

Energy monitoring and anomaly detection system

An Energy Monitoring and Anomaly Detection System for Residential Halls

March 26, 2025

wESLEY OMOKE- SCT212-0082/2021

BCT 2406; PROJECT – BSC CT YEAR 4

SUPERVISOR 1: DR. MICHAEL KIMWELE

SUPERVISOR 2: MR. JOHN WAINAINA

# Abstract

This project proposal outlines the development and implementation of an energy monitoring and anomaly detection system for residential halls on university campuses. The primary goal of this initiative is to enhance energy efficiency and improve safety through real-time monitoring, predictive analytics, and user engagement. The proposed system will include three main components: the Internet of Things (IoT), Machine Learning (ML), and Human-Computer Interaction (HCI).

IoT sensors will be strategically deployed to collect real-time data on energy usage, including patterns from electrical consumption. This data will be transmitted to a centralized cloud database for analysis using ML algorithms. The ML component will focus on identifying energy consumption patterns, forecasting usage trends, and detecting anomalies that may indicate inefficiencies or potential electrical faults. When such anomalies are detected, the system will generate alerts for facility managers.

An intuitive user interface, developed as part of the HCI Component will enable residents to monitor their energy usage in real-time, receive personalized insights, and adopt energy-saving behaviors. The system’s goal is to improve operational efficiency and promote safety while contributing to a sustainable energy future.

To validate the system’s effectiveness, the project will involve conducting a case study within the residential halls of Jomo Kenyatta University of Agriculture and Technology. This will involve collecting both quantitative data from IoT sensors and qualitative feedback through surveys and interviews with residents.

Table of Contents

[Abstract 1](#_Toc193862802)

[List of Abbreviations 6](#_Toc193862803)

[Chapter 1: Introduction 8](#_Toc193862804)

[1.1 Background and Motivation 8](#_Toc193862805)

[1.2 Problem Statement 8](#_Toc193862806)

[1.3 Proposed Solution 9](#_Toc193862807)

[1.4 Objectives 10](#_Toc193862808)

[Chapter 2: Literature Review 10](#_Toc193862809)

[2.1 Theoretical Review of Concepts Involved with Energy Monitoring and Anomaly Detection 10](#_Toc193862810)

[2.1.1 Unsupervised Learning for Energy Monitoring and Anomaly Detection 10](#_Toc193862811)

[2.1.2 Deep Learning in Building Energy Management 15](#_Toc193862812)

[2.1.3 The Role of Data-Driven Methodologies in Building Energy Consumption Forecasting 17](#_Toc193862813)

[2.1.4 Anomaly Detection and Diagnosis (ADD) in Buildings 20](#_Toc193862814)

[2.1.5 Hybrid Machine Learning Models for Energy Consumption Forecasting 23](#_Toc193862815)

[2.1.6 Predictive Analytics in Energy Monitoring Systems 24](#_Toc193862816)

[2.1.7 Time Series Forecasting for Energy Management 26](#_Toc193862817)

[2.2 Related Works 28](#_Toc193862818)

[2.2.1 Case Studies of Energy Monitoring and Anomaly Detection Systems 28](#_Toc193862819)

[Chapter 3: Methodology 30](#_Toc193862820)

[3.1 System Development 30](#_Toc193862821)

[Overview 30](#_Toc193862822)

[3.1.1 Data Collection and Preprocessing 31](#_Toc193862823)

[3.1.2 Anomaly Detection Framework 33](#_Toc193862824)

[3.1.3 Energy Consumption Forecasting Framework 34](#_Toc193862825)

[3.2 Technologies 35](#_Toc193862826)

[System Integration 36](#_Toc193862827)

[3. 3 Research Design 36](#_Toc193862828)

[3.3.1 Experimental Setup 36](#_Toc193862829)

[3.3.2 Pilot Deployment 37](#_Toc193862830)

[3.3.3 Comparison with Historical Data 37](#_Toc193862831)

[3.3.4 Controlled Anomaly Testing 37](#_Toc193862832)

[3.3.5Validation Metrics 37](#_Toc193862833)

[3.4 Ethical Considerations 38](#_Toc193862834)

[3.5 Expected Outcomes 38](#_Toc193862835)

[3.5.1 Intended Outputs 39](#_Toc193862836)

[Chapter 4: System Analysis and Design 40](#_Toc193862837)

[4.1 Design Methodology 40](#_Toc193862838)

[4.1.1 Design Approach 40](#_Toc193862839)

[4.1.2 Justification for Using Object-Oriented Design 40](#_Toc193862840)

[4.1.3 Comparison with Alternative Approaches 41](#_Toc193862841)

[4.2 Machine Learning Component Design 41](#_Toc193862842)

[4.3 Data Characteristics and Feature Engineering 43](#_Toc193862843)

[4.3.1 Data Description 43](#_Toc193862844)

[4.3.2 Preprocessing and Feature Extraction 44](#_Toc193862845)

[4.3.3 Integration in the ML Pipeline 45](#_Toc193862846)

[4.4 System Requirements 45](#_Toc193862847)

[4.4.1 Functional Requirements 45](#_Toc193862848)

[4.4.2 Non-Functional Requirements 46](#_Toc193862849)

[4.5 Technology Stack 47](#_Toc193862850)

[4.5.1 Software and Libraries 47](#_Toc193862851)

[4.5.2 Development Tools and Environments 47](#_Toc193862852)

[4.5.3 Data Storage and Integration 47](#_Toc193862853)

[4.5.5 Infrastructure and Deployment 48](#_Toc193862854)

[4.6 Visual Models and UML Diagrams 48](#_Toc193862855)

[4.6.1. ML Pipeline 48](#_Toc193862856)

[4.6.1 Use Case Diagram 51](#_Toc193862857)

[4.7.1Feasibility Analysis and Risk Assessment 53](#_Toc193862858)

[4.7.2 Risk Assessment and Mitigation Strategies 54](#_Toc193862859)

[Conclusion 57](#_Toc193862860)

[References 58](#_Toc193862861)

[Appendices 60](#_Toc193862862)

[Project Plan and Timeline 60](#_Toc193862863)

[Budget 60](#_Toc193862864)

**List of Figures**

[Figure 1 Envisioned System Architecture 9](#_Toc193313929)

[Figure 2 k-Means Clustering 11](#_Toc193313930)

[Figure 3 Hierarchical Clustering 12](#_Toc193313931)

[Figure 4 Network Architecture for Ordinary Autoencoder 12](#_Toc193313932)

[Figure 5 Symbolic aggregate approximation of a time series data. 13](#_Toc193313933)

[Figure 6 Illustration of Adaptive Symbolic Aggregation Approximation 14](#_Toc193313934)

[Figure 7 Recurrent Neural Network 15](#_Toc193313935)

[Figure 8:LSTM Network 19](#_Toc193313936)

[Figure 9 Boxplot Analysis 21](#_Toc193313937)

[Figure 10 Illustration of Time Series Forecasting using ARIMA Model 29](#_Toc193313938)

[Figure 11 Proposed System Architecture 32](#_Toc193313939)

[Figure 12: Machine Learning Framework 35](#_Toc193313940)

[Figure 13: ML Pipeline Diagram 47](#_Toc193313941)

[Figure 14:Use Case Diagram 48](#_Toc193313942)

[Figure 15: Class Diagram 49](#_Toc193313943)

[Figure 16: Project plan and timeline 53](#_Toc193313944)

[Figure 17: Project budget 54](#_Toc193313945)

### List of Abbreviations

ADD – Anomaly Detection and Diagnosis

AHU – Air Handling Unit

ANN – Artificial Neural Network

API – Application Programming Interface

ARIMA – Autoregressive Integrated Moving Average

ARM – Association Rule Mining

AWS – Amazon Web Services

BCT – Bachelor of Computer Technology

BEMS – Building Energy Management System

BMS – Building Management System

BPNN – Backpropagation Neural Network

CAD – Contextual Anomaly Detection

CNN – Convolutional Neural Network

CT – Classification Tree

DL – Deep Learning

EMS – Energy Management System

EIS – Energy Information System

FNR – False Negative Rate

FPR – False Positive Rate

FTS – Fuzzy Time Series

GPU – Graphics Processing Unit

HCI – Human-Computer Interaction

HVAC – Heating, Ventilation, and Air Conditioning

IoT – Internet of Things

IQR – Interquartile Range

JKUAT – Jomo Kenyatta University of Agriculture and Technology

LSTM – Long Short-Term Memory

ML – Machine Learning

MLP – Multi-Layer Perceptron

NEC – Nippon Electric Company

NN – Neural Network

PAA – Piecewise Aggregate Approximation

RF – Random Forest

RNN – Recurrent Neural Network

RMSE – Root Mean Squared Error

SAX – Symbolic Aggregate Approximation

SVM – Support Vector Machine

TPU – Tensor Processing Unit

WSN – Wireless Sensor Network

XAI – Explainable Artificial Intelligence

# Chapter 1: Introduction

## 1.1 Background and Motivation

Energy management, especially in educational institutions is a crucial concern since it impacts both operational costs and environmental sustainability. Inefficient energy management practices, coupled with the lack of real-time monitoring systems, have led to excessive energy consumption in the past. This not only increases operational costs but also contributes significantly to the institution’s carbon footprint, undermining sustainability efforts.

Moreover, the absence of systems capable of detecting and addressing anomalies in power consumption exacerbates these issues. Anomalies, such as sudden spikes or irregular patterns in energy usage, often indicate electrical faults or inefficiencies that, if left unresolved, can escalate into safety hazards. These hazards may include equipment failure or electrical fires, posing risks to the safety and well-being of residents and facility infrastructure (Bhargavi & Yashasvi, 2021).

Advancements in technology, particularly in IoT, ML, and HCI, present an opportunity to address these challenges. IoT devices enable real-time monitoring of energy consumption through sensor networks, while ML algorithms analyze collected data to identify patterns, forecast energy usage, and detect anomalies. The HCI component, through intuitive user interfaces, ensures that actionable insights are effectively communicated to residents and facility managers, encouraging energy-efficient behaviors, and enabling timely response to potential issues (Arias-Requejo et al., 2023). In Kenya, the implementation of Energy Management Systems (EMS) in households has shown great promise paving the way for its potential integration in institutional settings (Makaa, 2024).

The motivation for this project stems from the need to reduce energy wastage, improve safety and promote sustainability in residential halls. The implementation of an Energy Monitoring and Anomaly Detection System will help provide comprehensive insights into energy consumption patterns, ensure timely detection of anomalies, and foster a culture of energy conservation among the residents.

## 1.2 Problem Statement

Energy consumption in residential halls on university campuses presents several challenges that hinder sustainability, and safety. Despite growing concerns of energy efficiency, many residential halls still face high energy consumption due to outdated energy management practices and the absence of real-time monitoring systems. These inefficiencies contribute to elevated operational costs and an increased carbon footprint, negatively impacting both institutions and the environment.

Furthermore, the lack of an effective anomaly detection system adds to these challenges, as unaddressed anomalies in power consumption may signify potential electrical faults. If left unchecked, these faults can escalate into safety hazards, such as electrical fires or equipment failures, posing significant risks to residents and facility infrastructure.

## 1.3 Proposed Solution

To address these pressing issues, there is need for an energy monitoring and anomaly detection system that integrates Internet of Things (IoT) technology, machine learning, and human-computer interaction (HCI). Such a system will provide:

* Real-time Energy Monitoring: Using IoT sensors to collect and analyze energy consumption data.
* Anomaly detection: leveraging ML techniques to identify unusual energy usage patterns indicative of inefficiencies or potential electrical faults.
* Safety Alerts: automatically sending alerts to facility managers to enable proactive interventions before issues escalate.
* User engagement: Through an intuitive user interface, residents will gain access to personalized insights, encouraging energy-efficient behaviors and fostering sustainability.

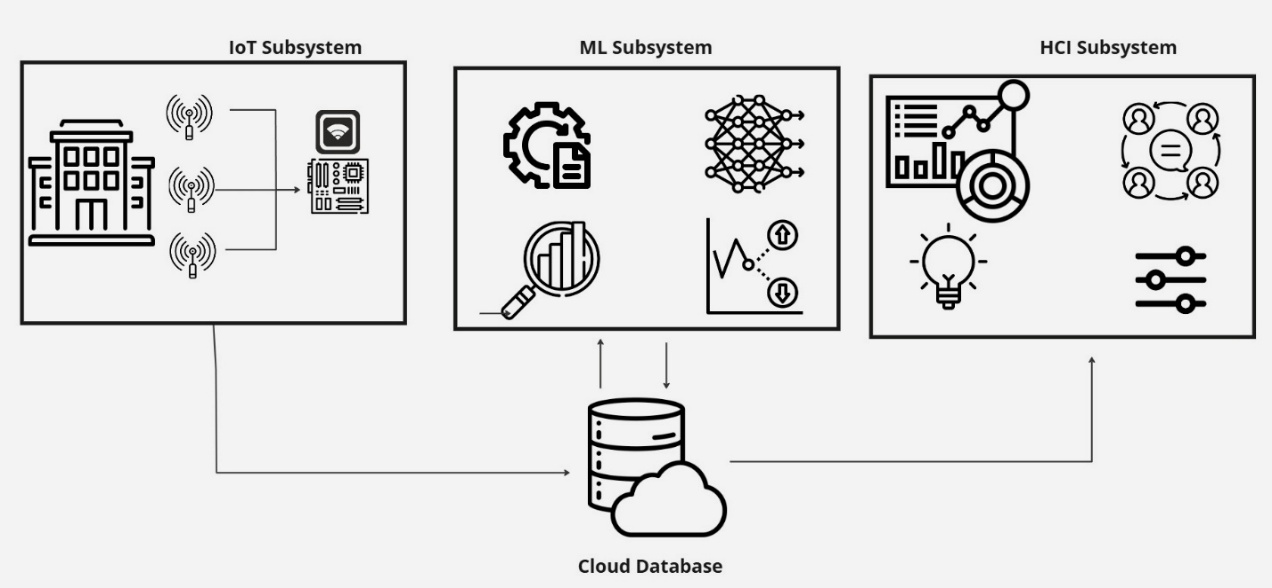


Figure 1 Envisioned System Architecture

The implementation of the system above will be a team of 3, with each having an area of focus tied to a certain computing aspect which when put together comprise the entire system.

