** ENERGY DEMAND FORECASTING**

**SOFTWARE USING MACHINE LEARNING**

**Submitted by**

**ALWIN ANTONY (810421243008)**

**GLAISON ANTONY P J (810421243017)**

# **SREEJITH S (810421243054)**

**Of**

## **DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE (AUTONOMOUS)**

**PERAMBALUR – 621 212**

**A PROJECT REPORT**

**Submitted to the**

## **FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING**

***In partial fulfilment of the requirements***

***for the award of the degree***

**of**

**BACHELOR OF TECHNOLOGY**

**in**

# **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**ANNA UNIVERSITY CHENNAI - 600 025**

**MAY 2025**

**BONAFIDE CERTIFICATE**

Certified that this Project report on **“ENERGY DEMAND FOREASTING SOFTWARE USING MACHINE LEARNING”** is the bonafide work of **GLAISON ANTONY P J (810421243017), ALWIN ANTONY (810421243008), SREEJITH S (810421243054)** who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| **M. PALANISAMY ASSISTANT PROFESSOR OF AI&DS** | **Dr. Shree K.V.M, M.E., Ph.D.,** |
| **SUPERVISOR,** | **HEAD OF THE DEPARTMENT,** |
| Department of Artificial Intelligence and Data Science, | Department of Artificial Intelligence and Data Science, |
| Dhanalakshmi Srinivasan Engineering College (Autonomous), Perambalur – 621212 | Dhanalakshmi Srinivasan Engineering College (Autonomous), Perambalur - 621212 |

Submitted for the ProjectViva-Voce Examination held on \_\_\_\_\_\_\_\_\_\_\_

**INTERNAL EXAMINER EXTERNAL EX****AMINER**

**ACKNOWLEDGEMENT**

We express our gratitude and thanks to **Our Parents** first for giving health and a sound mind for completing this project. We give all the glory and thanks to our almighty **GOD** for showering upon, the necessary wisdom and grace for accomplishing this project.

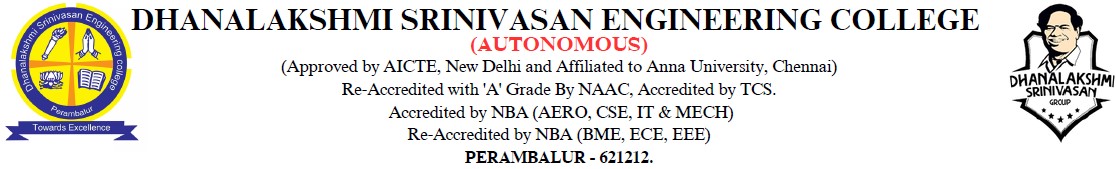
It is our pleasant duty to express deep sense of gratitude to our Honourable Chancellor **Shri. A. Srinivasan**, for his kind encouragement. We have unique pleasure in thanking our Principal **Prof. Dr. D. Shanmugasundaram, M.E., PhD.,** our Dean **Dr. K. Anbarasan, M.E., Ph.D.,** and our COE **Dr. M. Chellappan M.E., Ph.D.,** for their unflinching devotion, which leads us to complete this project.

We express our faithful and sincere gratitude to our Head of the Department **Dr. Shree K.V.M, M.E., Ph.D.,** Artificial Intelligence and Data Science, for her valuable guidance and support the gave us during the project time.

We express our faithful and sincere gratitude to our Project Coordinator**, Ms. Suganya S** of Artificial Intelligence and Data Science, for her valuable guidance and support the gave us during the project time.

We are also thankful to our internal project guide **Mr. M Palanisamy,** Assistant Professor of Department of Artificial Intelligence and Data Science for his valuable guidance and precious suggestion to complete this project work successfully.

We render our thanks to all **Faculty members** and **Programmers** of Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE** for their timely assistance.

****

**Vision**

To develop globally competitive professionals in the field of Artificial Intelligence (AI) & Data Science (DS) by imparting cognitive learning of AI&DS tools with basic Computer Science Knowledge and by encouraging Global Industrial collaboration towards serving the greater cause of Engineering Society.

**Mission**

**M1:** Impart knowledge in cutting edge Artificial Intelligence and Data Science technologies in par with industrial standards.

**M2:** Inculcate research and lifelong learning that benefit society at large which promotes ethical values and entrepreneurial skill.

**PROGRAM EDUCATIONAL OBJECTIVES (PEOs)**

**PEO1:** Graduates will have the ability to adapt, contribute and innovate new technologies and systems in the key domains of Artificial Intelligence and Data Science.

**PEO2:** Graduates will be able to successfully pursue higher education in reputed institutions with AI Specialization.

**PEO3:** Graduates will have the ability to explore research areas and produce outstanding contribution in various areas of Artificial Intelligence and Data science.

**ABSTRACT**

Efficient energy management is essential for balancing supply and demand in various sectors, including markets, industries, and residential areas. This project focuses on developing a machine learning-based energy demand forecasting system that predicts energy needs for specific times and scenarios, optimizing resource utilization. The system monitors real-time environmental parameters such as temperature, humidity, and wind speed, displaying these values on an LCD. Using these inputs, the AI model analyses and predicts energy demand patterns across different sectors. For instance, during peak market hours or industrial shifts, the system identifies high energy demand periods, enabling better resource allocation. Seasonal variations, such as increased cooling needs in summer or heating requirements in winter, are also considered. This predictive approach empowers energy managers to prepare for peak demand periods, such as during extreme weather conditions or specific industrial operations, ensuring reliable energy supply. By forecasting energy needs and recommending optimal energy usage strategies, the system promotes sustainability and energy conservation, reducing costs and minimizing waste. This innovative solution aligns with modern energy efficiency goals, providing a robust tool for informed decision-making in energy management.

**KEYWORDS:** Energy management, Machine learning, Energy demand forecasting, Supply and demand balance, Resource optimization, Real-time monitoring, Demand patterns, Peak energy periods, Seasonal variations, Energy allocation, Sustainability, Cost reduction, Smart energy system.

**TABLE OF CONTENTS**

**CHAPTER NO: CONTENT PAGE NO:**

**ABSTRACT v**

**TABLE OF CONTENTS vi**

**LIST OF FIGURES viii**

**LIST OF ABBREVIATIONS ix**

**1**  **INTRODUCTION 1**

* 1. INTRODUCTION 1
  2. AIM 2
  3. OBJECTIVES 2

**2**  **LITERATURE SURVEY** **3**

**3 SYSTEM ANALYSIS AND DESIGN 10**

3.1 EXISTING SYSTEM 10

3.2 PROPOSED SYSTEM 10

3.3 BLOCK DIAGRAM 12

**4**  **HARDWARE REQUIREMENT 13**

4.1 POWER SUPPLY 13

4.2 TRANSFORMER 14

4.3 RECTIFIER 16

4.4 SMOOTHING 20

4.5 REGULATOR 21

4.6 IOT ESP8266 23

4.7 LCD DISPLAY 25

4.8 ARDUINO 27

4.9 DIGITAL PINS 29

4.10 TEMPERATURE SENSOR AND 33

HUMIDITY SENSOR

4.11 WIND SENSOR 37

**5 SOFTWARE REQUIREMENT 39**

5.1 ARDUNIO IDE SKETCH 39

5.2 ARDUNIO 40

5.3 VARIABLES 41

5.4 FUNCTIONS 41

**6 ADVANTAGES AND APPLICATIONS 44**

6.1 ADVANTAGES 44

6.2 APPLICATIONS 44

**7 MODULES 46**

**8 RESULTS AND CONCLUSION 48**

7.1 RESULTS AND DISCUSSSION 48

7.2 CONCLUSION 50

**APPENDIX 1 51**

SAMPLE CODE 51

**APPENDIX 2 54**

PROJECT PHOTOGRAPH 54

**9 REFERENCES 55**

**LIST OF FIGURES**

**FIGURE NO: TITLE PAGE NO:**

3.1 SYSTEM ARCHITECTURE 12

4.1 REGULATED POWER SUPPLY SYSTEM 14

4.2 TRANSFORMER AND OUTPUT: LOW 15

VOTTAGE AC

4.3 RECTIFIER AND OUTPUT: VARYING DC 16

4.4 BRIDGE RECTIFIER 18

4.5 SINGLE DIODE RECTIFIER 19

4.7 SMOOTHING 21

4.8 REGULATOR 22

4.9 PIN DIG.OF IOT MODULE 23

4.11 LCD DISPLAY 25

4.13 CIRCUIT DIAGRAM OF LCD INTERFACING 27

WITH MICROCONTROLLER

4.14 ARDUINO 27

4.15 DHT11 SENSOR 33

4.16 2D MODEL OF THE SENSOR 37

**LIST OF ABBREVIATIONS**

EMS - ENERGY MANAGEMENT SYSTEM

ANNs - ARTIFICAL NEURAL NETWORK

SVMs - SUPPORT VECTOR MACHINES

PSU - POWER SUPPLY UNIT

RMS - ROOT MEAN SQUARE

ADC - ANALOG TO DIGITAL CONVERSION

LGPL - GNU LESSER GENERAL PUBLIC LICENSE

GPL - GNU GENERAL PUBLIC LICENSE

USB - UNIVERSAL SERIAL BUS

IDE - INTEGRATED DEVELOPMENT ENVIRONMENT

**CHAPTER- 1**

**INTRODUCTION**

**1.1 INTRODUCTION**

Efficient energy management has become increasingly vital in today’s world, as the demand for energy continues to rise while the availability of natural resources becomes more limited. Balancing energy supply and demand is essential for ensuring a reliable, cost-effective, and sustainable energy system. This is especially true across various sectors, including residential, industrial, and commercial areas, where energy consumption patterns can fluctuate based on several factors such as time of day, seasonal changes, and environmental conditions. To address these challenges, advanced forecasting tools and smart systems are necessary to predict energy needs with high accuracy, enabling better resource utilization and minimizing energy wastage.

