VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590018, KARNATAKA.

A PROJECT REPORT

On

"SMART ENERGY MANAGEMENT & BILL PREDICTION USING IOT AND MACHINE LEARNING"

Submitted in Partial Fulfillment for the Award of the Degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Submitted By:

1SG21CS048	**MANOJKUMAR N**
1SG21CS105	**SOURABH A KANDAGAL**
1SG21CS112	**TEJAS A M**
1SG22CS405	**MALHARI MAGADUM**

Under the Guidance of
Prof. Hemalatha K
Assistant Professor

Department of Computer Science and Engineering

(Accredited by NBA)

SAPTHAGIRI COLLEGE OF ENGINEERING

(Affiliated to Visvesvaraya Technological University, Belagavi& Approved by AICTE, New Delhi.)
ISO 9001-2015 & 14001-2015 Certified, Accredited by NAAC with 'A' Grade
14/5, Chikkasandra, Hesarghatta Main Road
Bengaluru – 560057.

2024-2025

SAPTHAGIRI COLLEGE OF ENGINEERING

(Affiliated to Visvesvaraya Technological University, Belagavi& Approved by AICTE, New Delhi.)

ISO 9001-2015 & 14001-2015 Certified, Accredited by NAAC with 'A' Grade

Bengaluru-560057

Department of Computer Science and Engineering

(Accredited by NBA)

Certificate

Certified that the Project Work entitled **"SMART ENERGY MANAGEMENT & BILL PREDICTION USING IOT AND MACHINE LEARNING "** carried out by **MANOJKUMAR N (1SG21CS048), SOURABH A KANDAGAL (1SG21CS105), TEJAS A M (1SG21CS112), MALHARI MAGADUM (1SG22CS405),** bonafide students of **Sapthagiri College of Engineering**, in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** of **Visvesvaraya Technological University**, **Belagavi** during the academic year 2024-2025. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the Department Library. The project report has been approved as it satisfies the academic requirements in respect of **Project Work (21CSP76)** prescribed for the said degree.

Signature of the Guide	Signature of the HOD	Signature of the Principal
Prof. Hemalatha K	**Dr. Kamalakshi Naganna**	**Dr. H Ramkrishna**
Assistant Professor	**Professor & Head**	**Principal**

EXTERNAL EXAMINATION:

Name of the Examiners	Signature with Date
1.______________________________	______________________________
2.______________________________	______________________________

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement ground my efforts with success.

I consider it is a privilege to express my gratitude and respect to all those who guided me in completion of Final Project.

I am, grateful to thank our Principal **Dr. H Ramakrishna**, Sapthagiri College of Engineering, who patronized throughout our career & for the facilities provided to carry out this work successfully.

It's a great privilege to place on record my deep sense of gratitude to our beloved HOD **Dr. Kamalakshi Naganna** of Computer Science & Engineering, who patronized throughout our career & for the facilities provided to carry out this work successfully.

I am also grateful to thank my Final Project Guide **Prof. Hemalatha K,** Assistant Professor of CSE department for her invaluable support and guidance.

I would also like to thank the teaching and non-teaching staff members who have helped me directly or indirectly during the project.

Lastly but most importantly I would like thank my family and friends for their co-operation and motivation to complete this project successfully.

MANOJKUMAR N **1SG21CS048**
SOURABH A KANDAGAL **1SG21CS105**
TEJAS A M **1SG21CS112**
MALHARI MAGADUM **1SG22CS405**

ABSTRACT

Traditional domestic energy meter reading systems face several challenges, including complex construction, limited bandwidth, low data rates, poor real-time monitoring, and lack of two-way communication. To address these limitations, this project proposes an IoT-based Automatic Meter Reading (AMR) system, which enables seamless communication between the Electricity Board and consumers through a wireless smart metering solution.

The proposed system utilizes IoT technology for real-time electricity consumption monitoring and automated billing. An Arduino-based energy meter, equipped with an ESP32 Wi-Fi module, records power usage and transmits the data securely to the cloud. Consumers can access their electricity usage, bill details, and payment history via a user-friendly Blynk mobile application, ensuring transparency and convenience. Additionally, an integrated Billing System enables users to make hassle-free bill payments directly through the platform, streamlining the traditional manual payment process.

To further enhance the system, Machine Learning (ML) using Linear Regression is implemented to predict future electricity bills based on historical consumption patterns. This predictive model allows users to anticipate their energy costs, optimize consumption habits, and plan expenses more efficiently. The fusion of IoT-based smart metering with ML-driven analytics results in a highly automated, scalable, and user-centric electricity billing solution, significantly improving accuracy, real-time tracking, and energy management.

This system ultimately bridges the gap between consumers and utility providers, modernizing energy metering with enhanced efficiency, automation, and intelligent forecasting to support a smarter and more sustainable power management system.

SAPTHAGIRI COLLEGE OF ENGINEERING

(Affiliated to Visvesvaraya Technological University, Belagavi& Approved by AICTE, New Delhi.)
ISO 9001-2015 & 14001-2015 Certified, Accredited by NAAC with 'A' Grade
#14/5, Chikkasandra, Hesaraghatta Main Road,
Bengaluru-560057

Department of Computer Science and Engineering

(Accredited by NBA)

DECLARATION

We **MANOJKUMAR N(1SG21CS048), SOURABH A KANDAGAL(1SG21CS105), TEJAS AM (1SG21CS112)** and **MALHARI MAGDUM(1SG22CS405),** bonafide students of **Sapthagiri College of Engineering**, hereby declare that the project entitled **"SMART ENERGY MANAGEMENT & BILL PREDICTION USING IOT AND MACHINE LEARNING"** submitted in partial fulfilment for the award of Bachelor of Engineering in **Computer Science & Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024-2025 is our original work and the project has not formed the basis for the award of any other degree, fellowship or any other similar titles.

Name of the Student	Signature of the Student with date
1) MANOJKUMAR N (1SG21CS048)	______________________________
2) SOURABH A KANDAGAL (1SG21CS105)	______________________________
3) TEJAS A M (1SG21CS112)	______________________________
4) MALHARI MAGADUM (1SG22CS405)	______________________________

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 Embedded system and RTOS

The role of Embedded Systems and Real-Time Operating Systems (RTOS) is crucial in enabling such advanced energy management solutions. Embedded systems are pre- designed, task-specific computing systems without user-configurable connections, unlike general-purpose operating systems, which are designed for multiple applications. They play a vital role in many real-time applications, providing reliable, efficient, and performance- driven solutions.

The Electricity Bill Prediction System is an advanced web-based application designed to assist users in estimating their electricity expenses for the upcoming month. With the continuous rise in energy demand and increasing electricity costs, it has become crucial for households and businesses to monitor their consumption and plan their budgets effectively. This system employs machine learning techniques to analyze historical electricity usage patterns and generate precise bill predictions, enabling users to make informed financial decisions.

Embedded systems are already integrated into numerous modern technologies, from household appliances such as washing machines and air conditioners to complex automotive systems such as anti-lock braking systems (ABS) and security mechanisms. These systems ensure seamless functionality with minimal hardware, avoiding driver conflicts and unnecessary system delays. In automobiles, embedded technology adjusts suspension settings based on road conditions, optimizes fuel efficiency, and enhances safety through real-time monitoring systems. Embedded systems and Real Time Operating systems are two among the several technologies that will play a major role in making these concepts possible. A large number of people are already depending on operating systems for real time applications, these ‘eyes in the sky’ are now going to make an impact on our everyday lives in a more significant manner. Embedded systems are pre-designed without connections and operate as per the required task. But in operating systems instruction is design-oriented.

These systems are basically platform-less systems. Embedded systems are the unsung heroes of much of the technology we use today the video game we play, or the CD player or the washing machines we use employ them. Without an embedded system we would not even be able to go online using modem. Embedded systems are specialized computing devices designed to perform dedicated functions, often without direct user interaction. They are integral to modern automobiles, household devices, and industrial applications, ensuring real-time responses and efficiency. The Electricity Bill Prediction System aims to help users estimate their energy costs using a machine learning-based regression model.

Energy Metering

Energy meters are essential devices used to measure and monitor electricity consumption in residential, commercial, and industrial setups. Over the years, traditional mechanical meters have been replaced by digital and smart meters, improving accuracy, efficiency, and real- time data access.

Electricity Bill Prediction System

The Electricity Bill Prediction System is a machine learning-based web application designed to help users estimate their electricity expenses. It uses historical electricity consumption data to predict future bills, allowing for better budget planning and energy management.

1.1.1 Advantages

1. **Efficiency & Speed –** Embedded systems are optimized for specific tasks, resulting in fast and reliable performance.
2. **Real-Time Processing** – Essential in automotive applications (e.g., antilock braking systems (ABS)) where immediate responses are required.
3. **Automation & Convenience –** Systems like electronic energy meters (EEMs) reduce manual efforts and improve accuracy.
4. **Cost Savings –** The Electricity Bill Prediction System enables better financial planning, helping users monitor energy consumption.
5. **Compact & Low Power Consumption** – Embedded systems consume less power and require minimal maintenance compared to traditional computing devices.
6. **Remote Monitoring & Control –** Digital energy meters and IoT-based embedded systems allow real-time data collection and remote monitoring.

1.1.2 Disadvantages

i. **Limited Upgradability** – Embedded systems are custom-built, making software or hardware upgrades difficult.

ii. **Complex Debugging & Repair** – Unlike general-purpose computers, diagnosing and fixing issues in embedded systems requires specialized knowledge.

iii. **High Development Costs** – Custom-built solutions, such as ASIC-based EEMs, can be expensive to design and implement.

iv. **Limited Functionality** – Most embedded systems are designed for single tasks, lacking flexibility.

v. **Dependence on Hardware** – System failures can render the entire device non-functional, leading to costly replacements.

1.1.3 Privacy Concerns

i. **Data Security Risks –** The Electricity Bill Prediction System collects and stores user data, posing risks of unauthorized access and data breaches.

ii. **Unauthorized Surveillance** – Embedded systems in smart meters, automobiles, and IoT devices can potentially be exploited for monitoring user activities.

iii. **Cyber Threats** – The increasing reliance on internet-connected devices makes them vulnerable to hacking and data theft.

iv. **Third-Party Data Sharing –** Companies managing electricity consumption data may share user information with third parties, raising concerns about data misuse.

1.2 The need of the smart meters

Smart meters play a crucial role in modernizing electricity management by ensuring accurate billing and real-time monitoring. Traditional meters often lead to billing errors due to manual readings, but smart meters eliminate these issues by automatically transmitting consumption data. This not only enhances accuracy but also reduces disputes between consumers and electricity providers.

One of the key benefits of smart meters is real-time monitoring, allowing users to track their electricity usage instantly. By understanding their consumption patterns, consumers can make informed decisions to reduce unnecessary power usage, leading to significant cost savings and better energy management. Additionally, Automatic Meter Reading (AMR) technology enables remote data collection, eliminating the need for physical meter checks and reducing operational costs for utility companies.

Smart meters also play a vital role in load management. By analyzing real-time consumption data, electricity providers can balance demand and supply more efficiently, preventing overloads and power outages. Moreover, they enable two-way communication between consumers and providers, allowing quick responses to faults, service issues, and demand fluctuations.

Energy conservation is another major advantage of smart meters. By providing insights into electricity usage, these devices encourage consumers to adopt energy-efficient practices, ultimately promoting sustainability. They also support prepaid and flexible billing options, helping users manage their expenses effectively while avoiding unexpected high bills.

Security and theft prevention are also strengthened with smart meters. They can detect irregularities, such as unauthorized usage or tampering, reducing electricity theft and revenue losses for utility companies. Furthermore, smart meters facilitate the integration of renewable energy sources like solar and wind power, enabling better grid management and a greener energy future.

Lastly, smart meters help governments and utility companies optimize electricity distribution, improve infrastructure, and implement better energy policies. By enhancing efficiency, reducing costs, and promoting sustainability, smart meters are an essential component of the future of electricity management.

1.3 Smart meter and bill prediction using machine learning

With the rapid advancement of technology, smart meters have become an essential component of modern energy management. Unlike traditional electricity meters, which require manual readings and are prone to errors, smart meters automatically record and transmit electricity consumption data in real time. This ensures accurate billing, minimizes discrepancies, and provides consumers with insights into their energy usage. Additionally, smart meters facilitate two-way communication between consumers and electricity providers, allowing for efficient load management, faster issue resolution, and greater transparency in billing.

However, while smart meters ensure precise readings, predicting future electricity bills remains a challenge due to fluctuating consumption patterns and changing tariff rates. This is where Machine Learning (ML) plays a transformative role. ML algorithms, particularly regression models, neural networks, and time-series forecasting techniques, analyze historical consumption data to predict future electricity bills with high accuracy. These models consider multiple factors, including past energy usage, peak demand hours, seasonal variations, and even external influences such as weather conditions and household occupancy, to generate reliable bill estimates.

Smart meters play a vital role in Machine Learning (ML)-based bill prediction systems by continuously capturing real-time electricity consumption data. This detailed data helps create a rich dataset for training predictive models, allowing ML algorithms to identify consumption trends, peak usage periods, and anomalies, resulting in more accurate bill predictions compared to traditional methods.

The process begins with data collection from smart meters, followed by data preprocessing, which involves cleaning and structuring the data to remove inconsistencies. In the next step, feature selection and engineering are performed, extracting relevant factors like past meter readings and weather conditions to improve prediction accuracy. Afterward, various ML models, such as Linear Regression and Random Forest, are trained on historical data to learn electricity consumption patterns. Finally, the trained model predicts future bills, offering users insights, alerts, and recommendations to optimize their energy usage and reduce excess consumption.

ML-based bill prediction enhances accuracy by analysing real-time data, reducing billing errors. It enables proactive energy management, helping consumers optimize usage and costs. Smart meters with ML detect anomalies, prevent fraud, and support efficient renewable energy integration.

1.3.1 Potential Applications

One key application is **proactive energy management**, where consumers can track real-time energy consumption, receive predictive bill estimates, and adjust usage to avoid high costs. This not only helps in optimizing consumption but also encourages responsible energy use. Additionally, **dynamic pricing models** can be implemented by utility providers, adjusting electricity rates based on consumption trends, peak times, or real-time demand, allowing consumers to benefit from more cost-effective pricing. ML can also enhance **demand forecasting and grid management**, enabling utilities to predict energy needs, prevent outages, and ensure efficient distribution.

Another significant application is in **fraud detection and prevention**. ML algorithms can detect unusual consumption patterns, helping to identify electricity theft or tampering with meters. Moreover, **energy efficiency and sustainability** are improved by using predictions to identify areas of excessive power consumption, leading to optimized energy use and lower carbon footprints. For **renewable energy integration**, ML can forecast energy generation from sources like solar or wind and balance it with demand, enhancing grid efficiency. Finally, integrating ML with **smart home systems** can allow automated adjustments based on predicted energy usage, further reducing waste and promoting sustainable living.

1.3.2 Embedded technology

Embedded technology plays a crucial role in the functioning of **smart meters**, which are advanced devices designed to measure electricity consumption in real-time. These smart meters rely on embedded systems for data collection, processing, and communication, enabling accurate and efficient billing and energy management.

At the heart of a smart meter's functionality is its **embedded processor**, which handles the core task of reading electricity consumption data from sensors. The **sensors** embedded in the meter record real-time energy usage, while the processor ensures that this data is processed and stored accurately. These embedded systems often include **communication modules**, such as **Wi-Fi, Zigbee, or cellular networks**, which transmit data to utility providers for remote monitoring, billing, and management. This allows for **two-way communication**, enabling not only the transmission of data to the utility provider but also the reception of commands, such as tariff updates or remote disconnections.

