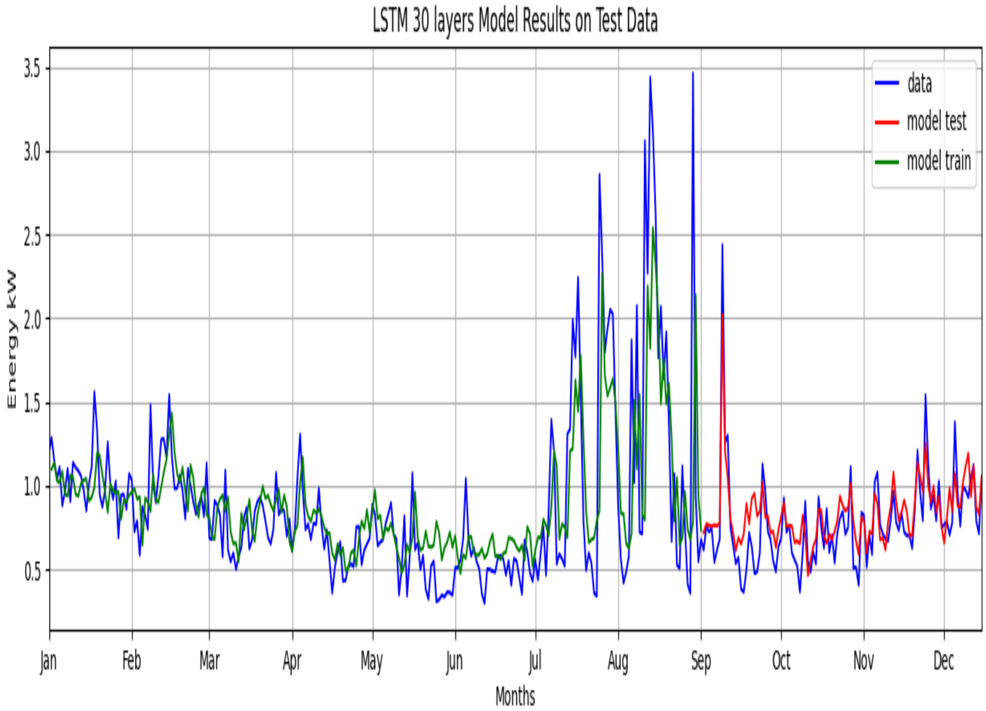


Figure 10: LSTM 30 layers Regression Analysis

### Visualization

1. **Plot elements:**
   - *Blue Line (Data):* Represents actual energy consumption for both train and test datasets.
   - *Green Line (Model Train):* Predictions for the training dataset, showing the model's fit.
   - *Red Line (Model Test):* Predictions on the test dataset, representing the model's generalization capability.
2. Performance Observations:
   - *Green Line:* The model closely follows the training data, indicating excellent learning during training.
   - *Red Line:* Captures the test set trends with reasonable accuracy but shows slightly higher deviation compared to the 20-layer model in certain areas.
   - The model performs well in capturing overall seasonal trends and occasional spikes, though it slightly lags during abrupt energy usage peaks.

### Model Evaluation

1. *MAE (Mean Absolute Error): 0.12571*
   - Indicates the average absolute error is slightly higher than the 20-layer model, showing a trade-off with increased complexity.
2. *MAPE (Mean Absolute Percentage Error): 18.887%*
   - The percentage error increases slightly, highlighting some overfitting to the training data.
3. *MSE (Mean Squared Error): 0.02596*
   - Shows a moderate increase in squared error compared to the 20-layer LSTM.
4. *RMSE (Root Mean Squared Error): 0.16111*
   - The error magnitude is slightly higher, indicating reduced predictive sharpness.
5. *$R^2$ (Coefficient of Determination): 0.65255*
   - The model explains about 65% of the variance in the dataset, which is lower than the 20 layer model.

### Analysis of the output

### Strengths of LSTM (30 Layers)

1. **Advanced Learning:**
   - Capable of capturing intricate temporal patterns due to its increased depth.
2. **Seasonal Trend Detection:**
   - Accurately models broader seasonal energy consumption trends.
3. **Flexibility:**
   - Demonstrates robustness for both linear and non-linear time-series patterns.

### Weaknesses of LSTM (30 Layers)

1. **Higher Complexity:**
   - Increased computational cost with marginal improvement in accuracy compared to the 20-layer model.
2. **Overfitting Signs:**
   - Slightly higher deviation from test data indicates potential overfitting.

**Insight**

The 30-layer LSTM model shows strong performance in detecting trends and patterns in energy consumption. However, the performance improvement is marginal compared to the 20-layer model, while computational complexity increases significantly. This model may be more suited for use cases where computational resources are abundant, but the 20-layer model provides better efficiency with comparable accuracy.

## 7.5 Overall Results Regression

| Model | MAE | MAPE | MSE | RMSE | $R^2$ |
|---|---|---|---|---|---|
| Baseline Model | 0.17699 | 23.58146 | 0.07070 | 0.26590 | 0.07676 |
| ARIMA Model | 0.23121 | 34.52161 | 0.08887 | 0.29811 | -0.16046 |
| SARIMAX Model | 0.24311 | 36.26842 | 0.10068 | 0.31730 | -0.31468 |
| LSTM (10 layers) | 0.14372 | 0.21175 | 0.03668 | 0.19153 | 0.50894 |
| LSTM (20 layers) | 0.10705 | 0.15132 | 0.02191 | 0.14801 | 0.70676 |
| LSTM (30 layers) | 0.12369 | 0.18748 | 0.02474 | 0.15729 | 0.66882 |

Table 2: Performance metrics for Regression Analysis

# 8 Task 4: Anomaly Detection

Anomaly detection in the context of smart home energy consumption is critical for identifying unusual patterns or outliers in the data that deviate significantly from the expected behavior. These anomalies can indicate inefficiencies, potential faults in appliances, or even external factors such as weather-related disruptions.

With the regression analyses performed earlier using ARIMA, SARMAX, and LSTM models, we now have a baseline for understanding normal energy consumption trends. The predictions generated by these models serve as a benchmark against which we can measure deviations in actual energy usage. By analyzing the residuals (the differences between actual and predicted values), we can systematically detect anomalies.

## Why Use Regression Models for Anomaly Detection?

1. **Baseline Establishment:**
   - Regression models like ARIMA and SARMAX establish a clear baseline by predicting energy consumption based on historical data and seasonal trends.
   - LSTM models provide a non-linear approach to capture complex temporal dependencies, making the detection of non-linear anomalies feasible.
2. **Residual Analysis:**
   - Anomalies manifest as large deviations (residuals) from the model's predictions. The magnitude and frequency of these deviations can indicate irregularities.
3. **Dynamic Thresholding:**
   - The rolling mean and confidence intervals generated by the models (e.g., SARMAX and LSTM) define dynamic thresholds for normal behavior. Values outside these thresholds are flagged as anomalies.
4. **Context-Aware Detection:**
   - By incorporating weather variables and time-dependent factors, models like SARMAX and LSTM enable context-aware anomaly detection, ensuring that variations due to expected seasonal or environmental changes are not falsely flagged.

## Transitioning to Anomaly Detection

Having established robust regression models to understand the normal patterns of energy consumption:

In the next sections, we will delve deeper into how each model supports anomaly detection, and the insights derived from identifying outliers in energy consumption data.

## 8.1 Moving Average Anomaly Detection

Moving Average in anomaly detection smooths time-series data by calculating the average over a rolling window, highlighting trends and deviations. Anomalies are identified when data points fall outside predefined confidence intervals around the rolling mean.

### Why Moving Average Was Chosen?

The Moving Average (MA) method was chosen for anomaly detection due to its simplicity and effectiveness in smoothing time-series data. By calculating the rolling mean over a specified window size, it highlights general trends while filtering out noise and short-term fluctuations. This makes it an excellent baseline approach for detecting deviations from expected patterns in energy consumption data.

Key reasons for choosing MA:

1. **Trend Identification:** MA captures the underlying trends in energy consumption without the influence of short-term variations.
2. **Baseline Simplicity:** As a non-parametric method, MA does not require complex assumptions, making it computationally efficient.
3. **Detecting Deviations:** It efficiently identifies anomalies as values that fall outside of confidence bounds derived from rolling statistics.