This project introduces a **machine learning-based energy demand forecasting system** designed to predict energy consumption patterns in real-time, using environmental parameters such as temperature, humidity, and wind speed as input. The system continuously monitors these parameters and displays the data on an LCD for easy access and monitoring. By leveraging machine learning algorithms, the system analyses historical and real-time environmental data to predict energy demand at various times and under different scenarios, such as peak market hours or industrial operation shifts.

The predictive capability of this system allows energy managers to anticipate periods of high demand and adjust energy distribution accordingly, helping avoid power shortages or inefficiencies. For example, during hot summer months, when cooling needs are high, or cold winter months, when heating is in demand, the system takes into account these seasonal variations to optimize energy usage.

By offering real-time energy demand predictions and suggesting optimized energy strategies, this system not only enhances energy efficiency but also promotes sustainability. It helps reduce energy consumption during peak times, lowers operational costs, and minimizes waste, aligning with the goals of modern energy management and sustainability. This approach empowers decision-makers with the tools necessary for smarter, more efficient energy management in an increasingly energy-conscious world.

**1.2 AIM**

To develop a machine learning-based system that forecasts energy demand using environmental data, optimizing resource allocation and promoting sustainability.

**1.3 OBJECTIVES**

* To Create an AI-based forecasting system that predicts energy demand patterns for specific times, scenarios, and sectors, based on environmental parameters.
* To Continuously track and display environmental factors like temperature, humidity, and wind speed, which influence energy consumption patterns.
* To Enable accurate predictions of energy demand during peak market hours, industrial shifts, or seasonal variations, ensuring efficient allocation of energy resources.
* To Incorporate seasonal factors (e.g., increased cooling in summer or heating in winter) to enhance energy demand forecasting and improve energy distribution strategies.
* To Use predictive analytics to anticipate periods of high energy demand due to extreme weather conditions or specific industrial activities, ensuring timely energy availability.
* To Provide recommendations on optimal energy usage strategies, aiming to reduce waste, conserve energy, and lower costs across residential, industrial, and commercial sectors.
* To Contribute to sustainability goals by minimizing energy consumption during low-demand periods and reducing the carbon footprint through efficient energy management.
* To Equip energy managers with accurate forecasting data and actionable insights to make informed decisions, leading to improved overall energy management practices.

**CHAPTER 2**

**LITERATURE SURVEY**

**[1] Willim Caicedo- Torres and Fabian Payares (2014). A Machine Learning Model for Occupancy Rates and Demand Forecasting in the Hospitality Industry.**

Occupancy rate forecasting is a very important step in the decision-making process of hotel planners and managers. Popular state gives as Revenue Management feature forecasting as a vital activity for dynamic pricing, and without accurate forecasting, errors in pricing will negatively impact hotel financial performance. However, having accurate enough forecasts is no simple task for a wealth of reasons, as the inherent variability of the market, lack of personnel with statistical skills, and the high cost of specialized software. In this paper, several machine learning techniques were surveyed in order to construct models to forecast daily occupancy rates for a hotel, given historical records of bookings and occupation. Several approaches related to dataset construction and model validation are discussed. The results obtained in terms of the Mean Ab solute Percentage Error (MAPE) are promising, and support the use of machine learning models as a tool to help solve the problem of occupancy rates and demand forecasting.

**[2] Sai Mani Krishna Sistla¹, Gowrisankar Krishnamoorthy (2024). Machine Learning for Demand Forecasting in Manufacturing.**

This research paper investigates the application of machine learning (ML) techniques in demand forecasting within the manufacturing sector. By analysing case studies, practical examples, and comparative studies, we explore the effectiveness and challenges of ML-driven demand forecasting. The paper discusses various ML techniques, including regression models, time series forecasting methods, neural networks, and ensemble methods, highlighting their strengths and limitations. Evaluation metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) are examined to assess forecasting accuracy. Additionally, challenges such as data quality, model interpretability, computational resources, and overfitting are discussed, along with recommendations for addressing these challenges. The paper concludes with recommendations for practitioners and suggestions for future research directions, emphasizing the importance of data quality improvement, model interpretability enhancement, and ethical considerations in MI-based demand forecasting.

**[3]** **Guoxuan Liu, Dragan Savic D and Guangtao Fua (2023).** **Short-term water demand forecasting using data centric machine learning approaches.**

Accurate water demand forecasting is the key for urban water management and can alleviate system prasstee brought by urbanisation, water starcity and climate change. However, axosting research on water demand forecasting using machine learning focused on modal-centric approaches, what are the various forecasting models are tasted to improve accuracy. The study undertakes a data-centric machine learning approach by analysing the impact of training data length, temporal resolution and data uncertainty on forecasting model results. The models evulated are Autoregressive (AR) Integrated Moving Average (ARIMA), Neural Network (NN), Random Forest (OF) and Prophet. The first two are commonly used forecasting models. RF has shown similar forecast accuracy to NN but has received less attention, Prophet is a new model that has not been applied to short-term water demand forecasting, though it has had successful applications in various fields. The results obtained from four case studies show that (1) dlatar-centric machine learning approaches offer promise for improving forecast accuracy of short-term water demands; (2) accurate forecasts are possible with short training data, (3) RF and NN models are superior at forecasting high-temporal resolution data; and (4) data quality improvement can achieve a level of accuracy increase comparable to model-centric machine learning approaches.

**[4] Lukas Hans1, Patrick Eichenseer (2023). Machine Learning Forecasting of Daily Delivery Positions: A Modern Take on Industrial Workforce Planning.**

The logistics industry plays a pivotal role in global trade, and efficient ware-house operations are essential for the seamless movement of goods. In recent years, prevailing job market conditions have presented significant difficulties in recruiting skilled workers in warehouse operations [1]. This shortage of skilled logistics workers has challenged companies to meet dynamic market demands. and hindered effective workforce planning. The labour shortage results in increased operating costs and decreased overall efficiency due to suboptimal resource utilization. The key challenge for warehouse managers is the fluctuating and unpredictable nature of customer demand. Short-term customer orders, seasonal fluctuations, and rapidly changing market demands make it difficult for companies to forecast and plan their logistics workforce accurately. These dynamic demands often result in overstaffing during off-peak periods and understaffing during peak periods. Understaffing leads to unmet customer demand and, in some cases, customer chun. It also runs counter to a common warehouse goal of maximizing service levels (le, the promise of fast and accurate delivery) as a measure of differentiation from competitors.

**[5] Tahir Aja Zarma, Emmanuel Ali2, Ahmadu Adamu Galadimal (2015). Energy Demand Forecasting for Hybrid Microgrid Systems Using Machine Learning Models.**

This study aims to design energy demand forecasting models for energy management in hybrid microgrid systems using optimized machine learning techniques. By incorporating temperature, humidity, season, hour of the day, and irradiance, the complex relationship between these input parameters and the yield of photovoltaics, generator, and grid energy sources is examined. Five different machine learning models including linear regression, random forest (RF), support vector regression, artificial neural network, and extreme gradient boosting models are adopted in this study. Evaluation of model performance shows that the RF model is the best candidate for the dataset, with a mean-squared error of 0.2023, mean absolute error of 0.0831, root-mean-squared error of 0.4498, and R score of 0.9992. Shapley additive explanations analysis identified key predictors such hour, irradiation, and season while highlighting the negative impact of humidity and day of the week on energy demand.