The use of embedded *technology* in smart meters also facilitates the integration of **real-time energy usage monitoring**, enabling consumers to track their consumption on a regular basis. These systems help in **dynamic pricing**, where electricity costs are adjusted based on usage patterns and peak demand times. Furthermore, the embedded systems can analyze **anomalous consumption patterns**, aiding in **fraud detection** and identifying potential faults or issues in the electrical grid. Embedded technology makes smart meters both **energy-efficient** and **cost- effective**, ensuring that these systems operate seamlessly with minimal human intervention while providing reliable data for energy management and optimization.

1.3.3 Applications

Clear and Accurate Consumption Readings: Smart meters provide consumers and property owners with precise and real-time readings of their electricity usage. This transparency allows users to monitor their consumption patterns closely, leading to better management of energy usage and more accurate billing.

i. **Automatic Disconnection for Non-Payment:** Smart meters enable automatic disconnection of electricity supply through the internet if the bill remains unpaid for a specified period. This automated feature ensures that consumers remain accountable for timely payments and helps utility providers efficiently manage billing cycles.

ii. **Secure Data Access:** The data collected by smart meters is securely stored and transmitted, preventing unauthorized access. Only authorized entities have access to this information, ensuring privacy and the integrity of usage data while preventing potential breaches or misuse of consumer information.

iii. **Tamper Detection and Prevention:** Smart meters are equipped with tamper detection technology that helps to identify unauthorized alterations or attempts at power theft. By monitoring abnormal usage patterns and irregularities, smart meters act as a deterrent against fraud, promoting fair usage and minimizing power theft.

iv. **Power Generation Control**: Smart meters are integrated with energy management systems to regulate and optimize power generation. They provide real-time data on consumption, enabling energy producers to adjust power generation based on demand and supply fluctuations.

v. **Residential Smart Metering:** For residential use, smart meters offer homeowners detailed insights into their electricity consumption, broken down by time of day or device usage. This information helps homeowners make informed decisions about their energy consumption, adopt energy-saving practices, and reduce electricity costs. Smart meters also support dynamic pricing, enabling consumers to save by adjusting their usage during off-peak hours

1.4 Background

Energy consumption has been increasing worldwide due to urbanization, industrialization, and the growing reliance on electrical appliances. Traditional energy metering and billing systems face several challenges, including manual meter readings, delays in bill generation, and inefficiencies in energy management. Moreover, issues such as electricity theft, tampering, and inaccurate billing further complicate the process.

To address these challenges, smart energy management and billing systems have emerged as a revolutionary approach, leveraging the Internet of Things (IoT) and Machine Learning (ML). IoT enables real-time monitoring and automation, allowing consumers and electricity providers to track consumption patterns accurately. Machine Learning techniques, particularly Linear Regression models, can predict future energy consumption based on historical data, helping users optimize usage and control costs.

A smart energy management system integrates real-time data collection, predictive analytics, and automated billing, providing a seamless, efficient, and secure energy distribution network. By incorporating features such as real-time alerts, energy theft detection, and automated billing, these systems not only enhance energy efficiency but also improve transparency between consumers and providers. With the growing adoption of smart grids and IoT technologies, the implementation of such systems is a crucial step toward achieving sustainable and optimized energy usage. Furthermore, the adoption of IoT-based smart energy management systems has the potential to transform traditional power distribution networks into intelligent, automated infrastructures. By integrating IoT-enabled smart meters, real-time data can be transmitted to both consumers and electricity providers, allowing for better demand-side management. This ensures that electricity is distributed efficiently, reducing energy wastage and enabling dynamic pricing strategies.

In addition to real-time monitoring and automated billing, the incorporation of Machine Learning in energy management introduces predictive capabilities that can help users make informed decisions about their electricity consumption. By analyzing historical data and identifying trends, ML algorithms can forecast upcoming energy demands, allowing users to adjust their usage and save on costs. This predictive approach is especially useful in industrial and commercial settings, where high electricity consumption can lead to significant expenses. Overall, by leveraging IoT and ML technologies, smart energy management systems contribute to a more sustainable and efficient power ecosystem, promoting responsible energy consumption and reducing the strain on national power grids.

1.5 Overview of the present work

The present work focuses on developing a Smart Energy Management and Billing System using IoT (Internet of Things) and Machine Learning (ML) to enhance the efficiency, reliability, and accuracy of electricity monitoring and billing processes. Traditional energy meters often suffer from inefficiencies such as manual reading errors, billing delays, and the inability to provide real- time data on power consumption. To overcome these limitations, this system integrates smart metering technology with real-time data acquisition and analysis, ensuring automated and accurate billing for consumers while improving grid management for utility providers.

The core functionality of this system revolves around IoT-enabled smart meters that monitor real- time electricity usage and transmit the data wirelessly to a centralized server. The use of Arduino- based energy monitoring ensures that power consumption is recorded with high precision, while the implementation of wireless communication eliminates the need for manual meter reading. The system also incorporates a web-based dashboard and mobile application that allows users to track their electricity usage, receive billing notifications, and make informed decisions about energy consumption.

One of the key innovations in this project is the use of Machine Learning algorithms for bill prediction. By analyzing historical electricity consumption data, an ML model based on Linear Regression is trained to forecast future energy bills. This predictive capability allows consumers to anticipate their upcoming electricity costs and take proactive measures to optimize their energy usage. Additionally, the system can generate real-time alerts and notifications for unusual consumption patterns, helping in early detection of power theft or appliance malfunctions. Another important aspect of this work is its focus on security and tamper detection. Power theft and unauthorized meter tampering are significant challenges faced by energy providers. This system integrates tamper detection mechanisms that analyze usage patterns for anomalies, triggering alerts in case of suspicious activities. This ensures a transparent and secure energy management process, benefiting both consumers and utility companies.

Overall, this project provides a comprehensive, automated, and intelligent energy management solution that enhances billing accuracy, improves energy efficiency, and empowers consumers with real-time insights. By integrating IoT, Machine Learning, and predictive analytics, the system contributes to a more sustainable and efficient power distribution network, paving the way for smart cities and intelligent power grids of the future.

1.6 PROBLEM STATEMENT

How can we create a real-time energy management system that optimizes monitoring, billing accuracy, theft detection, and analyzes the next month's bill to minimize costs and energy inefficiencies?

1.7 Objectives

i. **Efficient and Reliable Grid –** Develop an energy management system that optimizes electricity distribution and minimizes energy losses.

ii. **Accurate and Automated Billing –** Implement an IoT-based automatic meter reading system to ensure precise and real-time billing.

iii. **Power Theft Detection and Prevention –** Integrate tamper detection mechanisms to identify unauthorized access and prevent electricity theft.

iv. **Real-Time Energy Monitoring –** Enable consumers to track their electricity usage in real- time through a web or mobile dashboard.

v. **Machine Learning-Based Bill Prediction –** Use Linear Regression models to analyze past consumption data and predict future energy bills.

vi. **User Notifications and Alerts –** Provide real-time alerts for high consumption, unusual activity, and scheduled maintenance.

vii. **Seamless Online Payment Integration –** Support multiple payment options like UPI, credit/debit cards, and net banking for convenient bill payments.

.

1.8 Organization of report

I. **Chapter 1: Introduction**

This chapter introduces the smart meter management system, outlining the problem statement, objectives, limitations, and literature survey. It provides background information on energy metering systems and machine learning applications for billing predictions.

II. **Chapter 2: Analysis**

The analysis chapter reviews prior works in smart metering and machine learning models for energy prediction. It highlights the identified gaps in current systems and presents the proposed solution that integrates machine learning to predict next-month billing.

III. **Chapter 3: System Requirements**

This chapter defines the functional and non-functional requirements of the smart meter management system. It also specifies the hardware and software needed for data collection, processing, and prediction.

IV. **Chapter 4: System Design**

The system design chapter discusses the architecture of the proposed system, detailing the interaction between components like the smart meter, sensors, and machine learning model. It explains the control flow between these components to ensure efficient system operation.

V. **Chapter 5: Implementation**

This chapter covers the implementation of the system, detailing how various modules such as data collection, machine learning model training, and integration work together. It discusses the specific algorithms used for billing prediction.

VI. **Chapter 6: Testing**

The testing chapter focuses on unit testing of individual components, integration testing of subsystems, and overall system testing. It ensures the system works as intended and meets the requirements for real-time energy prediction and billing.

VII. **Chapter 7: Conclusion and Future Enhancements**

The conclusion summarizes the achievements and outcomes of the system, while the future enhancements section suggests improvements like advanced algorithms and additional features for real-time monitoring and dynamic pricing.

CHAPTER 2

LITERATURE REVIEW

CHAPTER 2

LITERATURE REVIEW

2.1 Summary of Prior Works

Smart energy management and billing systems have significantly evolved with the integration of IoT and machine learning technologies. Several studies have explored digital energy meters, automated meter reading (AMR) systems, GSM-based communication, and smart theft detection mechanisms.

One of the key studies on digital energy metering is by Shahidi et al. (2013), which proposed a wireless energy meter (WEM) utilizing GSM technology for data transmission. The system measured energy consumption using a microcontroller-based digital meter with phase difference calculations between current and voltage using zero-cross detection. This study highlighted the efficiency of real-time data transmission and cost reduction in meter reading but was limited by GSM network availability.

Another significant work by Mehmood et al. (2011) introduced an electronic meter reader and database management system that transmitted local area meter readings to a centralized billing station. The research focused on integrating analog meters with digital circuitry and GSM communication for real-time billing. Their approach reduced human errors in meter reading but relied on prepaid and postpaid models, requiring users to manually recharge their accounts.

Visalatchi et al. (2017) investigated smart energy metering and power theft control using Arduino and GSM technology. Their system leveraged Atmega328P microcontrollers, solid-state relays, and GSM modules for remote disconnection and reconnection of electricity supply. The study addressed non-technical losses and tampering issues, enabling real-time alerts to utility providers. However, it did not incorporate predictive analytics for energy consumption optimization.

Furthermore, a study on electricity billing and consumption prediction using machine learning utilized algorithms such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Artificial Neural Networks (ANN) to predict electricity usage and bill amounts. The research demonstrated that ML models can significantly enhance the accuracy of energy consumption forecasting, helping both consumers and utility providers.

These studies collectively highlight the progress in automated billing and energy management, showing the transition from traditional manual readings to smart metering systems with GSM and IoT-based communication. However, most systems lack predictive capabilities and adaptive learning for energy optimization, which our proposed work aims to address.

.

2.2 Outcome of the review

From the literature review, several key observations emerge:

i. **Automation in Energy Metering:** Traditional energy meters require manual reading, which is prone to errors and inefficiencies. The integration of GSM and IoT-based communication has enhanced the automation of meter reading and billing, reducing human dependency.

ii. **Wireless Communication:** GSM-based systems have improved data transmission efficiency, enabling real-time monitoring. However, they depend on network availability, and alternative communication methods such as LPWAN or LoRaWAN should be explored.

iii. **Energy Theft Detection:** Smart metering systems integrated with microcontrollers and solid-state relays have been effective in detecting unauthorized access and tampering. Automatic alerts and remote disconnection mechanisms have enhanced security and minimized losses.

iv. **Database Management and User Interaction:** Many existing models have focused on real-time data logging and customer interaction via prepaid models. However, predictive analytics and dynamic tariff adjustments based on consumption trends remain underdeveloped areas.

v. **Need for Machine Learning Integration:** While smart meters have enabled automation, most systems do not leverage machine learning algorithms for predictive energy consumption analysis, anomaly detection, and demand forecasting. This limitation hinders proactive energy management.

2.3 Proposed work

Our proposed system, "Smart Energy Management & Billing with IoT and Machine Learning Prediction," aims to overcome the limitations identified in prior research by integrating real-time monitoring, predictive analytics, and automated billing.

i. **IoT-Based Smart Metering:** The system will use IoT-enabled digital energy meters with Wi-Fi or LPWAN connectivity for real-time data transmission to cloud-based servers. This ensures uninterrupted data collection and minimizes dependence on GSM networks.

ii. **Machine Learning Prediction:** A machine learning model will be trained on historical energy consumption data to predict usage trends, detect anomalies, and forecast peak demands. This will help users optimize their energy consumption and enable dynamic tariff adjustments by utility providers.

iii. **Automated Billing & Theft Detection:** The system will integrate a prepaid and postpaid billing mechanism with automated invoice generation. A tamper-detection module using AI .

vi. **User Dashboard & Alerts:** A mobile application and web dashboard will provide real-time consumption tracking, bill estimates, and personalized recommendations for energy savings. Users will receive alerts for unusual consumption patterns, payment due dates, and power theft notifications.

vii **Sustainability & Scalability:** The system will support integration with renewable energy sources, such as solar panels, allowing users to monitor and optimize their hybrid energy usage. The modular design will enable scalability for different consumer segments, including residential, commercial, and industrial users.

CHAPTER 3

SYSTEM REQUIREMENTS

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 Functional Requirements

The functional requirements define the specific operations and capabilities that the Smart Energy Management and Billing System must fulfill. These requirements ensure that the system effectively monitors, analyzes, and automates energy consumption and billing processes.

i. **Real-Time Energy Monitoring –** The system must continuously track and display electricity consumption data using IoT-enabled smart meters.

ii. **Automatic Meter Reading (AMR) –** The system should automatically collect electricity usage data and send it to the central database without manual intervention.

iii. **IoT-Based Communication –** The system must enable seamless data transmission between smart meters, cloud storage, and the user dashboard via Wi-Fi or other wireless protocols.

iv. **Machine Learning for Bill Prediction –** The system should analyze historical consumption patterns using a Linear Regression model to predict future electricity bills.

v. **Tamper and Theft Detection –** The system must detect any unauthorized meter tampering or abnormal energy usage patterns and generate alerts.

vi. **User Dashboard (Web & Mobile App) –** Users should have access to a user-friendly interface to monitor energy usage, view billing history, and receive insights.

vii. **Billing and Payment System –** The system should generate electricity bills automatically and support multiple online payment methods.

viii. **Alert and Notification System –** Real-time alerts must be sent to users regarding high consumption, bill due dates, and unusual activities.

ix. **Security and Data Protection –** The system should ensure data encryption and secure authentication to prevent unauthorized access.

x. **Support for Renewable Energy Sources –** The system should accommodate the integration of solar panels or other renewable energy sources for efficient energy management.

3.2 Non Functional Requirements

Non-functional requirements define the quality attributes and constraints that ensure the **Smart Energy Management and Billing System** operates efficiently, securely, and reliably.

i. **Scalability** – The system should support multiple users and devices, allowing easy expansion as the number of smart meters and consumers increases.

ii. **Performance** – The system should process and transmit energy consumption data in real- time with minimal latency.

iii. **Reliability** – The system must ensure consistent operation with minimal downtime, providing accurate billing and monitoring services.

iv. **Security** – All data transmissions and user interactions should be encrypted and protected from unauthorized access and cyber threats.

v. **Usability** – The web and mobile dashboard should have an intuitive, user-friendly interface to make energy monitoring and bill payment seamless for users.

vi. **Maintainability** – The system should be modular and easy to update or modify to accommodate future improvements and technology advancements.

vii. **Interoperability** – The system should be compatible with different energy meters, IoT devices, and cloud platforms for easy integration.

viii. **Data Accuracy** – The system should ensure that sensor readings and billing calculations are precise and free from errors.

ix. **Efficiency** – The energy monitoring and billing process should consume minimal computational and network resources to optimize performance.

x. **Compliance** – The system should adhere to industry standards and regulations related to energy metering, billing, and data security.