* David Muchiri SCT212-0367/2020 Internet of Things
* Wesley Omoke SCT212-0082/2021 Machine Learning
* Carson Mwirigi SCT21-0180/2021 Human-Computer Interaction

## 1.4 Objectives

The primary objective of this project is to develop and implement an energy monitoring and anomaly detection system for residential halls. This system aims to enhance energy efficiency, reduce consumption, and improve safety through real-time monitoring, anomaly detection, and user engagement.

The specific objectives are as follows:

1. To deploy a network of sensors throughout the residential halls that will assist in obtaining the load consumption of the respective buildings.
2. To investigate the application of various machine learning algorithms such as Isolation Forest and Long Short-Term Memory networks (LSTM) in electrical load forecasting and anomaly detection.
3. To design and implement an intuitive mobile application that allows residents to monitor their energy usage in real time and receive personalized recommendations for reducing consumption as well as receiving alerts on predicted safety hazards arising from anomalies in energy flow.

# Chapter 2: Literature Review

## 2.1 Theoretical Review of Concepts Involved with Energy Monitoring and Anomaly Detection

### 2.1.1 Unsupervised Learning for Energy Monitoring and Anomaly Detection

Unsupervised learning is a subset of machine learning whereby the algorithm is given data without explicit labels or instructions as it aims to discover patterns, structures, and relationships within the data on its own unlike supervised learning. This has proved to be quite handy in the field of energy management as unsupervised learning has been applied to areas involving dimensionality reduction, clustering, and anomaly detection. Unsupervised learning operates on unlabeled datasets, identifying inherent patterns and anomalies without predefined labels (Arias-Requejo et al., 2023). This capability is particularly valuable in energy monitoring, where labeled data is often scarce or costly to acquire. By uncovering hidden trends in energy usage, unsupervised learning enables proactive management of inefficiencies, safety hazards, and demand fluctuations. For instance, in residential halls, unsupervised algorithms can detect anomalous power consumption that may indicate electrical faults or wasteful behaviors, allowing for timely interventions (Chiosa et al., 2021).

#### Clustering Techniques for Pattern Recognition

Clustering is a foundational unsupervised learning approach that groups data points into clusters based on similarity. This technique is instrumental in identifying typical energy consumption patterns and deviations, which are critical for optimizing building operations.

#### k-Means Clustering

The k-means algorithm partitions data into k clusters by iteratively minimizing within-cluster variance. Centroids are updated based on the mean of assigned data points until convergence (Arias-Requejo et al., 2023). In energy management, k-means clusters energy usage into distinct operational modes (e.g., high, medium, and low consumption periods). For example, Arias-Requejo et al. (2023) used k-means to segment historical environmental and energy data, enabling tailored demand forecasting models for different building conditions. The algorithm requires predefined cluster counts (k) and is sensitive to initial centroid placement, which can yield suboptimal results (Capozzoli et al., 2021).

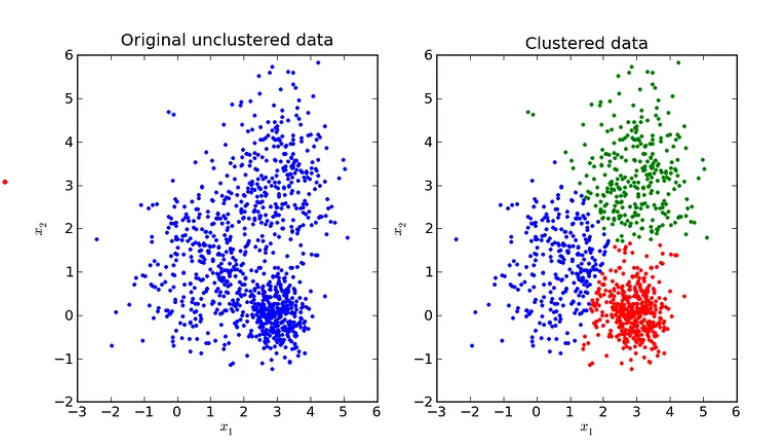


Figure 2 k-Means Clustering

##### Hierarchical Clustering

Hierarchical clustering builds a tree-like structure of nested clusters, either through agglomerative (bottom-up) or divisive (top-down) methods. This method identifies clusters of varying shapes and sizes, making it suitable for detecting nuanced energy consumption patterns in heterogeneous datasets (Arias-Requejo et al., 2023). Computational complexity increases with dataset size, limiting scalability for real-time applications.

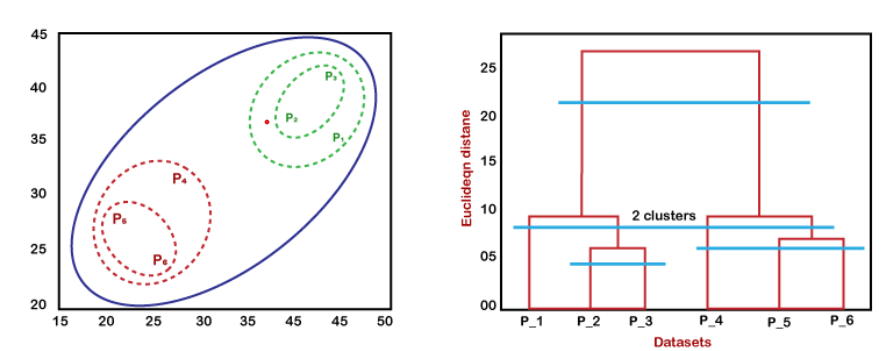


Figure 3 Hierarchical Clustering

#### Advanced Unsupervised Algorithms for Anomaly Detection

##### Autoencoders

Autoencoders are neural networks that compress input data into a latent representation and reconstruct it. Deviations between input and reconstruction signal anomalies (Elhabyb et al., 2024). Capozzoli et al. (2021) employed autoencoders to detect abnormal energy consumption in buildings, flagging inefficiencies linked to HVAC malfunctions.  
Training requires substantial computational resources, and model interpretability remains challenging (Das et al., 2024).

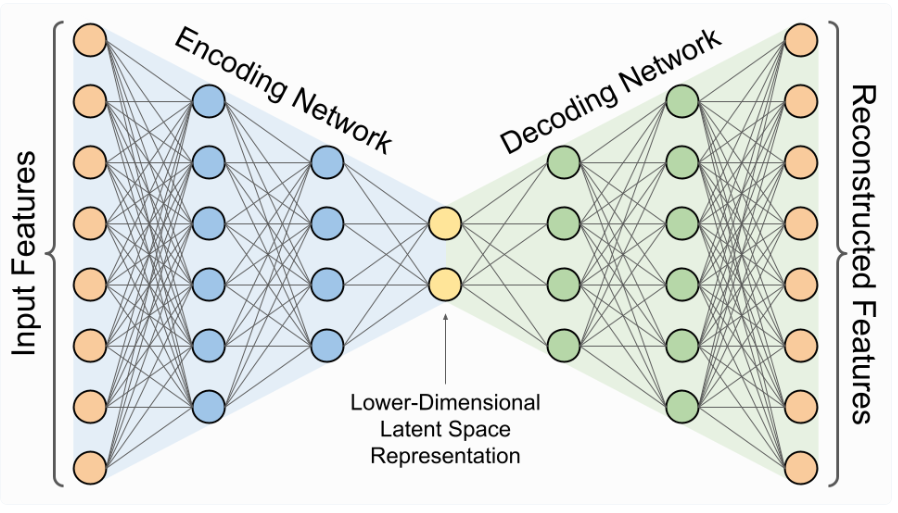


Figure 4 Network Architecture for Ordinary Autoencoder

##### Symbolic Aggregate Approximation (SAX)

SAX converts time-series data into symbolic sequences, enabling efficient pattern matching and anomaly detection. Chiosa et al. (2021) applied SAX to reduce dimensionality in energy datasets, identifying irregular consumption patterns in educational buildings. Oversimplification of data may obscure subtle anomalies.

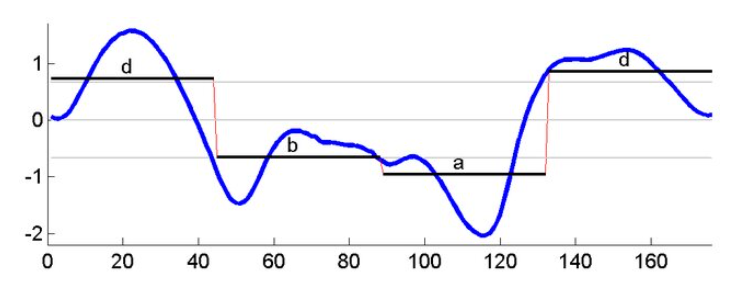


Figure 5 Symbolic aggregate approximation of a time series data.

##### Adaptive Symbolic Aggregate Approximation (aSAX)

Adaptive Symbolic Aggregate Approximation is an enhanced version of the Symbolic Aggregate Approximation (SAX) Technique used for time series data analysis. It aims to reduce the dimensionality of time series data while preserving its essential characteristics (Chiosa et al., 2021). The original time series is divided into non-overlapping sub-sequences. Each sub-sequence is further divided into segments called time windows. The length of these time windows can be equal or different based on user preference. This is typically known as chunking. The next step involves feature extraction, whereby for each time window, an aggregate numerical feature is calculated. This feature represents the data points within that window. Common features may include the mean, variance and trend angle which is particularly effective in describing the time series trend. After this, the aggregated numerical features are then mapped to a discrete set of symbols based on predefined breakpoints. The aSAX algorithm finds the optimal positions of these breakpoints through an iterative process which minimizes the distance among data points within each segment. The final step involves string representation whereby the sequence of symbols obtained from the encoding step forms the symbolic representation of the time series. This string representation can be used for various tasks such as classification and anomaly detection.

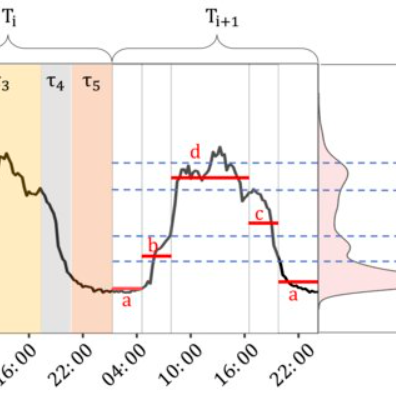


Figure 6 Illustration of Adaptive Symbolic Aggregation Approximation

#### Challenges

While unsupervised learning offers powerful tools for energy management, challenges persist:

1. Interpretability: Models like autoencoders act as "black boxes," complicating root-cause analysis (Das et al., 2024).
2. Data Quality: Noise and missing values in energy datasets degrade algorithm performance (Chiosa et al., 2021).
3. Scalability: Techniques like hierarchical clustering struggle with large, high-frequency data streams.

Future research should prioritize hybrid models that combine unsupervised learning with interpretable rule-based systems, such as association rule mining (ARM), to link anomalies to specific sub-load behaviors (Capozzoli et al., 2021).

### 2.1.2 Deep Learning in Building Energy Management

Deep learning has shown great potential in improving energy efficiency in smart buildings particularly in forecasting energy consumption. This is mainly attributed to its ability to extract intricate patterns and representations from data using artificial neural networks (ANNs) with multiple layers.

#### Deep Learning Architecture for Predicting Building Energy Usage

##### Recurrent Neural Networks (RNNs)

RNNs excel at processing sequential data like time series. Their inherent ‘memory’ enables them to learn from past data points, making them ideal for understanding temporal dependencies in energy consumption patterns (Arias-Requejo et al., 2023; Elhabyb et al., 2024). A specialized type of RNN, Long Short-Term Memory (LSTM) networks, are particularly effective at capturing long-term dependencies and have been successfully applied in numerous studies for building energy consumption prediction (Arias-Requejo et al., 2023; Elhabyb et al., 2024).

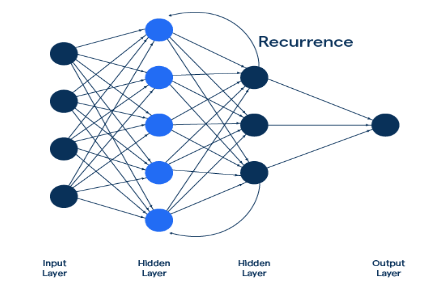


Figure 7 Recurrent Neural Network

##### Convolutional Neural Networks (CNNs)

While typically used in image recognition, CNNs are also being explored for energy consumption forecasting (Elhabyb et al., 2024). CNNs are proficient at extracting features from spatially correlated data, such as weather information or building layouts. Research suggests that combining CNNs with LSTM encode-decoder algorithms can enhance the accuracy of power consumption trend predictions, especially for low-powered devices (Elhabyb et al., 2024).

##### Hybrid Networks

Hybrid networks capitalize on the strengths of different deep learning techniques by combining them. for example, a hybrid network might integrate LSTMs and CNNs to simultaneously capture both temporal and spatial patterns in energy consumption data, leading to more accurate predictions (Elhabyb et al., 2024). Studies have shown that hybrid networks incorporating LSTMs, CNNs, feature engineering, and wavelet transforms can significantly improve short-term load prediction accuracy and efficiency (Elhabyb et al., 2024).