**[6] Ahmet Tezcan Tekit, Cem Sarl (2023). Energy Demand Forecasting for Türkiye: Comparison Between Traditional Machine Learning Algorithms and Ensemble Learning Algorithms.**

Energy demand forecasting is pivotal in modern society by enabling effective energy resource allocation, infrastructure planning, and policy formulation. Accurate predictions of energy demand facilitate efficient energy generation, distribution, and consumption, contributing to sustainability and economic viability. Accurate and timely energy consumption predictions enable utilities, governments, and businesses to make informed decisions to ensure a reliable and efficient energy supply. Energy Demand Forecasting holds significant importance in modern society for various reasons, such as resource allocation, infrastructure development, economic planning, growth and investment decisions, etc. Due to these reasons, prediction for future energy demand is crucial for countries. In literature, many prediction techniques, including machine learning and deep learning, were used for demand prediction. This study used the dataset of Türkiye's energy consumption information between 2016 and 2023, and traditional and ensemble learning approaches were applied for future demand prediction. The results of these algorithms were compared briefly

**[7] Satish Anchuri Walmart, USA (2024).** **Machine Learning-Driven Demand Forecasting: A Comparative Analysis of Advanced Techniques and Real-Time Integration.**

Recent advancements in machine learning have significantly improved demand forecasting, offering new ways to optimize supply chains. This article analyzes advanced methods like deep learning, ensemble models, and transfer learning, showing that hybrid models using real-time IoT data boost prediction accuracy by 34.6% over traditional methods. A review of 127 case studies in retail and manufacturing revealed benefits such as a 28% drop in inventory costs and a 42% reduction in stockouts. The study also highlights key success factors, including data quality, resource management, and system integration, and offers a framework for selecting suitable ML techniques based on industry needs.

**[8] Raphael Ibraimoh (2024) . Time Series Forecasting of Energy Demand Using Machine and Deep Learning Approach**

This study examines the use of machine learning and deep learning models for forecasting hourly national energy consumption. Models like Random Forest, XG Boost, LSTM, and TCN were tested, with XG Boost performing best, achieving an RMSE of 393.48 and a MAPE of 1.16%. SHAP analysis revealed the importance of features like lagged demand and time of day. The results confirm XG Boost’s reliability for short-term forecasting, offering valuable support for energy providers. A weekly forecast using the model showed high accuracy and unbiased predictions, proving its effectiveness for operational use.

**[9] Arun Kumar Mishra, Megha Sinha, & Sudhanshu Kumar Jha (2024). Comparative analysis of machine learning algorithms for demand forecasting under uncertainty.**

Doing business in present time is quite a challenging task. With the advent of technology, world is going through a complete transition. We are seeing 4th Industrial Revolution and Industry 5.0 is ready to hit the landscape. In this scenario managing Supply Chain (SC) for optimum performance is becoming a tedious task. Demand forecasting is playing a crucial role over the years for efficient and effective management and planning for competitive advantage. This paper aims to study the performance of various Machine Learning (ML) namely Linear Regression, Decision Tree Regression, Random Forest Regression, Support Vector Machine Regression, XG Boost Regression algorithms for demand forecasting under uncertainty.

**[10] Xavier Godinho Filipe Tadeu Oliveira (2020).** **Forecasting Heating and Cooling Energy Demand in an Office Building using Machine learning methods.**

This paper presents a study on forecasting hourly heating and cooling energy demand in an office building in Lisbon, Portugal, using machine learning models. The study analyses how exogenous variables like occupancy, solar gains, and outdoor temperature impact predictions. Traditional models such as linear and polynomial regression were compared with machine learning methods like artificial neural networks and support vector regression. Model performance was evaluated using MAE and RMSE against data from a calibrated building energy simulation. This work forms part of a broader research project exploring the use of machine learning for energy forecasting in buildings.

**[11] Emmanuel (2025). Leveraging machine learning techniques for precise energy demand forecasting in power sector.**

This paper explores the use of advanced machine learning techniques to improve energy demand forecasting in the power sector. Unlike traditional methods, ML models like LSTM networks, Gradient Boosting Machines, and hybrid approaches can handle complex, non-linear patterns and adapt to real-time data. The study evaluates forecast accuracy using MAE and RMSE, incorporating factors such as weather, economic trends, and regional consumption. Results show that ML enhances prediction precision and supports better grid management, resource allocation, and load balancing. The paper also highlights the future role of explainable AI and real-time predictive maintenance in smart grids.

**[12] Osinachi Deborah Segun-Falade, Olajide Soji Osundare, Wagobera Edgar Kedi. (2024). Developing innovative software solutions for effective energy management systems in industry.**

This review highlights the role of innovative software solutions in improving energy management systems (EMS) within the industrial sector. Leveraging AI, machine learning, and IoT, modern EMS tools optimize energy use, cut costs, and reduce environmental impact. These systems utilize real-time data and predictive analytics to forecast demand, manage energy loads, and identify inefficiencies. IoT devices enable seamless data collection and monitoring, while user-friendly dashboards support quick decision-making. Automated controls further enhance energy efficiency. Case studies show significant energy savings, such as a 15% reduction in one manufacturing plant. Overall, advanced EMS software is key to achieving industrial sustainability and efficiency goals.

**[13] Adebayo Olusegun Aderibigbe, Emmanuel Chigozie Ani, Peter Efosa Ohenhen. (2023), Enhancing energy efficiency With AI: A review of machine learning Models in electricity demand forecasting.**

This study reviews the impact of AI and machine learning on improving energy efficiency, with a focus on electricity demand forecasting. It highlights how ML models, particularly those using deep learning and big data, outperform traditional methods in accuracy and adaptability. The review emphasizes the importance of selecting models based on accuracy, data handling, and environmental impact. It also explores the broader technological, economic, and environmental benefits of ML, while addressing challenges like data privacy and the need for skilled professionals. Strategic recommendations and future research directions are offered to support sustainable, AI-driven energy forecasting.

**[14] R. A. Swief, Noha H. El-Amary & H. K. Temraz (2025). Seasonal forecasting of the hourly electricity demand applying machine and deep learning algorithms impact analysis of different factors.**

This paper proposes a short-term seasonal forecasting model for hourly electricity demand in the ISO New England Control Area, using machine and deep learning methods such as ANFIS, LSTM, GRU, and ANN. The dataset is divided by seasons, and the model considers the impact of temperature on demand during workdays and holidays. Two scenarios—forecasting on a workday and on a holiday—are analysed across all seasons. Results show that ANFIS achieved the highest accuracy (99.90%) in winter for workdays and (96.50%) in autumn for holidays. Model performance was evaluated using MAE, RMSE, NRMSE, and MAPE, confirming strong forecasting capabilities with minimal error

**[15] Sarunyoo Boriratrit, Chitchai Srithapon, Pradit Fuangfoo and Rongrit Chatthaworn ,(2022).** **Metaheuristic Extreme Learning Machine for Improving Performance of Electric Energy Demand Forecasting.**

This study focuses on improving electric energy demand forecasting using Extreme Learning Machine (ELM) models, which are valued for their speed and forecasting accuracy. However, standard ELM models often face overfitting issues. To address this, the research introduces three optimized models—JS-ELM, HH-ELM, and FP-ELM—by combining ELM with metaheuristic algorithms. Using Thailand’s energy demand data from 2018 to 2020, the models were tested and compared. Results show that JS-ELM achieved the lowest RMSE and offered efficient processing time, making it the most effective among the models tested**.**

**CHAPTER 3**

**SYSTEM ANALYSIS AND DESIGN**

**3.1 EXISTING SYSTEM**

Existing energy demand forecasting systems have been widely used across various sectors, but they often have limitations in terms of adaptability and accuracy. Traditional energy management systems (EMS) typically rely on statistical models like regression analysis and time-series forecasting to predict energy consumption. While useful in stable conditions, these methods struggle to account for real-time environmental factors such as weather changes, which can significantly impact energy demand. Similarly, smart grids, though equipped with sensors and automation for real-time energy monitoring, are more focused on energy distribution and management rather than sophisticated forecasting. While they can track usage patterns, they do not employ advanced predictive models based on external variables like temperature or humidity.

In more advanced systems, machine learning models, such as artificial neural networks (ANNs) or support vector machines (SVMs), are used to predict energy demand with greater precision. However, these systems require substantial computational power and high-quality data, making them challenging for broader implementation. IoT-based systems, which integrate real-time data from environmental sensors, have improved the monitoring process. Still, they are typically limited in their ability to predict long-term energy demand patterns accurately. Despite these advancements, the existing systems often fail to provide a comprehensive, adaptable solution that accounts for both environmental conditions and dynamic energy usage patterns in real time.

**3.2 PROPOSED SYSTEM**

The proposed system aims to enhance energy demand forecasting by integrating machine learning algorithms with real-time environmental data, enabling more accurate and dynamic predictions for energy consumption across various sectors. The system uses a combination of sensors to monitor environmental parameters such as temperature, humidity, and wind speed, which significantly influence energy demand. This real-time data is then fed into a machine learning model, which has been trained to recognize patterns in energy consumption based on environmental and historical usage data.

By utilizing the power of artificial intelligence, the system not only forecasts energy demand but also adapts to changes in weather conditions and seasonal variations, such as increased cooling needs during summer or heating demands in winter. The system can predict energy consumption for different sectors, such as residential, commercial, and industrial, allowing energy managers to optimize resource allocation based on predicted demand during peak hours or extreme conditions.

The key innovation of this system is its ability to provide real-time, data-driven insights that help optimize energy use. Through a cloud-based interface, energy managers can access predictions and actionable recommendations for reducing energy waste, lowering operational costs, and improving overall efficiency. By leveraging machine learning, IoT technology, and cloud integration, this system offers a cost-effective, scalable, and accurate solution for forecasting and managing energy demand in real time, contributing to sustainable energy use and better resource management.