3.3 System requirements

Requirement specification, also known as documentation, is a process of jotting down all the system and user requirements in the form of a document. These requirements must be clear, complete, comprehensive, and consistent. A requirements specification is consistent if no subset of requirements within the specification conflict with each other. If there are inconsistencies in the requirements specification, it is likely that unless detected and corrected, the design of the product will itself contain inconsistencies software does and how we expect it to perform.

3.3.1 Hardware Requirements are:

1. **Arduino microcontroller:**

i. Arduino boards are relatively inexpensive compared to other microcontroller platforms

ii. The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. And provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins. These pins are on the top of the board, several plug-in application shields are also commercially available.

2. **LDR** : The general purpose photoconductive cell is also known as LDR – light dependent resistor. It is a type of semiconductor and its conductivity changes with proportional change in the intensity of light.

   Construction of a Light Dependent Resistor: there two common types of materials used to manufacture the photoconductive cells. They are Cadmium Sulphide (CdS) and Cadmium Selenide (CdSe).

3. **Wifi:** wireless internet access can be added to any microcontroller-based design with simple connectivity through UART interface. can be used pc Wi-Fi or Wi-Fi module to the microcontroller, Wi-Fi model -802.11 b/g/n protocol

4. **Energy meter** : The conventional mechanical energy meter is based on the phenomenon of "Magnetic Induction". It has a rotating aluminium Wheel called Freewheel and many toothed wheels. Based on the flow of current, the Freewheel rotates which makes rotation of other wheels. This will be converted into corresponding measurements in the display section.

5. **Load:** load anything which works on AC supply, like bulb, fan., etc.

6. **Relay:** When a current is passed through the wire which is wound in form of a coil an electro-magnetic field is developed this acts as a temporary magnet. This phenomenon is used in working of relay.

7. **LCD:** To display the units and cost.

3.3.2 SOFTWARE REQUIREMENTS:

A software requirement specification (SRS) is an in-depth document that describes what the software does and how we expect it to perform. An SRS includes information about all the functional and non-functional requirements for a given piece of software.

Software Requirements are:

i. Arduino IDE
ii. Embedded C
iii. Blynk app

I. **Arduino IDE :** Arduino IDE is platform independent and open source, which we can modify the hardware or software, Arduino microcontroller will get all the supported libraries in Arduino IDE, other we won't get it.

a. The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

b. The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it.

II. **Embedded C** : This is programming technology we have to use and implement the functionality, normally any microcontrollers supports embedded c programming to control all the peripherals.

III. **Android app :** Blynk is a toolset for all makers, badass inventors, designers, teachers, nerds and geeks who would love to use their smartphones to control electronics like Arduino, RaspberryPi and similar ones. We've done all the hard work of establishing internet connection, building an app and writing hardware code.

a. With Blynk, you simply snap together an amazing interface from various widgets we provide, upload the example code to your hardware and enjoy seeing first results in under 5 minutes! It works perfectly for newbie makers and saves tons of time for evil geniuses.

b. Blynk will work with all popular boards and shields. We wanted to give you full freedom when deciding how to plug Blynk into your existing or new project.You will also enjoy the convenience of Blynk Cloud. Which is, by the way is free and open-source.

c. Imagine a prototyping board on your smartphone where you drag and drop buttons, sliders, displays, graphs and other functional widgets. And in a matter of minutes these widgets can control Arduino and get data from it.

d. Blynk is not an app that works only with a particular shield. Instead, it's been designed to support the boards and shields you are already using. And it works on iOs and Android.

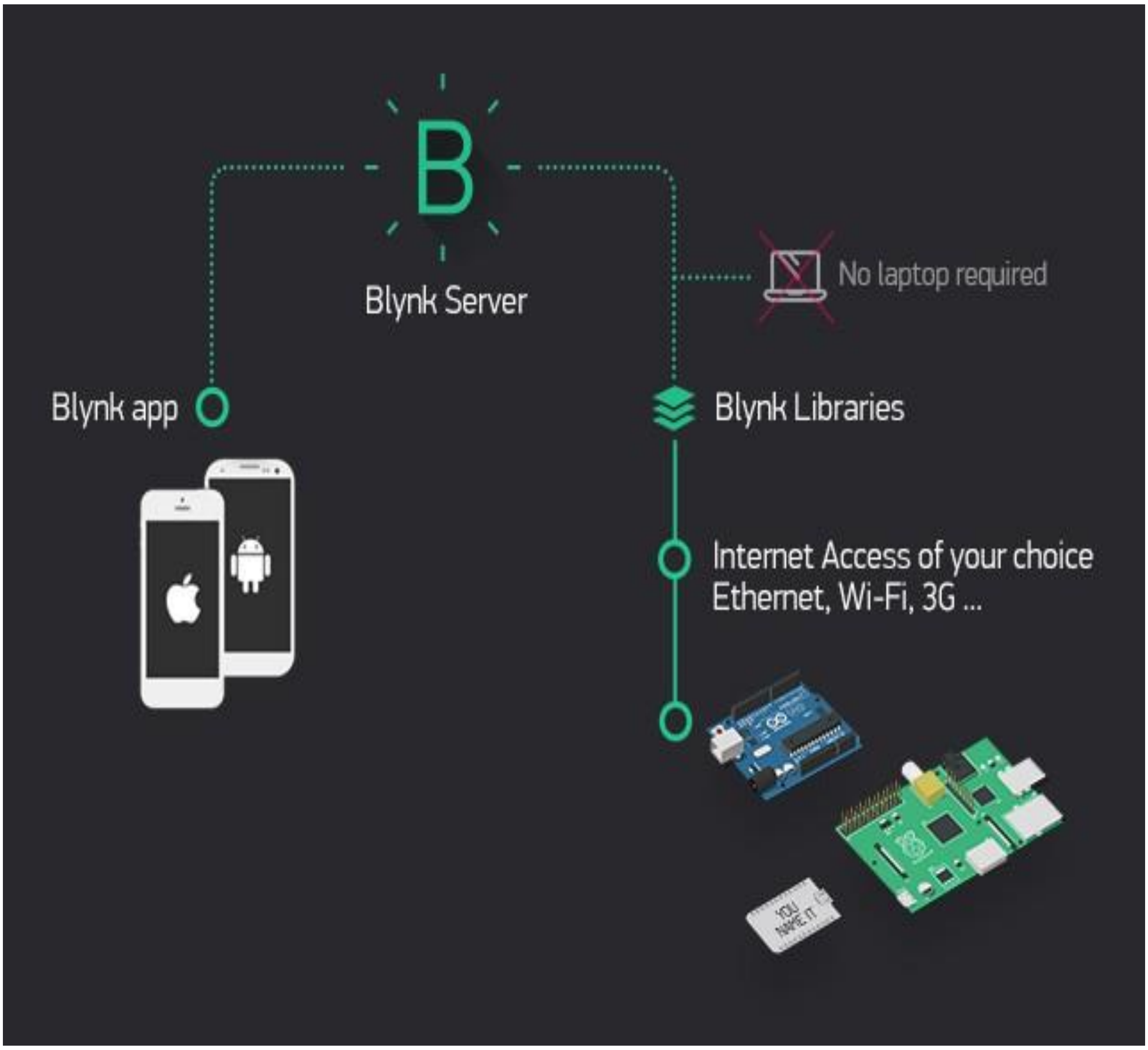


Fig 1.2 Blynk Working

3.3.3 PIN CONFIGURATION

Fig 1.3 28-pin MLF Top view

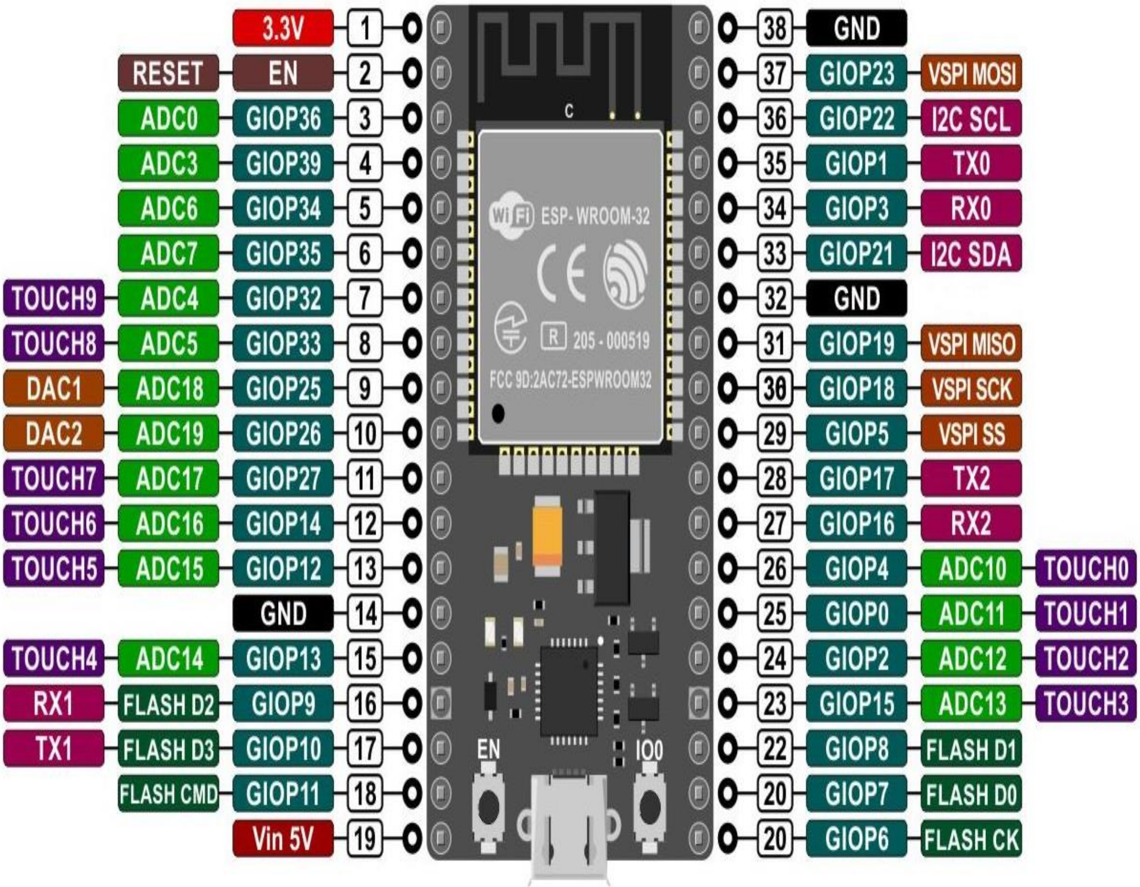


Fig 1.4 ESP32 pin configuration

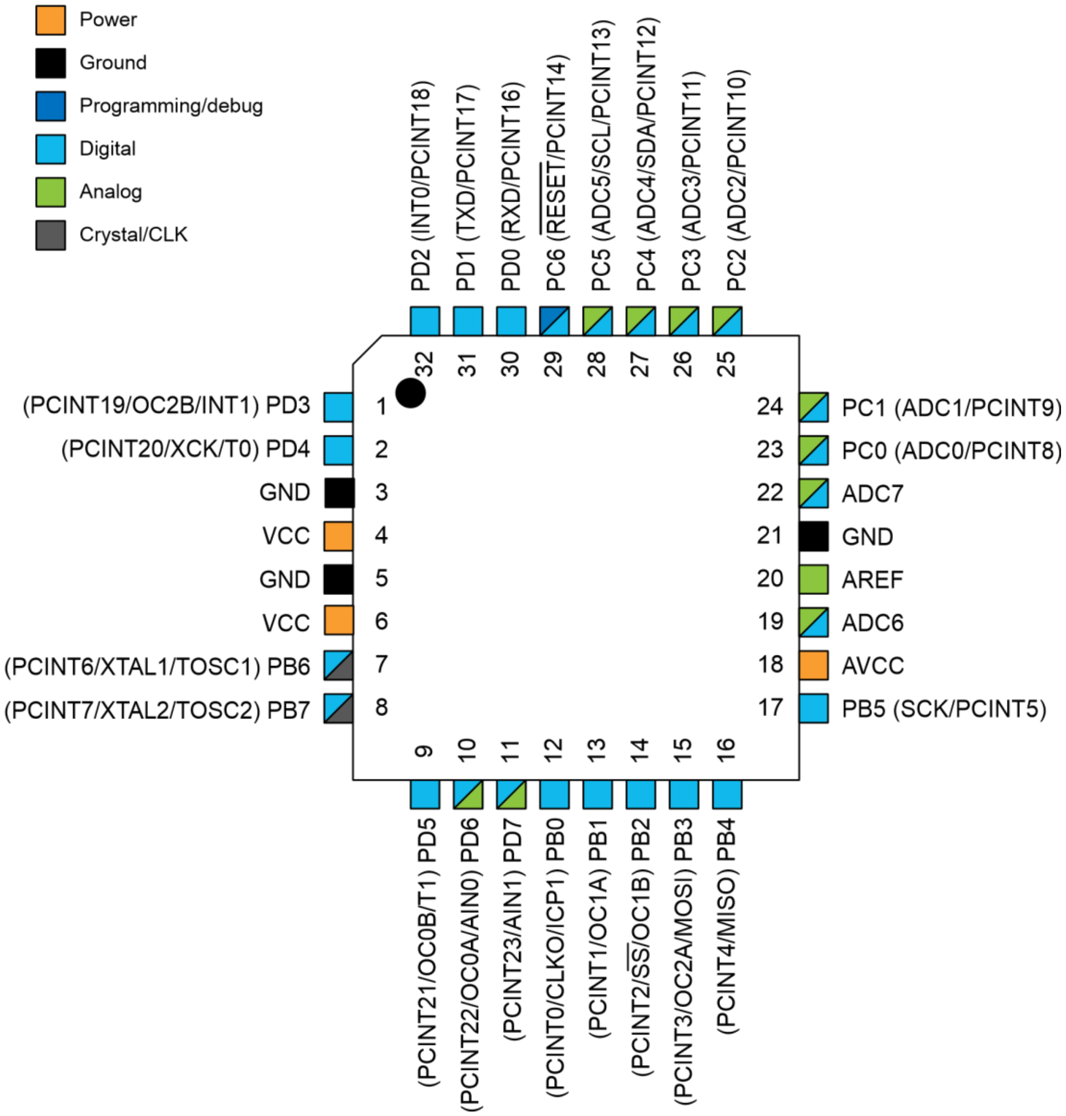


Fig 1.5 32-pin TQFP Top View

3.3.4 Machine Learning Software Requirements

The implementation of Machine Learning (ML) in the Smart Energy Management and Billing System requires specific software tools, frameworks, and libraries for data processing, model training, and prediction. Below are the key ML software requirements:

1. **Programming Language:**
   - **Python –** Preferred for its extensive ML libraries and ease of implementation.