#### Applications of Deep Learning in Building Energy Management

###### Short-term Load Forecasting

Accurately predicting energy consumption in short term, usually hours or days ahead, is vital for optimizing energy scheduling and implementing effective real-time control strategies (Arias-Requejo et al., 2023; Das et al., 2024). For example, one study used deep neural networks to forecast the energy consumption of specific HVAC system components, demonstrating the applicability of deep learning in real-world scenarios (Kim et al., 2020).

###### Medium-term Demand Forecasting

Forecasting energy demand over weeks or months enables better energy planning, procurement, and the implementation of demand response programs (Arias-Requejo et al., 2023).

###### Fault Detection and Diagnosis

This capacity of deep learning to analyze patterns in energy consumption data makes it invaluable for identifying anomalies that may indicate equipment malfunctions or inefficiencies (Das et al., 2024). This can lead to early detection of potential problems, preventing energy wastage and ensuring the smooth operations of building systems.

###### Occupancy Prediction

By predicting occupancy levels, building managers can optimize the control of systems like lighting, heating, and ventilation, resulting in significant energy savings (Das et al., 2024).

#### Gaps in Deep Learning in Smart Buildings

While deep learning shows promise, there are challenges to address such as:

###### Data Availability and Quality

The effectiveness of deep learning models is highly dependent on the availability of extensive, high-quality data for training (Das et al., 2024). Acquiring sufficient data with accurate labels remains a significant challenge.

###### Model Interpretability

Deep learning models are often considered ‘black boxes’ because their decision-making processes are opaque and difficult to understand (Das et al., 2024). Research into improving the interpretability of these models is crucial for building trust and gaining valuable insights from their predictions.

###### Computational Complexity

Training deep learning models often requires significant computational resources (Das et al., 2024). This necessitates the use of specialized hardware and software, which can be a barrier to wider adoption.

Future research directions should prioritize:

* Developing more interpretable deep learning models.
* Exploring transfer learning techniques to efficiently utilize data from multiple buildings.
* Integrating deep learning existing buildings management systems for seamless operation.

Deep learning is transforming the way we manage energy in buildings by enabling accurate predictions of energy consumption, detecting potential faults, and optimizing energy use (Das et al., 2024). Continued research in this field holds immense promise for creating smarter and more energy-efficient buildings for a sustainable future.

### 2.1.3 The Role of Data-Driven Methodologies in Building Energy Consumption Forecasting

Building energy consumption forecasting has become increasingly crucial in the pursuit of energy efficiency and sustainability in smart buildings. Data-driven methodologies, particularly those leveraging machine learning, play a pivotal role in achieving accurate and reliable predictions. This review examines the methodologies and approaches employed in predicting building energy consumption, highlighting the contributions of data-driven approaches.

#### Data Analysis and Preprocessing

The foundation of any data-driven forecasting model lies in the quality and characteristics of the data used. Raw data often require preprocessing to enhance the performance and accuracy of the models.

###### Data Understanding

The initial step involves understanding the patterns and characteristics of the energy consumption data (Elhabyb et al., 2024). This includes examining the distribution of the data, identifying trends, and analyzing potential correlations with other variables.

###### Feature Engineering

Selecting the most relevant features for forecasting energy consumption is critical. This might involve using techniques such as correlation analysis, ensemble analysis, and tree-based models to identify the most impactful factors (Elhabyb et al., 2024).

###### Data Transformation

Transforming data into a suitable format for the selected model is essential. For example, using symbolic aggregate approximation (aSAX) can reduce the dataset size and effectively identify patterns in time series data (Chiosa et al., 2021).

#### Addressing Missing Data

Handling missing values in the dataset is crucial for model accuracy. This involves imputation techniques or using models that can handle data effectively.

#### Predictive Modeling Techniques

A variety of predictive models can be employed for forecasting building energy consumption. Data-driven methodologies, particularly machine learning algorithms, have shown promise in achieving accurate predictions.

###### Traditional Statistical Approaches

Techniques like Autoregressive Integrated Moving Average (ARIMA) have been traditionally used for time series forecasting. ARIMA models are based on statistical relationships within the time series data but may not capture complex non-linear patterns effectively.

###### Machine Learning (ML) Algorithms

ML Algorithms offer a data-driven approach to forecasting, learning patterns and relationships from historical data. Some commonly used algorithms include:

* **Random Forest**: a robust ensemble learning method that combines multiple decision trees to improve prediction accuracy (Elhabyb et al., 2024).
* **Gradient Boosting Regressor (GBR)**: Another ensemble technique that sequentially builds trees, correcting errors from previous trees to enhance prediction performance (Elhabyb et al., 2024).
* **Support Vector Machines (SVMs)**: Powerful algorithms for both classification and regression tasks, capable of handling high-dimensional data.

#### Deep Learning Models

A subset of ML, deep learning models leverage artificial neural networks with multiple layers to learn complex patterns from data. Commonly used DL models include:

* **Long Short-Term Memory (LSTM) Networks:** A type of Recurrent Neural Network (RNN) well-suited for handling long-term dependencies in time series data, making them effective for energy consumption forecasting (Arias-Requejo et al., 2022).
* **Encoder-Decoder LSTM Networks:** A specific LSTM architecture that has shown effectiveness in predicting energy consumption (Aria-Requejo et al., 2023). These networks encode input sequences into a fixed-length vector and then decode this vector to generate output sequences.

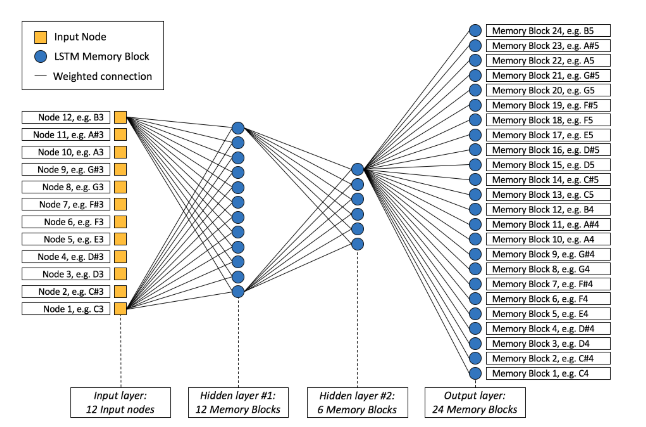


Figure 8:LSTM Network

#### Model Evaluation and Selection

Evaluating the performance of different models is essential in selecting the most suitable one for the specific forecasting task.

Cross-Validation: employing cross-validation techniques, such as splitting the data into training and validation sets, helps assess the model’s ability to generalise to unseen data and prevent overfitting.

Data-driven methodologies have revolutionized building energy consumption forecasting, enabling more accurate and reliable predictions. By leveraging the power of machine learning and deep learning, building managers and operators can gain valuable insights into energy consumption patterns, optimize building performance, and contribute to a more sustainable built environment. Addressing the challenges related to data quality, model interpretability, and computational complexity will be key to the continued advancement and widespread adoption of these powerful techniques.

### 2.1.4 Anomaly Detection and Diagnosis (ADD) in Buildings

Anomaly detection plays a crucial role in optimizing energy consumption and identifying potential issues within buildings. It involves identifying unusual patterns or events that deviate significantly from what is considered normal with reference to energy consumption data.

#### How Anomaly Detection Works

An energy management system gathers data from various sources after which the system learns what ‘normal’ energy consumption looks like. It analyzes the historical data to create a baseline of expected behavior. The system will continuously compare the current energy usage against the established baseline and if the system detects a significant deviation from the norm, it flags it as an anomaly. The system can then alert the parties involved or take automated actions to address the anomaly.

#### Importance of Anomaly Detection

* Identifying inefficiencies: ADD helps pinpoint operational inefficiencies, equipment malfunctions, or unusual occupancy patterns that lead to increased energy consumption (Chiosa et al., 2021).
* Proactive maintenance: Detecting anomalies can provide early warnings of potential equipment failures, enabling proactive maintenance and reducing downtime (Chiosa et al., 2021).
* Occupant Comfort and Safety: Anomalies in energy consumption patterns related to heating, ventilation, and air conditioning (HVAC) systems can indicate issues affecting occupant comfort and safety.
* Security and Fault Detection: ADD can be used to detect abnormal energy consumption patterns that might indicate energy theft or other security breaches.

#### Methodologies and Techniques for ADD

###### Statistical Methods

Boxplot Analysis: this simple method identifies outliers by defining upper and lower bounds based on the interquartile range of the data. Data points beyond those bounds are considered anomalies (Liu et al., 2020).

Statistical Process Control (SPC): SPC techniques monitor process variables over time and detect deviations from expected behavior using control charts.

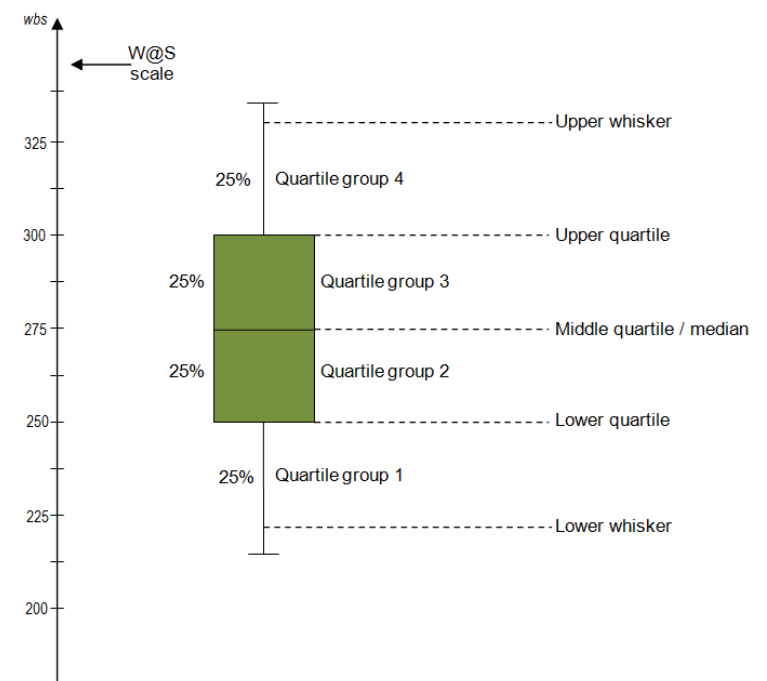


Figure 9 Boxplot Analysis

#### Machine Learning (ML) and Deep Learning (DL)

Clustering: Clustering algorithms, such as k-means or hierarchical clustering, can group data points based on similarities. Data points that do not belong to any cluster or form small, isolated clusters can be considered anomalies (Chiosa et al., 2021).

Classification: Supervised ML algorithms, like Support Vector Machines (SVMs) or decision trees, can be trained on labeled data to classify energy consumption patterns as normal or anomalous.

Time-Series Analysis: Deep learning models such as Long Short-Term Memory (LSTM) networks, can effectively capture temporal patterns in energy consumption data and identify deviations from expected trends (Liu et al., 2020).

#### Rule-Based Systems

Rule-based systems use predefined rules to detect anomalies. These rules can be based on expert knowledge or derived from data analysis techniques like association rule mining (Chiosa et al., 2021).

#### Data Preprocessing for ADD

Similar to forecasting models, data preprocessing is essential for enhancing the effectiveness of ADD techniques.

* Data Cleaning: removing outliners, handling missing values, and addressing inconsistencies of ADD techniques.
* Data Transformation: Techniques like Symbolic Aggregate approXimation (SAX) of Adaptive SAX (aSAX) can reduce the dimensionality of time-series data while preserving essential patterns, making it easier to detect anomalies (Chiosa et al., 2021).

##### Anomaly Diagnosis

Identifying the root cause of an anomaly is crucial for taking appropriate corrective actions.

* Association Rule Mining (ARM): ARM can uncover relationships between anomalous energy consumption at the aggregate level e.g., whole-building and specific sub-load behaviors (Chiosa et al., 2021). This helps pinpoint the sub-loads responsible for the detected anomaly.
* Expert knowledge: incorporating expert knowledge of building systems and operational practices can aid in interpreting detected anomalies and identifying their underlying causes.

##### Contextual Anomaly Detection

Recognizing that energy can vary significantly depending on external factors (e.g., weather, occupancy), contextual anomaly detection aims to identify anomalies within specific contexts (Liu et al., 2020).

* Concept Drift Detection: this involves monitoring changes in data distribution over time, indicating a shift in context. When concept drift is detected, the anomaly detection model is updated.

#### Challenges in ADD

* Lack of Labeled Data: Training supervised ML models for anomaly detection can be challenging due to the scarcity of labeled anomaly data.
* Defining Normal Behavior: Establishing a clear baseline for normal energy consumption patterns can be difficult, as building usage and environmental conditions can vary significantly.
* Real-time Detection: Implementing real-time anomaly detection requires efficient algorithms and systems capable of processing streaming data.

#### Applications and Future Directions

ADD has broad applications in building management, ranging from fault detection and predictive maintenance to optimizing energy efficiency and improving occupant comfort. Future research directions include:

* Developing More Robust and Adaptive Algorithms: Addressing challenges related to data scarcity, concept drift, and the dynamic nature of building environments.
* Integrating ADD with Building Management Systems (BMS): Enabling automated responses to detected anomalies, such as adjusting control setpoints or triggering maintenance alerts.
* Leveraging Advanced Visualization Techniques: Enhancing the interpretability of detected anomalies for building operators and facility managers.