**3.3 BLOCK DIAGRAM**

**POWER SUPPLY FOR ALL**

**UNIT**

**MICRO**

**CONTROLLER**

**TEMPERATURE**

**USB TO UART**

**PC**

**WIND SPEED**

**HUMIDITY**

**IOT**

**LCD**

Fig 3.1 System architecture

**CHAPTER 4**

**HARDWARE REQUIREMENT**

**4.1 POWER SUPPLY**

Power supply is a reference to a source of [electrical power](http://en.wikipedia.org/wiki/Electrical_power). A device or system that supplies [electrical](http://en.wikipedia.org/wiki/Electrical) or other types of [energy](http://en.wikipedia.org/wiki/Energy) to an output [load](http://en.wikipedia.org/wiki/External_electric_load) or group of loads is called a power supply unit or PSU. The term is most commonly applied to electrical energy supplies, less often to mechanical ones, and rarely to others.

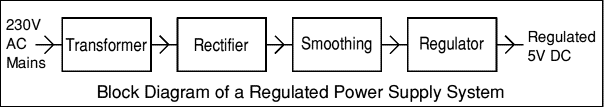
Power supplies for electronic devices can be broadly divided into linear and switching power supplies. The linear supply is a relatively simple design that becomes increasingly bulky and heavy for high current devices; voltage regulation in a linear supply can result in low efficiency. A switched-mode supply of the same rating as a linear supply will be smaller, is usually more efficient, but will be more complex.

**4.1.1 Linear Power supply:**

An [AC](http://en.wikipedia.org/wiki/Alternating_current) powered linear power supply usually uses a [transformer](http://en.wikipedia.org/wiki/Transformer) to convert the voltage from the wall outlet (mains) to a different, usually a lower voltage. If it is used to produce [DC](http://en.wikipedia.org/wiki/Direct_current), a [rectifier](http://en.wikipedia.org/wiki/Rectifier) is used. A [capacitor](http://en.wikipedia.org/wiki/Capacitor) is used to smooth the pulsating current from the rectifier. Some small periodic deviations from smooth direct current will remain, which is known as [ripple](http://en.wikipedia.org/wiki/Ripple_(electrical)). These pulsations occur at a frequency related to the AC [power frequency](http://en.wikipedia.org/wiki/Utility_frequency) (for example, a multiple of 50 or 60 Hz).

The voltage produced by an unregulated power supply will vary depending on the load and on variations in the AC supply voltage. For critical electronics applications a [linear regulator](http://en.wikipedia.org/wiki/Linear_regulator) will be used to stabilize and adjust the voltage. This regulator will also greatly reduce the ripple and noise in the output direct current. Linear regulators often provide current limiting, protecting the power supply and attached circuit from over current.

Adjustable linear power supplies are common laboratory and service shop test equipment, allowing the output voltage to be set over a wide range. For example, a bench power supply used by circuit designers may be adjustable up to 30 volts and up to 5 amperes output. Some can be driven by an external signal, for example, for applications requiring a pulsed output.



.

Fig 4.1 Regulated Power supply System

### **4.2 Transformer:**

### 

### transformer symbol

### Transformers convert AC electricity from one voltage to another with little loss of power. Transformers work only with AC and this is one of the reasons why mains electricity is AC.

### Step-up transformers increase voltage, step-down transformers reduce voltage. Most power supplies use a step-down transformer to reduce the dangerously high mains voltage (230V in UK) to a safer low voltage.

The input coil is called the primary and the output coil is called the secondary. There is no electrical connection between the two coils; instead they are linked by an alternating magnetic field created in the soft-iron core of the transformer. The two lines in the middle of the circuit symbol represent the core.

### Transformers waste very little power so the power out is (almost) equal to the power in. Note that as voltage is stepped down current is stepped up.

The ratio of the number of turns on each coil, called the turn’s ratio, determines the ratio of the voltages. A step-down transformer has a large number of turns on its primary (input) coil which is connected to the high voltage mains supply, and a small number of turns on its secondary (output) coil to give a low output voltage.

Turns ratio=Vp/Vs=Nn/Ns and Power out=Power in

Vs\*Is=Vp \* Ip

|  |  |  |
| --- | --- | --- |
| Vp = primary (input) voltage Np = number of turns on primary coil Ip  = primary (input) current |  | Vs = secondary (output) voltage Ns = number of turns on secondary coil Is  = secondary (output) current |

### AC power supply, transformer only Fig 4.2 Transformer and Output: low voltage AC

A low voltage AC output can be very useful for powering simple electrical devices such as lamps, heaters, and certain types of special AC motors. These devices do not require a constant, unidirectional flow of current; instead, they can operate efficiently with alternating current (AC), even when the voltage is relatively low. Lamps and heaters, for instance, rely on the heating effect of the current and are not sensitive to the direction in which it flows. Similarly, some AC motors are specifically designed to run on AC supply and can function properly at lower voltages.

However, low voltage AC is generally unsuitable for electronic circuits. Electronic devices—such as microcontrollers, digital sensors, amplifiers, and communication equipment—require a stable and consistent DC voltage to operate correctly. AC voltage, with its continuous changes in direction and magnitude, can cause serious disruptions in the operation of sensitive electronic components, leading to malfunctions, erratic behavior, or even damage.

### **4.3 Rectifier:**

### There are several ways of connecting diodes to make a rectifier to convert AC to DC. The [bridge rectifier](http://www.kpsec.freeuk.com/powersup.htm#bridgerectifier) is the most important and it produces full-wave varying DC. A full-wave rectifier can also be made from just two diodes if a centre-tap transformer is used, but this method is rarely used now that diodes are cheaper. A [single diode](http://www.kpsec.freeuk.com/powersup.htm#singlediode) can be used as a rectifier but it only uses the positive (+) parts of the AC wave to produce half-wave varying DC.

### DC power supply, transformer + rectifier

Fig 4.3 Rectifier and Output: varying DC

The varying DC output is suitable for lamps, heaters and standard motors. It is not suitable for electronic circuits unless they include a smoothing capacitor.

One of the most widely used types is the bridge rectifier, which employs four diodes arranged in a specific configuration. This setup enables the conversion of both the positive and negative halves of the AC waveform into a full-wave varying DC output. The full-wave rectification process ensures that the output maintains a more continuous flow of current compared to half-wave rectification, making it more efficient and powerful.

Alternatively, a full-wave rectifier can also be constructed using just two diodes when paired with a center-tap transformer. The center tap provides a neutral point that effectively allows the two diodes to handle opposite halves of the AC cycle. However, this method has become less common in modern circuits, primarily because diodes have become inexpensive and the bridge rectifier offers a more compact and cost-effective solution without the need for specialized transformers.

### **4.3.1 Bridge rectifier:**

### A bridge rectifier is a circuit commonly used to convert alternating current (AC) into direct current (DC) efficiently. It can be assembled manually using four individual diodes connected in a specific configuration, or it can be purchased as a ready-made integrated package that contains all four diodes pre-connected internally. These ready-to-use bridge rectifier modules simplify circuit design and save space on circuit boards.

### The bridge rectifier is often called a full-wave rectifier because it utilizes both the positive and negative halves of the AC input waveform. This means that no part of the AC cycle is wasted: during both the positive and negative swings of the AC input, the output current flows in the same direction through the load. This results in a smoother and more efficient DC output compared to a half-wave rectifier, which only uses one half of the AC cycle. However, a bridge rectifier does have a small voltage drop due to the nature of the diodes: Each silicon diode, when conducting, typically has a forward voltage drop of about 0.7 volts.

In a bridge rectifier, at any given time during conduction, two diodes are in series (one for the forward path and one for the return path). Therefore, the total voltage drop across the bridge is approximately 1.4 volts (0.7V + 0.7V). This voltage drop must be taken into account when designing power supplies, especially for low-voltage applications where a 1.4V loss might be significant.

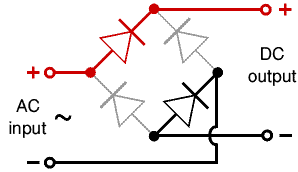
Bridge rectifiers are rated based on two key factors:

1. Maximum Current Rating — the highest current the rectifier can safely pass through without damage. This must match or exceed the current needs of the load.

2. Maximum Reverse Voltage Rating — the highest voltage the diodes can block in the non-conducting direction without breaking down.

As a general rule, the reverse voltage rating should be at least three times the RMS (Root Mean Square) value of the AC input voltage. This ensures the rectifier can safely handle the peak voltages present in the AC waveform, especially during voltage surges or spikes.

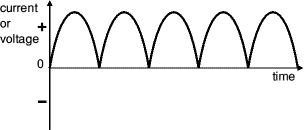
For more practical insights, diagrams, and pictures showing how bridge rectifiers look and how they are connected, you can refer to the Diodes page.