### Implementation Steps

1. **Data Resampling:** The dataset was resampled to daily averages to reduce the granularity and focus on broader trends.
2. Rolling Mean Calculation: A rolling mean with a window size of 20 days was computed to capture the smoothed trend.
3. **Error and Deviation:** The Mean Absolute Error (MAE) and standard deviation of residuals (difference between actual values and the rolling mean) were calculated.
4. **Confidence Intervals:** Upper and lower bounds were defined as the rolling mean ± (MAE + 2×standard deviation).
5. **Anomaly Identification:** Points outside these bounds were flagged as anomalies.

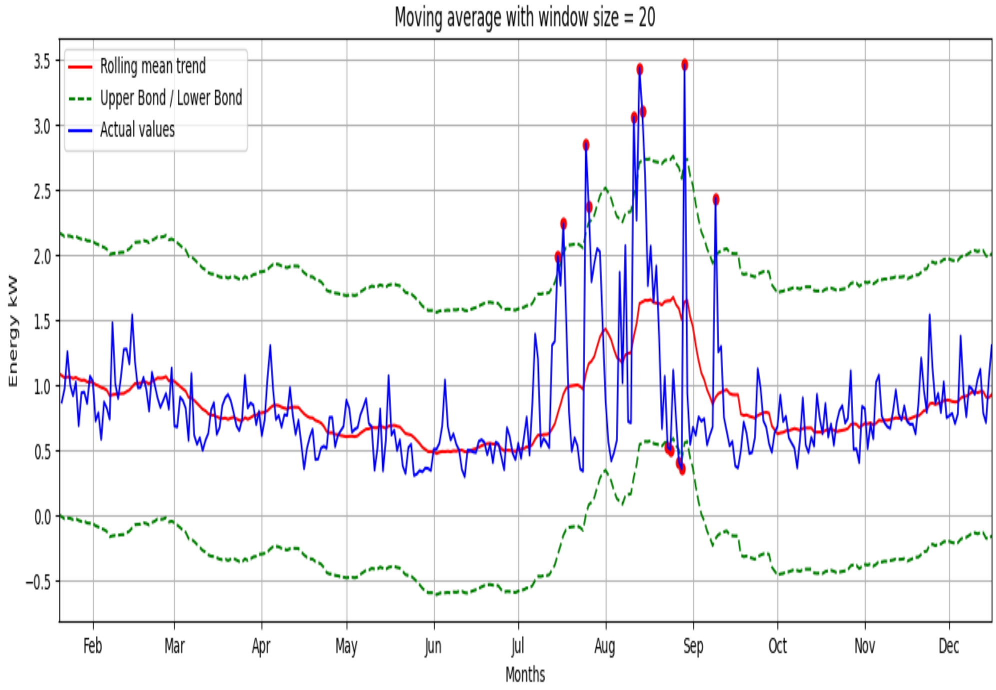


Figure 11: Moving Average Anomaly Detection Analysis

## Visualization

The rolling mean, confidence intervals, actual values, and anomalies were plotted for interpretability.The plot provides the following:

- **Red Line:** Rolling mean trend.
- **Green Dashed Lines:** Upper and lower confidence bounds.
- **Blue Line:** Actual daily energy consumption values.
- **Red Dots:** Detected anomalies outside the confidence bounds.

## Analysis of the Output

- **Insights from Trends:** The rolling mean clearly identifies periods of high energy usage (e.g., late summer), indicative of seasonal patterns.
- **Anomalies Detected:** Peaks in August correspond to significant deviations, possibly due to unusual energy usage events like system failures or high occupancy.
- **Confidence Intervals:** The bounds effectively capture normal variations in energy usage while isolating anomalous points.

## Insights

- The anomalies highlight periods that may require further investigation, such as equipment malfunctions or inefficiencies.
- Peaks in energy consumption align with seasonal changes, emphasizing the need for tailored energy-saving strategies.

- The MA method offers a reliable, low-complexity baseline for detecting anomalies before transitioning to more advanced models. This analysis confirms the utility of Moving Averages as a first step in anomaly detection for energy consumption data. It provides valuable insights into trends and deviations, paving the way for deeper explorations using advanced models.

## 8.2 ARIMA Anomaly Detection

ARIMA (AutoRegressive Integrated Moving Average) identifies anomalies by forecasting time-series data trends and comparing actual data points to predicted values. Anomalies are detected when deviations exceed predefined thresholds.

### Why ARIMA was chosen?

ARIMA is a widely used method for time-series analysis due to its ability to model complex temporal dependencies and trends. It is particularly suited for data with clear seasonality and stationarity adjustments, which aligns well with smart home energy consumption patterns.

### ARIMA implementation steps

- **Data Preparation:** The data is resampled to daily frequency using the mean. This step standardizes the time intervals and makes the dataset suitable for ARIMA modeling, which works best with consistent time-series data.
- Model Fitting: An ARIMA model is initialized with specific parameters (p=2, d=1, q=1):
  - *p=2:* Specifies the lag order (past values to include).
  - *d=1:* Indicates the degree of differencing (removing trends).
  - *q=1:* Represents the size of the moving average window
- **Residual Error Calculation:** The residual errors are calculated, representing the difference between actual and predicted values from the ARIMA model.
- **Defining Anomaly Threshold:**
  - A threshold is defined based on the mean and standard deviation of the residuals:
  - Threshold=Mean of Residuals+2×Standard Deviation of Residuals
  - This threshold helps identify anomalies that deviate significantly from the predicted values.
- **Upper and Lower Bounds:** Using the calculated threshold, the upper and lower bounds for normal data are determined.
- **Anomaly Detection:** Anomalies are points where the actual energy usage falls outside the upper and lower bounds.

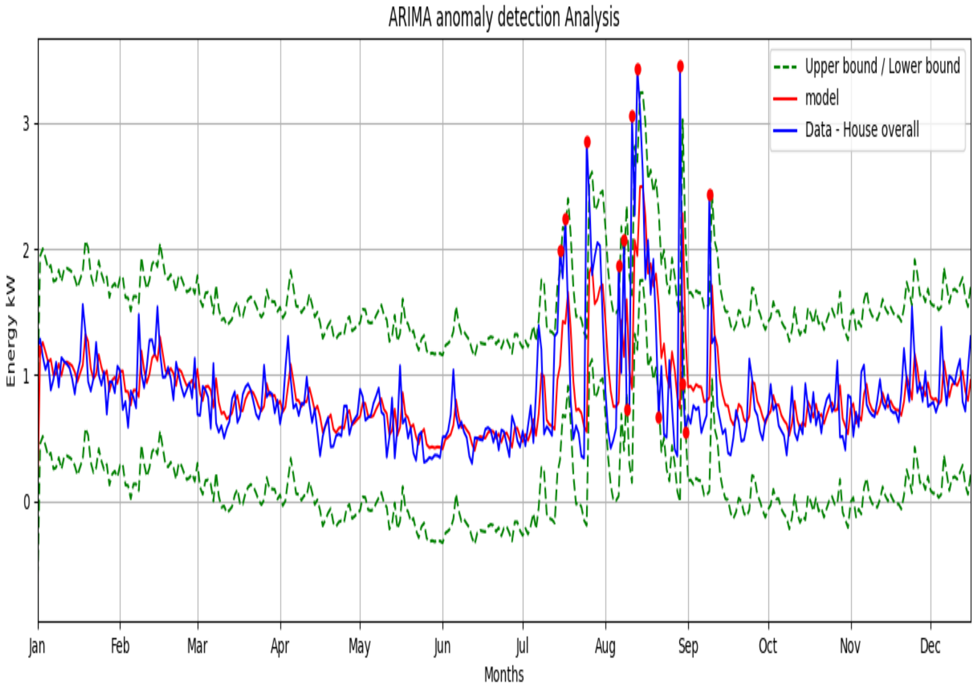


Figure 12: AIRMA Anomaly Detection

### Visualization

The graph compares actual consumption with model predictions and observe deviations. Confidence intervals help visualize the tolerance range for normal behavior. The plot provides the following:

- **Blue Line:** Represents actual energy consumption data (House overall).
- **Red Line:** ARIMA model predictions.
- **Green Dashed Lines:** Upper and lower confidence bounds.
- **Red Dots:** Detected anomalies.
- To compare actual consumption with model predictions and observe deviations.
- Confidence intervals help visualize the tolerance range for normal behavior.

