AI-ENABLED DIGITAL TWIN FRAMEWORK FOR REAL-TIME MONITORING AND OPTIMIZATION OF HOME ELECTRICITY USAGE

Project Reference No.: *49S_BE_2351*

College	:	*SEA College of Engineering and Technology, Bengaluru*
Branch	:	*Department of Computer Science and Engineering (CSE)*
Guide(s)	:	*Dr. Krishna Kumar P R*
		Mrs. Saswati Behera
Student(s)	:	*Mr. Srujan K S*
		Mr. Tejas H A
		Mr. Pruthvi H J
		Mr. Abhishek I C

Keywords:

Digital Twin, Energy Management System, Machine Learning, Energy Consumption Prediction, Anomaly Detection, Smart Home Analytics, Predictive Modeling, XGBoost, LightGBM, FastAPI, React.js, Data Visualization, Time-Series Analysis, Energy Optimization, Sustainable Computing.

Introduction:

1. Energy consumption in residential sectors has been increasing significantly due to rapid urbanization and the growing use of electrical appliances. Managing this energy efficiently has become a major challenge, especially with rising electricity costs and environmental concerns. Traditional energy monitoring systems mainly provide aggregate consumption data through monthly bills, which do not offer detailed insights into usage patterns or areas of inefficiency. As a result, users are unable to make informed decisions to optimize their energy consumption.
2. With the advancement of Artificial Intelligence and Machine Learning, more intelligent and data-driven approaches can be used to address these challenges. One such approach is the concept of a Digital Twin, which creates a virtual representation of a physical system. In the context of energy management, a digital twin can model household energy usage and provide real-time insights, predictions, and recommendations.
3. This project, titled **"AI-Enabled Digital Twin Framework for Real-Time Monitoring and Optimization of Home Electricity Usage,"** focuses on developing a software-based solution to monitor, analyze, and optimize household energy consumption. The system uses historical energy data to build machine learning models that can predict future energy usage and detect abnormal consumption patterns. It eliminates the need for additional hardware by working with existing datasets and user inputs.
4. The system is designed with a user-friendly web interface that allows users to visualize energy consumption trends, receive personalized suggestions, and perform what-if analysis to understand the impact of different usage scenarios. Technologies

such as FastAPI for backend development and React.js for frontend visualization are used to ensure scalability and performance.

5. Overall, this project aims to bridge the gap between basic energy monitoring and intelligent energy management by leveraging AI-driven insights. It contributes to reducing energy costs, improving efficiency, and promoting sustainable living practices.

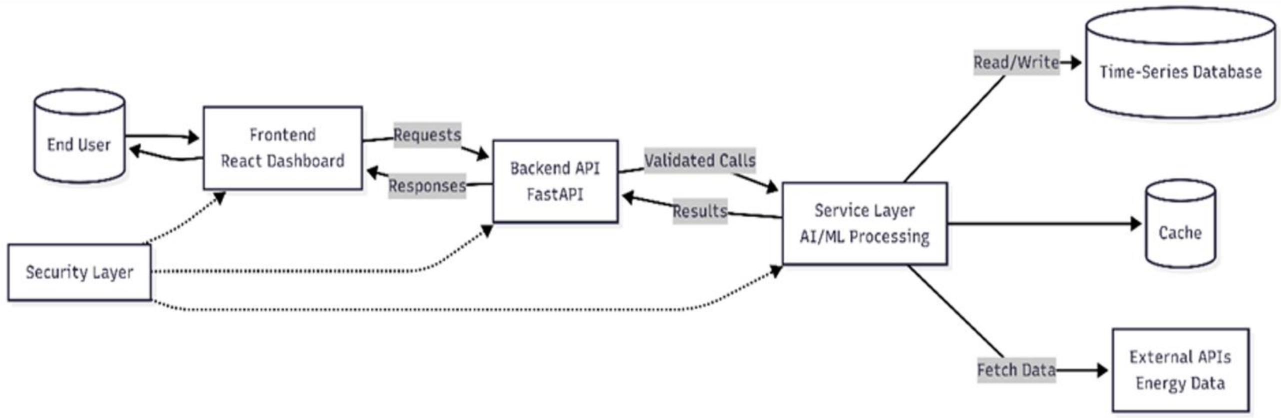

Fig 1: This figure illustrates the overall system architecture, showing the interaction between the user interface, backend API, AI/ML processing layer, and data storage components.