### Fig 4.4 Bridge Rectifier

### Alternate pairs of diodes conduct, changing over the connections so the alternating directions of AC are converted to the one direction of DC n a bridge rectifier, four diodes are arranged in a special configuration that allows them to convert the alternating current (AC)—which naturally flows in two directions—into direct current (DC), which flows in only one direction.

Output: full-wave varying DC: (using the entire AC wave)



**4.3.2 Single diode rectifier:**

A single diode can be used to build the simplest form of a rectifier, known as a half-wave rectifier. In this setup, the diode allows only the positive half-cycles of the alternating current (AC) to pass through, while blocking the negative half-cycles. As a result, the output is a pulsating direct current (DC) that consists of a series of positive voltage pulses separated by gaps whenever the AC voltage is negative.

However, this method has significant limitations. The presence of these gaps causes the output voltage to drop to zero during each negative half-cycle. This creates a highly irregular and "choppy" DC signal, making it unsuitable for powering most electronic circuits, which usually require a steady and continuous DC supply.

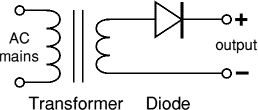
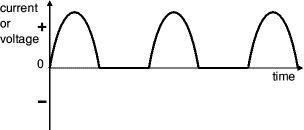


Fig 4.5 Single diode rectifier

output: half-wave varying DC (using only half the AC wave):



### **4.4 Smoothing:**

### Smoothing is performed by a large value electrolytic capacitor connected across the DC supply to act as a reservoir, supplying current to the output when the varying DC voltage from the rectifier is falling. The diagram shows the unsmoothed varying DC (dotted line) and the smoothed DC (solid line). The capacitor charges quickly near the peak of the varying DC, and then discharges as it supplies current to the output.

### Smoothing

Fig 4.6

Note that smoothing significantly increases the average DC voltage to almost the peak value (1.4 × [RMS](http://www.kpsec.freeuk.com/acdc.htm#rms) value). For example 6V RMS AC is rectified to full wave DC of about 4.6V RMS (1.4V is lost in the bridge rectifier), with smoothing this increases to almost the peak value giving 1.4 × 4.6 = 6.4V smooth DC. Smoothing is not perfect due to the capacitor voltage falling a little as it discharges, giving a small ripple voltage. For many circuits a ripple which is 10% of the supply voltage is satisfactory and the equation below gives the required value for the smoothing capacitor. A larger capacitor will give fewer ripples. The capacitor value must be doubled when smoothing half-wave DC.

Smoothing Capacitor for 10% ripple, C=5\*10/vs.\*f

C = smoothing capacitance in farads (F)

Io = output current from the supply in amps (A)

Vs = supply voltage in volts (V), this is the peak value of the unsmoothed DC

f    = frequency of the AC supply in hertz (Hz), 50Hz in the UK.

### Smooth DC power supply, transformer + rectifier + smoothing

### Fig 4.7 Smoothing

The smooth DC output has a small ripple. It is suitable for most electronic circuits.

**4.5 Regulator:**

Voltage regulator ICs are available with fixed (typically 5, 12 and 15V) or variable output voltages. They are also rated by the maximum current they can pass. Negative voltage regulators are available, mainly for use in dual supplies. Most regulators include some automatic protection from excessive current ('overload protection') and overheating ('thermal protection').

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and current.

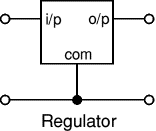
### Many of the fixed voltage regulator ICs has 3 leads and look like power transistors, such as the 7805 +5V 1A regulator shown on the right. They include a hole for attaching a [heat sink](http://www.kpsec.freeuk.com/components/heatsink.htm) if necessary.

1. Positive regulator
   1. input pin
   2. ground pin
   3. output pin

It regulates the positive voltage

1. Negative regulator
   1. ground pin
   2. input pin
   3. output pin

It regulate the negative voltage



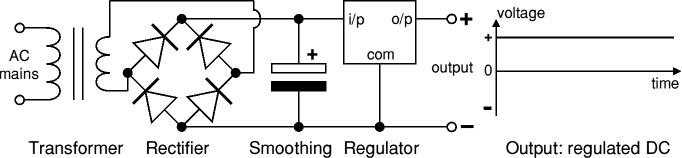


Fig 4.8 Regulator

The regulated DC output is very smooth with no ripple. It is suitable for all electronic circuits.

**4.6 IOT ESP8266**

An IoT module is a small electronic device embedded in objects, machines and things that connect to wireless networks and sends and receives data. Sometimes referred to as a "radio chip" or "IoT chip", the IoT module contains the same technology and data circuits found in mobile phones but without features like a display or keypad.

Another key differentiator of IoT modules is that they provide always-on connectivity. This is because IoT applications need to send data automatically, in real-time without someone hitting a send button.

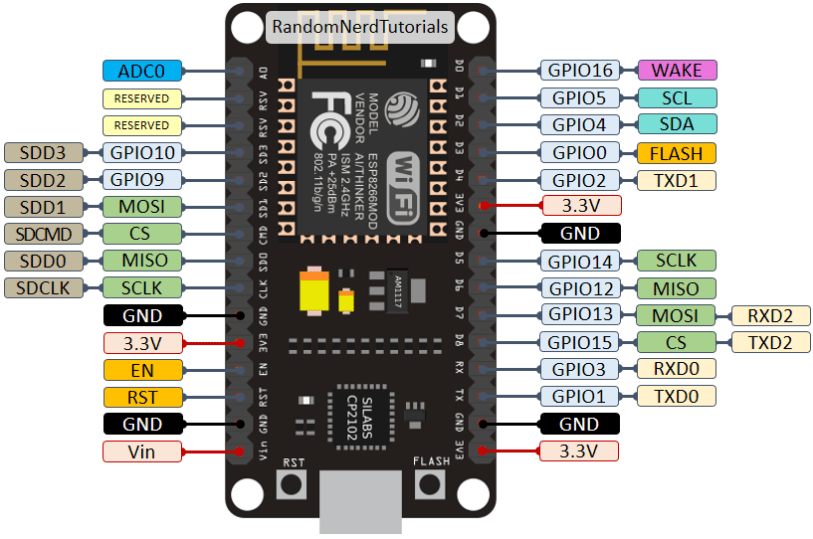


Fig4.9 pin dig. of IOT module

**4.6.1 PIN DESCRIPTION**

|  |  |
| --- | --- |
| **Name** | **Function** |
| VCC | Power 3.0 ~ 3.6V |
| GND | Ground |
| RESET | External reset signal (Low voltage level: Active) |
| ADC(TOUT) | ADC Pin Analog Input 0 ~ 1V |
| CH\_PD | Chip Enable. High: On, chip works properly; Low: Off, small current |
| GPIO0(FLASH) | General purpose IO, If low while reset/power on takes chip into serial programming mode |
| GPIO1(TX) | General purpose IO and Serial TXd |
| GPIO3(RX) | General purpose IO and Serial RXd |
| GPIO4 | General purpose IO |
| GPIO5 | General purpose IO |
| GPIO12 | General purpose IO |
| GPIO13 | General purpose IO |
| GPIO14 | General purpose IO |
| GPIO15(HSPI\_CS) | General purpose IO, Connect this pin to ground through 1KOhm resistor to boot from internal flash. |

Tab 4.10 pin description of IOT module

**4.7 LCD DISPLAY**

LCD display is used for displaying the continually monitored values from the sensors. All the sensor values will be displayed on the LCD. Based on that user can observe all the temperature, humidity and light intensity values. Without user involvement automatically controlling actions were performed whenever the present greenhouse values exceeds the user predefined values. In this project we use 16\*2 LCD display. Based on our requirement we can select any type of LCDs.

LCDs are of two types:

1. Dynamic scattering type
2. II. Field effect type

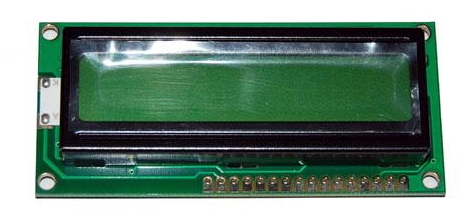
****

Fig 4.11 Dig. of LCD display

**4.7.1 PIN DESCRIPTION OF LCD**

|  |  |  |
| --- | --- | --- |
| **PIN NO** | **SYMBOL** | **FUNCTION** |
| 1 | Vss | Ground terminal of Module |
| 2 | Vdd | Supply terminal of Module, +  5v |
| 3 | Vo | Power supply for liquid crystal drive |
| 4 | RS | Register select  RS=0…Instruction register  RS=1…Data register |
| 5 | R/W | Read/Write  R/W=1…Read  R/W=0…Write |
| 6 | EN | Enable |
| 7-14 | DB0-DB7 | Bi-directional Data Bus.  Data Transfer is performed once ,thru DB0-DB7,incase of interface data length is 8-bits;and twice, thru DB4-DB7 in the case of interface data length is 4-bits.Upper four bits first then lower four bits. |
| 15 | LAMP-(L-) | LED or EL lamp power supply terminals |
| 16 | LAMP+(L+)  (E2) | Enable |