2. **Machine Learning Frameworks and Libraries:**
   - Scikit-Learn – Used for implementing the Linear Regression model for bill prediction.
   - NumPy – Supports numerical computations required for data analysis.
   - Pandas – Used for data manipulation and preprocessing of electricity consumption data.
   - Matplotlib / Seaborn – Used for visualizing consumption patterns and trends.
3. **Data Processing and Storage:**
   - SQLite / MySQL / Firebase – For storing historical energy consumption data and predictions.
   - CSV / JSON – Data formats used for importing and exporting dataset files.
4. **Cloud & IoT Integration:**
   - Google Colab / Jupyter Notebook – For developing and testing ML models.
   - AWS / Google Cloud / Microsoft Azure – Optional cloud services for hosting and deploying the ML model.
5. **Development Environment:**
   - Anaconda / Jupyter Notebook – Preferred for an interactive ML development environment.
   - PyCharm / VS Code – Python IDEs used for writing and debugging code.
6. **Deployment Tools:**
   - Flask / FastAPI / Django – If integrating ML predictions into a web-based or mobile dashboard.
   - TensorFlow Lite (Optional) – If deploying ML on edge devices for lightweight prediction models

CHAPTER 4

SYSTEM DESIGN / METHODOLOGY

CHAPTER 4

METHODOLOGY

4.1 Architecture

4.1.1 Design

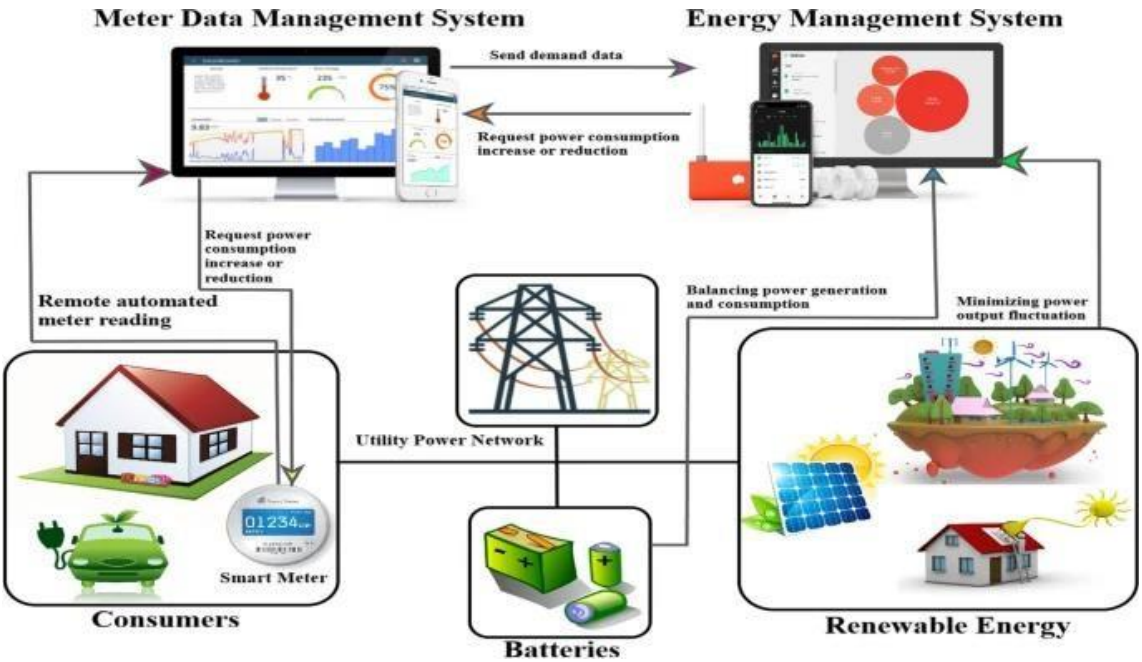


Figure 4.1 – Architecture of the Smart Meter Management

4.1.2 Modules Overview

1. **User Side:** This section describes the front-end modules of the electricity billing system, focusing on the user interface and functionality accessible to endusers. Let's examine each module in detail:

Landing Page: This is the initial page users encounter when visiting the platform. Its primary purpose is to introduce the system, highlighting its key features and benefits. It serves as a welcoming point of entry, providing a general overview of the platform's functionality and encouraging users to explore further. It might include marketing information, testimonials, or a brief explanation of how the system works. Crucially, it provides clear pathways to the Login/Register page for accessing the core system.

Login/Register Page: This module handles user authentication.

It provides two essential functions:

Login: Allows existing users to access their accounts by entering their credentials (username and password). This process verifies the user's identity before granting access to their personalized data and functionalities.

Register: Enables new users to create accounts on the platform. This involves collecting necessary information (e.g., name, address, meter number) and setting up login credentials. This page is critical for security, ensuring that only authorized users can access the system.

Dashboard: After successful login, users are directed to their personalized dashboard. This acts as the central hub for user interaction.

User Details: Basic information about the user, such as name, address, and meter number.

Bill History: A record of past bills, both paid and pending. This allows users to track their electricity expenses over time.

Pending Bills: Information about current outstanding bills, including the amount due and payment deadlines. This provides a clear view of the user's current financial obligations to the electricity provider.

Predict Bill Page: This module enables users to utilize the bill prediction feature. It provides a form where users can enter their latest meter reading. Upon submission, this reading is sent to the system's back-end (likely involving the machine learning model) for processing. The predicted bill amount is then displayed to the user, allowing them to anticipate their upcoming expenses. This page is crucial for empowering users to manage their electricity consumption and budget effectively.

2. **Admin Side:** This section describes the administrative modules of the electricity billing system, focusing on the functionalities available to administrators for managing the system and user data.

Here's a breakdown:

Add Meters: This module allows administrators to add new electricity meters to the system. This typically involves entering details such as the meter's unique identifier, location, and potentially other technical specifications. This functionality is essential for onboarding new customers and expanding the service area. It also likely includes assigning the newly added meter to a specific user.

View Meters: This module provides administrators with a comprehensive view of all existing meters in the system. It allows them to check the status of each meter, its location, the user it's assigned to, and potentially historical consumption data. This centralized view is crucial for monitoring the electricity network, identifying potential issues, and managing meter assignments.

Pay Electricity Bill: This module allows administrators to manually record payments made by users. An administrator would typically enter the meter number associated with the payment and record the payment amount. This action would then mark the corresponding bill as "paid" in the system, updating the user's account balance and billing history. This functionality is important for handling payments received through various channels (e.g., offline payments, bank transfers) that may not be automatically integrated with the system.

View Bills: This module provides administrators with access to all billing information, including both paid and pending bills. Administrators can view bill details, payment history, and outstanding balances for all users. This module is essential for managing accounts, generating reports, and handling billing inquiries.

Total Power Consumption: This module provides administrators with tools to monitor the overall electricity consumption across the network. This could include aggregated data on total consumption, consumption trends over time, and potentially breakdowns by region user group. This information is invaluable for capacity planning, identifying peak demand periods, and optimizing electricity distribution. It may also be used for detecting anomalies or potential energy theft.

4.1.3 Machine Learning Methodology for the Electricity Bill

The methodology for the Electricity Bill Prediction System follows a structured approach, ensuring accuracy, efficiency, and user-friendliness. Initially, data is collected from historical electricity usage records, including past meter readings and bill amounts. This data undergoes preprocessing, where missing values and outliers are handled, and essential features such as seasonal trends and usage patterns are extracted. The dataset is then split into training and testing sets to evaluate the model's accuracy.

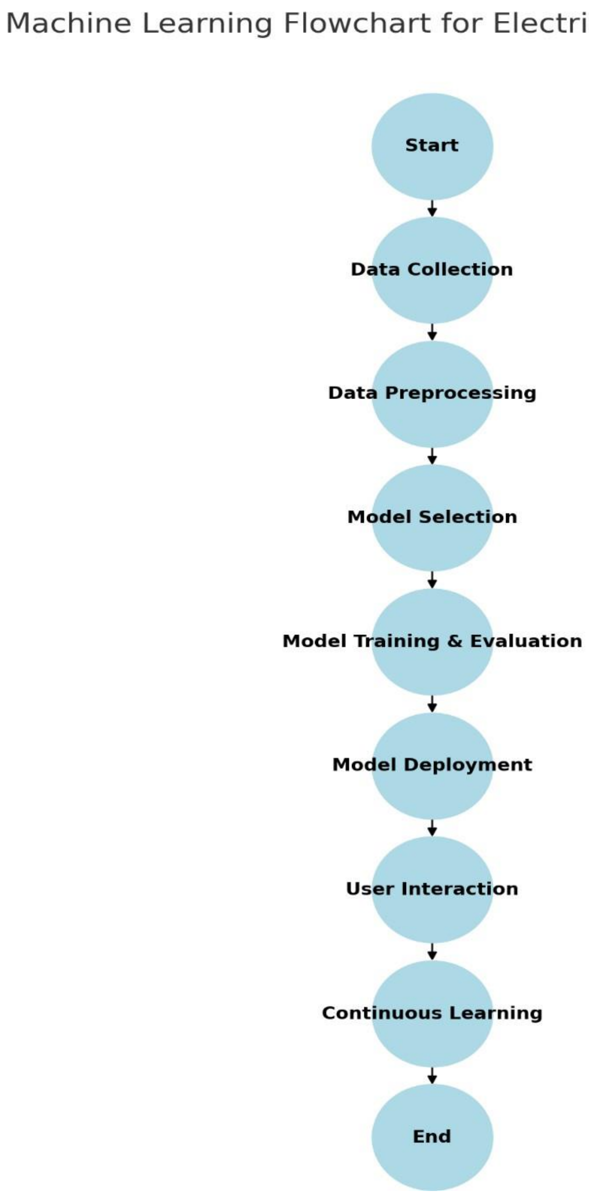
For machine learning model development, regression techniques such as Linear Regression and Decision Tree Regression are employed to predict electricity bills based on historical consumption trends. The model is trained using Scikit-learn, with hyperparameter tuning to optimize performance. Evaluation metrics like Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are used to assess prediction accuracy. Once trained, the model is serialized in a Pickle (.pkl) format for seamless integration into the web application.

The system's web application is developed using HTML, Tailwind CSS, and CSS for a responsive and visually appealing user interface. Flask is implemented as the backend framework to handle user inputs, process machine learning predictions, and manage database interactions. A relational database, such as MySQL or SQLite, is used to store user information, meter readings, billing history, and payment records.

The workflow of the system includes user registration and authentication, allowing secure access to the platform. Users can enter their current meter readings, and the system predicts their upcoming electricity bill based on past consumption patterns. The platform also provides bill management features, enabling users to view their pending and paid bills, while administrators oversee meter management, track power consumption, and process payments.

For deployment, the system is first tested locally before being hosted on cloud platforms for broader accessibility. Performance testing, including unit testing, integration testing, and user acceptance testing, ensures reliability. Security measures such as authentication protocols and database encryption safeguard user data.

Overall, this methodology ensures that the Electricity Bill Prediction System is accurate, efficient, and user-friendly, providing valuable insights into electricity consumption and enabling better financial planning for users.



4.1.4 Existing System

Efficient use of energy becomes more crucial when increase in the cost of energy is observed. Since energy management is required to define the amount of consumed energy in a specific period of time, utilization of Energy Meters is essential. It is possible to measure the consumed energy by using a simple energy meter. But sometimes the limited functionality of these meters restricts their area of application; especially in inaccessible positions or in the situations where visibility of the meter is poor, it is not possible to use such an appliance. Energy theft is very common problem in countries like India where consumers of energy are increasing consistently as the population increases. Utilities in electricity system are destroying the amounts of revenue each year due to energy theft. The newly designed AMR used for energy measurements reveal the concept and working of new automated power metering system but this increased the Electricity theft forms administrative losses because of not regular interval checkout at the consumer's residence. It is quite impossible to check and solve out theft by going every customer's door to door.

4.1.5 Proposed System:

i. **Real-Time Sensor Data Processing :** Implement Arduino board to read sensor data in real-time .Utilize advanced algorithms for efficient processing of sensor information.
ii. **Smart Meters:** Deploy IoT-enabled smart meters at user premises to measure real-time electricity consumption .These meters should be capable of collecting and transmitting data over the internet.
iii. **User Interface:** Develop a user-friendly web or mobile application for consumers to monitor their power usage. Display real-time usage, historical data, and billing information. Provide alerts for abnormal consumption or potential issues.
iv. **Remote Control:** Allow users to remotely control certain devices or appliances to optimize energy consumption. Implement features like scheduling, turning off devices, or setting power limits.
v. **Automated Billing System:** Generate accurate bills based on real-time usage data. Send bills to users electronically through email or the application. Provide detailed breakdowns of the consumption patterns and charges.
vi. **Notifications and Alerts:** Send notifications to users for high consumption, power outages, or abnormal patterns. Provide proactive alerts for upcoming maintenance or system updates.
vii. **Security Measures:** Implement robust security protocols to safeguard user data and prevent unauthorized access. Use encryption techniques for data transmission and storage.
viii. **Energy Efficiency Recommendations:** Provide personalized recommendations for energy efficiency based on user usage patterns. Encourage users to adopt energy-saving practices.

.

4.1.6 Sequence Diagram of the Application:

The sequence diagram for the Smart Energy Management & Billing System using IoT and ML Prediction begins with the user logging into the application. Once authenticated, the user accesses the main dashboard, where they can monitor real-time energy consumption, view billing details, or analyze usage trends. The IoT-enabled smart meter continuously collects energy consumption data and transmits it to the cloud server via MQTT or HTTP protocols. If the user selects the billing section, the system retrieves the latest consumption data and calculates the bill using predefined tariffs and ML-based predictions for future consumption. Users can also receive alerts for unusual energy usage or due bill payments. The system ensures real-time updates and communication through cloud services, enabling seamless monitoring and smart billing automation.

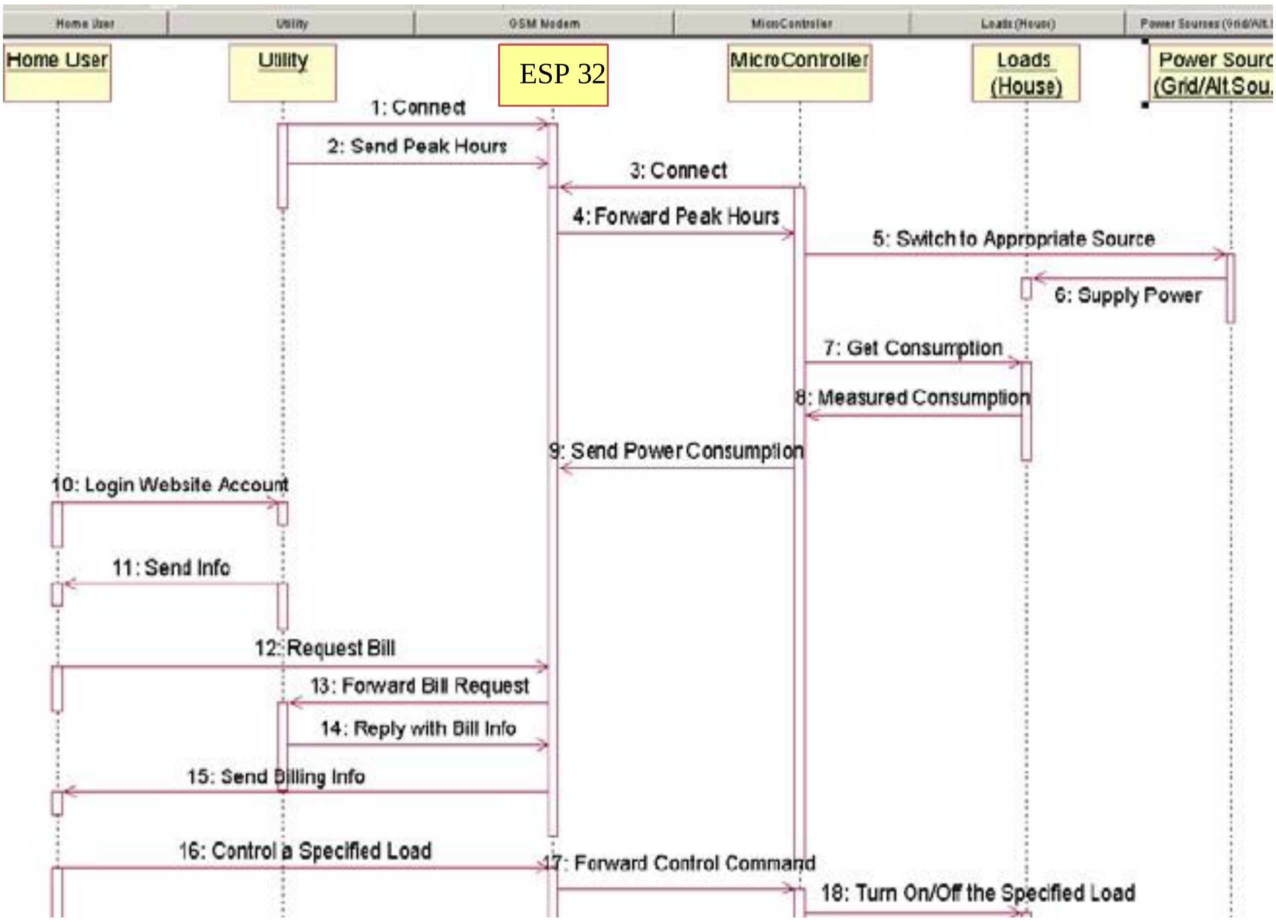


Figure 4.3 – Sequence diagram of the Smart Meter Management

Smart energy management and billing using IoT and machine learning prediction is an advanced approach to optimizing electricity consumption and automating the billing process. With the increasing demand for electricity and the need for sustainable energy solutions.