Objectives:

1. To develop an AI-powered digital twin for modeling and monitoring household energy consumption.
2. To analyze historical data and predict future energy usage using machine learning techniques.
3. To detect anomalies and identify inefficient energy usage patterns.
4. To provide an interactive and user-friendly dashboard for real-time monitoring and visualization.
5. To generate personalized recommendations and enable what-if analysis for energy optimization.
6. To design a scalable, software-based system that promotes cost savings and sustainable energy practices.

Methodology:

1. The proposed system follows a systematic and structured methodology to design and develop an AI-powered digital twin for efficient household energy management. Initially, historical energy consumption data is collected from available datasets or through manual user inputs. This data typically includes time-based energy readings such as hourly, daily, and monthly consumption of various household appliances. The collected data is stored in a suitable format such as time-series databases to support further analysis and model training. In the preprocessing stage, the raw data undergoes cleaning to remove missing, duplicate, or inconsistent values. Feature engineering is performed by extracting meaningful attributes such as hour of the day, day of the week, and seasonal trends, which significantly influence energy consumption. The data is then normalized and scaled to improve the efficiency and accuracy of machine learning algorithms. This processed data is split into training

and testing datasets for model development. Machine learning models such as XGBoost, LightGBM, and regression algorithms are implemented to analyze energy consumption patterns. These models are trained using the training dataset to learn relationships between different variables affecting energy usage. After training, the models are tested on unseen data to evaluate their performance. Evaluation metrics such as Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) are used to measure prediction accuracy. The best-performing model is selected and saved for deployment to avoid repeated training.

2. In addition to prediction, anomaly detection techniques are incorporated to identify abnormal or inefficient energy usage patterns. These techniques help in detecting sudden spikes or unusual behavior in energy consumption, enabling users to take corrective actions. Alerts and notifications can be generated based on these anomalies to improve energy efficiency and reduce wastage. The backend of the system is developed using FastAPI, which is responsible for handling data processing, executing machine learning models, and managing API requests. The backend communicates with the database to store and retrieve energy data, predictions, and user information. Efficient data handling and caching mechanisms are used to ensure fast response times and scalability of the system.
3. The frontend is developed using React.js, which provides a dynamic and interactive user interface. Users can visualize their energy consumption through graphs, charts, and dashboards. The interface allows users to view real-time data, historical trends, predictions, and anomaly alerts in an easy-to-understand format. This improves user engagement and helps in better decision-making.
4. The system follows a client-server architecture where the frontend communicates with the backend through RESTful APIs. When a user requests data, the frontend sends a request to the backend, which processes the request, retrieves the required data, performs predictions if needed, and sends the results back to the frontend for display. This modular architecture ensures scalability, maintainability, and efficient system performance.
5. A what-if analysis module is also implemented to allow users to simulate different energy usage scenarios. Users can modify parameters such as appliance usage or time duration and observe the impact on overall energy consumption. This feature helps users make informed decisions regarding energy optimization.
6. The overall system design is supported by various diagrams such as system architecture diagrams, data flow diagrams (DFD), and entity-relationship (ER) diagrams. These diagrams clearly illustrate the interaction between different components such as users, devices, energy readings, prediction models, and alert systems (as shown in the project report).

Result and Conclusion:

Test	Description	Expected Outcome	Outcome Result	Status
Data Collection	Collects historical energy consumption data	Accurate and complete dataset	Data successfully collected and stored	Pass
Data Preprocessing	Cleans and prepares raw energy data	Clean and normalized dataset	Missing values handled and data normalized	Pass
ML Model Training	Trains models (XGBoost, LightGBM)	Accurate learning of patterns	Models trained with good accuracy	Pass
Energy Prediction	Predicts future energy consumption	Reliable and accurate forecasts	Predictions generated with low error (RMSE/MAPE)	Pass
Anomaly Detection	Identifies unusual energy usage	Detect abnormal consumption patterns	Abnormal usage successfully detected	Pass
Dashboard Visualization	Displays energy data and insights	Clear and interactive visualization	Data shown correctly using charts/graphs	Pass
API & Backend	Handles data processing and requests	Smooth backend performance	FastAPI handled requests efficiently	Pass
What-if Analysis	Simulates different usage scenarios	Accurate simulation results	Scenario analysis working correctly	Pass