Anomaly detection and diagnosis hold significant potential for optimizing energy consumption, improving operational efficiency, and enhancing the overall performance of smart buildings. Continued research and development in this area will lead to more sophisticated and effective ADD systems, contributing to a more sustainable and intelligent built environment.

### 2.1.5 Hybrid Machine Learning Models for Energy Consumption Forecasting

Hybrid machine learning models integrate the strengths of various algorithms to address the complexity of energy consumption data, which often involves non-linear patterns, temporal dependencies, and spatial variability. These models are increasingly being used in energy systems to enhance prediction accuracy and system performance.

#### Key Components of Hybrid Models

##### Dimensionality Reduction and Feature Selection

Hybrid models often incorporate feature selection techniques such as Random Forests (RF) to identify key predictors of energy consumption. Reducing the dimensionality of datasets ensures that the models focus on the most impactful variables, improving their accuracy and interpretability.

##### Neural Networks and Backpropagation

Backpropagation Neural Networks (BPNN) are a common component of hybrid models. They excel at modeling complex, non-linear relationships within data, making them particularly effective when combined with other techniques such as feature selection or statistical forecasting methods.

##### Temporal and Spatial Integration

Advanced hybrid models, such as those combining Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs), handle both temporal and spatial features. For example, CNNs extract spatial correlations from weather data, while LSTMs model sequential energy consumption patterns over time.

#### Applications of Hybrid Models

###### Short-Term Load Forecasting

Hybrid models are widely used for short-term energy consumption predictions. A recent study achieved a 96% accuracy rate in forecasting household energy consumption by integrating lifestyle data with hybrid machine learning techniques (Jiang et al., 2023).

###### Electricity Demand Prediction

Hybrid models combining statistical and machine learning methods have been successfully applied to forecast national electricity demand, offering long-term forecasts with hourly resolution.

###### Anomaly Detection

Leveraging the strengths of multiple algorithms enables hybrid models to detect anomalies in energy data, such as unexpected consumption spikes or potential equipment malfunctions, with higher precision compared to standalone models.

#### Advantages of Hybrid Models

* Improved Accuracy: the integration of complementary algorithms enhances prediction performances across various scenarios.
* Flexibility and Robustness: Hybrid models adapt well to diverse datasets and can handle missing or noisy data effectively.

#### Challenges faced when using Hybrid Models

Hybrid models, while powerful, require significant computational resources and expertise for implementation. Future research should focus on simplifying these models, improving interpretability, and enhancing efficiency to facilitate broader adoption in real-world applications.

### 2.1.6 Predictive Analytics in Energy Monitoring Systems

#### Theoretical Foundations of Predictive Analytics

Predictive analytics is grounded in the principles of statistical modeling and machine learning, which enable the analysis of historical data to forecast future energy trends. Its applications in energy monitoring systems draw on theoretical concepts of data-driven modeling and advanced algorithmic frameworks. These theories emphasize the importance of energy consumption, the prediction of energy demand, and the detection of inefficiencies within energy systems (Aria-Requejo et al., 2023)

#### Core Theoretical Concepts and Methodologies

#### Data-Driven Approach

Predictive analytics relies on historical data, such as energy consumption patterns, occupancy data, and environmental conditions, to create accurate forecasting models. The theoretical principle underpinning this approach is that high quality and contextually relevant data enhances predictive accuracy. This foundation is critical for energy applications where consumption patterns vary widely across contexts.

##### Machine Learning Algorithms

The use of machine learning algorithms represents a theoretical and applied knowledge. Several key algorithms form the backbone of predictive analytics in energy systems:

* **Time Series Models**: Theoretical models like ARIMA are built on the principle of temporal dependency, making them suitable for energy consumption forecasting (Wang & Srinivasan, 2015).
* **Regression Models**: Linear regression, support vector machines, and random forest are rooted in statistical learning theory and are effective for both short-and long-term energy predictions (Elhabyb et al., 2024).
* **Neural Networks**: The deep learning framework, particularly Long Short-Term Memory (LSTM) networks, capitalizes on advanced theories of temporal correlation to model non-linear dependencies in energy data (Arias-Requejo et al., 2023).
* **Ensemble Methods**: Theoretical concepts of bias-variance trade-offs are leveraged in methods like gradient boosting to combine predictions from multiple models and enhance accuracy.

##### Anomaly Detection

Anomaly detection in energy systems is founded on theories of classification and clustering. Techniques like classification trees and association rule mining help identify deviations in energy consumption patterns, enabling efficient diagnosis of system inefficiencies (Zanetti et al., 2021). This approach is particularly relevant in smart grids and industrial energy systems, where even minor anomalies can lead to significant energy losses.

##### Contextual Awareness

Contextual awareness is a critical theoretical dimension, emphasizing the incorporation of external factors such as weather conditions, occupancy dynamics, and market-driven energy pricing into predictive models (Powroznik & Szczesniak, 2024). These factors enhance model robustness and applicability across diverse energy systems.

#### Applications of Predictive Analytics in Energy Systems

Predictive analytics has demonstrated practical relevance across a variety of energy management scenarios:

* Smart buildings: The application of predictive analytics in smart buildings aligns with theoretical principles of proactive energy management, enabling real-time HVAC adjustments to optimize energy use (Arias-Requejo et al., 2023).
* Industrial Energy Management: Theories of historical pattern analysis are applied to optimize industrial energy operations, minimizing energy waste (Kumar et al., 2021).
* Demand Response Programs: Predictive models provide theoretical baselines for counterfactual energy consumption, improving the effectiveness of demand response programs.

#### Challenges and Research Gaps

Despite significant advancements, several theoretical challenges remain in the application of predictive analytics:

* **Data Quality and Availability**: The effectiveness of predictive models is constrained by the availability of large, high-quality datasets. This issue is particularly pronounced in developing regions where data infrastructure is limited (Okonkwo, 2023).
* **Model Interpretability**: while deep learning models provide high accuracy, their theoretical complexity often makes them ‘black boxes’. Future research must prioritize explainable AI (XAI) to enhance interpretability (Aria-Requejo et al., 2023).
* **Behavioral Factors**: Occupant behavior introduces unpredictable variability in energy analytics frameworks to improve prediction accuracy.

Emerging research should focus on the development of hybrid models that integrate statistical and machine-learning approaches to enhance prediction accuracy (Elhabyb et al., 2024). Additionally, advancements in real-time data collection through IoT sensors will enable more responsive energy management systems (Brunelli et al., 2015). Incorporating theories of user-centric design will further enhance the adoption of predictive analytics in energy systems (Powroznik & Szczesniak., 2024).

### 2.1.7 Time Series Forecasting for Energy Management

A literature review of time series forecasting for energy consumption reveals a variety of methods and techniques employed across different contexts, including smart homes, buildings, industrial settings, and power grids. These methods aim to model and predict energy usage effectively, with a focus on enhancing sustainability and efficiency (Sumorek et al., 2024).

#### Time Series Forecasting Methodologies

* **Long Short-Term Memory (LSTM) Networks**: LSTM models are frequently used for energy consumption due to their ability to capture complex patterns in time-series data (Abdel-Nasser & Mahmoud, 2019). These models require substantial data for accurate predictions and can be structured in an encoder-decoder architecture, which is particularly useful for medium-term forecasting.
* **Fuzzy Time Series (FTS) Models**: Embedding FTS methods, such as yFTS, are designed for forecasting high-dimensional multivariate time series by applying embedding transformations to reduce dimensionality (Bitencourt & Guimaraes, 2021; dos Santos et al., 2021). This enables efficient pattern recognition and the induction of fuzzy rules, offering a data-driven and flexible modeling approach (de Lima Silva et al., 2020).
* **Autoregressive Integrated Moving Average (ARIMA) Models**: These conventional statistical techniques are used in time series analysis and are effective in areas where data frequency is low (Yaffee & McGee, 2000). However, for hourly values, the non-linear behavior of data might be difficult to predict using these models (Deb et al., 2017).
* **Hybrid Models:** These models combine different approaches to improve forecasting performance. Examples include integrating machine learning and statistical techniques or combining a genetic algorithm with LSTM (Bouktif et al., 2018). Hybrid models are designed to improve the performance of physical or statistical techniques.
* **Transformer Models:** The seq2seq architecture used in transformer models is suitable for modeling complex relationships in sequence data and for multi-step time series forecasting (Lezmi & Xu, 2023; Yang & Lu, 2022).

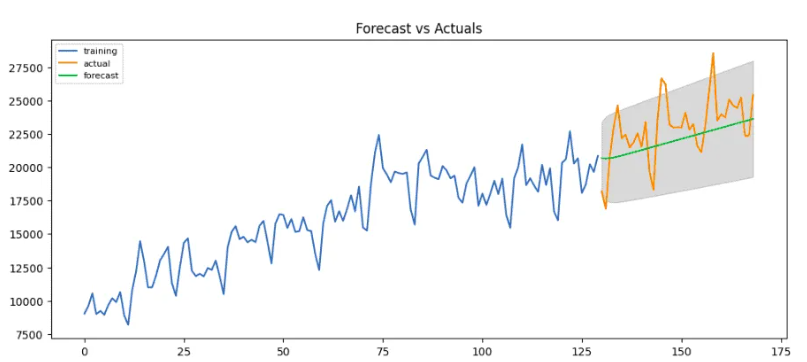


Figure 10 Illustration of Time Series Forecasting using ARIMA Model

#### Applications

Energy forecasting has many practical applications:

* Smart Buildings: Energy forecasting is essential for optimizing operations, reducing costs, and minimizing environmental impact (Seyendzadeh et al., 2018).
* Smart Homes: Energy decomposition techniques and machine learning can model and forecast energy consumption for individual appliances.
* Smart Manufacturing: Predictive analysis helps model and forecast energy consumption in smart manufacturing settings (Kumar et al., 2021).
* Renewable Energy: Time series forecasting is critical for managing energy production from renewable sources, especially those dependent on weather conditions (Sumorek & Idzkowski, 2023; Alaraj et al., 2021).

Effective time series forecasting of energy consumption involves employing a range of methodologies, from statistical models to advanced machine learning techniques (Athanasopoulos, 2018). The choice of the optimal approach depends on the specific context, data characteristics, and desired outcomes. The literature highlights that robust data preprocessing, careful feature selection, and thorough model evaluation are essential for developing accurate and reliable energy forecasting systems (Powroznik & Szczesniak, 2024).

## 2.2 Related Works

### 2.2.1 Case Studies of Energy Monitoring and Anomaly Detection Systems

The increasing importance of energy efficiency has led to the development and implementation of various energy monitoring and anomaly detection systems in different settings. These systems utilize a variety of techniques, including machine learning, statistical analysis, and data visualization, to optimize energy consumption and identify unusual patterns. This case study will explore how such systems have been applied in educational buildings, residential environments and other contexts from various sources.

#### Energy Monitoring and Management in Educational Buildings

The application of energy monitoring systems in educational buildings involves the deployment of wireless sensor networks (WSNs) to monitor energy consumption of various devices and systems, such as lighting and computers.

##### King Saud University

Al-daraiseh et al., (2013) implemented an energy management system at King Saud University using sensors to measure energy consumption every 10 seconds. They found that 17% of energy consumption could be reduced by addressing standby energy loss of lab computers. This system also incorporated a timetabling system to optimize HVAC operation. The system used a rule base, populated with expert knowledge, to diagnose abnormal consumption. The monitoring system was installed in 10 computers labs and 10 offices, with 500 sensors deployed. They were able to reduce energy wastage by 8.53% through awareness programs and a further 8.46% through automated actions that switched off unused devices.

##### Alice Perry Engineering Building

Arias-Requejo et al. (2021) utilized data from the Alice Perry Engineering Building at the National University of Ireland in Galway, a smart building with a building management system (BMS) to monitor and control its electrical systems. They developed a methodology that combines embedding transformation and fuzzy time series forecasting to predict energy consumption. The building has 14,250 square meters, four floors, and 400 rooms. They focused on the heating energy demand of eleven air handling units (AHUs) in the building. The system also considers weather conditions to improve prediction accuracy.

##### University Campus of Politecnico di Torino

Chiosa et al. (2021) developed a data analytics-based Energy Information System (EIS) tool that performs meter-level anomaly detection and diagnosis in buildings. The system was tested at the university campus of Politecnico di Torino. They used an adaptive symbolic aggregate approximation (aSAX) process to analyze the data. They employed classification trees (CT) and association rule mining (ARM) to detect and diagnose anomalies. The system was designed to detect anomalies at the whole-building and diagnose the causes at the sub-level. They found that the system was able to detect infrequent behaviors in energy consumption, indicating previously unknown issues. The aSAX method transforms and reduces the time series data, which then is used for pattern recognition.

Cui and Wang (2017) investigated five models for detecting anomalies in school electricity consumption data, including autoregressive moving-average, polynomial regression, Gaussian Kernel distribution, Gaussian distribution. They proposed a hybrid model combining polynomial regression and Gaussian distribution, achieving high precision. A data detection and visualization system was developed to help facility managers to identify anomalies.

#### Energy Monitoring in Residential Environments.

Energy monitoring systems are also used in residential environments, often focusing on home automation and personalized energy management.

##### Contextual Anomaly Detection

Liu et al. (2020) proposed a contextual anomaly detection (CAD) framework for residential energy consumption using a sliding window approach, prediction-based detection, and concept drift detection to identify unusual consumption in different contexts. This approach uses streaming data from smart meters and is integrated with a concept drift detector to train the detection model.

##### Commercial Buildings

Qureshi et al. (2024) used deep learning for forecasting energy consumption in commercial buildings using an LSTM based time series approach. The system uses a BEMS that tracks and manages energy use. A data-driven method is used to create a model of normal consumption behavior that can then be used to identify unusual patterns. The system can also be used to detect gas leaks and other safety issues.