Tab 4.12 pin description of lcd

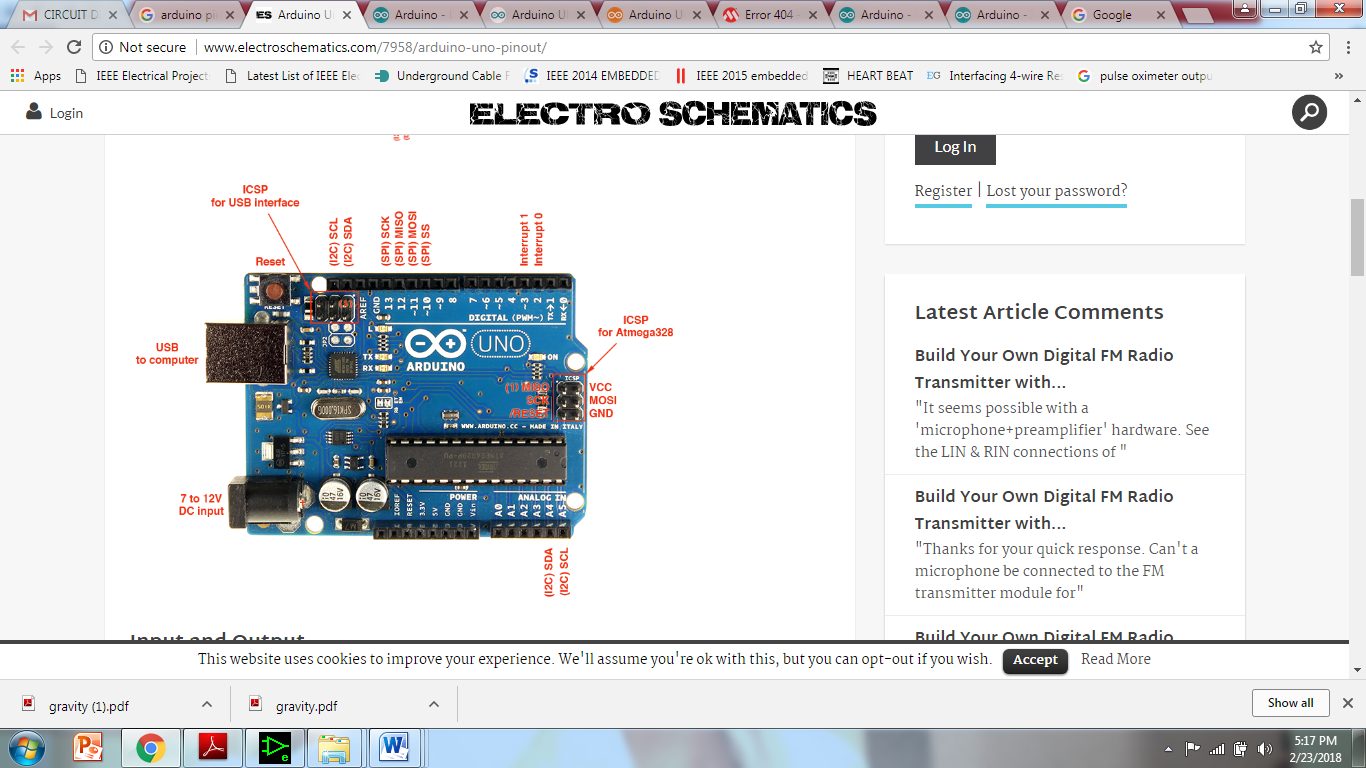
**4.7.2 LCD INTERFACING WITH MICROCONTROLLER:**



Fig 4.13 Circuit Diagram. of LCD interfacing with microcontroller

**4.8 ARDUINO**

Arduino is [open-source hardware](https://en.wikipedia.org/wiki/Open-source_hardware). The hardware reference designs are distributed under a [Creative Commons](https://en.wikipedia.org/wiki/Creative_Commons) Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available. The source code for the IDE is released under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), version 2.[[8]](https://en.wikipedia.org/wiki/Arduino#cite_note-8) Nevertheless, an official [Bill of Materials](https://en.wikipedia.org/wiki/Bill_of_Materials) of Arduino boards has never been released by Arduino staff.

Fig 4.14 Arduino

**4.8.1 SPECIFICATION**

Microcontroller ATmega328

Operating Voltage 5V Input Voltage (recommended) 7-12V

Input Voltage (limits) 6-20V

Digital I/O Pins 14 (of which 6 provide PWM output)

Analog Input Pins 6

DC Current per I/O Pin 40 mA

DC Current for 3.3V Pin 50 mA

Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader SRAM 2 KB (ATmega328) EEPROM 1 KB (ATmega328) Clock Speed 16 MHz

**4.8.2 Input and Output**

Each of the 14 digital pins on the Arduino Uno can be used as an input or output, using pin Mode(), digital Write(), and digital Read() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 k Ohms. In addition, some pins have specialized functions:  
**Serial:** pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

**External Interrupts:** pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt() function for details.

**PWM:** 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog Write() function.

**SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.

**LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it’s off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analog Reference() function. Additionally, some pins have specialized functionality:

**TWI:** A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

There are a couple of other pins on the board:

**AREF.** Reference voltage for the analog inputs. Used with analog Reference().

**Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

### **4.9 DIGITAL PINS**

In addition to the specific functions listed below, the digital pins on an Arduino board can be used for general purpose input and output via the [pin Mode()](https://www.arduino.cc/en/Reference/PinMode), [digital Read()](https://www.arduino.cc/en/Reference/DigitalRead), and [digital Write()](https://www.arduino.cc/en/Reference/DigitalWrite) commands. Each pin has an internal pull-up resistor which can be turned on and off using digital Write() (w/ a value of HIGH or LOW, respectively) when the pin is configured as an input. The maximum current per pin is 40 mA.

**Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. On the Arduino Diecimila, these pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip. On the Arduino BT, they are connected to the corresponding pins of the WT11 Bluetooth module. On the Arduino Mini and Lily Pad Arduino, they are intended for use with an external TTL serial module (e.g. the Mini-USB Adapter).

**External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attach Interrupt ()](https://www.arduino.cc/en/Reference/AttachInterrupt) function for details.

**PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analog Write()](https://www.arduino.cc/en/Reference/AnalogWrite) function. On boards with an ATmega8, PWM output is available only on pins 9, 10, and 11.

**BT Reset: 7.** (Arduino BT-only) Connected to the reset line of the bluetooth module.

**SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

**LED: 13.** On the Diecimila and LilyPad, there is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

### **4.9.1 Analog Pins**

In addition to the specific functions listed below, the analog input pins support 10-bit analog-to-digital conversion (ADC) using the [analogRead()](https://www.arduino.cc/en/Reference/AnalogRead) function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19. Analog inputs 6 and 7 (present on the Mini and BT) cannot be used as digital pins.

**I2C: 4 (SDA) and 5 (SCL).** Support I2C (TWI) communication using the [Wire library](http://wiring.org.co/reference/libraries/Wire/index.html) (documentation on the Wiring website).

### **4.9.2 Power Pins**

**VIN** (sometimes labelled "9V"). The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. Note that different boards accept different input voltages ranges, please see the [documentation for your board](http://www.arduino.cc/en/Main/Hardware). Also note that the LilyPad has no VIN pin and accepts only a regulated input.

**5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

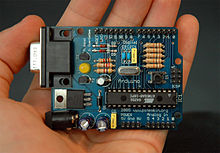
**3V3.** (Diecimila-only) A 3.3 volt supply generated by the on-board FTDI chip.

**4.9.3 GND. Ground pins.**

### **Other Pins**

* **AREF.** Reference voltage for the analog inputs. Used with [analog Reference](https://www.arduino.cc/en/Reference/AnalogReference)().
* **Reset.** (Diecimila-only) Bring this line LOW to reset the microcontroller.

Typically used to add a reset button to shields which block the one on the board. Although the hardware and software designs are freely available under [copyleft](https://en.wikipedia.org/wiki/Copyleft) licenses, the developers have requested that the name Arduino be [exclusive to the official product](https://en.wikipedia.org/wiki/Generic_trademark) and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in -duino.

[](https://en.wikipedia.org/wiki/File:Arduino316.jpg)

An early Arduino board[[11]](https://en.wikipedia.org/wiki/Arduino#cite_note-11) with an [RS-232](https://en.wikipedia.org/wiki/RS-232) [serial](https://en.wikipedia.org/wiki/Serial_communication) interface (upper left) and an Atmel ATmega8 microcontroller chip (black, lower right); the 14 digital I/O pins are at the top, the 6 analog input pins at the lower right, and the power connector at the lower left.