The cloud server processes the collected data, retrieves past energy usage records from a database, and forwards the information to a machine learning model for analysis. The model predicts future electricity bills based on consumption trends, allowing users to plan their expenses and avoid unexpected charges.

The system also facilitates smart load management by enabling the automated switching of power sources between grid electricity and renewable energy alternatives such as solar or battery storage. Users can monitor their real-time energy usage and billing estimates through a web or mobile application, ensuring better control over consumption patterns. By leveraging predictive analytics, the system not only provides accurate billing insights but also promotes energy conservation by encouraging users to optimize their electricity usage. Additionally, automated billing eliminates manual errors and streamlines the process for both consumers and utility providers.

One of the key advantages of this technology is its ability to manage peak load demands efficiently, reducing strain on the power grid and minimizing electricity wastage. By integrating AI-driven predictive analysis, the system enhances the reliability and sustainability of energy distribution while encouraging the adoption of smart energy solutions. Overall, smart energy management and billing with IoT and machine learning prediction represents a significant step toward a more efficient and environmentally friendly approach to electricity consumption, benefiting both users and energy providers.

4.2 Methodology

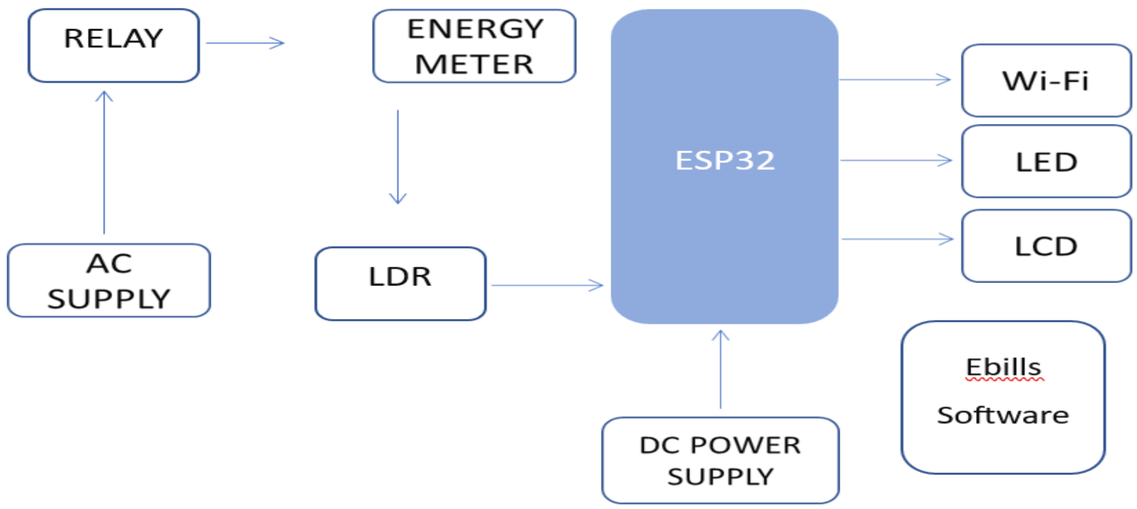


Figure 4.4 Methodology of Smart Meter

The Smart Energy Management and Billing System integrates IoT and Machine Learning (ML) to optimize power consumption, automate billing, and enhance energy efficiency. The system utilizes smart energy meters, ESP32 microcontrollers, IoT sensors, and cloud-based platforms to enable real-time monitoring and control of electricity usage. The energy meter continuously tracks voltage, current, and power consumption, transmitting this data to the ESP32, which processes it and forwards it to a cloud-based E-bills software via Wi-Fi connectivity. Users securely register and log in to access their energy data, while administrators manage meter assignments and payment statuses in real time.

The ML model, trained on historical consumption patterns, tariff structures, and real-time meter readings, predicts future electricity bills, helping users anticipate and optimize their energy expenses. Additionally, ML-driven predictive analytics identify abnormal power consumption, detect potential faults, and suggest energy-saving measures, such as load balancing strategies and peak usage alerts. IoT-enabled LDR sensors and relay modules further enhance efficiency by automating lighting control based on natural daylight and enabling remote switching of electrical appliances to prevent energy wastage.

The system ensures transparent and secure billing through blockchain-based smart contracts, eliminating fraud and reducing disputes. Users can access a digital payment interface and track their payment history, while IoT dashboards provide real-time consumption insights. Furthermore, ML-based optimization algorithms allow seamless integration with renewable energy sources, such as solar power, enabling intelligent switching between grid and solar electricity to minimize costs and improve sustainability.

By combining IoT for real-time data collection with ML-driven analytics and automation, this system enhances billing accuracy, optimizes power distribution, and promotes sustainable energy management. It provides a scalable and intelligent solution for residential, commercial, and industrial applications, transforming energy usage into a more efficient, secure, and eco- friendly process.

4.3 Major Algorithms

4.3.1 Linear Regression Algorithm for Electricity Bill Prediction

Step 1: Data Collection

- Gather historical energy consumption data and corresponding billing records.

Step 2: Data Preprocessing

- Handle missing values (e.g., mean imputation or interpolation).
- Normalize/standardize numerical data for better model performance.
- Encode categorical variables (e.g., tariff type) if necessary.

Step 3: Splitting Data

- Divide dataset into Training (80%), Validation (10%), and Testing (10%) sets.

Step 4: Model Training

- Import necessary libraries:
  from sklearn.linear_model

  import LinearRegression

  from sklearn.model_selection

  import train_test_split

  import pandas as pd

1. **Load dataset and select features:**
   data = pd.read_csv('energy_data.csv')

   X = data[['Previous_Consumption']] # Independent variable y = data['Current_Bill'] #

   Dependent variable
2. **Split dataset into training and testing sets:**
   X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
3. **Train the Linear Regression model:**
   model = LinearRegression() model.fit(X_train, y_train)

Step 5: Making Predictions

- Use the trained model to predict electricity bills: y_pred = model.predict(X_test)

Step 6: Model Evaluation

- Evaluate the model using performance metrics:

from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

mae = mean_absolute_error(y_test, y_pred) mse = mean_squared_error(y_test, y_pred) r2 = r2_score(y_test, y_pred)

print(f"MAE: {mae}, MSE: {mse}, R² Score: {r2}")

Step 7: Model Deployment

- Save the trained model using Pickle for future predictions: import pickle

with open('model.pkl', 'wb') as file: pickle.dump(model, file)

CHAPTER 5

IMPLEMENTATION

CHAPTER 5

IMPLEMENTATION

5.1 Detailed Explanation of Technologies and Advantages

5.1.1 Internet of Things (IoT)

The backbone of the system is the Internet of Things (IoT), which involves connecting physical devices to the internet for remote monitoring and control.

IoT in Smart Energy Management

Smart Meters: Smart meters are equipped with sensors that can measure power consumption in real-time and send data wirelessly to a central system for analysis and billing.

IoT Gateways: These serve as intermediaries between local devices (like smart meters) and the cloud, transmitting data securely over the internet.

Real-time Monitoring: Using IoT, data can be collected continuously and analyzed to give insights into energy consumption patterns and billing.

Advantages of IoT in this project

Remote Monitoring: Energy consumption can be monitored remotely, eliminating the need for manual meter reading.

Automated Control: Allows automatic control over appliances based on usage patterns, improving energy efficiency.

Data Transparency: Consumers can access real-time data regarding their energy usage, enhancing transparency and trust.

5.1.2 ESP32 Wi-Fi Module

The **ESP32** is a powerful microcontroller with **built-in Wi-Fi and Bluetooth capabilities**, ideal for wireless communication between smart meters and the cloud.

Advantages of ESP32

Low Power Consumption: Its low power features are perfect for continuously transmitting data from the energy meters without draining battery life.

Dual-core Processor: This allows ESP32 to perform complex tasks like running a web server and processing sensor data simultaneously.

Wi-Fi Connectivity: Direct connection to the internet enables remote control and monitoring of devices.

5.1.3 Machine Learning (ML) - Linear Regression

Machine Learning (ML) is implemented using **Scikit-learn** and is used to predict energy consumption patterns based on historical data. The **Linear Regression** model can help forecast the amount of energy usage based on various parameters like voltage, current, and past consumption trends.

Advantages of ML in Energy Management

Energy Prediction: Helps predict future energy consumption, allowing consumers to optimize their usage and reduce unnecessary expenditure.

Dynamic Pricing: The ML model can be used to suggest dynamic pricing based on predicted usage during peak or off-peak hours.

Pattern Recognition: Identifies unusual usage patterns that might indicate faulty appliances or power theft.

Implementation of Linear Regression

Collect energy consumption data from smart meters and preprocess it. Use Scikit-learn to train a **linear regression model** on past data.

The model can predict future energy usage by analyzing patterns in voltage, current, and other related variables.

Predictions can help users make decisions regarding energy conservation and efficiency.

Code Example for Energy Prediction

```
from sklearn.linear_model import LinearRegression import pandas as pd
# Example dataset

data = pd.read_csv('energy_data.csv')

X = data[['Voltage', 'Current']]  # Features y = data['PowerUsage']  # Target


# Train a linear regression model
 model = LinearRegression() model.fit(X, y)


#  Predict future energy usage
predicted_usage = model.predict([[230, 5]]) # Example input for 230V, 5A print(f"Predicted Energy
    Usage: {predicted_usage[0]} kWh")
```

5.1.4 Cloud Computing and Data Analytics

Cloud platforms are used to store and process energy usage data collected from smart meters. They allow for centralized data management, analysis, and visualization.

Advantages of Cloud Computing

Scalability: Cloud platforms can easily scale to handle large amounts of data from millions of smart meters.

Real-time Processing: Data from smart meters is processed in real-time to provide immediate insights into energy usage.

Data Security: Cloud platforms often come with built-in security features to protect sensitive consumer data.

Implementation

The energy consumption data is sent to the cloud via IoT-enabled devices like ESP32.

Data analytics tools in the cloud help in real-time analysis of energy usage patterns, providing actionable insights for consumers and utilities.

The cloud system also handles **billing** and generates accurate, transparent energy bills.

5.2 Blynk App for Real-time Monitoring

Blynk is a versatile mobile app development platform for IoT applications. It allows users to monitor and control connected devices via smartphones, and is especially suited for smart home systems.

Advantages of Using Blynk

- **Quick Setup**: No need for complex app development; you can use drag-and-drop widgets to build dashboards.
- **Mobile Platform**: It works on both **iOS** and **Android**, making it widely accessible.
- **Real-time Monitoring**: Blynk allows users to view energy consumption in real-time, as well as track historical data.
- **Customization**: Users can customize their dashboard to show energy consumption metrics, billing details, and alerts.
- **Implementation**
- Users can install the Blynk app on their smartphones.
- The app connects to the smart meter via the cloud to fetch and display data.

5.3 Energy Meter and Sensors

The **energy meter** is an essential hardware component that measures voltage, current, and power consumption.

Advantages of Digital Energy Meters

- **Accuracy**: Digital energy meters are more accurate than analog meters and do not have moving parts, reducing mechanical failures.
- **Remote Monitoring**: The energy meter can send readings directly to the cloud for remote monitoring and billing.
- **Tamper Protection**: The energy meter is equipped with sensors like **LDR** (Light Dependent Resistor) to detect any tampering attempts.
- **Implementation**
- **Voltage Sensor** (ZMPT101B) and **Current Sensor** (ACS712) are used to measure voltage and current, respectively.
- **Analog-to-Digital Conversion** is done via the ESP32, which transmits the data to the cloud for processing.
- **Tamper Detection**: The system can use **LDR sensors** to detect any physical interference with the meter and trigger alerts.

5.3.1 Components Required

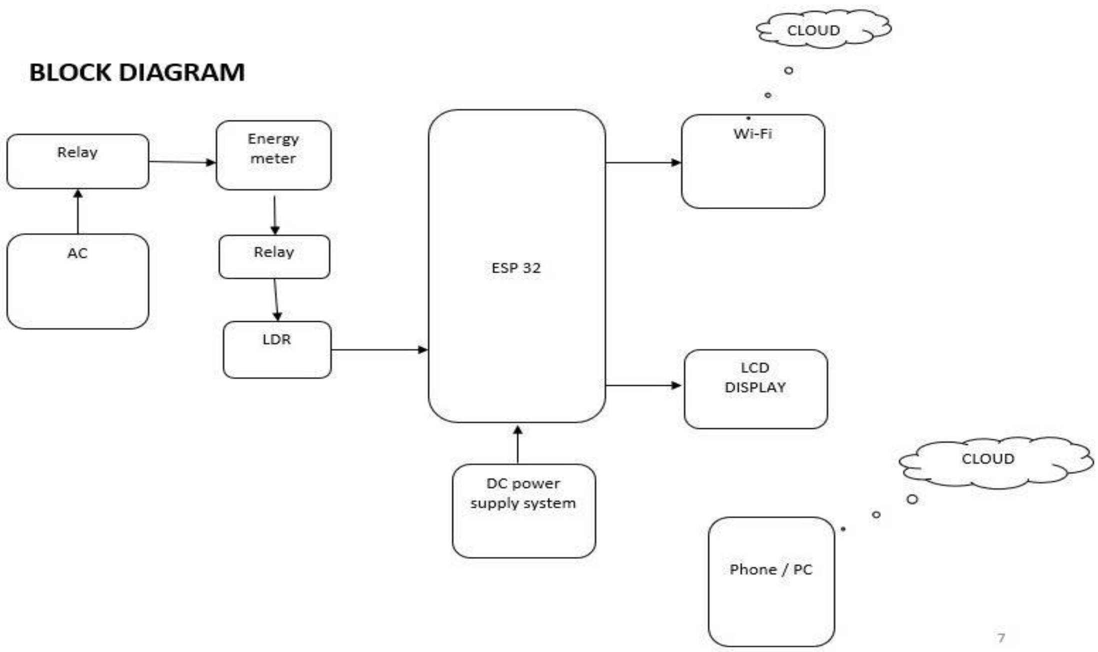


Figure 5.1 -Smart Energy Meter Block Diagram

This block diagram represents a Smart Energy Meter system using an ESP32 microcontroller. The energy meter monitors power usage, while relays and an LDR (Light Dependent Resistor) help detect electricity consumption. The ESP32 connects to Wi-Fi, sending data to the cloud and displaying readings on an LCD or a phone/PC. A DC power supply ensures stable operation.