##### Predictive Analysis in Energy Management Systems

NEC researchers have explored the use of predictive analysis integrated with energy management systems using machine learning, to predict power outages. The system uses past data about power consumption, outages, weather, and temperature to develop a mathematical prediction model.

# Chapter 3: Methodology

This chapter outlines the methodology employed in developing the machine learning component of the Energy Monitoring and Anomaly Detection System.

## 3.1 System Development

### Overview

This section talks about how the proposed project aims to develop a small-scale energy monitoring and anomaly detection system for residential halls using Internet of Things (IoT) sensors and machine learning techniques. The system will collect real-time electrical load data, identify patterns, and detect anomalies indicative of potential faults. Due to the limited equipment available, the study will focus on a streamlined implementation that prioritizes cost-effectiveness and efficiency while ensuring reliable anomaly detection.

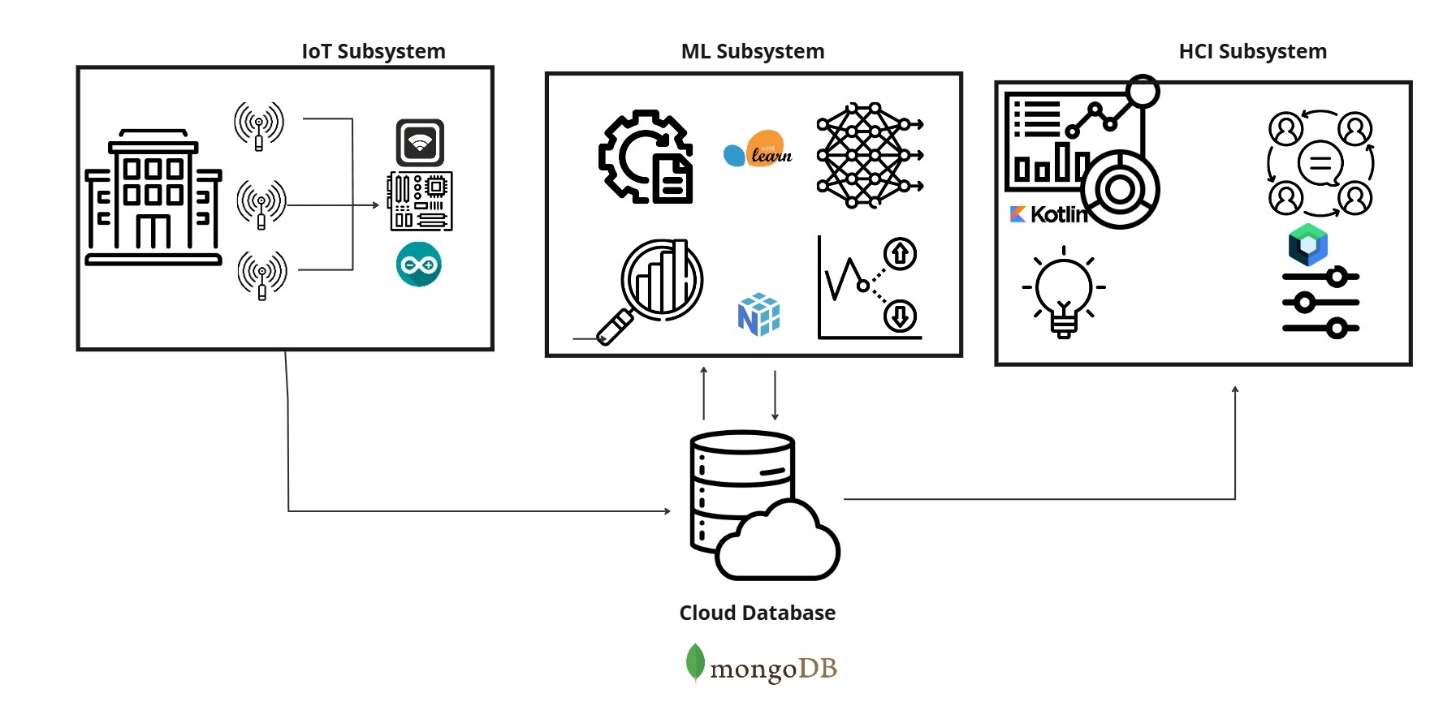


Figure 11 Proposed System Architecture

### 3.1.1 Data Collection and Preprocessing

Electrical load data will be collected through a series of processes that begin with the use of IoT-based current sensors installed in strategic locations within residential halls in JKUAT. These sensors will measure current flow through electrical circuits, and the voltage values will be obtained through collaboration with the hall electricians upon obtaining request from the relevant school authority overseeing the electrical supply in the residential halls. Electrical load will be determined using the standard electrical formula where:

**V = is the voltage value (volts) and**

**I = is the measure current (amperes).**

**Load(P) = power consumed by a device**

**P = V \* I (Power = Voltage \* Current)**

**Energy Consumption (wh) = Power X Time**

**Energy Consumption (kwh) = Energy Consumption (wh) / 1000**

#### Sensor Deployment Strategy

Sensors will be installed at key distribution points, such as main electrical panels and high-consumption appliances, ensuring representative coverage of energy use. The sensors will relay real-time current measurements wirelessly to an Arduino microcontroller equipped with a Wi-Fi module, which will compute the power consumption using the assumed voltage and send the values to a centralized cloud database hosted on AWS (Amazon Web Services) through MongoDB. The relevant columns will be ‘Timestamp’, and ‘TotalLoad’. Data will be recorded at an interval of 30 minutes in the cloud database. This frequency was chosen to balance the need for detailed information with computational feasibility for the anomaly detection model. Higher frequency data e.g., every 15 minutes would provide more granular insights but significantly increase the data volume and processing requirements, potentially exceeding the available resources.

#### Preprocessing Techniques

* Cleaning: Missing values regarding the ‘TotalLoad’ column will be handled using linear interpolation. This method has been selected as it is suitable for time-series data and preserves the overall trend of the load profile. Outliers will be identified using the Interquartile Range (IQR) method. Values exceeding 1.5 times the IQR above the 75th percentile or below the 25th percentile will be considered outliers and capped at those respective values. This approach has been chosen to limit the impact of extreme, potentially erroneous readings, without completely removing them, as some extreme values might represent genuine peak demand.
* Normalization: The load values will be scaled to a range of 0-1 using Min-Max scaling which preserves the relative relationships between load values and is suitable for decision tree models. The formula used will be:
* **ScaledLoad = (TotalLoad – MinLoad) / (MaxLoad - MinLoad), where MinLoad and MaxLoad are the minimum and maximum loads observed for each HallID in the dataset.**
* Filtering Outliers: Statistical boxplot analysis will detect anomalies caused by sensor errors or sudden spikes unrelated to typical usage.
* Time Synchronization: All data entries will be time-stamped and synchronized to ensure consistency across multiple sensors.

#### Symbolic Representation and Feature Engineering

To optimize computational efficiency given the small-scale scope, Adaptive Symbolic Aggregate Approximation (aSAX) will be applied to convert continuous electrical load readings into symbolic representations. Data reduction will be carried out through Piecewise Aggregate Approximation which will compress the time-series data while preserving its key consumption trends to ensure reduced storage and processing requirements. Additionally, the following features will be engineered to capture relevant temporal patterns:

* HourOfDay: Extracted from the ‘TimeStamp’ column (0-23)
* DayOfWeek: Extracted from the ‘TimeStamp’ column (0-6, Monday=0).
* IsWeekend: A binary feature (1 for weekend, 0 for weekday).
* RollingAverageLoad\_24h: The average load over the previous 24 hours. This feature will capture daily load trends.

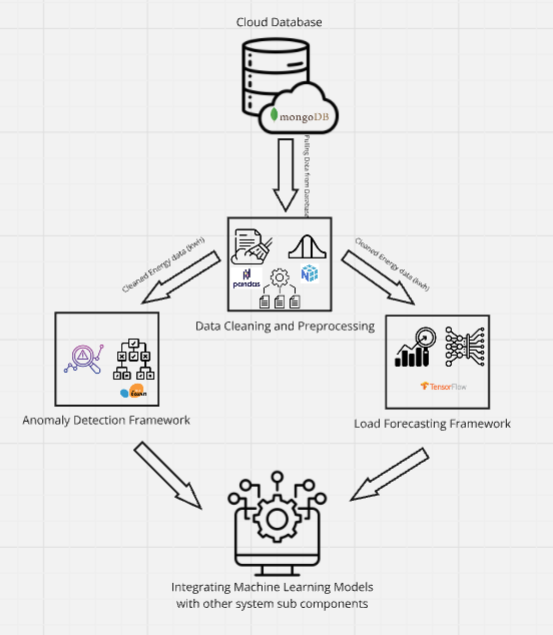


Figure 12: Machine Learning Framework

### 3.1.2 Anomaly Detection Framework

#### Building-Level Anomaly Detection

The primary goal of the anomaly detection component is to identify deviations in electrical load that may indicate inefficiencies, potential faults, or safety hazards. Here, I employ the isolation forest algorithm which operates by randomly portioning the data and isolating observations. Because anomalies are generally few and distinct, they tend to be isolated with fewer splits than normal data points, making this algorithm particularly effective for energy consumption data where anomalies are rare.

#### Training the Decision Tree Model

Historical electrical load data is used to establish a baseline of normal energy usage. The data is first preprocessed, cleaned, normalized, and enriched with engineered features such as time-based attributes and rolling averages, to ensure that it is in a suitable format for analysis. The isolation forest algorithm is then trained on this preprocessed data. During training, the algorithm learns to isolate data points, and the resulting model is used to predict anomaly scores for new data. Each data point is assigned a label, where -1 indicates an anomaly and 1 denotes normal behavior.

#### Deploying the Decision Tree Model

Once trained, the Isolation Forest model is deployed to operate on real-time or near-real-time data inputs. As new data is ingested, the model classifies each data point based on its anomaly score. Alerts are generated when a sequence of consecutive data points is flagged as anomalous. These alerts are structured for integration with downstream systems, ensuring that facility managers are notified promptly of any potential issues.

This transition to Isolation Forest enhances the robustness and efficiency of the anomaly detection process, particularly in handling high-dimensional time series data with potentially noisy measurements.

### 3.1.3 Energy Consumption Forecasting Framework

The energy consumption forecasting component aims to predict usage patterns in residential halls, enabling proactive energy management and optimization. Long Short-Term Memory (LSTM) networks, a specialized type of recurrent neural network (RNN), will be employed due to their ability to model temporal dependencies in time-series data. This framework ensures accurate short-term load forecasting, which is critical for demand response and reducing peak energy consumption.

#### LSTM Model Architecture

The LSTM network will be designed to capture both short-term fluctuations and long-term trends in energy consumption. The input sequence length will be set to 24 hours to capture daily periodicity. Each input sample will include:

* Historical Load Data: ‘TotalLoad’ which may be resampled to hourly values for the past 24 hours.
* Engineered Features:
  + HourOfDay (0-23)
  + DayOfWeek (0-6, Monday = 0)
  + IsWeekend (binary)
  + RollingAverageLoad\_24h (24-hour moving average)

#### Training the LSTM Model

The model will be trained on historical load data. The dataset will be split into training (80%), validation (10%), and testing (10%) sets using a time-series split to preserve temporal order.

#### Deploying the LSTM Model

Once trained, the LSTM model will be integrated into the mobile application platform to generate 24-hour-ahead forecasts. Predictions will be updated hourly using the most recent load data. The forecasts will be visualized through the HCI subsystem, enabling facility managers to:

* Anticipate peak demand periods.
* Schedule preventive maintenance during low-usage intervals.
* Communicate energy-saving recommendations to residents.

## 3.2 Technologies

* Programming Language: Python is a popular choice for data science, machine learning and IoT applications due to its readability, extensive libraries, and strong community support.
* Development Tools:
  + Visual Studio Code (VS Code): A powerful and versatile code editor that provides features like debugging, and integration with Git for version control. It’s excellent for general Python development and smaller machine learning projects.
  + Google Colab: A cloud-based platform that provides free access to computing resources, including GPUs and TPUs, making it ideal for computationally intensive tasks like training decision trees. It is also great for collaboration and sharing which will be key in this project.
* Libraries and Frameworks:
  + pyts: This library provides implementation of various time series analysis algorithms, including the aSAX algorithm. It’s used for symbolic representation of time series data, which can be helpful for anomaly detection, pattern recognition and data compression.
  + scikit-learn: A comprehensive library for various machine learning algorithms such as decision trees and classification models.
  + Pandas: a powerful library for data manipulation and analysis. It provides data structures like DataFrames, which are excellent for handling tabular data like CSV files.
  + NumPy: the fundamental library for numerical computing in Python. It provides support for arrays, matrices and mathematical functions, which are essential for machine learning and scientific computing.
  + Matplotlib, and Seaborn: These are data visualization libraries. Matplotlib is a basic plotting library, while Seaborn builds on top of Matplotlib to provide a higher-level interface for creating statistically informative and visually appealing plots.

### System Integration

Teamwork and communication in the project are important hence we need to ensure that the system is always in a state whereby the data is easily accessible and transferable between different parts of the system e.g., from the IoT devices to the database to the machine learning models.

###### IoT Ecosystem:

* Real-time collection of electrical load data: This describes the core function of the IoT devices. They measure the electrical current and together with the voltage obtained the energy usage is calculated and converted to digital data.
* Transmitting it to a centralized database: the collected data is sent wirelessly via Wi-Fi to a central database. This centralized database will be designed for sensor data allowing for data storage, retrieval and analysis.