Most Arduino boards consist of an [Atmel](https://en.wikipedia.org/wiki/Atmel) 8-bit AVR [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) (ATmega8, ATmega168, [ATmega328](https://en.wikipedia.org/wiki/ATmega328), ATmega1280, ATmega2560) with varying amounts of flash memory, pins, and features[[12]](https://en.wikipedia.org/wiki/Arduino#cite_note-12). The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012[[13]](https://en.wikipedia.org/wiki/Arduino#cite_note-13). The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed shields. Multiple, and possibly stacked shields may be individually addressable via an [I²C](https://en.wikipedia.org/wiki/I%C2%B2C) [serial bus](https://en.wikipedia.org/wiki/Serial_bus). Most boards include a 5 V [linear regulator](https://en.wikipedia.org/wiki/Linear_regulator) and a 16 MHz [crystal oscillator](https://en.wikipedia.org/wiki/Crystal_oscillator) or [ceramic resonator](https://en.wikipedia.org/wiki/Ceramic_resonator). Some designs, such as the LilyPad, run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions. Arduino microcontrollers are pre-programmed with a [boot loader](https://en.wikipedia.org/wiki/Boot_loader) that simplifies uploading of programs to the on-chip [flash memory](https://en.wikipedia.org/wiki/Flash_memory). The default bootloader of the Aduino UNO is the optiboot bootloader.[[14]](https://en.wikipedia.org/wiki/Arduino#cite_note-14) Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between [RS-232](https://en.wikipedia.org/wiki/RS-232) logic levels and [transistor–transistor logic](https://en.wikipedia.org/wiki/Transistor%E2%80%93transistor_logic) (TTL) level signals. Current Arduino boards are programmed via [Universal Serial Bus](https://en.wikipedia.org/wiki/Universal_Serial_Bus) (USB), implemented using USB-to-serial adapter chips such as the [FTDI](https://en.wikipedia.org/wiki/FTDI) FT232. Some boards, such as later-model Uno boards, substitute the [FTDI](https://en.wikipedia.org/wiki/FTDI) chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, [Bluetooth](https://en.wikipedia.org/wiki/Bluetooth) or other methods, when used with traditional microcontroller tools instead of the Arduino IDE, standard AVR [in-system programming](https://en.wikipedia.org/wiki/In-system_programming) (ISP) programming is used. The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila, Duemilanove, and current Uno[[](https://en.wikipedia.org/wiki/Arduino#cite_note-N1-17) provide 14 digital I/O pins, six of which can produce [pulse-width modulated](https://en.wikipedia.org/wiki/Pulse-width_modulation) 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, via female 0.1-inch (2.54 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board[[15]](https://en.wikipedia.org/wiki/Arduino#cite_note-18) and Boarduino[[16]](https://en.wikipedia.org/wiki/Arduino" \l "cite_note-19) boards may provide male header pins on the underside of the board that can plug into solderless [breadboards](https://en.wikipedia.org/wiki/Breadboard). Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots. Others are electrically equivalent but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use different processors, of varying compatibility.

**4.10 TEMPERATURE SENSOR AND HUMIDITY SENSOR**

### **4.10.1 Pin Identification and Configuration:**

|  |  |  |
| --- | --- | --- |
| **No:** | **Pin Name** | **Description** |
| **For DHT11 Sensor** | | |
| 1 | Vcc | Power supply 3.5V to 5.5V |
| 2 | Data | Outputs both Temperature and Humidity through serial Data |
| 3 | NC | No Connection and hence not used |
| 4 | Ground | Connected to the ground of the circuit |
| **For DHT11 Sensor module** | | |
| 1 | Vcc | Power supply 3.5V to 5.5V |
| 2 | Data | Outputs both Temperature and Humidity through serial Data |
| 3 | Ground | Connected to the ground of the circuit |



Fig 4.15 DHT11 Sensor

**DHT11 Specifications:**

* Operating Voltage: 3.5V to 5.5V
* Operating current: 0.3mA (measuring) 60uA (standby)
* Output: Serial data
* Temperature Range: 0°C to 50°C
* Humidity Range: 20% to 90%
* Resolution: Temperature and Humidity both are 16-bit
* Accuracy: ±1°C and ±1%



DHT11–Temperature and Humidity Sensor DHT11 Sensor Pinout

**Difference between DHT11 Sensor and module**:

The **DHT11 sensor** can either be purchased as a sensor or as a module. Either way, the performance of the sensor is same. The sensor will come as a 4-pin package out of which only three pins will be used whereas the module will come with three pins as shown above. The only difference between the sensor and module is that the module will have a filtering capacitor and pull-up resistor inbuilt, and for the sensor, you have to use them externally if required.

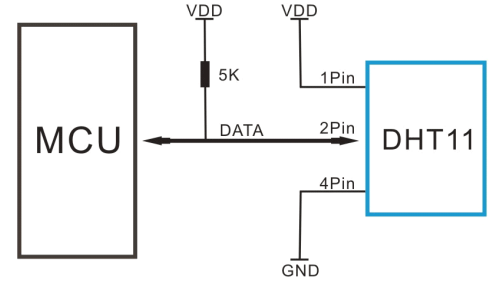
**Where to use DHT11:**

The **DHT11**is a commonly used **Temperature and humidity sensor.** The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers.

The sensor can measure temperature from 0°C to 50°C and humidity from 20% to 90% with an accuracy of ±1°C and ±1%. So if you are looking to measure in this range then this sensor might be the right choice for you.

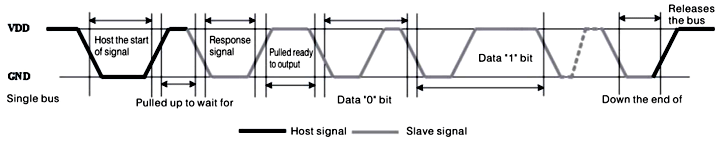
**How to use DHT11 Sensor:**

The DHT11 Sensor is factory calibrated and outputs serial data and hence it is highly easy to set it up. The connection diagram for this sensor is shown below.



As you can see the data pin is connected to an I/O pin of the MCU and a 5K pull-up resistor is used. This data pin outputs the value of both temperature and humidity as serial data. If you are trying to interface DHT11 with Arduino then there are ready-made libraries for it which will give you a quick start.

If you are trying to interface it with some other MCU then the datasheet given below will come in handy. The output given out by the data pin will be in the order of 8bit humidity integer data + 8bit the Humidity decimal data +8 bit temperature integer data + 8bit fractional temperature data +8 bit parity bit. To request the DHT11 module to send these data the I/O pin has to be momentarily made low and then held high as shown in the timing diagram below



The duration of each host signal is explained in the DHT11 datasheet, with neat steps and illustrative timing diagrams

**Applications:**

* Measure temperature and humidity
* Local Weather station
* Automatic climate control
* Environment monitoring

**2D–model of the sensor:**

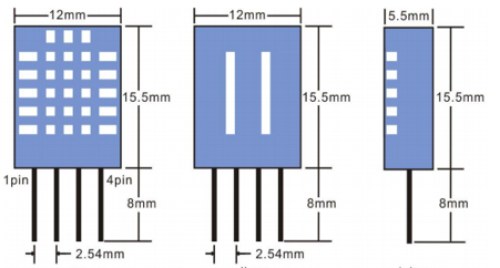
****

Fig 4.16 2D model of the sensor

**4.11 WIND SENSOR**

A wind sensor, also known as an anemometer, is an instrument used to measure the speed and direction of wind. It is crucial in various fields, including meteorology, environmental monitoring, aviation, and renewable energy.

Wind sensors typically come in two main types: **anemometers** for wind speed measurement and **wind vanes** for wind direction. The most common anemometer design features rotating cups or blades that are driven by the wind. The wind's force causes the cups or blades to spin, and the rotation speed is directly proportional to the wind speed. More advanced sensors use ultrasonic or laser technology to measure wind speed without moving parts, which increases durability and reduces maintenance.

Wind vanes, used in conjunction with anemometers, measure the direction of wind by utilizing a simple design with a pointer that aligns with the wind's direction. These vanes are often used in combination with the anemometer to provide a comprehensive view of the wind conditions.

The data collected by wind sensors is critical for weather forecasting, climate studies, and the optimization of wind energy turbines. For example, wind sensors on turbines help adjust their position and operation to maximize energy production based on wind speed and direction.

In addition to measuring wind conditions, modern wind sensors can communicate with weather stations, satellites, or other devices to relay real-time information. They are often equipped with features like temperature and humidity sensors to enhance their functionality.

In summary, wind sensors are essential tools for understanding wind behavior, supporting various industries, and contributing to safety, energy production, and environmental monitoring.

**CHAPTER 5**

**SOFTWARE REQUIREMENT**

**5.1 ARDUINO IDE SKETCH**

In the getting started guide ([Windows](https://www.arduino.cc/en/Guide/Windows), [Mac OS X](https://www.arduino.cc/en/Guide/MacOSX), [Linux](http://www.arduino.cc/playground/Learning/Linux)), you uploaded a sketch that blinks an LED. In this tutorial, you'll learn how each part of that sketch works.

A sketch is the name that Arduino uses for a program. It's the unit of code that is uploaded to and run on an Arduino board.