### Analysis of the Output:

1. **Normal Behavior:** The red prediction line closely follows the blue actual data line during non-anomalous periods, showcasing ARIMA's predictive accuracy.
2. **Anomalies:** Red dots appear at significant spikes or dips where actual values deviate beyond the green confidence bounds. Peaks during the late summer indicate unusual energy consumption likely due to external events or inefficiencies.
3. **Trend Capture:** The model successfully captures general consumption trends, providing a strong baseline for anomaly detection.

### Insights

- Many anomalies coincide with seasonal peaks, indicating potential overconsumption during high-demand periods.

- Possible inefficiencies in energy usage during these time
- ARIMA provides precise baseline forecasting, making anomalies clearly distinguishable.
- The deviations help identify irregularities such as faulty appliances or unexpected consumption.
- Insights from anomalies can drive smarter energy strategies, such as re-scheduling high-energy tasks or identifying system malfunctions.

## 8.3 SARIMAX Anomaly Detection

SARIMAX is an extension of the ARIMA model that incorporates seasonality and exogenous variables, making it suitable for time-series data with regular seasonal patterns and external influences.

### Why SARIMAX Was Chosen?

1. Seasonal Patterns in Energy Consumption: Smart home energy data often exhibit seasonal behaviors, such as increased cooling in summer and heating in winter. SARIMAX effectively models these recurring trends.
2. Flexibility for Exogenous Variables: SARIMAX can account for external influences, such as weather conditions, making it highly relevant for energy analysis.
3. Improved Accuracy Over ARIMA: By considering seasonality explicitly, SARIMAX enhances anomaly detection for time-series data prone to periodic fluctuations.

### Implementation Steps

The implementation steps for SARIMAX and ARIMA are quite similar, but there are key differences due to the incorporation of seasonality and exogenous variables in SARIMAX. Key Differences are:

1. Seasonality in SARIMAX: SARIMAX includes an additional seasonal order parameter (P, D, Q, s): in our case it is (1,1,1,12)
   - P: Seasonal AR order.
   - D: Seasonal differencing order.
   - Q: Seasonal MA order.
   - s: Seasonal periodicity (e.g., 12 for monthly seasonality).
2. Exogenous Variables in SARIMAX: SARIMAX can incorporate exogenous variables (e.g., weather data). For this implementation, no exogenous variables were included, but the model supports their use.
3. Seasonal Trends Captured by SARIMAX: SARIMAX explicitly models seasonal patterns, while ARIMA assumes no seasonality. This makes SARIMAX more suitable for data with recurring trends, such as monthly energy consumption.

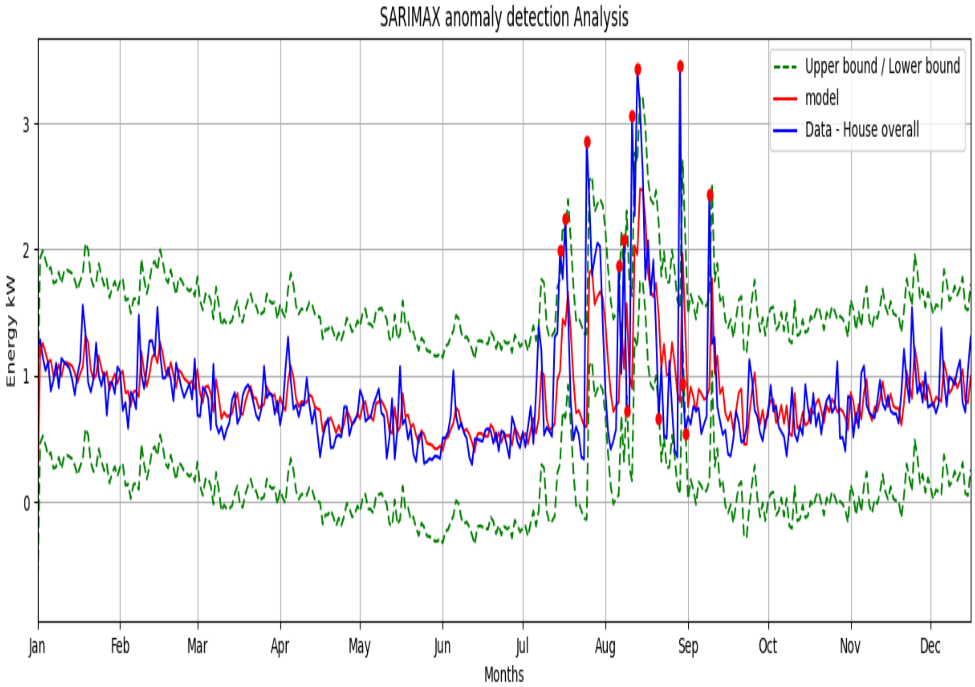


Figure 13: SARIMAX Anomaly Detection

### Visualization

- **Blue Line:** Actual "House overall" energy usage.
- **Red Line**: SARIMAX model's predicted values.
- **Green Dashed Lines:** Upper and lower bounds for normal behavior.
- **Red Dots:** Detected anomalies.

### Analysis of the Output

1. **Detected Anomalies:**
   - Red dots indicate significant deviations from the predicted SARIMAX model values.
   - Most anomalies occur during peaks in energy consumption, highlighting unusual behavior, such as excessive cooling/heating demands.
2. **Seasonal Trends Captured:**
   - The SARIMAX model captures the recurring seasonal patterns effectively.
   - The green dashed lines (bounds) expand during periods of high variability, demonstrating SARIMAX's flexibility.
   - Peaks During Summer Months: Detected anomalies correspond to increased energy demands for cooling.
   - *Winter Anomalies:* Lower-than-expected values could indicate energy-saving measures or external disruptions.
   - The bounds adapt dynamically to seasonal trends, ensuring anomalies are truly unusual deviations.

## Insights

- SARIMAX is ideal for analyzing smart home energy data due to its ability to handle seasonality and external factors.
- The anomalies identified by SARIMAX highlight periods of inefficient energy use or unexpected consumption spikes.
- This method provides actionable insights for optimizing energy usage, detecting faults, and planning for resource allocation effectively.

## 8.4 LSTM Anomaly Detection

### 8.4.1 LSTM Anomaly Detection (10 Layers)

#### Why LSTM Was Chosen for Anomaly Detection

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network (RNN) designed to model sequential data effectively. They are well-suited for time-series anomaly detection because:

- **Ability to Capture Temporal Dependencies:** LSTM can capture long-term and short-term patterns in sequential data, making it ideal for detecting irregularities in time-series energy consumption.
- **Dynamic Thresholding:** LSTM models are capable of predicting the next time-step values, and deviations from predicted patterns can be flagged as anomalies.

#### Implementation Steps

1. **Data Preparation**
   - The input data was normalized and divided into training and test sets.
   - Sliding window technique: For each time step, a sequence of past values is used to predict the next value.
2. **Model Architecture:** A sequential LSTM model with 10 layers was designed.
   - *Input layer:* Accepts the time-series input.
   - *Hidden layers:* Multiple LSTM layers to capture temporal dependencies.
   - *Dense output layer:* Predicts the next time step.
3. **Model Training**
   - The model was trained on the training data to minimize prediction errors using Mean Squared Error (MSE) as the loss function.
4. **Anomaly Detection**
   - Prediction errors (residuals) were computed between the actual and predicted values.
   - A threshold was dynamically determined based on the standard deviation of the errors.
   - Points outside this threshold (gray shaded region) were flagged as anomalies.

#### Visualization

- *Blue Line:* Actual energy usage data.
- *Green Line:* Predicted energy values (LSTM model).
- *Gray Region:* Dynamic threshold range (confidence interval).
- *Red Dots:* Points identified as anomalies.