###### HCI Subsystem:

* Visualizing energy forecasts and anomaly alerts for end-users through an intuitive interface. This is where the results of the data analysis and machine learning models are presented to the users. An intuitive interface is crucial for making the information understandable and actionable. With the case of our project, the interface will consist of a mobile application.
* Anomaly alerts: the system can detect unusual patterns in energy consumption, which could indicate a faulty appliance, a power surge, or other issues. Alerts are sent to the facility managers through the interface to notify them of these anomalies.

## 3. 3 Research Design

This research employs a case study approach, investigating the effectiveness of the proposed energy monitoring and anomaly detection system within residential halls at Jomo Kenyatta University of Agriculture and Technology (JKUAT). The study will involve collecting and analyzing electrical load data over a defined period to evaluate the system’s performance and identify its potential benefits.

### 3.3.1 Experimental Setup

The system will be deployed and tested in 1-2 residential halls at JKUAT. The halls to be selected are to represent a diverse range of energy consumption profiles. The data collection period will be around 3-4 months, which is deemed sufficient to capture variations in energy daily usage trends.

### 3.3.2 Pilot Deployment

Before full-scale deployment, a pilot test will be conducted in within the chosen residential halls. This pilot phase will serve several key purposes:

* Sensor Validation: the accuracy and reliability of the current sensors will be rigorously verified by their readings. Any necessary adjustments or replacements will be made before proceeding to full-scale data collection.
* System integration: the pilot deployment will also allow for testing the integration of the sensors with the data acquisition system and the cloud database. This will identify and resolve any potential communication or data transfer issues.
* Refinement of Data Preprocessing: The pilot data will be used to refine the data preprocessing steps, such as outlier handling and feature engineering, to ensure optimal data quality for model training.

### 3.3.3 Comparison with Historical Data

Where available, pre-existing historical data in previous years will serve as a benchmark to assess the system’s accuracy in measuring energy consumption. This comparison will provide insight into the system’s ability to accurately reflect actual energy usage.

### 3.3.4 Controlled Anomaly Testing

To validate the anomaly detection framework, controlled anomaly testing will be conducted. This will involve simulating various types of anomalies, such as:

* Increased Load: Artificially increasing the load on specific appliances for example by turning on multiple devices simultaneously to simulate equipment malfunctions or unusual usage patterns.
* Reduced Load: Simulating a reduction in load which could be done by unplugging appliances to mimic power outages or equipment failures.
* Fluctuating Load: Introducing rapid and unpredictable changes in load to test the system’s ability to detect transient anomalies.

The simulated anomalies will be introduced at specific times and locations, and the system’s ability to detect and classify these anomalies will be evaluated.

### 3.3.5Validation Metrics

The system’s performance will be assessed using the following metrics:

* **Root Mean Squared Error (RMSE):** RMSE will be used to evaluate the accuracy of the load forecasting component. It quantifies the difference between the predicted and actual load values.
* **Precision, Recall, and F1-score:** These metrics will measure the accuracy of the anomaly detection component. Precision indicates the proportion of detected anomalies that are actually anomalies, while recall measures the proportion of actual anomalies that were detected. The F1-score combines precision and recall into a single metric.
* **False Positive Rate (FPR) and False Negative Rate (FNR):** FPR measures the proportion of normal data points that are incorrectly classified as anomalies, while FNR measures the proportion of actual anomalies that are missed by the system. These metrics are crucial for evaluating the reliability of the system in distinguishing between normal and abnormal consumption.
* **Cross-validation:** k-fold cross-validation will be employed during model training and evaluation to enhance the robustness of the machine learning models and prevent overfitting.

## 3.4 Ethical Considerations

Ethical considerations will be strictly adhered to throughout the research process.

* **Data Privacy:** Data collected from the residents as feedback will be anonymized and securely stored in the cloud database, with access restricted to authorized research personnel only.
* **User Consent:** Participants (residents of the selected halls) will be fully informed about the purpose of the study, the data collection process, and how the data will be used. Written informed consent will be obtained from all participants before their data is included in the study.
* **Fairness and Bias Mitigation:** The machine learning models will be rigorously tested for potential biases related to factors such as appliance usage, or other demographic characteristics. If any biases are identified, appropriate mitigation strategies will be implemented to ensure fairness and equal representation of different energy consumption patterns.

## 3.5 Expected Outcomes

The successful implementation of the proposed system is expected to achieve the following outcomes:

* **Real-time Monitoring:** The system will provide real-time or near real-time monitoring of electrical load trends in the residential halls, enabling building management personnel to gain insights into energy usage patterns.
* **Improved Energy Efficiency:** By identifying and addressing anomalies in energy consumption, the system will contribute to improved energy efficiency in the residential halls as well as reduced energy costs.
* **Enhanced Safety:** The system's ability to detect potential electrical faults early through unusual load patterns can enhance safety in the residential halls by preventing electrical hazards.
* **Scalability:** The system is designed with scalability in mind, allowing for future expansion to include more residential halls or other buildings as resources become available.
* **Data-Driven Insights:** The collected data and the analysis performed by the system will provide valuable data-driven insights into energy consumption behavior in residential settings. These insights can be used to develop targeted energy conservation programs and policies.

### 3.5.1 Intended Outputs

1. Predictive Models:

* Anomaly detection models that can identify unusual energy usage patterns indicative of inefficiencies or potential safety hazards.

1. Actionable Insights:

* Real-time notifications and alerts for anomalies in energy consumption.
* Energy-saving recommendations based on behavioral patterns derived from qualitative data.

##### Integrated System Outputs:

* Seamless integration of machine learning prediction with the HCI subsystem, allowing residents and facility managers to visualize energy usage trends and receive personalized insights.

# Chapter 4: System Analysis and Design

This section provides a concise introduction to the machine learning (ML) component of the Energy Monitoring and Anomaly Detection System. The ML component is tasked with two primary functions:

* Anomaly Detection: Utilizing the Isolation Forest algorithm to identify unusual energy consumption patterns that may signal inefficiencies or potential safety hazards.
* Energy Forecasting: Employing an LSTM network to predict future energy usage trends based on historical time-series data.

The ML module operates as an independent, yet integral, component of the larger system. It interacts with the IoT component, which collects energy data, and with the HCI module, which visualizes and communicates the insights derived from ML models. This design ensures that data flows efficiently between the sensor network, cloud database, and user interface, facilitating real-time decision-making and prompt responses to anomalies.

## 4.1 Design Methodology

### 4.1.1 Design Approach

The design of the ML component is founded on an object-oriented methodology. This approach emphasizes modularity, encapsulation, and reusability, which are crucial for managing the complexity of a system that integrates multiple functionalities such as anomaly detection and energy forecasting, model training, and output generation, the system ensures that each module is independent yet interoperable. This structure allows for easier debugging, testing, and future enhancements. Moreover, the object-oriented paradigm supports abstraction, enabling the design to hide complex implementation details behind simple interfaces, which is beneficial when integrating with the IoT and HCI components of the broader system.

### 4.1.2 Justification for Using Object-Oriented Design

#### Modularity and Maintainability

Each component (e.g., DataLoader, Preprocessor, AnomalyDetector, EnergyForecaster) is encapsulated as a class, allowing for independent development, and maintenance. This separation of concerns simplifies updates and modifications, which is vital for a system that may evolve as new ML algorithms of data sources are incorporated.

#### Reusability

Object-oriented design promotes code reuse through inheritance and polymorphism. Common functionalities across different modules can be implemented in base classes and extended or overridden as needed, reducing redundancy and enhancing consistency.

#### Scalability

As the project grows, additional functionalities can be added without disrupting the existing system. The modular structure facilitates integration of new models or algorithms e.g., switching the Isolation Forest with another anomaly detection method with minimal changes to the overall architecture.

#### Ease of Collaboration

In a team setting, object-oriented design allows team members to work on separate modules concurrently. This clear delineation of responsibilities streamlines development and integrates seamlessly with version control practices.

### 4.1.3 Comparison with Alternative Approaches

Procedural programming: While a procedural approach can be simpler for small-scale applications, it becomes less manageable when dealing with complex systems that require frequent updates and integration with various components.

Service-Oriented Architecture (SOA): SOA focuses on communication between distinct services and is more suitable for distributed systems. In contrast, the object-oriented paradigm offers a more natural fit for tightly integrated ML component, where components must share and manipulate data directly in a cohesive environment.

### 4.2 Machine Learning Component Design

#### 4.2.1 Anomaly Detection Framework

The anomaly detection framework is designed to identify unusual patterns in energy consumption that may signal inefficiencies, equipment malfunctions, or potential safety hazards. The core of this framework is the Isolation Forest algorithm, chosen for its effectiveness in handling high-dimensional time-series data and its ability to isolate rare anomalies through random partitioning.

##### Isolation Forest Algorithm Overview

The Isolation Forest works by recursively partitioning the data space, randomly selecting a feature and a split value, until it isolates individual data points. Since anomalies are fewer and distinct from normal observations, they require fewer splits to isolate. This characteristic makes the Isolation Forest particularly suitable for identifying sporadic irregularities in energy usage.

##### Rationale for Selection

* **Efficiency and Scalability:** Isolation Forest is computationally efficient and scales well with large datasets, an essential feature for handling continuous streams of time-series energy data.
* **Robustness to Noise:** the algorithm is less sensitive to the presence of noisy data, which is common in real-world energy monitoring scenarios, ensuring that sporadic sensor errors do not lead to false-positives.
* **Unsupervised learning advantage:** Given the scarcity of labeled anomaly data in energy consumption, the unsupervised nature of Isolation Forest allows for effective anomaly detection without the need for extensive pre-labeled datasets.

##### Anomaly Detection Workflow Details

The framework begins with data preprocessing, where energy readings are cleaned, normalized and enriched with engineered features such as time-based-attributes and rolling averages. The preprocessed data is then used to train the Isolation Forest model. Once trained, the model assigns an anomaly score to each data point, with a threshold set to classify observations as anomalous (typically labeled as -1) or normal (labeled as 1). These scores are further used to trigger alerts, which are integrated into the system’s dashboard for real-time monitoring and decision-making.

The anomaly detection module is designed to operate in real time, continuously analyzing incoming data from IoT sensors. Its output is structured in a standardized format (e.g., JSON or CSV), ensuring seamless integration with both the data storage and HCI components of the system. This integration enables facility managers to receive prompt notifications and take corrective action when unusual energy consumption patterns are detected.

#### 4.2.2 Energy Forecasting Framework

The energy forecasting framework is designed to predict energy consumption patterns enabling proactive management of energy demand optimization of resource allocation. This component leverages Long Short-Term Memory (LSTM) networks, a specialized type of recurrent neural network (RNN) adept at modeling temporal dependencies in time-series data.

##### LSTM Network Overview

LSTM networks are specifically engineered to capture both short-term fluctuations and long-term trends within sequential data. By incorporating memory cells that preserve information over extended periods, LSTMs effectively model the complex temporal dependencies in energy consumption data.

##### Rationale for Choosing LSTM

* **Handling Temporal Dependencies:** Energy consumption data, characterized by daily, weekly, and seasonal patterns, requires a model that can learn from historical trends and context. LSTM networks excel in this regard, outperforming traditional models like ARIMA in scenarios with non-linear relationships and complex temporal dynamics.
* **Flexibility and Robustness:** LSTM models can adapt to various types of time-series data, including those with irregular patterns or missing values, making them suitable for real-world energy monitoring applications.
* **Enhanced Predictive Accuracy:** By learning from a continuous stream of historical data, LSTM networks provide more accurate short-term forecasts, which are critical for demand response strategies and scheduling maintenance activities.

##### Energy Forecasting Workflow

The forecasting process begins with extensive data preprocessing, which includes cleaning, normalization, and feature engineering to capture essential temporal attributes such as hour of day, day of week, rolling averages. These features are then combined with historical load data to form input sequences for the LSTM model.

Model Architecture: The LSTM network is structured to process input sequences e.g., past 24 hours of energy data and generate predictions for the subsequent 24 hours. The network typically comprises one or more LSTM layers followed by dense layers that output the forecasted energy consumption.

Training and Validation: The model is trained using historical energy data, with careful splitting into training, validation, and testing sets to preserve temporal order and prevent data leakage. Performance metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are used to evaluate and fine-tune the model.

Once Trained, the LSTM model is deployed to generate forecasts in near real time. The predictions are updates on an hourly basis, allowing the system to provide timely insights that inform energy management decisions.

The energy forecasting module operates in tandem with the anomaly detection framework, contributing a complementary perspective by anticipating future consumption patterns. Forecasted data is formatted in standardized outputs (e.g., JSON or CSV) and integrated with the HCI component, where it is visualized to help facility managers plan for peak usage periods, schedule maintenance and implement energy-saving measures.

### 4.3 Data Characteristics and Feature Engineering

### 4.3.1 Data Description

The foundation of the ML component lies in the quality and nature of the energy consumption data. In this system, the data is primarily time-series data collected via IoT sensors installed in residential halls. Key aspect include:

* Temporal Nature:
  + Data is recorded at fixed intervals (e.g., every 30 minutes) to capture both short-term fluctuations and long-term trends in energy usage.
  + Time stamps are critical to align and analyze over different periods (hourly, daily, weekly).
* Energy Readings:
  + The primary metric is the electrical load, measured in kilowatt-hours (kWh), which is derived from sensor measurements of voltage and current.
  + Additional metadata, such as sensor location or hall identifier, aids in contextualizing the consumption patterns.
* Data Volume and Frequency:
  + The continuous collection of data ensure a rich dataset, though it also necessitates efficient preprocessing and feature extraction to manage computational overhead.