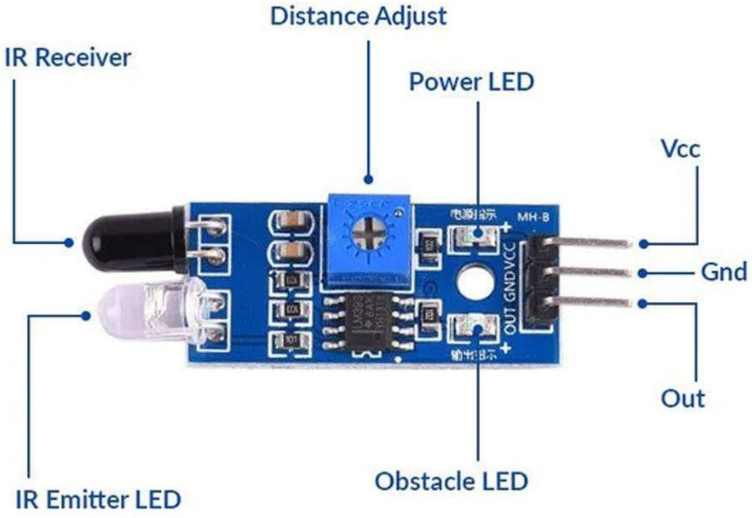


Figure 5.2-Photoresistor Sensor Module: A Light-Sensitive Solution

This module features an LDR (light-dependent resistor) to detect varying light levels. The LM393 comparator ensures accurate digital output, while the adjustable potentiometer allows for customizable sensitivity. With both digital and analog outputs, it's versatile for various applications. Commonly used in light-sensitive circuits, alarms, and automated control systems.

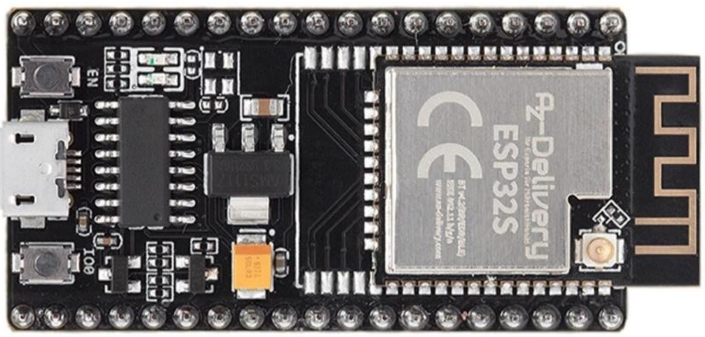


Figure 5.3-ESP32-WROOM-32: Powerful and Versatile Microcontroller

This module features the ESP32 microcontroller, boasting Wi-Fi and Bluetooth connectivity for a wide range of applications. With a dual-core processor and ample memory, it's well-suited for complex tasks. Numerous GPIO pins offer flexibility for connecting sensors and actuators. Ideal for IoT projects, home automation, and embedded systems.

Figure 5.4 - Relay Module: Electrically Operated Switch for High-Power Control

A relay is an electromechanical switch that controls high-power circuits using a low-power signal. It has coil terminals (pins 1 & 2) for activation, COM (pin 3) as the common contact, NC (pin 4) for normally closed connection, and NO (pin 5) for normally open connection. When the coil is energized, the COM contact shifts between NC and NO, enabling circuit switching.

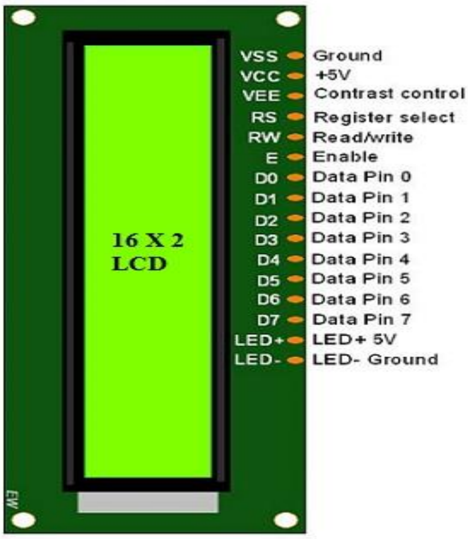


Figure 5.5 - ILI9341: High-Performance TFT LCD Display

The 16x2 LCD module is a widely used display in embedded systems, capable of showing 16 characters per line across 2 lines. It has 16 pins, including power (VCC, GND), contrast control (VEE), control pins (RS, RW, E), and 8 data pins (D0-D7) for communication. The LED+ and LED- pins control the backlight, enhancing visibility.

Figure 5.6 - Buzzer Module: Audio Signaling Device for Alerts and Notifications

A buzzer module is an audio signaling device used for alarms and notifications. It typically has three pins: **VCC** (Power), **GND** (Ground), and **SIG** (Signal) for triggering sound.

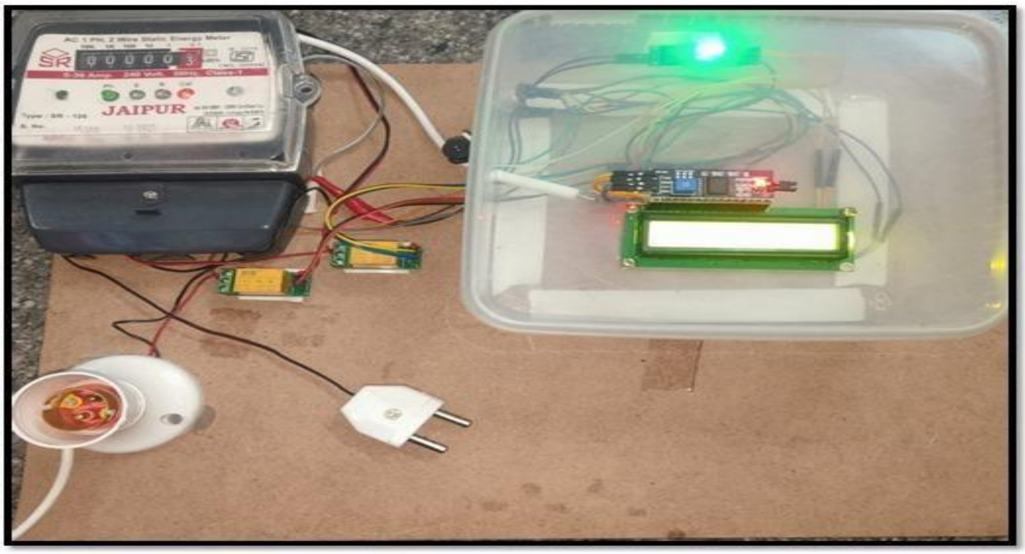


Figure 5.7-Automated Energy Monitoring and Control System

This project combines energy measurement with automated control, likely using a relay controlled by the microcontroller. The ACS712 sensor monitors AC current, while the ESP8266 (NodeMCU) processes data and makes control decisions. The system likely automates energy consumption based on programmed parameters or thresholds. The clear enclosure houses the electronics for protection and safety.

5.4 Tamper Detection and Theft Prevention

Power theft is a significant concern in many regions, especially in emerging economies. **Tamper detection** ensures that any unauthorized attempts to alter or bypass the meter are quickly identified.

Advantages of Tamper Detection

- **Prevents Revenue Loss**: By detecting unauthorized tampering, the system prevents power theft and ensures that utilities get paid for the electricity consumed.
- **Real-time Alerts**: Alerts are sent immediately to the utility company and the user, ensuring quick action.
- **Improved Trust**: Consumers can be assured that the system is fair, as it minimizes the possibility of manual errors or fraudulent activities.
- **Implementation**
- Use of **LDR** sensors to detect changes in lighting conditions around the meter, which may indicate tampering.
- **Relay Module** can be used to disconnect the power supply remotely if tampering is detected. Data from these sensors is sent to the **cloud** for immediate analysis.

5.5 Flask

Flask is a micro web framework in Python that provides a lightweight yet powerful way to deploy machine learning models as web applications. It allows developers to create RESTful APIs that can handle user inputs, process them using a trained model, and return predictions in real time.

When implementing a **Linear Regression** model for bill prediction, Flask can be used to serve the model and accept relevant input parameters, such as energy consumption, service duration, or past billing amounts. The framework supports data preprocessing, model loading, and prediction generation within a structured API endpoint.

Users can interact with the model via HTTP requests, sending input data through a front-end interface or API call, and receive output predictions in JSON format. Flask also enables scalability by allowing integration with databases for storing historical billing data, user authentication for secure access, and logging mechanisms for monitoring performance.Its flexibility makes it an excellent choice for deploying machine learning models in production environments while ensuring smooth interaction between the back end, model, and users.

Advantages of Flask in Python and Web Development

1. **Lightweight**
   Flask is a minimalistic framework, making it easy to set up and use. It doesn't enforce strict rules, allowing developers to structure the app as they see fit. This simplicity is perfect for small to medium-sized projects.
2. **Flexibility**
   Flask offers great flexibility, allowing you to add only the components you need. It supports different extensions, making it easy to integrate additional features like databases, authentication, and REST APIs without unnecessary overhead.
3. **Easy**
   Being a Python-based framework, Flask seamlessly integrates with Python libraries, making it ideal for projects that involve data science, machine learning, or automation.
4. **Scalability**
   While Flask is simple, it can be scaled for larger projects.

5.6 Code Snippet

5.6.1 Embedded C code

C: > Users > TEJAS A M > OneDrive > Desktop > final_code.ino

```
#define BLYNK_TEMPLATE_ID "TMPL37Fypmelc"
#define BLYNK_TEMPLATE_NAME "smart energy meter"
#define BLYNK_AUTH_TOKEN "axkW9BqrsE2I5uHUmJaN3sJ0xZkRjn5P"

// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

char auth[] = BLYNK_AUTH_TOKEN;

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Bharath";
char pass[] = "12345678";

BlynkTimer timer;

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
  Blynk.virtualWrite(V5,cost);
  lcd.setCursor(11,1);
  lcd.print(cost);

delay(500);

//digitalWrite(RELAY, HIGH);
if(unit>=60)
 {
   if(cost>=480)
      {


       lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Bill is Overdue");
        delay(5000); //1000 msec = 1sec
        Blynk.virtualWrite(V2,"Bill is Overdue");
        delay(5000);
        Blynk.virtualWrite(V2,"Power Cut");
        delay(5000);
```

5.6.2 Machine learning code

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import joblib

# Create a dataset
data = {
    'hours_of_use': [2, 5, 8, 10, 15, 20],
    'rate_per_unit': [0.12, 0.15, 0.10, 0.20, 0.18, 0.25],
    'electricity_bill': [0.24, 0.75, 0.80, 2.00, 2.70, 5.00]
}

# Convert the data into a pandas dataframe
df = pd.DataFrame(data)

# Features (independent variables)
X = df[['hours_of_use', 'rate_per_unit']]

# Target (dependent variable)
y = df['electricity_bill']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize the Linear Regression model
model = LinearRegression()