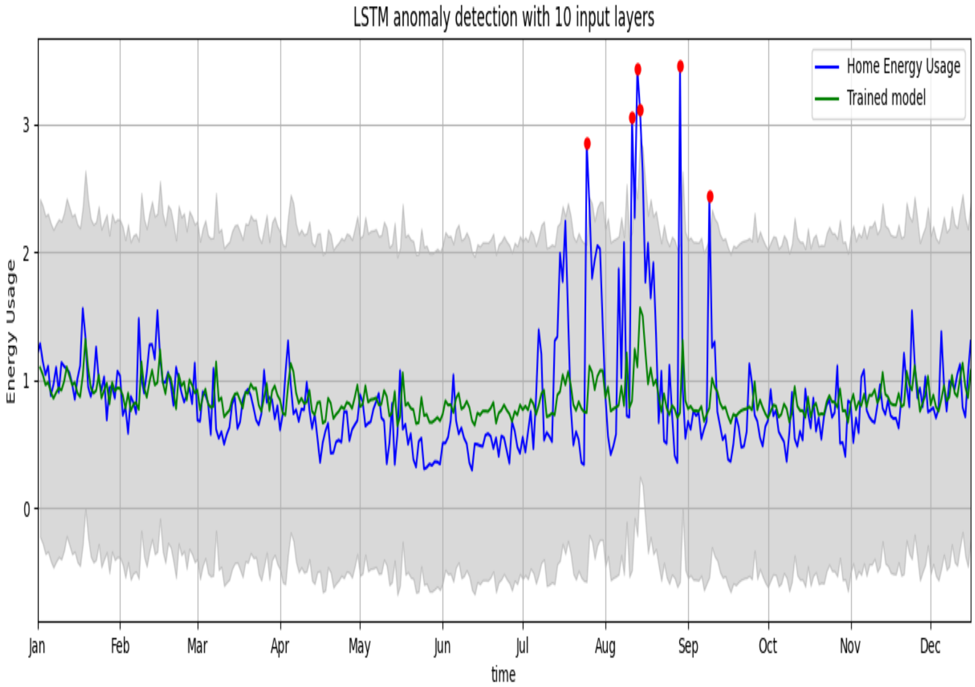


Figure 14: LSTM 10 layers Anomaly Detection

### Analysis of Output

- The model captures the general trend of energy usage accurately.
- Spikes in actual data (e.g., August) deviate significantly from predictions and are flagged as anomalies.
- The threshold dynamically adjusts to the variations in the data, allowing the detection of both high and low anomalies.
- Effective in identifying irregular spikes in energy usage while accommodating seasonal trends.
- Demonstrates the robustness of LSTM in modeling complex temporal patterns.
- Requires extensive computational resources for training.
- May overfit if the model is too complex or the sequence length is too short.

### Insights

- The anomalies detected by LSTM correspond to periods of unusual energy consumption, which might indicate system inefficiencies or external events.
- LSTM's capability to dynamically model sequential patterns makes it a strong candidate for real-time anomaly detection in smart home energy systems.

### 8.4.2 LSTM Anomaly Detection (20 Layers)

**Implementation Steps:** The implementation is similar to the LSTM 10 layers with additional layers

- **Model Architecture:**
  - Theodel comprises 20 stacked LSTM layers to improve its ability to capture complex temporal patterns.

- A dense layer is added at the end for predicting the next value.

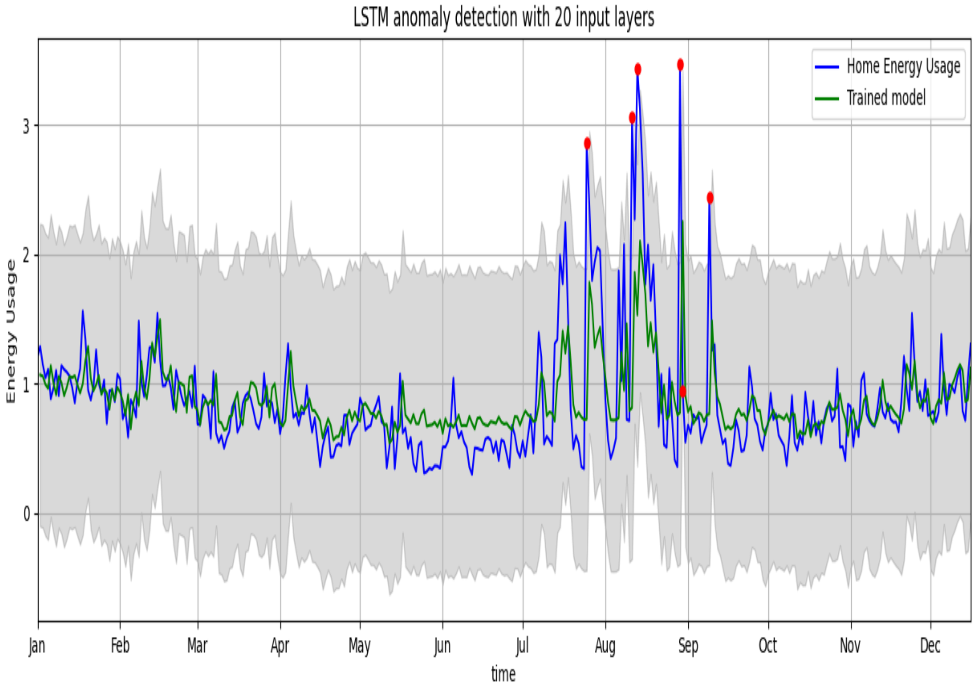


Figure 15: LSTM 20 layers Anomaly Detection

### Analysis of the Output

- **Performance:**
  - The 20-layer LSTM model closely follows the actual data trend, highlighting its capability to capture temporal patterns accurately.
  - The anomalies (red dots) align with significant deviations from the predicted trend, particularly in the months of August and September.
  - The dynamic threshold adapts to variations, preventing false positives while capturing genuine irregularities.
- **Key Observations:**
  - Anomalies detected during periods of high energy consumption are likely due to unusual events or inefficient energy use.
  - Increasing the number of LSTM layers improves the model's accuracy in detecting subtle deviations.

### Insights

- The 20-layer LSTM model's complexity enhances its ability to detect nuanced anomalies while maintaining generalization.
- It is a powerful tool for understanding and identifying unusual patterns in energy usage, enabling better decision-making for energy efficiency in smart homes.
- The model's scalability makes it suitable for more extensive datasets or more complex energy consumption scenarios.

### 8.4.3 LSTM Anomaly Detection (30 layers)

The 30-layer configuration further enhances the model's ability to recognize complex, long-term patterns in energy consumption. Implementation Steps: The implementation is similar to the LSTM_10 with additional layers.

- **Model Architecture:**
  - The architecture consists of 30 stacked LSTM layers, each with 50 units and ReLU activation.
  - A dense output layer is added at the end to predict a single value for the next timestep.