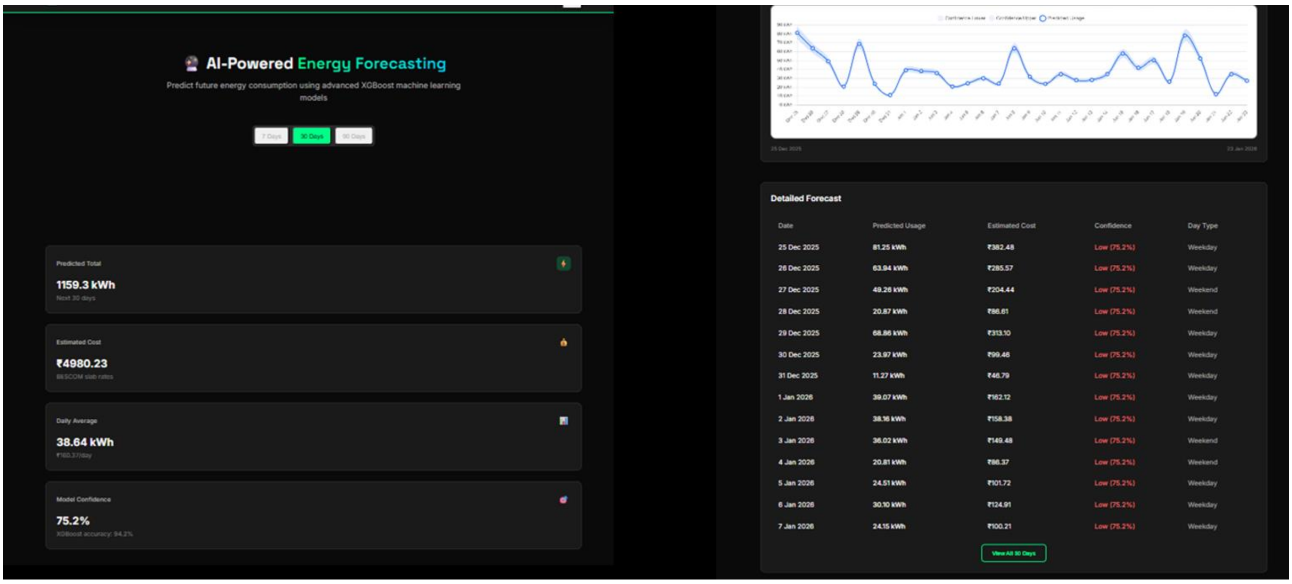

Fig 2: The figure shows the energy forecasting dashboard, where the system predicts future energy consumption, estimated cost, and daily usage trends using machine learning models.

In conclusion, the “AI-Enabled Digital Twin Framework for Real-Time Monitoring and Optimization of Home Electricity Usage” successfully demonstrates an intelligent system for monitoring and optimizing household energy consumption. The project effectively uses machine learning models to predict future energy usage and detect anomalies in consumption patterns. The interactive dashboard provides clear visualization and actionable insights for users. The system helps in reducing energy wastage and improving cost efficiency. Its scalable and software-based design makes it practical for real-world applications.

References:

- Jones et al., “Characterising the Digital Twin: A Systematic Literature Review,” CIRP Journal of Manufacturing Science and Technology, 2020.
- Helmstaedter et al., “Connectomic Reconstruction of the Inner Plexiform Layer in the Mouse Retina,” Nature, 2013.
- Silver et al., “Mastering the Game of Go with Deep Neural Networks and Tree Search,” Nature, 2016.
- Jumper et al., “Highly Accurate Protein Structure Prediction with AlphaFold,” Nature, 2021.
- Fahim et al., “Machine Learning-Based Digital Twin for Predictive Modeling in Wind Turbines,” IEEE Access, 2022.

Future Scope:

The future scope of this project includes:

1. Enhancement of prediction models using advanced deep learning techniques for better long-term forecasting.
2. Development of a mobile application for real-time monitoring, alerts, and user-friendly visualization.
3. Implementation of automated appliance control for intelligent energy optimization without manual intervention.
4. Integration with renewable energy sources such as solar systems for efficient and sustainable energy management.