# Train the model
model.fit(X_train, y_train)

# Save the trained model to a file using joblib
joblib.dump(model, 'electricity_bill_model.pkl')

print("Model trained and saved successfully!")
```

Figure 5.8 Code snippet for both Embedded C and ML

CHAPTER 6

TESTING

CHAPTER 6

TESTING

6.1 Testing

Testing is a critical phase in software development that ensures a system functions correctly, meets user requirements, and operates efficiently. It involves evaluating the software against predefined test cases to identify defects, security vulnerabilities, and performance issues. By conducting rigorous testing, developers can minimize errors, enhance system reliability, and improve overall user experience. In the Smart Energy Management and Billing System, testing plays a crucial role in validating real-time energy monitoring, billing accuracy, machine learning predictions, and security features like tamper detection.

Software testing is typically divided into unit testing, integration testing, and system testing. Unit testing verifies individual components, such as IoT-based smart meters and data transmission mechanisms, ensuring they function correctly. Integration testing checks the interaction between modules, such as the connection between the billing system and the machine learning model. System testing evaluates the entire application's functionality, performance, and security, ensuring a smooth and reliable experience for users. Additionally, user acceptance testing (UAT) is conducted to ensure the system meets real-world expectations before deployment.

By following standardized testing principles, the project ensures high software quality, reduced maintenance costs, and improved performance. Testing not only identifies existing issues but also enhances the system's scalability, security, and efficiency, making it more robust and user- friendly. Proper testing ensures that users can reliably track their energy consumption, receive accurate bills, and optimize their power usage, leading to an effective and sustainable energy management solution.

6.1.1 Aim of Testing

The aim of testing in this project is to ensure that the Smart Energy Management and Billing System functions correctly, efficiently, and securely. The primary goals include:

i. **Verifying System Accuracy –** Ensuring smart meters accurately capture energy consumption and generate precise billing.

ii. **Validating Machine Learning Predictions –** Checking if the ML model correctly forecasts future bills using historical data.

iii. **Ensuring Reliability –** Testing the stability and performance of the system under different conditions.

iv. **Detecting and Fixing Errors –** Identifying bugs and fixing system vulnerabilities before deployment.

v. **Enhancing Security** – Ensuring proper authentication, data encryption, and tamper detection.

vi. **Improving User Experience –** Validating that the web and mobile dashboard are user-friendly and efficient.

6.1.2 Testing principles

The following software testing principles are followed to ensure effective testing:

i. **Testing Shows the Presence of Defects** – It helps identify errors but cannot prove a system is entirely bug-free.

ii. **Exhaustive Testing is Impossible** – Not all possible test cases can be covered, so testing focuses on high-risk areas.

iii. **Early Testing Saves Costs** – Detecting issues early reduces time and expenses required for fixes.

iv. **Defect Clustering** – A small portion of the system often contains the most defects; priority is given to high-risk components.

v. **Pesticide Paradox** – Repeating the same tests reduces effectiveness, so test cases are regularly updated.

vi. **Testing is Context-Dependent** – Different testing methods (unit, integration, system) are applied based on the project needs.

vii. **Absence of Errors is a Fallacy** – Even if the system is bug-free, it may fail if it does not meet user requirements.

6.1.3 Test Plan

The test plan outlines the strategy, scope, and objectives for evaluating the Smart Energy Management and Billing System to ensure its accuracy, reliability, and security. The testing process is designed to validate core functionalities, detect potential issues, and ensure seamless system performance. Key areas of focus include:

i. **Functionality Testing –** Ensures that smart meters accurately measure energy consumption, billing calculations are precise, and machine learning predictions align with actual usage trends.

ii. **Performance and Security Testing –** Evaluates the system's response time, scalability under different loads, and security mechanisms to prevent unauthorized access and data breaches.

iii. **User Interface and Experience Testing –** Assesses the responsiveness and usability of the web and mobile dashboards to ensure a smooth and intuitive user experience.

A well-structured testing plan ensures that the Smart Energy Management and Billing System operates efficiently, meets user expectations, and minimizes errors. By performing functional, performance, and security testing, the system is validated for accuracy, robustness, and scalability. This ensures that users can reliably track energy consumption, receive automated billing, and benefit from predictive analytics, ultimately leading to an optimized and sustainable energy management solution.

6.2 Unit Testing

The table 6.1 gives information about the test cases and their results of the Unit testing. Unit testing focuses on verifying the functionality of individual modules of the system**.**

Table 6.1 – Unit testing Test cases

Test Case ID	Module	Test Scenario	Expected Output	Actual Output	Result
UT-01	Smart Meter Sensor	Check if the energy meter reads real-time data	Displays correct readings	Displays correct readings	Pass

UT-02	Data Transmission	Verify if data is sent to the cloud database	Data successfully stored	Data successfully stored	Pass
UT-03	Billing Calculation	Validate bill generation based on usage	Correct bill amount displayed	Correct bill amount displayed	Pass
UT-04	Machine Learning Model	Test bill prediction accuracy using past data	Prediction is close to actual bill	Prediction is close to actual bill	Pass
UT-05	User Dashboard	Verify real-time consumption updates	Dashboard updates instantly	Dashboard updates instantly	Pass

6.3 Integration Testing

Integration testing ensures that different system components interact correctly.

Table 6.2 – Integration testing Test cases

Test Case ID	Modules Integrated	Test Scenario	Expected Output	Actual Output	Result
IT-01	Smart Meter + Database	Verify data transmission to the cloud	Data stored correctly	Data stored correctly	Pass
IT-02	Database + Billing System	Check if usage data converts to a bill	Correct bill generated	Correct bill generated	Pass
IT-03	Billing System + Payment	Ensure successful online payments	Payment processed successfully	Payment processed successfully	Pass
IT-04	User Dashboard + API	Test if users can retrieve past bills	Bills displayed correctly	Bills displayed correctly	Pass
IT-05	ML Model + Billing	Validate predicted vs. actual bills	Prediction is reasonable	Prediction is reasonable	Pass

6.4 System Testing

System testing evaluates the complete system’s functionality.

Table 6.3 – System testing Test cases

Test Case ID	Feature Tested	Test Scenario	Expected Output	Actual Output	Result
ST-01	Real-Time Monitoring	Check if energy usage updates live	Live consumption visible	Live consumption visible	Pass
ST-02	Billing Automation	Verify if the bill generates automatically	Correct amount calculated	Correct amount calculated	Pass
ST-03	Machine Learning Model	Validate predicted bill vs. actual usage	Reasonable bill forecast	Reasonable bill forecast	Pass
ST-04	Tamper Detection	Check if unauthorized access is detected	Security alert triggered	Security alert triggered	Pass
ST-05	Mobile App UI	Ensure mobile dashboard works properly	Smooth and responsive UI	Smooth and responsive UI	Pass
ST-06	Payment Integration	Ensure that users can pay bills via UPI, credit/debit card, or net banking.	Payment is processed successfully.	Payment is processed successfully.	Pass
ST-07	User Notifications	Check if users receive alerts for high usage, billing reminders, and outages.	Notifications are sent via email/SMS/app alerts.	Notifications are sent via email/SMS/app alerts.	Pass
ST-08	Mobile & Web Dashboard	Test the usability and responsiveness of the user interface.	Dashboard loads smoothly with all features accessible.	Dashboard loads smoothly with all features accessible.	Pass
ST-09	Data Storage & Retrieval	Verify if the system correctly stores energy consumption data and retrieves it for reports.	Users can access past energy usage records without errors.	Users can access past energy usage records without errors.	Pass
ST-10	System Load Handling	Test the system’s performance when multiple users access the dashboard	System remains responsive without crashes or slowdowns.	System remains responsive without crashes or slowdowns.	Pass

CHAPTER 7

RESULTS

CHAPTER 7

RESULTS

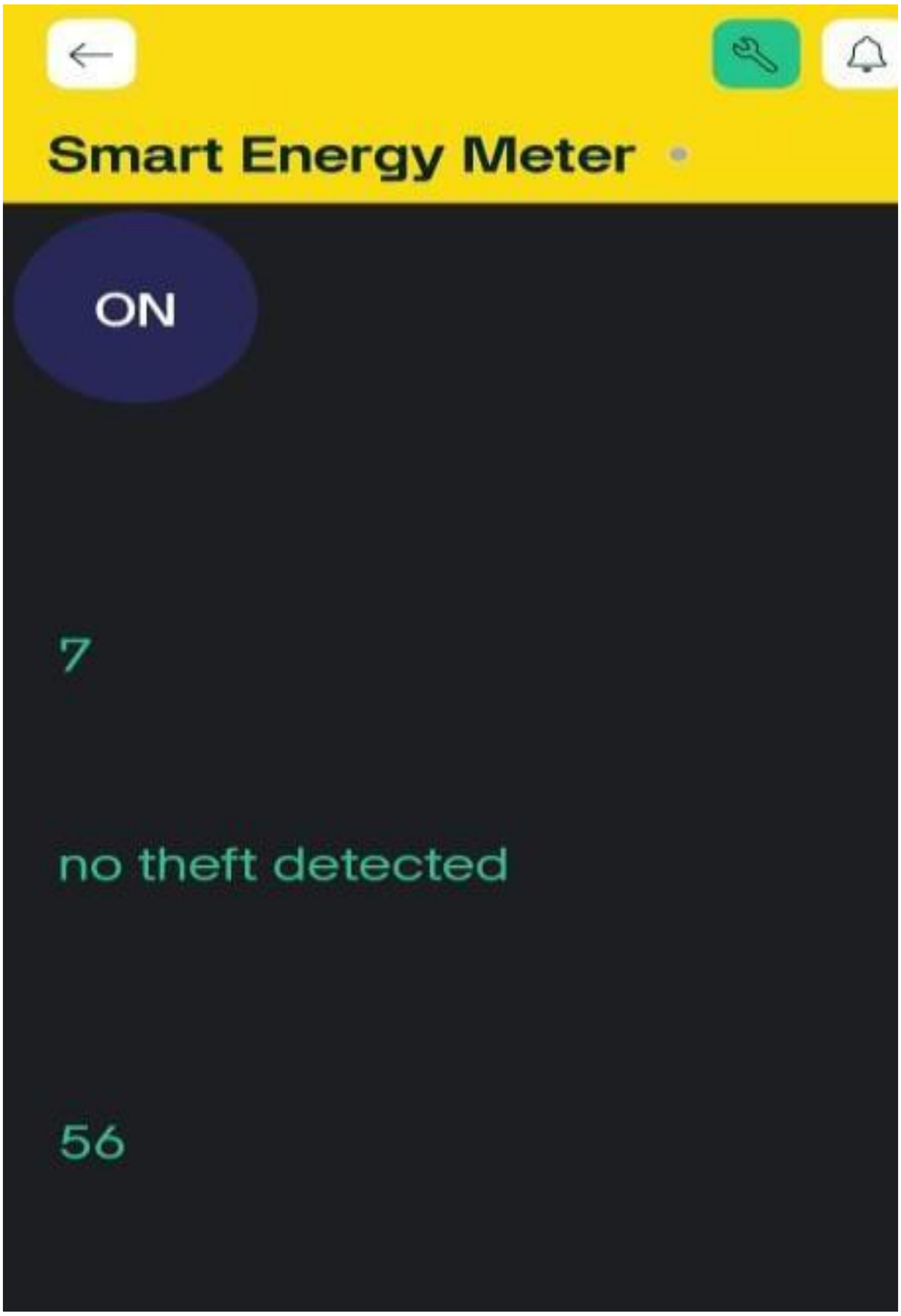


Figure 7.1- Blynk IOT APP

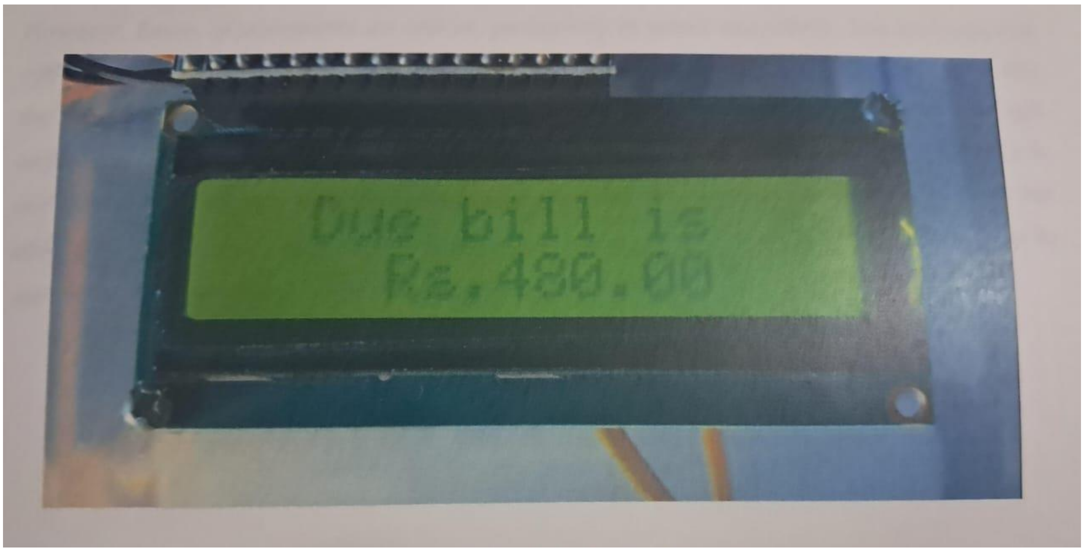


Figure 7.2- Due Bill Display