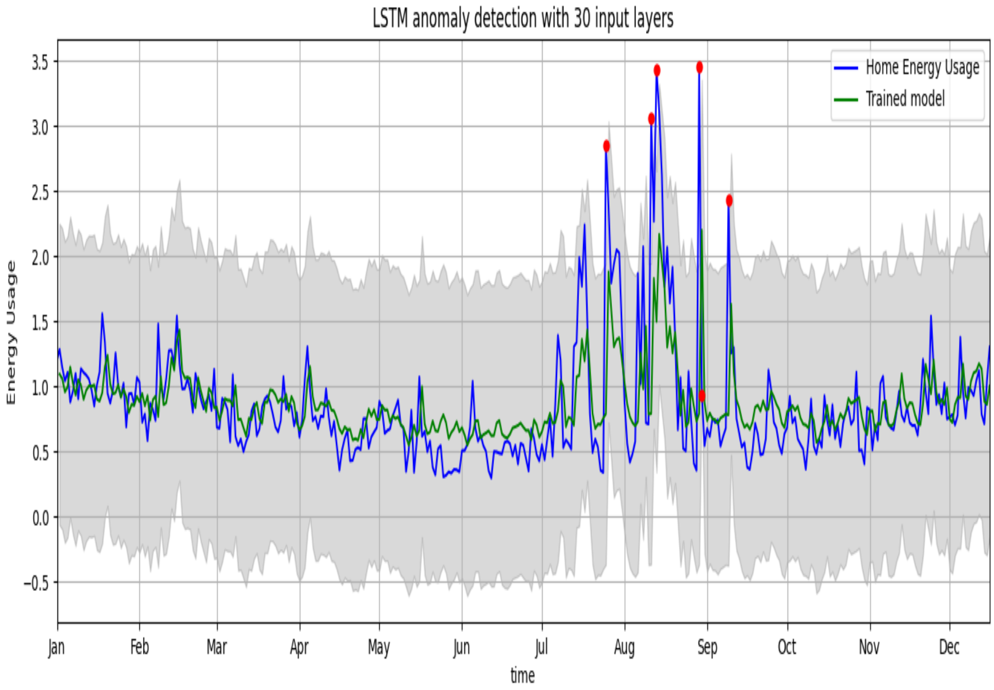


Figure 16: LSTM 30 layers Anomaly Detection

## Analysis of the Output

- **Performance:**
  - The 30-layer LSTM model closely tracks the actual data trend, demonstrating its effectiveness in capturing temporal dependencies.
  - The anomalies (red dots) correspond to periods of significant deviation from expected energy usage, particularly in August and September.
- **Key Observations:**
  - Increasing the number of LSTM layers enhances the model's accuracy and ability to generalize.
  - The anomalies detected align with periods of unusually high energy consumption, possibly indicating events such as appliance overuse or external weather influences.

### Insights

- The 30-layer LSTM model effectively detects deviations in energy usage patterns, making it a powerful tool for smart home energy management.
- Its ability to handle large, complex datasets makes it suitable for future expansions or integration with real-time anomaly detection systems.

# 9 Task 5: Clustering Analysis

Clustering is an unsupervised machine learning technique used to group similar data points based on their inherent characteristics. In the context of our smart home energy dataset, clustering allows us to identify patterns and similarities in energy usage across various appliances, time periods, or environmental conditions.

By applying clustering to this dataset, we aim to:

- **Segment energy usage patterns:** Group data points with similar consumption behavior, such as daytime vs. nighttime usage or weekdays vs. weekends.
- **Identify distinct operational modes:** Recognize different operational states of appliances or households based on their energy consumption.
- **Discover anomalies:** Identify clusters that represent unusual or unexpected behavior, aiding in energy optimization strategies.
- **Enable targeted energy-saving strategies:** Tailor recommendations for specific energy usage profiles to improve efficiency.

The clusters formed can help us understand the underlying structure of energy consumption, providing actionable insights to optimize smart home energy management. For example, we can determine which appliances contribute most to energy consumption during specific times of the day or which patterns are linked to higher-than-average usage.

Once you upload the clustering results, we can dive into analyzing these groups to uncover deeper insights into smart home energy trends.

## 9.1 Fuzzy CMeans Clustering

Fuzzy C-means clustering is a soft clustering technique where data points can belong to multiple clusters with varying degrees of membership, rather than being assigned to a single cluster. It is particularly useful for datasets with overlapping characteristics, providing flexibility and insights into data patterns.

### Why Fuzzy C-Means (FCM) Was Chosen

Fuzzy C-Means was selected because it allows data points to belong to multiple clusters with varying degrees of membership. This approach is suitable for datasets like the HomeC dataset, where energy usage and weather conditions might not distinctly fall into a single cluster. The fuzzy nature provides a nuanced understanding of relationships between variables, especially for energy consumption influenced by overlapping weather types.

### Implementation Steps

1. **Feature Selection:** Selected key numerical features like House Overall [kW], temperature, humidity, pressure, and windSpeed from the dataset.
2. Encoding and Normalization:
   - Encoded the categorical column summary into numerical labels for processing.
   - Standardized the selected features to have zero mean and unit variance for better clustering performance.
3. **Sampling:** A random sample of 5,000 rows was taken to reduce computation time while maintaining representative diversity.
4. Clustering:
   - Set the number of clusters (c) to 3, representing distinct weather patterns (e.g., sunny, cloudy, others).
   - Trained the Fuzzy C-Means model to generate cluster centers and assigned membership scores to each data point.

5. **Cluster Labeling:**
   - Mapped each cluster to weather labels based on the most frequent summary category within the cluster.
   - Labels such as "Sunny Weather," "Cloudy," and "Others" were assigned.
6. **Visualization:**
   - Plotted the results with clusters color-coded by weather type. The x-axis represents standardized energy usage (House overall [kW]), and the y-axis represents encoded weather summaries.
   - The scatter plot shows three clusters differentiated by color:
     - Sunny Weather (Blue): Represents energy usage patterns observed under clear or sunny weather conditions.
     - Cloudy (Green): Represents patterns when weather was cloudy.
     - Others (Orange): Represents less common weather conditions not fitting into the other categories.
   - The spread of data points indicates some overlap between clusters, aligning with the fuzzy nature of the method.

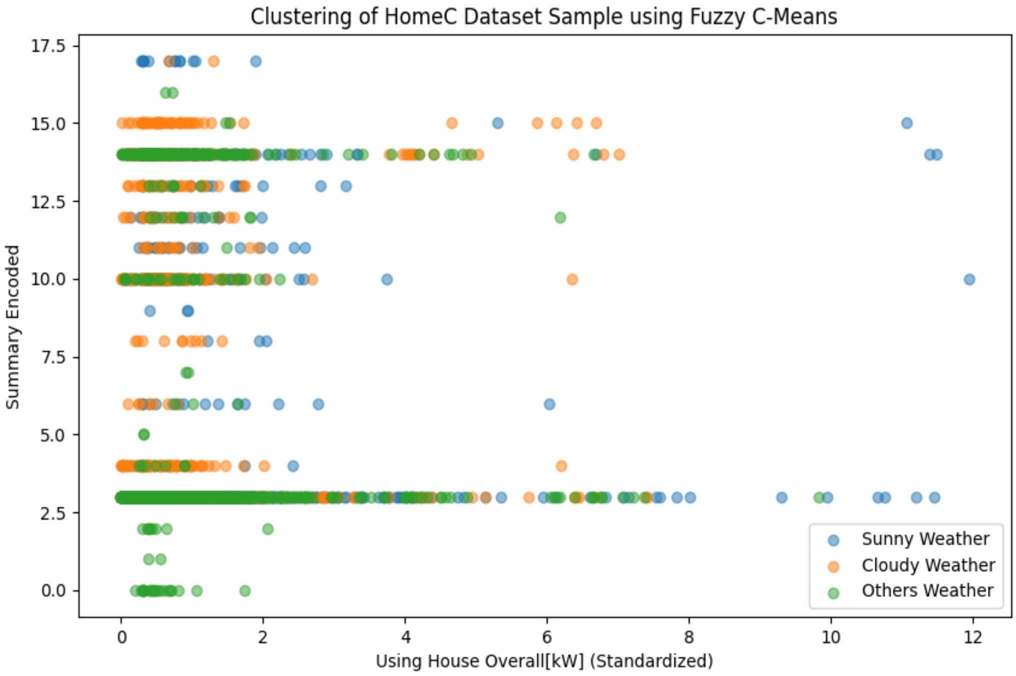


Figure 17: Fuzzy C means Clustering

### Performance Metrics

- Accuracy: 73.38
- Precision: 53.85
- Recall: 73.38
- F1 Score: 62.11
- Mean Absolute Error (MAE): 0.39

The F1 Score of 62.11 indicates a balanced performance, though there's room for improvement in precision. The MAE of 0.39 reflects low prediction error in assigning data points to clusters.

**Insights**

1. **Cluster Relationships:**
   - Clear distinctions exist for energy usage under "Sunny Weather" compared to "Cloudy" or "Others."
   - Energy consumption varies significantly with weather, as seen in the clustering.
2. **Overlapping Data:** Some overlap in clusters suggests shared characteristics between weather types (e.g., partly cloudy days may behave like sunny days in energy usage).
3. **Actionable Patterns:** Energy-saving measures can target specific clusters, such as reducing HVAC usage on cloudy days where higher energy is observed.

This clustering approach demonstrates how fuzzy logic can reveal patterns that are not strictly binary, offering richer insights into energy usage behavior.

