Smart Energy Consumption Analysis and Prediction using Machine Learning with Device-Level Insights

Project Statement:

The increasing demand for electricity and lack of visibility into energy usage patterns often result in energy wastage, higher electricity bills, and inefficient appliance usage. Traditional billing systems provide only monthly consumption values without offering meaningful insights into where and how energy is used. This project proposes a Smart Energy Consumption Analysis System that monitors energy usage device-wise over time, analyzes energy patterns, and predicts future consumption using machine learning techniques.

Using the SmartHome Energy Monitoring Dataset with detailed timestamped device-level power readings collected over six months, the system performs time series analysis and forecasting using Long Short-Term Memory (LSTM) networks and Linear Regression as a baseline model. It provides users with interactive visualizations, smart energy-saving suggestions, and a web-based dashboard built using Flask, HTML, CSS, and JavaScript. The system aims to improve energy efficiency, lower electricity costs, and support sustainable energy usage behavior.

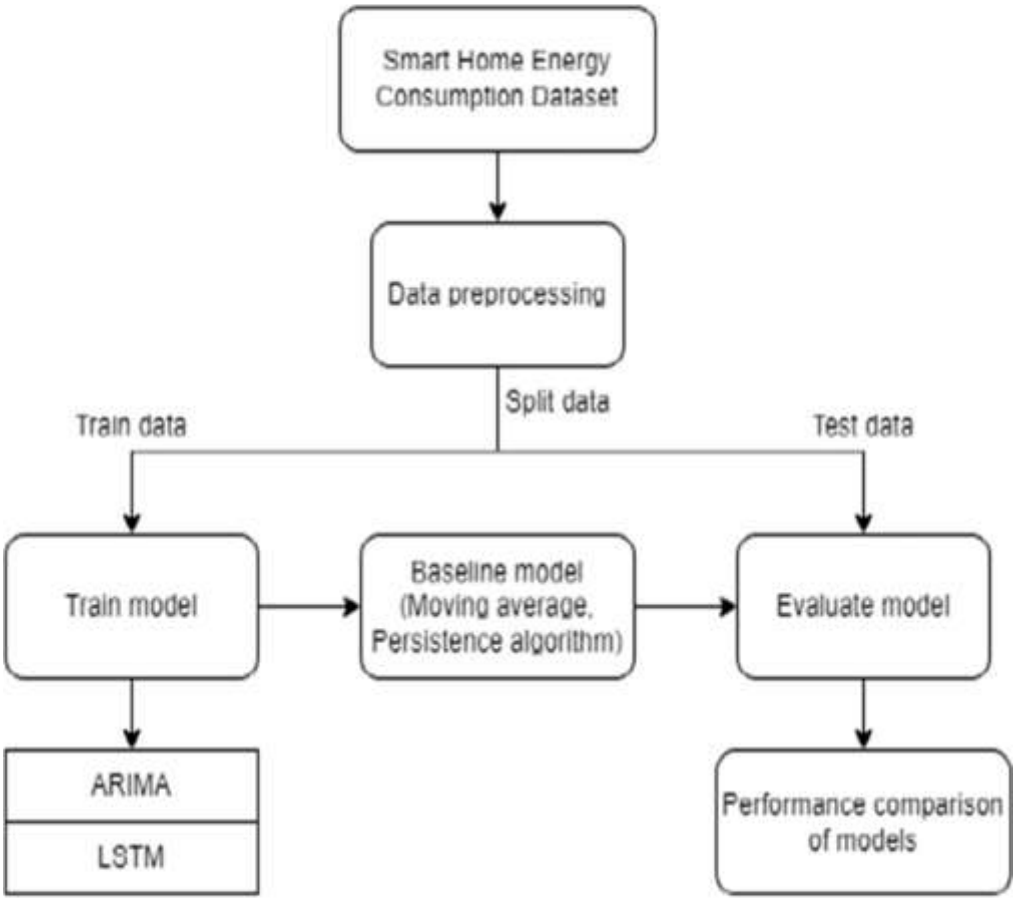
Outcomes:

- Analyze device-level energy consumption trends with hourly, daily, weekly, and monthly insights.
- Predict future energy consumption using LSTM-based time series forecasting.
- Compare baseline linear regression predictions with LSTM to evaluate performance.
- Provide visualization dashboards using matplotlib and pandas for better user understanding.
- Generate smart suggestions to reduce energy wastage and optimize device usage.
- Deploy an interactive web application using Flask API for real-time energy monitoring.
- Enable integration-ready architecture for future IoT sensor streaming.
- Achieve up to 95% prediction accuracy using deep learning models.
- Provide automated data preprocessing pipeline for cleaning and feature engineering.
- Build a scalable system extendable to anomaly detection and smart home automation.