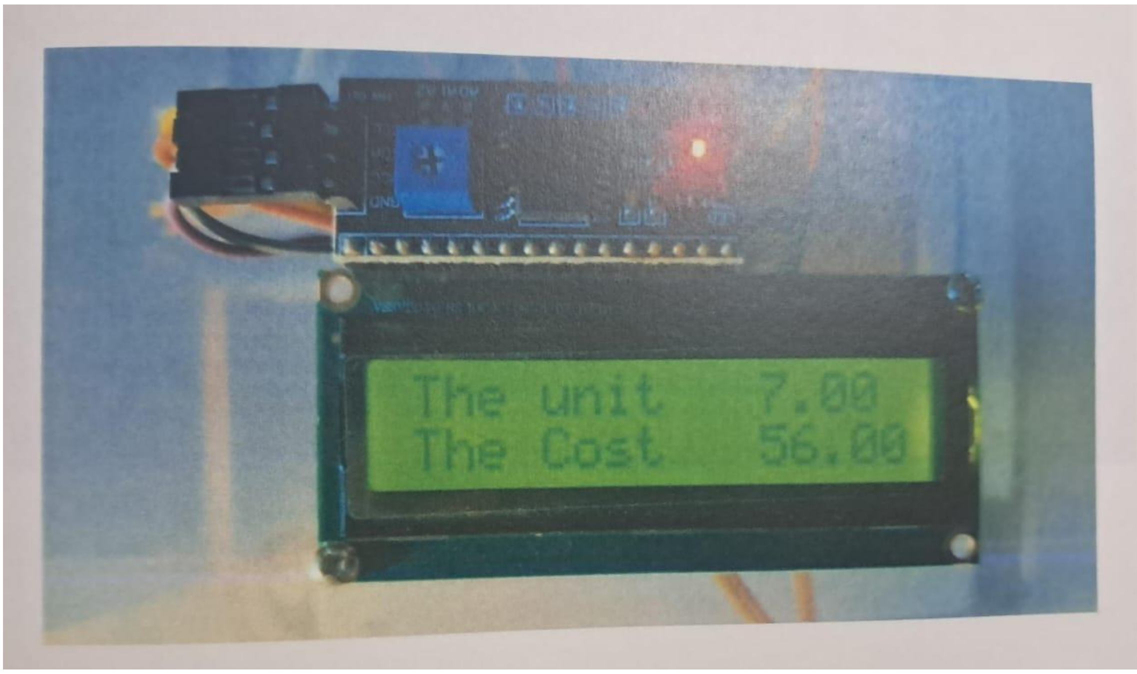


Figure 7.3- LED Meter Reading Display

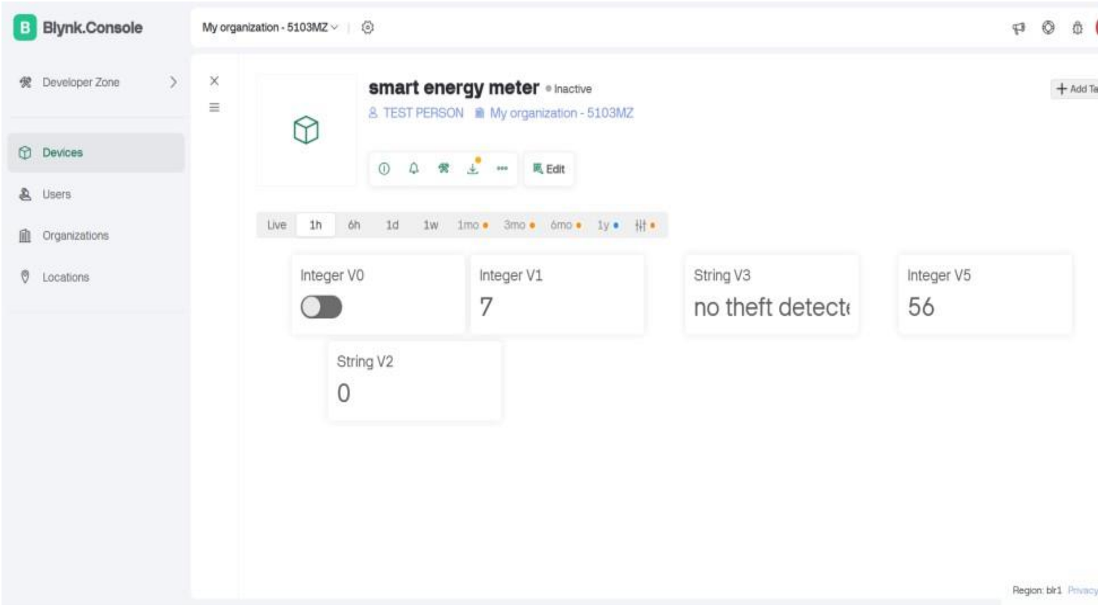


Figure 7.4-Smart Energy Meter Interface: Mobile App Display

Figure 7.5- Help & Support Section

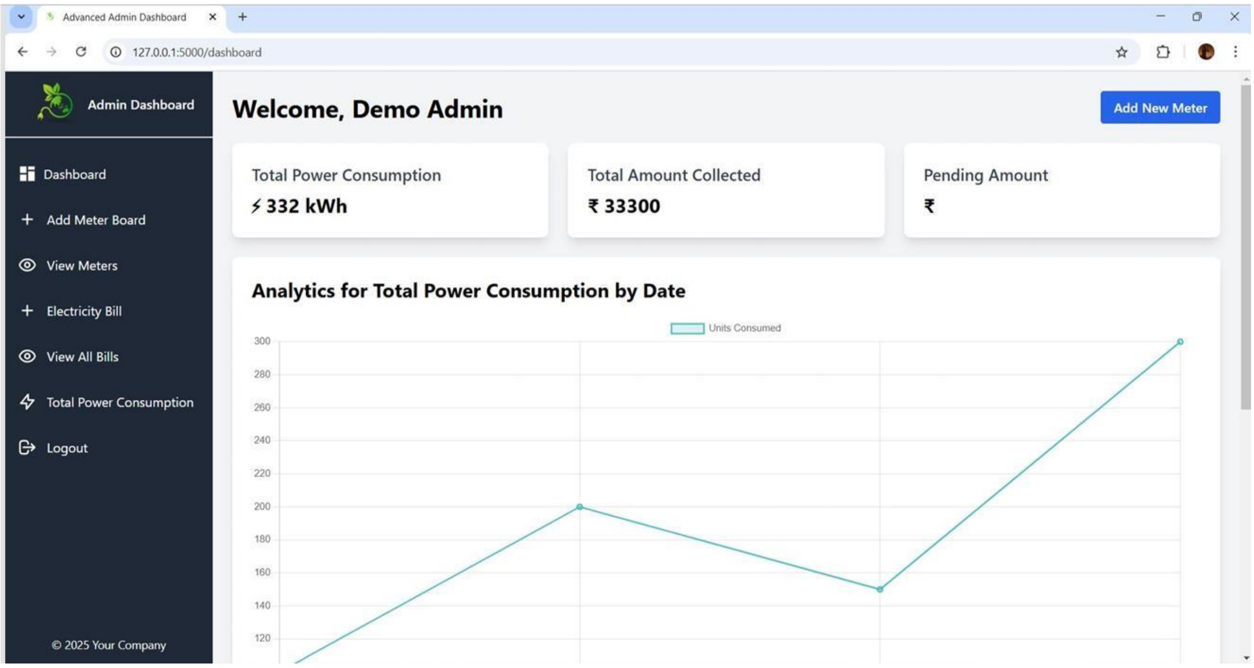


Figure 7.6-Dashboard Overview

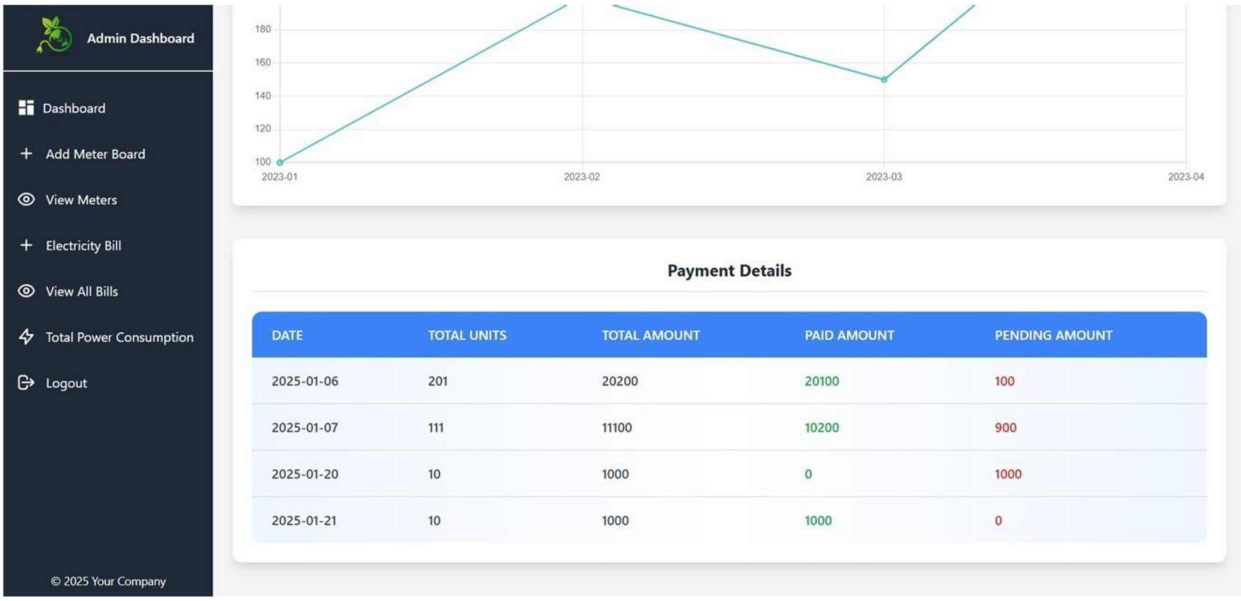


Figure 7.7-Admin Dashboard - Payment Overview

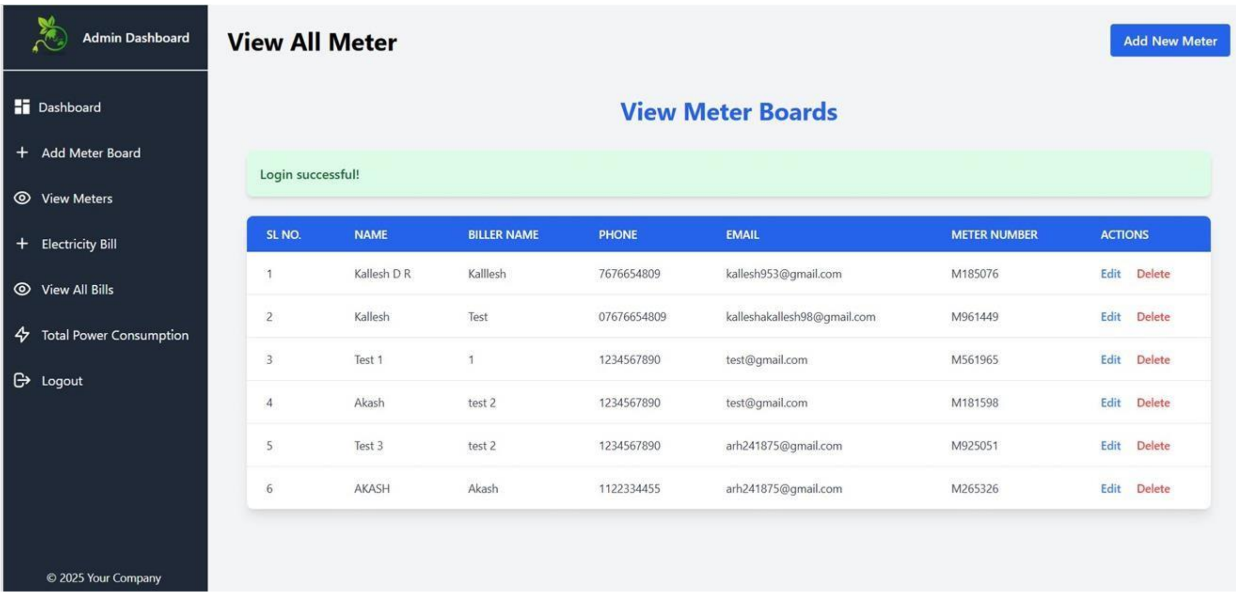


Figure 7.8-Transaction History

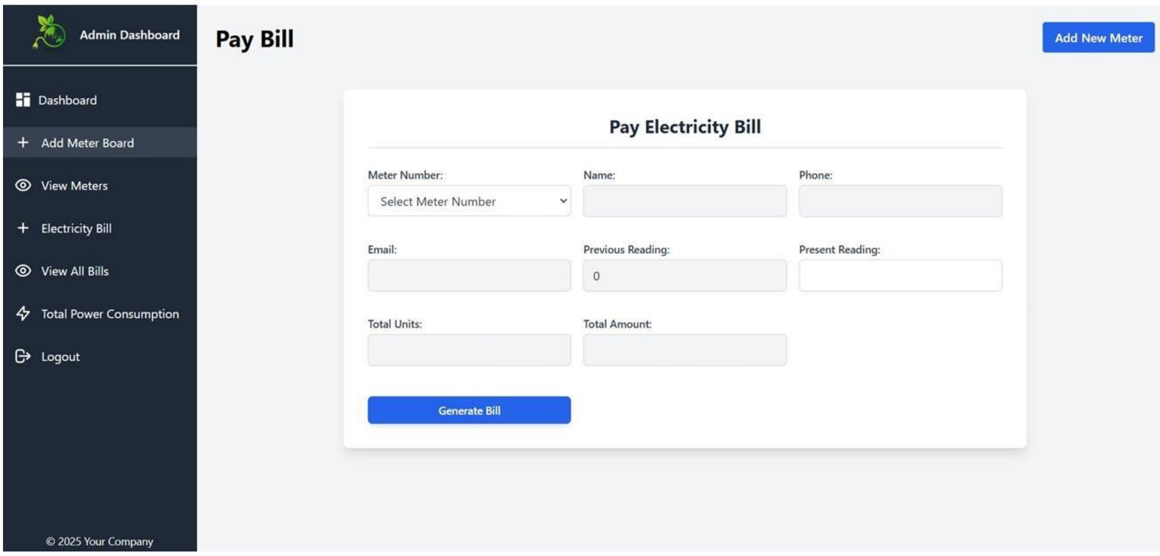


Figure 7.9- Bill Payment Section

Figure 7.10-Login Screen

CHAPTER 8

CONCLUSION AND FUTURE WORK

CHAPTER 8

CONCLUSIONS

Embedded Systems and Real-Time Operating Systems (RTOS) are integral to modern technological advancements, enhancing efficiency, automation, and reliability. The Electricity Bill Prediction System exemplifies how embedded systems and machine learning collaborate to provide accurate electricity cost estimations, enabling users to monitor and optimize power consumption. With rising energy costs, predictive analytics plays a crucial role in financial planning and energy conservation.

This system incorporates hardware components like Arduino microcontrollers, energy meters, Wi-Fi modules, relays, and LCD displays to monitor real-time energy usage and transmit data for analysis. The software framework, including Arduino IDE, Embedded C, and the Blynk app, ensures seamless data processing and visualization. This integration simplifies electricity billing and promotes responsible energy usage.

Despite advantages like efficiency, real-time processing, automation, and cost savings, embedded systems pose challenges such as limited upgradability, complex debugging, and privacy concerns. The increasing adoption of IoT and smart metering raises cybersecurity risks, but advancements in encryption, secure data transmission, and privacy policies aim to mitigate these issues.

The future of embedded systems in energy management is promising, with IoT-driven smart grids, AI-powered analytics, and real-time monitoring optimizing electricity distribution. As industries and households adopt smart meters, automated billing, and predictive analytics, traditional electricity monitoring will become obsolete. Additionally, integrating renewable energy sources, such as solar and wind, with embedded monitoring systems will enhance sustainability.

In conclusion, the Electricity Bill Prediction System is a significant step toward intelligent energy management, demonstrating the potential of embedded systems, IoT, and machine learning in transforming energy consumption. While challenges exist, ongoing advancements in embedded technology will continue to drive innovation, making power usage more efficient, cost-effective, and environmentally sustainable.

Future enhancements

The Electricity Bill Prediction System has significant potential for future enhancements, making energy management more efficient, intelligent, and sustainable. As technology advances, the integration of Artificial Intelligence (AI), the Internet of Things (IoT), smart grids, and renewable energy sources will further revolutionize electricity monitoring and billing. Below are some key future scopes of this system:

i. **AI & IoT Integration** – Enhances prediction accuracy and enables real-time energy monitoring with smart meters.
ii. **Renewable Energy & Smart Grids** – Optimizes solar/wind energy usage and adjusts pricing dynamically based on demand.
iii. **Anomaly Detection & Security** – Identifies energy wastage, prevents theft, and secures transactions using blockchain.
iv. **Automation & User Convenience** – Enables smart appliance control, automated billing, and mobile app integration with voice assistants.
v. **Government & Utility Collaboration** – Supports energy policy-making, demand forecasting, and incentive-based pricing models.

References

[1] Mehmood, N., Ali, Z. A., Sidiqui, A. A., Asif, M., & Wasi, S. (2011). Electronic meter reader & data base management system. 2011 6th IEEE Joint International Information Technology and Artificial Intelligence Conference.

[2] Shahidi, S., Gaffar, M. A., & Salim, K. M. (2013). Design and implementation of digital energy meter with data sending capability using GSM network. 2013 2nd International Conference on Advances in Electrical Engineering (ICAEE). doi:10.1109/icaee.2013.6750333.

[3] “Internet of Things(IoT) Based Smart Energy Tracking and Bill Estimation System” authorised by Rishi Mathur , Kamlesh Kalbande Electronics and Telecommunication Engineering G H Raisoni College of Engineering Nagpur published in 2020 4th International conference on I- SMAC . DOI-10.1109/ISMAC49090.2020.9243480.

[4] "Smart Energy Metering and Power Theft Control using Arduino & GSM” authorised by Visalatchi S and Kamal Sandeep K published in 2018 2nd International Conference for Convergence in Technology (I2CT).

[5] “Electricity Billing and Consumption Prediction using Machine Learning” authorised by Mrunalini Bhandarkar Ashok Suryawanshi published in 2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA)

[6] **Predictive Modeling for Electricity Billing: A Data-Driven Approach** (2024): This study utilizes advanced data analytics approaches to create prediction models for power bills. The algorithms predict future energy demand with high accuracy by combining previous power consumption data with contextual elements such as weather patterns, time-of-use pricing, and demographic information. The study's goal is to streamline energy billing systems, allowing utilities to better forecast demand variations and improve operational efficiency.

[7] **Role of Artificial Intelligence in Smart Meters** (2023): This article discusses how artificial intelligence, particularly machine learning algorithms, can analyze data from smart meters to reduce energy consumption. It emphasizes the use of time-series machine learning algorithms to predict future energy usage, aiding in demand-response and ensuring security.

analyticsvidhya.com

[8] **Smart Electricity Billing System Using IoT & ML** (2025): This paper proposes a system that utilizes advanced algorithms to analyze consumption data for patterns and trends. Machine learning models are employed to predict future electricity usage, identify inefficiencies, and optimize energy management strategies.
irjmets.com

[9] **Energy Consumption Forecasting for Smart Meters Using Extreme Learning Machine** (2021): This study proposes a data-driven ensemble that combines five well-known models in the forecasting literature, including statistical linear autoregressive models and four types of artificial neural networks. The ensemble employs extreme learning machines as the combination model due to their simplicity, learning speed, and generalization ability. Experiments conducted on real consumption data collected from a smart meter demonstrate that this solution outperforms other statistical, machine learning, and ensemble models proposed in the literature.
mdpi.com

[10] **Machine Learning-Based Energy Management System for Prosumer** (2020): This paper explores and compares various system architectures to store and process smart meter reading data, aiming to establish a solid foundation for future intelligent systems. The study highlights the potential of machine learning algorithms in developing innovative solutions for energy management, emphasizing the need for further research in this area to achieve significant energy savings and a more sustainable future.
researchgate.net

[11] **Improved Energy Efficiency in IoT-Based Smart Energy Meter** (2023): This study highlights the potential of machine learning algorithms and IoT technologies in developing innovative solutions for energy management. It emphasizes the need for further research in this area to achieve significant energy savings and a more sustainable future.
riverpublishers.com

[12] **Modelling and Predicting Energy Usage from Smart Meter Data and Consumer Behaviors in Residential Houses** (2022): This thesis proposes a novel approach of using smart meter data obtained through profiling users remotely, enabling the detection of abnormal user behaviors. Advanced analytics tools for machine learning are utilized to study data patterns and profile user

behavior, contributing to more accurate energy usage predictions and efficient energy management.
researchonline.ljmu.ac.uk

[13] **The Energy Prediction Smart-Meter Dataset: Analysis of Previous Competitions and Beyond** (2023): This paper presents a real-world smart-meter dataset and offers an analysis of solutions derived from the Energy Prediction Technical Challenges. It focuses on accurate energy consumption forecasting and the importance of interpretability in understanding underlying factors. The study addresses the challenge of predicting monthly and yearly estimated consumption for households, contributing to accurate billing with limited historical smart meter data.
arxiv.org

[14] IoT Based Smart Energy Metering System with Usage Prediction ,youtube url : https://youtu.be/WKPRzeW-xwM.

[15] Smart Energy Management System ,youtube url : https://youtu.be/2E1JD2N6qzA.