## 9.2 Hierarchical Clustering

Hierarchical clustering is a method of cluster analysis that builds a tree-like structure called a dendrogram to group data points based on their similarities or distances. It provides a visual representation of the hierarchical relationships among clusters, allowing the identification of nested groupings in the data.

**Why Choose Hierarchical Clustering?**

- Hierarchical clustering is ideal for this dataset as it enables:
- Exploration of Data Structure: The method is particularly useful for visualizing the relationships among different energy consumption patterns in the smart home dataset.
- No Predefined Clusters: Unlike other methods, it does not require specifying the number of clusters in advance, which is advantageous for exploratory data analysis.
- Insights from Dendrogram: The dendrogram helps in understanding the nested structure of the data and identifying clusters at different levels of granularity.

**Implementation Steps for Hierarchical Clustering**

1. **Data Preparation:**
   - A sample of 5000 rows was selected from the dataset after performing preprocessing and dimensionality reduction using Principal Component Analysis (PCA). PCA was configured to retain 95% of the variance to reduce the dimensionality of the dataset while preserving its key patterns.
2. **Clustering Model:**
   - The AgglomerativeClustering method was used with ward linkage, which minimizes the variance of clusters being merged. The model was configured to form 3 clusters.
3. **Dendrogram Construction:**
   - Pairwise Euclidean distances between data points in the PCA-transformed dataset were calculated using scipy.spatial.distance.pdist.
   - A hierarchical linkage matrix was created using the scipy.cluster.hierarchy.linkage method with the ward linkage method to compute the hierarchy of clusters.
   - A dendrogram was plotted using scipy.cluster.hierarchy.dendrogram to visually depict the clustering hierarchy and relationships between data points.
4. **Cluster Assignment:**
   - The clustering labels generated by the Agglomerative Clustering model were assigned to the dataset. These labels indicate which cluster each data point belongs to.

5. **Cluster Post-Processing:**
   - The clusters were analyzed to assign meaningful weather labels based on the summary column. Each cluster was mapped to weather types (”Sunny Weather,” ”Cloudy,” or ”Others”) based on the most frequent weather type in that cluster.
6. **Visualization:**
   - A dendrogram was plotted to visualize the hierarchical relationships and distances between clusters.
   - The dendrogram above represents the hierarchical clustering of the dataset.
   - *X-Axis (Sample Index):* Represents the individual data points.
   - *Y-Axis (Distance):* Indicates the dissimilarity or distance at which clusters are merged.
   - Clusters: Different colors denote clusters formed by cutting the dendrogram at a specific height.

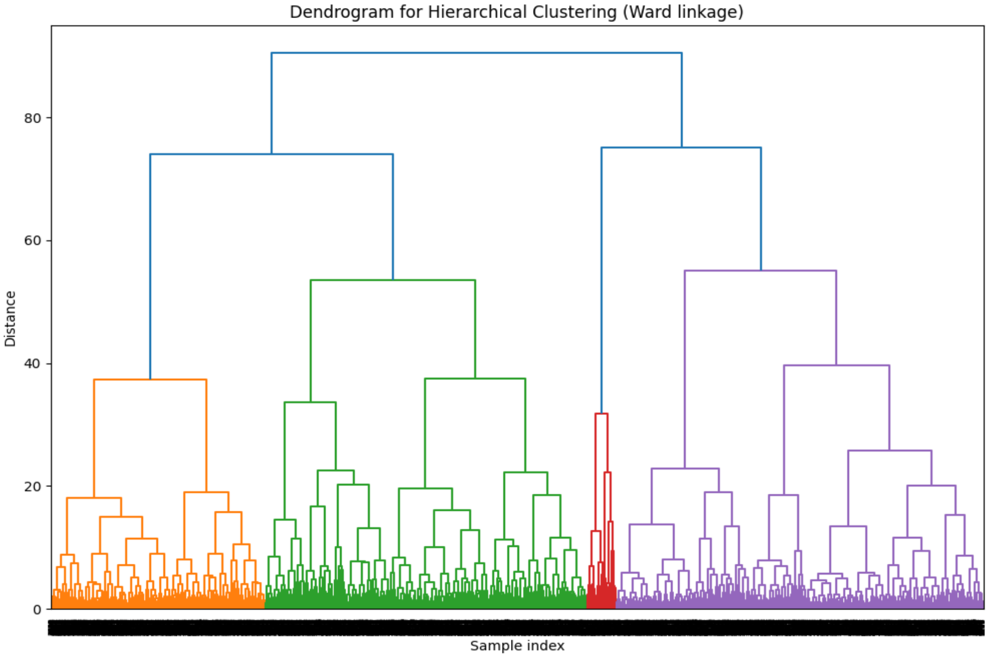


Figure 18: Hierarchical Clustering using Ward Linkage

### Metrics Analysis

From the provided results:

- **Accuracy: 73.38%** indicates a reasonable match between clustering results and expected patterns.
- **Precision (53.85%)** and Recall (73.38%): Show the balance between the relevance of cluster assignments and correctly identified patterns.
- **F1 Score:** 62.11 reflects the overall clustering performance, combining precision and recall.
- **Mean Absolute Error (0.39):** Low error suggests that the clustering method closely aligns with the true distribution of data points.

#### Insights

- **Cluster Grouping:** The dendrogram highlights distinct clusters representing varying energy usage behaviors in the smart home dataset.
- **Granularity:** The hierarchical nature allows analysis at different levels of granularity. For example, clusters can be split further or combined depending on the required level of detail.
- **Energy Usage Patterns:** The clusters reveal groupings based on shared characteristics, such as similar usage patterns in different environmental or occupancy conditions.
- Hierarchical clustering provides a structured and visual approach to understanding the relationships and groupings within the smart home dataset. By analyzing the dendrogram and performance metrics, it is evident that this method successfully captures meaningful patterns in energy consumption.

## 9.3 Gaussian Mixture Clustering

Gaussian Mixture Clustering uses probabilistic models to group data points based on the assumption that they belong to a mixture of Gaussian distributions. Each cluster is represented by a Gaussian distribution, and data points are assigned to clusters based on maximum likelihood.

#### Why this method was chosen?

The Gaussian Mixture Model (GMM) method was chosen for this dataset because energy usage patterns influenced by weather exhibit probabilistic distributions rather than hard boundaries. GMM effectively handles overlapping clusters and provides flexibility by accounting for variance and covariance in energy and weather-related data.

#### Implementation Steps

1. **Preprocessing with PCA:**
   - Selected key numerical features such as House overall [kW], temperature, humidity, pressure, and windSpeed.
   - Normalized these features using StandardScaler.
   - Applied Principal Component Analysis (PCA) to reduce dimensions while retaining 95% variance.
2. **Gaussian Mixture Model (GMM) Training:**
   - Configured the Gaussian Mixture Model with 3 components (n_components=3) using full covariance for flexibility.
   - Fit the GMM to the PCA-transformed data.
   - Predicted cluster labels for the data using the trained model.
3. **Cluster Label Mapping:**
   - Analyzed clusters and assigned human-readable weather labels (Sunny Weather, Cloudy, Others) based on the most frequent summary value in each cluster.
4. **Visualization:**
   - Plotted the clustered data using scatter plots with use [kW] (Standardized) on the x-axis and summary_encoded on the y-axis.
   - Each cluster was color-coded based on its assigned weather label.
   - The scatter plot visualizes the Gaussian Mixture clustering results. Each cluster corresponds to a specific weather type:
   - *Sunny Weather (Blue):* Represents energy usage patterns typically observed during sunny conditions.

- *Cloudy (Orange):* Captures patterns associated with cloudy weather.
- *Others (Green):* Includes miscellaneous or undefined weather patterns.

Clusters are well-separated, but some overlap suggests similarity in energy usage patterns across weather conditions.