### 4.3.2 Preprocessing and Feature Extraction

Preparing the raw data for ML models involves several key steps designed to improve data quality and highlight important features:

* Data Cleaning:
  + Handling Missing Values: Missing energy readings are addressed using linear interpolation to maintain the continuity of the time series.
  + Outlier Treatment: The interquartile Range (IQR) method is applied to detect and cap extreme values, mitigating the influence of sensor errors or anomalous spikes.
* Normalization:
  + To ensure that all feature contribute equally to the model, energy readings are scaled, using Min-Max normalization, which preserves the relative differences in consumption.
* Feature Engineering:
  + Temporal Features:
    - Hour of Day and Day of Week: These features help capture periodic patterns in energy usage.
    - IsWeekend: A binary feature to distinguish between weekday and weekend consumption behaviors.
  + Statistical Features:
    - Rolling Averages: Calculated over defined windows such as 24-hour moving average to smooth out short-term fluctuations and highlight underlying trends.
  + Symbolic Representation:
    - The Adaptive Symbolic Aggregate Approximation (aSAX) method is employed to reduce the dimensionality of high-frequency time-series data. This approach converts continuous readings to symbolic representations, thereby emphasizing key consumption patterns while filtering out noise.

### 4.3.3 Integration in the ML Pipeline

The cleaned and enriched dataset forms the basis for both the anomaly detection and forecasting modules. By transforming the raw data into a structured and informative format, these preprocessing steps enhance model performance and facilitate seamless integration with subsequent analysis components.

## 4.4 System Requirements

### 4.4.1 Functional Requirements

This section outlines the core functions that the ML component must perform to support the overall energy monitoring and anomaly detection system:

#### Data Ingestion

* The system must be capable of accepting energy consumption data via file uploads or through an API endpoint.
* It should validate incoming data to ensure it meets the expected format and integrity standards.

#### Data Preprocessing

* Automate data cleaning procedures, including handling missing values through interpolation and managing outliers via the IQR method.
* Normalize data using techniques such as Min-Max scaling.
* Extract relevant features including temporal (hour, day) flags and statistical attributes (moving averages).

#### Anomaly Detection

* Implement the Isolation Forest algorithm to analyze preprocessed data and identify anomalies.
* Assign anomaly scores to each data point and label them accordingly (e.g., -1 for anomalies, 1 for normal observations).

#### Energy Forecasting

* Utilize an LSTM network to forecast future energy consumption based on historical time-series data.
* Generates predictions on an hourly basis and update forecasts in near real time.
* Provide forecast outputs in a structured format (e.g., JSON, CSV) for integration with other system components.

#### Output Generation and Integration

* Export processed results, reports and forecasted values in standardized formats

### 4.4.2 Non-Functional Requirements

#### Performance

* The ML models must process data efficiently to provide real-time or near-real-time insights.
* For anomaly detection, evaluation metrics such a precision, recall, and F1-score should meet predefined thresholds.
* For energy forecasting, error metrics like Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) should be minimized.

#### Scalability

* The system should be designed to handle increasing volumes of data from multiple sensors without degradation in performance.
* It must support efficient querying and data aggregation, leveraging optimized data storage and processing techniques.

#### Security

* Implement robust encryption for data at rest and in transit to protect sensitive energy consumption data.
* Enforce access control and role-based privileges to prevent unauthorized access to the system.
* Include audit logs to monitor and record data access and modifications.

#### Maintainability

* The modular design should allow for straightforward updates and improvements.
* Comprehensive documentation and automated testing frameworks should be in place to facilitate ongoing maintenance.

#### Reliability and Availability

* Ensures that the system is robust against failures, with mechanisms for error handling and recovery.
* The system should maintain high availability to continuously support real-time monitoring and forecasting.

The ML component will integrate with other system modules (IoT data collection and HCI visualization), using standardized data formats and communication protocols.It should support automated data pipelines that seamlessly connect all components of the energy monitoring system.

## 4.5 Technology Stack

## 4.5.1 Software and Libraries

The ML component is primarily developed using Python due to its extensive support in data science and machine learning. Key libraries include:

* Pandas and NumPy: For efficient data manipulation, cleaning, and numerical computations.
* Scikit-learn: Utilized for implementing the Isolation Forest algorithm, enabling robust unsupervised anomaly detection
* TensorFlow: The framework used for constructing and training the LSTM network for energy forecasting.
* pytz: Employed for time series transformation, particularly for applying the Adaptive Symbolic Aggregate Approximation (aSAX) method to reduce data dimensionality.
* Matplotlib and Seaborn: Used for data visualization, allowing for detailed plotting of energy consumption trends and model performance metrics.
* pymongo: allows system to connect to and interact with MongoDB database.

### 4.5.2 Development Tools and Environments

To support collaborative development and efficient model training, the following tools and environments are employed:

* Visual Studio Code (VS Code): Serves as the primary integrated development environment (IDE) for coding, debugging, and version control.
* Google Colab: Provides cloud-based computing resources, including GPU acceleration, which are critical for training computationally intensive models such as LSTMs.
* Version Control: Git is used for source code management, ensuring that team collaboration is streamlined and that code changes are effectively tracked.

### 4.5.3 Data Storage and Integration

Efficient data handling and seamless integration across system components are achieved through:

* Data Formats: Standardized file formats such as CSV and JSON are used for data interchange between the ML component, IoT sensor, and the HCI module.
* Database Integration: Data ingested from IoT sensors is stored in a centralized MongoDB cloud database which facilitates real-time access and processing.
* API Endpoints: RESTful APIs are implemented to enable secure and efficient data transfer between the ML module and other components of the system, ensuring interoperability and modularity.

### 4.5.5 Infrastructure and Deployment

The system’s deployment strategy is designed to support both development and production environments:

* Local Development Environment: Initial development and testing are conducted on local machines using VS Code, allowing for rapid prototyping and debugging.
* Cloud-Based Training and Deployment: Critical components such as the LSTM network are trained on cloud platforms like Google Colab, which provide the necessary computational power and scalability.
* Modular Deployment: The ML component is designed as a standalone module that can be easily integrated with the IoT and HCI systems. This modularity ensures that updates and maintenance can be performed without impacting the overall system functionality.
* Security Measures: Data security is enforced through encryption protocols for data at rest and data in transit, and strict access controls, ensuring that sensitive energy consumption data is protected throughout the pipeline.

## 4.6 Visual Models and UML Diagrams

### 4.6.1. ML Pipeline

##### Data Flow and Preprocessing

To enhance the quality of our time series energy data and improve model performance, the preprocessing module includes several key steps: cleaning, normalization, and feature extraction. One significant enhancement in our approach is the integration of the pyts library, which implements Adaptive Symbolic Aggregate Approximation (aSAX).

The aSAX algorithm is used to reduce the dimensionality of our high-frequency energy consumption data by converting continuous numerical readings into symbolic representations.

This transformation is beneficial in several ways:

* Dimensionality Reduction: aSAX compresses the data while preserving essential consumption patterns, reducing the computational load on subsequent models.
* Noise Reduction and Smoothing: By representing the data symbolically, the algorithm helps filter our noise, making trends and anomalies more apparent.
* Enhanced Feature Extraction: The symbolic features generated by aSAX provide a robust basis for both the Isolation Forest used in anomaly detection and the LSTM model used for forecasting. These features capture temporal patterns such as seasonal variations and periodic trends more effectively than raw data.

Once the data has been transformed via aSAX, the resulting features are fed into the ML pipeline. The Isolation Forest algorithm uses the enhanced features to accurately identify and isolate anomalous energy consumption events, while the LSTM networks benefits from the reduced noise and complexity when forecasting future usage.

##### Anomaly Detection Module:

For anomaly detection, the system employs the Isolation Forest algorithm. This algorithm is chosen for its ability to isolate outliers by randomly partitioning data, thereby effectively identifying rare anomalous events. The model is trained on the preprocessed data, each data point is assigned a label which e.g., can be –1 for anomalies and 1 for normal observations. The module outputs a structured anomaly report.

##### Energy Forecasting Module:

The forecasting module uses an LSTM network to predict future energy consumption. LSTM networks are well-suited for time series data due to their ability to capture long-term dependencies. The model is trained using historical energy data (accumulated for over a month or more) in a cloud-based environment such as Google Colab. The forecasted values are then produced as structured output.

##### Database and Output Design:

The entire subsystems in this project share a common real-time database created using firebase hence there is need to ensure standardized data formats. Outputs, including anomaly reports and forecasted energy consumption, are exported in CSV or JSON format, making them easily consumable by other project components.

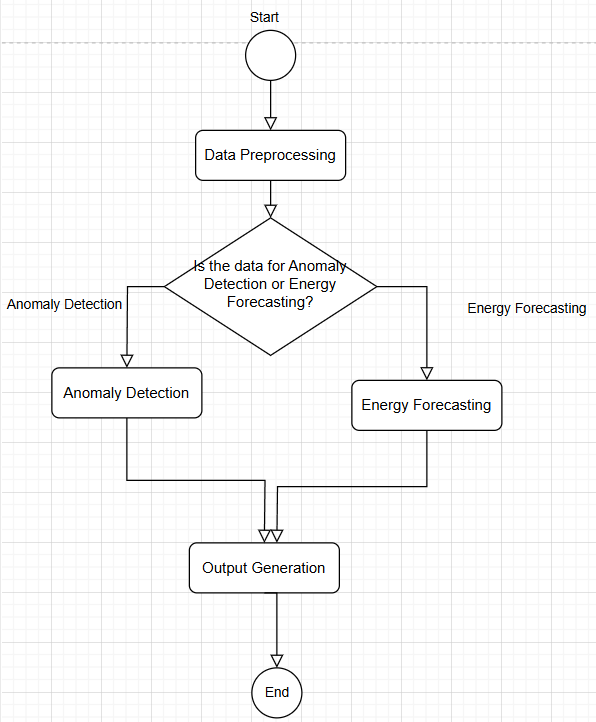


Figure 14: ML Pipeline Diagram

### 4.6.1 Use Case Diagram

The Use Case Diagram provides an overview of the interactions between the primary actors—such as the ML engineer, facility manager, or system user—and the ML component. This diagram highlights the following key use cases:

* Data Ingestion and Preprocessing
* Anomaly Detection and Alert Generation
* Energy Forecasting
* Output Generation and Reporting

This visualization helps illustrate how users interact with the system, ensuring that the functional requirements are clearly communicated and understood by all stakeholders.

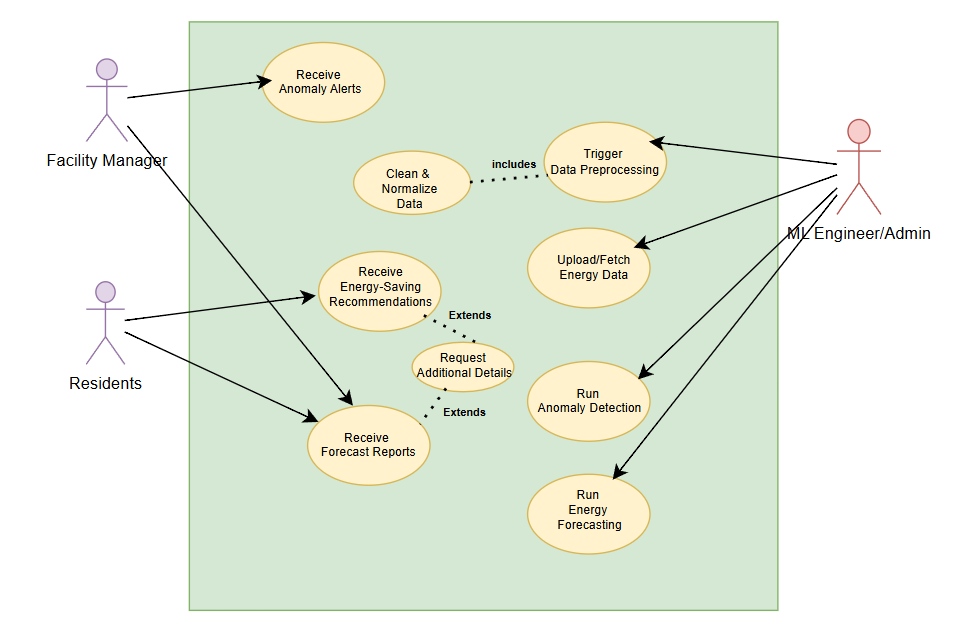


Figure 15:Use Case Diagram

*4.6.2 Class Diagram*  
The Class Diagram details the object-oriented structure of the ML component, emphasizing the modular design. Key classes and their relationships include:

* **DataLoader:** Handles data ingestion from various sources (CSV, JSON, API).
* **TimeSeriesPreprocessor:** Responsible for data cleaning, normalization, and feature engineering (including aSAX transformation).
* **AnomalyDetector:** Implements the Isolation Forest algorithm for anomaly detection, including methods for training and scoring.
* **EnergyForecaster:** Encapsulates the LSTM network for forecasting energy consumption, managing model training and prediction outputs.
* **OutputGenerator:** Manages the export of results into standardized formats (e.g., JSON, CSV) for integration with the HCI module.

This diagram reinforces the system’s modularity, showing how each class encapsulates a specific functionality while interacting with other components through well-defined interfaces.

A diagram with text and symbols

AI-generated content may be incorrect.

Figure 16: Class Diagram

The ML component for energy monitoring and anomaly detection has been carefully analyzed and designed to meet technical, operational, and legal feasibility criteria. The systems fulfill all functional requirements, ranging from data ingestion and preprocessing to anomaly detection via Isolation Forest and energy forecasting using LSTM while also addressing non-functional aspects such as performance, scalability, security and maintainability. The design is conveyed through a high-level diagram and detailed UML diagrams, ensuring that every stage of this process is clearly understood.