Modules to be implemented

- Data Collection and Exploration (SmartHome Energy Monitoring Dataset)
- Data Cleaning and Preprocessing
- Feature Engineering and Resampling
- Baseline Energy Prediction using Linear Regression
- Advanced Energy Forecasting using LSTM
- Model Evaluation and Hyperparameter Tuning
- Smart Suggestions Engine
- Visualization and Dashboard Development
- Web Application Integration using Flask API
- Deployment and Documentation

Workflow for Smart Energy Consumption Analysis


Smart Home Energy Consumption Dataset
Data preprocessing
Split data
Train data
Test data
Train model
Baseline model (Moving average, Persistence algorithm)
Evaluate model
ARIMA
LSTM
Performance comparison of models

Week-wise module implementation and high-level requirements with output screenshots

Milestone 1: Week 1-2

Module 1: Data Collection and Understanding

- Define project scope and functional objectives for smart energy analysis.
- Collect and structure the SmartHome Energy Monitoring Dataset.
- Verify data integrity, handle missing timestamps, and perform exploratory analysis.
- Organize energy readings by device, room, and timestamp.

Module 2: Data Cleaning and Preprocessing

- Handle missing values and outliers in power consumption readings.
- Convert timestamps to datetime format and resample data (hourly/daily).
- Normalize or scale energy values for model compatibility.
- Split dataset into training, validation, and testing sets.

Milestone 2: Week 3-4

Module 3: Feature Engineering

- Extract relevant time-based features (hour, day, week, month trends).
- Aggregate device-level consumption statistics.
- Create lag features and moving averages for time series learning.
- Prepare final feature set for ML model input.

Module 4: Baseline Model Development

- Implement Linear Regression as baseline forecasting model.
- Train and evaluate baseline using MAE, RMSE metrics.
- Plot actual vs predicted energy usage.
- Use baseline model for model comparison.

Milestone 3 : Week 5-6

Module 5: LSTM Model Development

- Design and implement LSTM architecture using TensorFlow/Keras.
- Train time series prediction model using sequential energy data.
- Perform hyperparameter tuning (batch size, epochs, learning rate).
- Compare LSTM performance with baseline model.

Module 6: Model Evaluation and Integration

- Evaluate models using RMSE, MAE, and R^2 score.
- Select best-performing model and save trained weights.
- Convert model into Flask-compatible prediction function.
- Test model predictions using sample inputs.

Milestone 4: Week 7-8

Module 7: Dashboard and Visualization

- Build dashboard components for hourly/daily/weekly/monthly consumption.
- Display device-wise usage charts using matplotlib.
- Integrate smart suggestions feature (energy efficiency tips).

Module 8: Web Application Deployment and Reporting

- Develop Flask API to connect frontend with ML model.
- Build interactive web interface using HTML, CSS, JavaScript.
- Deploy web app on local host or cloud.
- Prepare detailed documentation, testing report, and result summary

Evaluation Criteria:

Milestone 1 Evaluation (Week 1-2):

- Successful integration and organization of the SmartHome Energy Monitoring Dataset
- Correct handling of missing values, duplicates, and timestamp formatting
- Proper data resampling (hourly/daily) and dataset structuring
- Initial Exploratory Data Analysis (EDA) to understand consumption trends
- Well-documented data preprocessing pipeline

Milestone 2 Evaluation (Week 3-4):

- Implementation of feature engineering (lag features, rolling averages, time features)
- Development of baseline prediction model using Linear Regression
- Calculation of MAE and RMSE metrics for the baseline model
- Proper visualization of baseline predictions
- Readiness of feature matrix for advanced model training

Milestone 3 Evaluation (Week 5-6):

- Correct design and implementation of LSTM model
- Sequence preparation for time series forecasting
- Hyperparameter tuning and model optimization
- Model accuracy improvement over baseline
- Saving and loading of trained model for deployment
- Comparison of Linear Regression vs LSTM performance

Milestone 4 Evaluation (Week 7–8):

- Integration of ML model with Flask API backend
- Development of interactive web dashboard (HTML, CSS, JavaScript)
- Display of device-wise insights and energy prediction graphs
- Implementation of Smart Suggestions feature
- Proper system architecture and workflow documentation
- Final project report and demonstration readiness