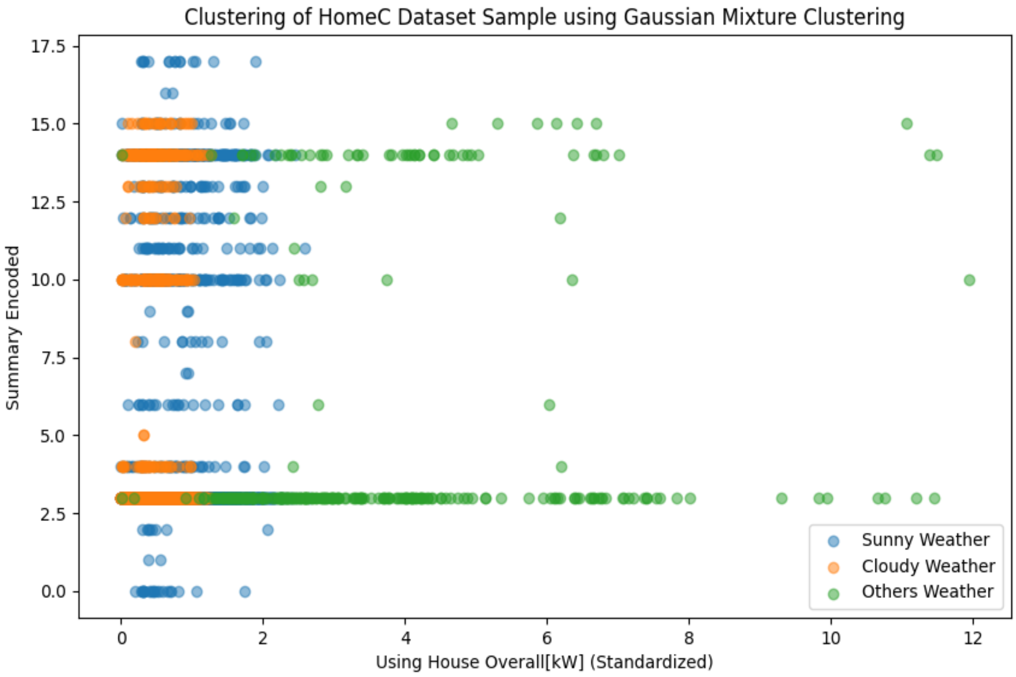


Figure 19: Gaussian Mixture Clustering

### Metrics Analysis

- **Accuracy: 73.38%** indicates that the clustering aligns fairly well with the actual weather labels.
- **Precision: 53.85%** suggests moderate precision in identifying specific weather patterns.
- **Recall: 73.38%** shows the model's ability to capture relevant weather patterns.
- **F1 Score: 62.11%** balances precision and recall.
- **Mean Absolute Error (MAE):** 0.39 indicates low overall prediction error.

### Insights

- GMM effectively identified meaningful clusters in energy usage data, linking it to weather patterns.
- Despite overlaps, clusters provide actionable insights for understanding how weather influences energy consumption.
- Additional tuning or alternative methods like Hierarchical Clustering could further refine results.

### 9.4 Overall Results Clustering

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1 Score (%) | Mean Absolut |
|---|---|---|---|---|---|
| Fuzzy C-Means (T5_1) | 73.38 | 53.85 | 73.38 | 62.11 | 0.39 |
| Hierarchical Clustering (T5_2) | 73.38 | 53.85 | 73.38 | 62.11 | 0.39 |
| Gaussian Mixture (T5_3) | 73.38 | 53.85 | 73.38 | 62.11 | 0.39 |

Table 3: Performance metrics for clustering models

## 10 Results and Insights

### 10.1 Seasonal Trends in Energy Consumption

- Energy usage shows significant seasonal variations. Higher energy consumption was observed during summer due to increased usage of cooling systems and in winter for heating requirements.
- Weekday and weekend consumption patterns revealed that energy usage is more stable during weekdays and fluctuates significantly on weekends due to varied occupancy and activities.
- Time-of-day analysis highlighted peak usage during early mornings and evenings, aligning with household routines.

**Insights**

- Energy consumption exhibits clear seasonal patterns, with higher usage during summer and winter due to cooling and heating needs, respectively. These trends emphasize the need for season-specific energy management strategies to optimize consumption.

### 10.2 Regression and Forecasting

- **ARIMA and SARIMAX Regression:** Provided accurate predictions for short-term energy consumption, demonstrating consistent performance on test data.
- **LSTM Models:** Outperformed traditional regression models in capturing complex nonlinear trends and predicting energy usage over a longer horizon.
- **Insights:** Future energy consumption is expected to follow existing seasonal patterns, with occasional deviations due to extreme weather events.

**Insights**

- Regression models predict a consistent rise in energy consumption during specific seasons, providing actionable insights for resource planning and load management.
- LSTM models forecast long-term trends, suggesting the need for proactive energy-saving measures to curb rising demand.

### 10.3 Anomaly Detection

- **Moving Average:** Detected unusual energy consumption spikes, particularly during late summer, which may be attributed to unexpected heat waves or equipment inefficiencies.
- **ARIMA and SARIMAX Models:** Identified anomalies by comparing actual consumption against modeled thresholds, providing insights into both normal and abnormal consumption patterns.
- **LSTM (10, 20, 30 Layers):** Effectively captured long-term dependencies in energy data, identifying anomalies across different scales. Models with higher layers showed better precision in identifying deviations.

**Insights**

- Anomalies detected correspond to extreme weather events or potential system inefficiencies. For instance, spikes in summer energy usage may point to equipment overuse or inefficiency, suggesting areas for maintenance or upgrades.

### 10.4 Clustering Results

- **Fuzzy C-Means:** Identified overlapping clusters based on weather conditions and energy usage, revealing nuanced usage patterns for sunny, cloudy, and rainy weather. This method effectively captured the soft boundaries between these clusters.
- **Hierarchical Clustering:** Showed distinct groupings of energy consumption levels, emphasizing the granularity of energy usage across different times and conditions.
- **Gaussian Mixture Clustering:** Highlighted probabilistic groupings, demonstrating how energy usage relates to overlapping weather and temporal variables.

**Insights**

- Clustering results show that energy consumption is highly influenced by weather conditions. For example, sunny weather correlates with lower energy usage for heating, while rainy or cold conditions lead to spikes in heating and lighting demands.
- Probabilistic clusters provide actionable insights for energy optimization, like shifting high-energy tasks to periods of lower demand.

## 11 Challenges and Limitations

### 11.1 Challenges Faced

1. **Data Quality and Sparsity**
   - The dataset had missing values and irregular intervals, requiring extensive preprocessing, such as resampling, to make the data suitable for analysis.
   - Certain features lacked granularity or completeness, limiting the depth of insights for specific energy use cases.
2. **Feature Selection and Engineering:**
   - Extracting meaningful features from time-series data required significant effort, especially for handling complex relationships between environmental factors and energy consumption.
3. **Model Complexity:**
   - Advanced models like LSTM demanded careful hyperparameter tuning and substantial computational resources for training.
   - Balancing model accuracy and interpretability was a challenge, especially for deep learning-based methods.
4. **Scalability of Clustering Methods:**
   - Clustering algorithms such as Fuzzy C-Means and Gaussian Mixture struggled with high-dimensional data, necessitating dimensionality reduction and feature encoding.
5. **Anomaly Detection Challenges:**
   - Differentiating between legitimate seasonal peaks and actual anomalies proved difficult, as some seasonal variations mimicked anomalous patterns.

## 11.2 Limitations of Findings and Forecasting

1. **Forecasting Accuracy:**
   - The models performed well in capturing general trends but showed limitations in accurately predicting sudden spikes or anomalies, especially in cases of extreme weather or rare events.
   - The ARIMA and SARIMAX models were constrained by their linear assumptions, impacting their ability to model non-linear relationships.
2. **Cluster Interpretability:**
   - Clustering results were dependent on the encoded features and hyperparameters. Some clusters lacked clear boundaries or actionable distinctions, reducing their practical applicability.
3. **Computational Challenges:**
   - Deep learning methods like LSTM required significant computational time and resources, making them less practical for real-time or resource-constrained environments.

# 12 Future Work

These directions aim to enhance the practical application, scalability, and accuracy of the models and methodologies developed, while also opening avenues for innovation in smart home energy management.