### 4.7.1Feasibility Analysis and Risk Assessment

#### 4.7.1.1 Technical Feasibility

#### Implementation Viability:

The ML component is developed using well-supported libraries in Python that facilitate robust data manipulation, anomaly detection, and forecasting. The use of cloud platforms such as Google Colab for model training and GPU acceleration ensures the computational requirements are met even with large-scale time-series data.

* Modular Architecture:

The object-oriented design enhances maintainability and scalability. Each module operates independently yet integrates seamlessly with the overall system, simplifying updates and facilitating troubleshooting.

#### 4.7.1.2 Operational Feasibility

* **Real-Time Processing:**  
  The system is designed to process and analyze incoming data in near real time. This responsiveness is crucial for timely anomaly detection and the generation of energy forecasts, which support proactive decision-making by facility managers.
* **Integration with Existing Systems:**  
  The ML component is engineered to interact efficiently with the IoT sensors for data collection and the HCI module for visualization, ensuring that all components of the energy monitoring system communicate effectively. Standardized data formats (CSV, JSON) and API endpoints further support smooth integration.
* **User Training and Documentation:**  
  Comprehensive documentation and clear UML diagrams provide guidance for maintenance and future system enhancements. Training materials ensure that users and system administrators can operate and manage the system effectively.

#### 4.7.1.3 Legal Feasibility

* **Data Privacy and Security:**  
  Energy consumption data is processed in compliance with relevant data protection regulations. Data is anonymized, encrypted at rest (using protocols like AES-256) and in transit (via TLS/SSL), and access is controlled through role-based privileges.
* **Use of Open-Source Tools:**  
  The reliance on open-source libraries and frameworks with permissive licenses minimizes legal risks related to intellectual property and copyright, ensuring that the system adheres to legal and ethical standards.

### 4.7.2 Risk Assessment and Mitigation Strategies

##### Data Quality Risks:

* *Risk:* Inaccurate sensor data or missing values may lead to incorrect model predictions.
* *Mitigation:* Implement rigorous data cleaning and interpolation methods, and use robust outlier detection (e.g., IQR) to maintain data integrity.

##### Model Performance Risks:

* *Risk:* The Isolation Forest and LSTM models may underperform if not properly tuned or if the data exhibits unforeseen patterns.
* *Mitigation:* Use cross-validation techniques and regularly update models with new data. Monitor performance using key metrics (precision, recall, RMSE, MAE) and adjust parameters as needed.

##### Integration Risks:

* *Risk:* Potential mismatches in data formats or communication protocols between the ML component and other system modules.
* *Mitigation:* Ensure strict adherence to standardized data formats and implement comprehensive testing during integration phases.

##### Security Risks:

* *Risk:* Unauthorized access to sensitive energy consumption data could result in data breaches.
* *Mitigation:* Enforce strong encryption, implement multi-factor authentication (MFA), and restrict access through network whitelisting and role-based access control (RBAC).
* **Class Diagram:**  
   Provides a detailed view of the core classes (e.g., DataLoader, TimeSeriesPreprocessor, AnomalyDetector, EnergyForecaster) and their relationships, emphasizing object-oriented design and modularity.

A diagram with text and symbols

AI-generated content may be incorrect.

Figure 16: Class Diagram

The ML component for energy monitoring and anomaly detection has been carefully analyzed and designed to meet technical, operational, and legal feasibility criteria. The systems fulfill all functional requirements, ranging from data ingestion and preprocessing to anomaly detection via Isolation Forest and energy forecasting using LSTM while also addressing non-functional aspects such as performance, scalability, security and maintainability. The design is conveyed through a high-level diagram and detailed UML diagrams, ensuring that every stage of this process is clearly understood.

# Conclusion

The proposed Energy Monitoring and Anomaly Detection System is designed to enhance energy efficiency, improve safety, and promote sustainable energy usage within university residential halls. By leveraging IoT technology, machine learning algorithms, and a user-centered HCI platform, the system provides real-time insights into energy consumption, detects anomalies that may indicate inefficiencies or faults, and generates actionable recommendations.

The anomaly detection framework, implemented using the Isolation Forest algorithm, effectively identifies unusual consumption patterns. This early detection enables proactive maintenance and supports energy conservation efforts. Concurrently, the energy forecasting model, powered by Long Short-Term Memory (LSTM) networks, delivers short-term predictions of energy usage, thereby facilitating demand optimization and the reduction of peak consumption periods. The cloud-based infrastructure ensures seamless data collection, storage, and analysis, which support both scalability and system integration.

Pilot deployment and experimental validation will assess the system’s accuracy and reliability in real-world conditions. Key performance metrics will be used to evaluate the effectiveness of the models.

Successful implementation of this system is expected to reduce energy wastage, lower operational costs, and enhance safety by identifying potential electrical hazards in advance. Moreover, the interactive user platform will empower residents with personalized insights, fostering responsible energy consumption behaviors. Overall, this project addresses a significant gap in energy management within institutional settings and provides a robust foundation for future enhancements, including expansion to additional buildings and the integration of advanced AI-driven energy optimization techniques.

# References

Abdel-Nasser, M., & Mahmoud, K. (2019). Accurate photovoltaic power forecasting models using deep LSTM-RNN. Neural Computing and Applications, 31(7), 2727–2740. DOI:[10.1007/s00521-017-3225-z](https://link.springer.com/article/10.1007/s00521-017-3225-z)

Alaraj, M., Kumar, A., Alsaidan, I., Rizwan, M., & Jamil, M. (2021). Energy Production Forecasting from Solar Photovoltaic Plants Based on Meteorological Parameters for Qassim Region, Saudi Arabia. IEEE Access, 9, 83241-83251. DOI:[10.1109/ACCESS.2021.3087345](http://dx.doi.org/10.1109/ACCESS.2021.3087345)

Arias-Requejo, D., Alonso-González, C. J., Pulido, B., & Keane, M. M. (2023). Clustering and deep-learning for energy consumption forecast in smart buildings. IEEE Access, 11, 322000–322019. <http://dx.doi.org/10.1109/ACCESS.2023.3331329>

Bitencourt, G. A., & Guimarães, F. G. (2021). High-dimensional Multivariate Time Series Forecasting in IoT Applications using Embedding Non-stationary Fuzzy Time Series, 38(1), e12545. DOI: [10.48550/arXiv.2107.09785](http://dx.doi.org/10.48550/arXiv.2107.09785)

Bouktif, S., Fiaz, A., Ouni, A., & Serhani, M. A. (2018). Optimal deep learning lstm for short-term load forecasting using feature selection and genetic algorithm. Energies, 11(7), 1690. <https://doi.org/10.3390/en11071636>

Bhargavi, V. S., & Yashasvi, S. (2021). Campus energy monitoring system. International Research Journal of Engineering and Technology (IRJET), 8(12). [IRJET-V8I12137.pdf](https://www.irjet.net/archives/V8/i12/IRJET-V8I12137.pdf)

Chiosa, R., Piscitelli, M. S., & Capozzoli, A. (2021). A Data Analytics-Based Energy Information System (EIS) Tool to Perform Meter-Level Anomaly Detection and Diagnosis in Buildings. Energies, 14(1), 237. <https://doi.org/10.3390/en14010237>

Das, H. P., Lin, Y.-W., Agwan, U., Spangher, L., Devonport, A., Yang, Y., Drgoňa, J., Chong, A., Schiavon, S., & Spanos, C. J. (2024). Machine learning for smart and energy-efficient buildings. Environmental Data Science, 3, e1. DOI: [10.1017/eds.2023.43](https://10.1017/eds.2023.43)

Deb, C., Zhang, F., Yang, J., Zhang, S. E., & Lee, W. J. (2017). A review on time series forecasting techniques for building energy consumption. Renewable and Sustainable Energy Reviews, 74, 901–924. <https://doi.org/10.1016/j.rser.2017.02.085>

de Lima Silva, P. C., Junior, C. A. S., Alves, M. A., Silva, R., Weiss-Cohen, M., & Guimarães, F. G. (2020). Forecasting in non-stationary environments with fuzzy time series. Applied Soft Computing, 97, 106825. <https://doi.org/10.1016/j.asoc.2020.106825>

dos Santos, R. B., de Lima Silva, P. C., & Guimarães, F. G. (2021). A fuzzy time series forecasting model based on data embedding and nonlinear transformation. IEEE Transactions on Fuzzy Systems, 30(3), 715-725. <https://doi.org/10.1109/TFUZZ.2021.3058765>

Elhabyb, K., Baina, A., Bellafkih, M., & Deifalla, A. F. (2024). Machine learning algorithms for predicting energy consumption in educational buildings. International Journal of Energy Research, 2024, 6812425. <https://doi.org/10.1155/2024/6812425>

He, Y., & Tsang, E. C. (2021). A hybrid forecasting model combining statistical and machine learning techniques for day-ahead wind power prediction. Renewable Energy, 172, 717-730. <https://doi.org/10.1016/j.renene.2021.03.097>

Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: principles and practice. OTexts. <https://otexts.com/fpp3/>

Jiang, Y., Li, X., & Zhou, H. (2023). Forecasting household energy consumption based on lifestyle data using hybrid machine learning. *Journal of Electrical Systems and Information Technology, 10*(3), 123–132. <https://doi.org/10.1186/s43067-023-00104-2>

Kim, J.-H., Seong, N.-C., & Choi, W. (2020). Forecasting the energy consumption of an actual air handling unit and absorption chiller using ANN models. Energies, 13(17), 4361. <https://doi.org/10.3390/en13174361>

Kumar, M., Shenbagaraman, V. M., Shaw, R. N., & Ghosh, A. (2021). Predictive Data Analysis for Energy Management of a Smart Factory Leading to Sustainability. In Lecture Notes in Electrical Engineering (pp. 569-579). Springer, Singapore. <https://doi.org/10.1007/978-981-15-4692-1_58>

Lezmi, E., & Xu, J. (2023). Time Series Forecasting with Transformer Models and Application to Asset Management. DOI: [10.2139/ssrn.4375798](http://dx.doi.org/10.2139/ssrn.4375798)

Liu, X., Lai, Z., Wang, X., Huang, L., & Nielsen, P. S. (2020). A contextual anomaly detection framework for energy smart meter data stream. In Proceedings of the International Conference on Neural Information Processing (pp. 733–742). Springer. <https://doi.org/10.1007/978-3-030-63823-8_83>

Makaa, B. (2024). Energy Management Systems (EMS): Transforming energy use for Kenyan households. Electrical Engineering in Kenya.https://www.eeekenya.com/energy-management-systems-ems-transforming-energy-use-for-kenyan-households/

Mansattha, M., Dao, H., & Jikaraji, A. (2023). Smart meter design for energy consumption monitoring of residential premises. ResearchGate. Doi link: <https://doi.org/10.60101/jarst.2023.250745>

Piscitelli, M. S., Brandi, S., & Capozzoli, A. (2019). Recognition and classification of typical load profiles in buildings with non-intrusive learning approach. Applied Energy, 255, 113727. <https://doi.org/10.1016/j.apenergy.2019.113727>

Powroźnik, P., & Szcześniak, P. (2024). Predictive Analytics for Energy Efficiency: Leveraging Machine Learning to Optimize Household Energy Consumption. Energies, 17(23), 5866. <https://doi.org/10.3390/en17235866>

Sumorek, M., & Idzkowski, A. (2023). Time Series Forecasting for Energy Production in Stand-Alone and Tracking Photovoltaic Systems Based on Historical Measurement Data. Energies, 16(17), 6367. <https://doi.org/10.3390/en16176367>

# Appendices

## Project Plan and Timeline

Below is a structured breakdown of tasks and deadlines for implementing the Energy and Anomaly Detection System. This helps in tracking progress and ensuring timely completion.

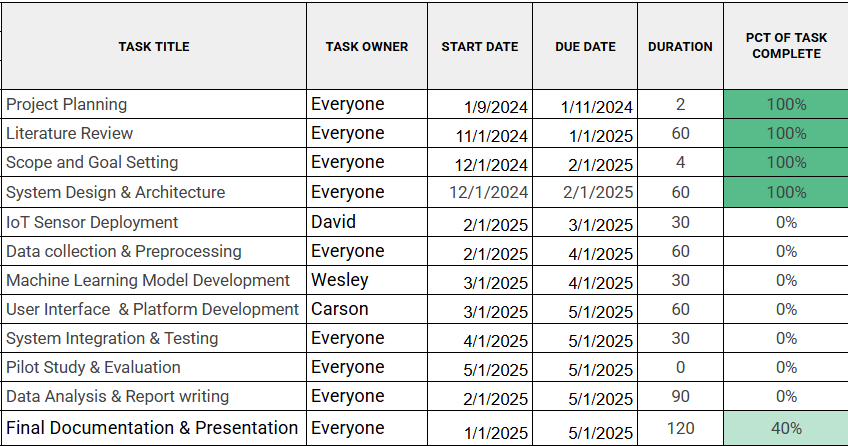


Figure 17: Project plan and timeline

## Budget

Below is an outline of the financial aspects of the project, including costs related to IoT sensors, cloud storage, and system deployment. Some of the equipment has already been acquired.

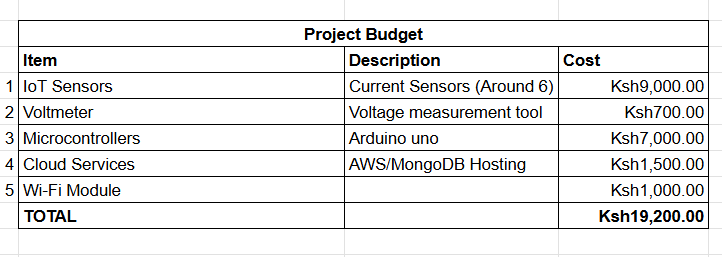


Figure 18: Project budget