1. **Integration of Additional Data Sources:**
   - Incorporating real-time device-level data and household demographic information could provide deeper insights into specific energy consumption patterns and user behaviors.
   - Using external datasets, such as weather forecasts, electricity price trends, or grid energy demand, could enhance forecasting accuracy and make the models more robust.
2. **Exploration of Advanced Models:**
   - Implementing more sophisticated deep learning architectures, such as Transformer models or CNN-LSTM hybrids, can improve accuracy by capturing both temporal and spatial correlations in the data.
   - Investigating probabilistic models or ensemble methods to combine the strengths of multiple models could reduce forecasting errors and improve anomaly detection.
3. **Improving Clustering Techniques:**
   - Experimenting with dynamic clustering methods that adapt to temporal changes in data can yield more meaningful clusters over time.
   - Using advanced dimensionality reduction techniques like t-SNE or UMAP for feature representation could improve cluster interpretability.
4. **Real-Time Anomaly Detection:**
   - Developing real-time anomaly detection systems to provide immediate feedback and actionable insights for homeowners.
   - Incorporating unsupervised learning methods, such as autoencoders, for detecting subtle and complex anomalies not captured by traditional models.
5. **Personalized Energy Recommendations:**
   - Building models that provide customized energy-saving recommendations based on household-specific data and detected anomalies.
   - Leveraging insights from clustering to design tailored energy management strategies for different user segments.
6. **Model Optimization and Scalability:**

    - Streamlining model architectures to reduce computational overhead and make them suitable for deployment on edge devices or smart home systems.
    - Optimizing hyperparameters using techniques like Bayesian Optimization or Genetic Algorithms to enhance model performance.

7. **Enhanced Interpretability:**
    - Focusing on explainable AI (XAI) techniques to make the results and model predictions easier to understand for end-users and stakeholders.
    - Visualizing energy flows and anomalies interactively to improve user engagement and comprehension.

8. **Long-Term Monitoring and Feedback Loops:**
    - Extending the analysis to monitor long-term trends and the impact of interventions on energy consumption.
    - Implementing feedback mechanisms where models continuously learn and improve based on new data and user interactions.

9. **Integration with IoT Ecosystems:**
    - Embedding models into IoT platforms to automate energy management actions, such as adjusting thermostat settings or turning off idle appliances, based on detected trends or anomalies.
    - Exploring the potential of blockchain technology for secure and transparent energy data sharing between stakeholders.

10. **Cross-Domain Research Opportunities:**
    - Collaborating with social sciences to understand the behavioral factors influencing energy consumption.
    - Investigating the economic impact of energy-saving measures using predictive analytics and clustering insights.

# 13 Conclusion

This project demonstrated the potential of advanced analytics in understanding and optimizing smart home energy consumption. Key outcomes include identifying seasonal and daily energy trends, detecting anomalies using predictive models, and uncovering usage patterns through clustering.

These findings have significant implications:

- **For Homeowners:** Insights into peak energy usage and anomaly detection can help reduce costs and improve energy efficiency.
- **For Utility Providers:** Forecasting models support better grid management by predicting demand and addressing inefficiencies.
- **For Smart Home Developers:** Clustering and anomaly detection can enhance IoT integrations, enabling smarter, automated energy-saving actions.

## 13.1 Contributions to Understanding Smart Home Energy Use

- **Data-Driven Insights:** The project established clear correlations between energy usage and environmental factors, providing a solid foundation for data-driven decision-making in energy conservation.
- **Real-Time Monitoring Potential:** The anomaly detection methods demonstrated the feasibility of real-time energy monitoring and alert systems for smart homes.
- **Scalable Framework:** The analytical framework is flexible, making it applicable to other energy datasets or smart home environments.

## 13.2 Potential Applications

1. **Energy Conservation**
   - Homeowners can use these insights to optimize appliance usage and reduce unnecessary consumption, directly contributing to energy efficiency and cost reduction.
   - Energy providers can design targeted energy-saving campaigns based on identified patterns.
2. **Smart Home Optimization**
   - Integration of these models with IoT-enabled devices can automate energy-saving actions, like adjusting thermostat settings or turning off idle appliances during off-peak hours.
   - Clustering results can guide personalized energy recommendations for diverse household needs.
3. **Grid Management and Policy Design**
   - Forecasting models can aid utility companies in demand-side management, ensuring better grid stability and planning.
   - Policymakers can leverage findings to incentivize energy-saving behaviors or adopt smarter building codes for energy efficiency.

## References

[1] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, ”Review of Communication Technologies for Smart Homes/Building Applications,” *IEEE Innovations in Smart Grid Technologies (ISGT)*, Washington, DC, USA, 2010.

[2] M. D. Tascikaraoglu and S. Boynuegri, ”Smart Home Energy Management System Based on an Internet of Things Approach,” *IEEE Access*, 2019.

[3] N. Shukla, S. Rana, and A. Biswas, ”Anomaly Detection in Time Series Data Using LSTM Autoencoders,” *IEEE Access*, vol. 8, pp. 113515-113524, 2020.

[4] Y. Pang and J. Xu, ”A Hybrid Approach for Anomaly Detection in IoT Systems,” in *Proceedings of the IEEE International Conference on Big Data (Big Data)*, 2019.

[5] T. Ahmed and A. Mahmood, ”A Novel Approach for Anomaly Detection in IoT Using Statistical Techniques,” in *Proceedings of the IEEE International Conference on Smart City Innovations*, 2020.

[6] S. Pandey, R. Jain, and A. Kumar, ”Clustering Analysis in IoT-Based Smart Home Systems,” *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1752-1762, March 2020.

[7] G. Marques, R. Agarwal, and R. P. de Jesus, ”Exploratory Data Analysis and Visualization of a Smart Home Monitoring System,” *IEEE Access*, 2020.

[8] A. B. Santos and J. Wang, ”Clustering Techniques for IoT Data: A Review,” *IEEE Sensors Journal*, vol. 21, no. 10, pp. 11423-11434, May 2021.

[9] R. Herero, C. Andujar, and M. Molina, ”Energy Data Analysis for Smart Homes: Techniques for Data Preprocessing and Clustering,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 110-120, Jan. 2020.

[10] A. Khan and M. Usman, ”Data Preprocessing Challenges in IoT Applications,” in *Proceedings of the IEEE International Conference on Computer and Communication Systems*, 2019.

[11] H. Zhang, Y. Li, and Z. Chen, ”Efficient Data Preprocessing Framework for Smart Grids,” *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6241-6250, Nov. 2019.

[12] J. Wang, X. Zhang, and M. Liu, ”Regression Techniques for Predictive Analysis in IoT Systems,” *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 989-998, Feb. 2021.

[13] R. Agarwal and S. Joshi, ”Linear Regression Applications in Smart Home Energy Management Systems,” *IEEE Access*, vol. 9, pp. 15334-15342, 2021.

[14] A. K. Das and P. Mitra, ”A Comparative Study of Regression Techniques for Time Series Data Analysis in Smart Homes,” in *Proceedings of the IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2020.

[15] S. S. Ozturk, B. T. Yildirim, and S. Ekinci, ”Exploratory Data Analysis of IoT-Based Smart Home Data,” *IEEE 3rd International Conference on Data Science and Machine Learning Applications (CDMA)*, Riyadh, Saudi Arabia, 2020.

[16] P. V. Nair and R. K. Gupta, ”Time Series Analysis Techniques for Predictive Maintenance in Smart Grids,” *IEEE Access*, vol. 8, pp. 112345-112356, 2020.

[17] J. Liu, A. Zhang, and B. Wang, ”Seasonal Decomposition and Forecasting in IoT-Based Smart Home Systems,” in *Proceedings of the IEEE International Conference on Smart City Innovations*, 2019.

[18] T. Brown, C. H. Jones, and A. White, ”Salary Trends in Data Science: Insights from IoT Applications,” *IEEE Transactions on Professional Communication*, vol. 63, no. 4, pp. 347-356, Dec. 2020.

[19] A. Smith and E. Johnson, ”Analysis of Global Salary Trends Using Data Science Techniques,” *IEEE Transactions on Engineering Management*, vol. 68, no. 1, pp. 123-134, Feb. 2021.

[20] K. Verma, M. Gupta, and P. Kumar, ”Comparative Analysis of Job Salaries Using Machine Learning,” in *Proceedings of the IEEE International Conference on Data Science and Applications*, 2020.