**Arduino** is an open source, computer hardware and software company, project, and user community that designs and manufactures [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as [open-source hardware](https://en.wikipedia.org/wiki/Open-source_hardware) and [software](https://en.wikipedia.org/wiki/Open-source_software), which are licensed under the [GNU Lesser General Public License](https://en.wikipedia.org/wiki/GNU_Lesser_General_Public_License) (LGPL) or the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License) (GPL),[[1]](https://en.wikipedia.org/wiki/Arduino#cite_note-1) permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as [do-it-yourself](https://en.wikipedia.org/wiki/Do-it-yourself) kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) pins that may be interfaced to various expansion boards (*shields*) and other circuits. The boards feature serial communications interfaces, including [Universal Serial Bus](https://en.wikipedia.org/wiki/Universal_Serial_Bus) (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B). In addition to using traditional compiler toolchains, the Arduino project provides an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) based on the [Processing](https://en.wikipedia.org/wiki/Processing_(programming_language)) language project. The Arduino project started in 2003 as a program for students at the [Interaction Design Institute Ivrea](https://en.wikipedia.org/wiki/Interaction_Design_Institute_Ivrea) in [Ivrea](https://en.wikipedia.org/wiki/Ivrea), Italy,[[2]](https://en.wikipedia.org/wiki/Arduino#cite_note-kushner-2) aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using [sensors](https://en.wikipedia.org/wiki/Sensor) and [actuators](https://en.wikipedia.org/wiki/Actuator). Common examples of such devices intended for beginner hobbyists include simple [robots](https://en.wikipedia.org/wiki/Robot), [thermostats](https://en.wikipedia.org/wiki/Thermostat), and [motion detectors](https://en.wikipedia.org/wiki/Motion_detector).

### **5.2 Arduino**

Arduino is an open-source electronics platform based on easy-to-use hardware and software. [Arduino boards](https://www.arduino.cc/en/Main/Products) are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the [Arduino programming language](https://www.arduino.cc/en/Reference/HomePage) (based on [Wiring](http://wiring.org.co/)), and [the Arduino Software (IDE)](https://www.arduino.cc/en/Main/Software), based on [Processing](https://processing.org/).

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of [accessible knowledge](http://forum.arduino.cc/) that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The [software](https://www.arduino.cc/en/Main/Software), too, is open-source, and it is growing through the contributions of users worldwide.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

**Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50

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

**Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

**Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

**Open source and extensible hardware** - 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. Even relatively inexperienced users can build the [breadboard version of the module](https://www.arduino.cc/en/Main/Standalone) in order to understand how it works and save money.

### **5.3 Variables**

A variable is a place for storing a piece of data. It has a name, a type, and a value. For example, the line from the Blink sketch above declares a variable with the name ledPin, the type int, and an initial value of 13. It's being used to indicate which Arduino pin the LED is connected to. Every time the name ledPin appears in the code, its value will be retrieved. In this case, the person writing the program could have chosen not to bother creating the ledPin variable and instead have simply written 13 everywhere they needed to specify a pin number. The advantage of using a variable is that it's easier to move the LED to a different pin: you only need to edit the one line that assigns the initial value to the variable.

### **5.4 Functions**

A function (otherwise known as a procedure or sub-routine) is a named piece of code that can be used from elsewhere in a sketch. For example, here's the definition of the setup() function from the Blink example:

void setup()  
{  
   pinMode(ledPin, OUTPUT);         
}

The first line provides information about the function, like its name, "setup". The text before and after the name specify its return type and parameters: these will be explained later. The code between the { and } is called the body of the function: what the function does.

**5.4.1 pinMode(), digitalWrite(), and delay()**

The pinMode() function configures a pin as either an input or an output. To use it, you pass it the number of the pin to configure and the constant INPUT or OUTPUT. When configured as an input, a pin can detect the state of a sensor like a pushbutton; As an output, it can drive an actuator like an LED.

The digitalWrite() functions outputs a value on a pin.

For example, the line:

digitalWrite(ledPin, HIGH);

The delay() causes the Arduino to wait for the specified number of milliseconds before continuing on to the next line. There are 1000 milliseconds in a second, so the line:

delay(1000);

### **5.4.2 setup() and loop()**

There are two special functions that are a part of every Arduino sketch: setup() and loop(). The setup() is called once, when the sketch starts. It's a good place to do setup tasks like setting pin modes or initializing libraries. The loop() function is called over and over and is heart of most sketches. You need to include both functions in your sketch, even if you don't need them for anything.

Everything between the /\* and \*/ is ignored by the Arduino when it runs the sketch (the \* at the start of each line is only there to make the comment look pretty, and isn't required). It's there for people reading the code: to explain what the program does, how it works, or why it's written the way it is. It's a good practice to comment your sketches, and to keep the comments up-to-date when you modify the code. This helps other people to learn from or modify your code.

**CHAPTER 6**

**ADVANTAGES AND APPLICATIONS**

**6.1 ADVANTAGES**

* The system accurately forecasts energy demand, allowing for better resource allocation and reducing energy waste, leading to significant energy savings.
* By optimizing energy consumption, the system helps reduce operational costs for industries, residential areas, and commercial buildings, resulting in lower utility bills and maintenance costs.
* The integration of real-time environmental data and machine learning provides immediate insights into energy demand, enabling quicker, data-driven decisions for energy management.
* By forecasting energy needs and promoting efficient usage, the system supports sustainability goals by reducing unnecessary energy consumption and minimizing the carbon footprint.
* The system is scalable and adaptable to various sectors, from residential to industrial, and can easily be integrated with existing energy infrastructure, making it a versatile solution for diverse energy management needs.

**6.2 APPLICATIONS**

* Smart Grid Optimization: The proposed system can be integrated with smart grids to predict energy demand and optimize distribution. By forecasting peak demand periods, it helps manage grid load, prevent outages, and reduce energy wastage.
* Industrial Energy Management: In industrial settings, the system can forecast energy needs during peak operation hours or shifts, allowing for better resource allocation, reduced operational costs, and optimized energy consumption in factories and manufacturing plants.
* Residential Energy Efficiency: The system can be applied in smart homes to optimize energy usage based on real-time environmental conditions, adjusting heating, cooling, and lighting systems for improved energy conservation and reduced utility bills.
* Renewable Energy Integration: By predicting energy demand and supply, the system can help manage renewable energy sources, such as solar or wind power, ensuring that energy production is balanced with consumption, minimizing waste, and enhancing sustainability.
* Urban and Commercial Energy Planning: For urban areas and commercial buildings, the system can predict seasonal and daily energy demand fluctuations, helping facility managers optimize HVAC systems, lighting, and other energy-intensive operations, contributing to overall energy savings.

**CHAPTER 7**

**MODULES**

**Modules**

1. Data Ingestion Module
2. Data Preprocessing & Cleaning Module
3. Feature Engineering Module
4. Forecasting Engine (ML Model Module)
5. Explainability & Interpretation Module
6. Multi-Horizon Forecasting Module
7. Demand Response & Optimization Module
8. Visualization & Dashboard Module

**Data Ingestion Module**

The Data Ingestion Module is responsible for connecting to various data sources such as smart meters, IoT devices, weather APIs, and energy databases. It efficiently handles both batch processing and real-time streaming data collection to ensure the system is always updated with the latest information.

**Data Preprocessing & Cleaning Module**

The Data Preprocessing & Cleaning Module ensures that all incoming data is clean, consistent, and ready for analysis. This module deals with missing values, noisy entries, and aligns time series data while performing scaling, encoding, and detecting outliers to maintain data quality.

**Engineering Module**

The Feature Engineering Module plays a crucial role in enhancing model performance by generating useful features like lag variables, moving averages, weather-related factors, and calendar events. It also incorporates geographical and behavioral data, making the features more meaningful and rich.

**Feature Engineering Module**

At the core of the system, the Forecasting Engine (ML Model Module) hosts machine learning models such as Random Forest and XGBoost, as well as deep learning models like LSTM and GRU. This module handles the complete cycle of model training, validation, and testing to deliver accurate energy demand forecasts.

**Explainability & Interpretation Module**

The Explainability & Interpretation Module provides transparency by using tools like SHAP and LIME to explain the reasoning behind the model’s predictions. It generates human-readable interpretations, helping stakeholders understand and trust the forecasts.

**Multi-Horizon Forecasting Module**

The Multi-Horizon Forecasting Module adds flexibility by supporting forecasts over multiple time horizons such as hourly, daily, weekly, and monthly. It also enables forecasts at varying granularities, from city-wide predictions to neighbourhood-level insights.

**Demand Response & Optimization Module**

The Demand Response & Optimization Module integrates with grid management systems to utilize forecast outputs. It recommends demand response actions, helping utilities balance loads during peak periods and avoid potential blackouts.

**Visualization & Dashboard Module**

Finally, the Visualization & Dashboard Module presents the forecasts and historical trends through interactive charts and graphs. It also visualizes error metrics like MAPE and RMSE, allowing users to monitor model performance and track energy demand effectively.

**CHAPTER 8**

**RESULTS AND CONCLUSION**

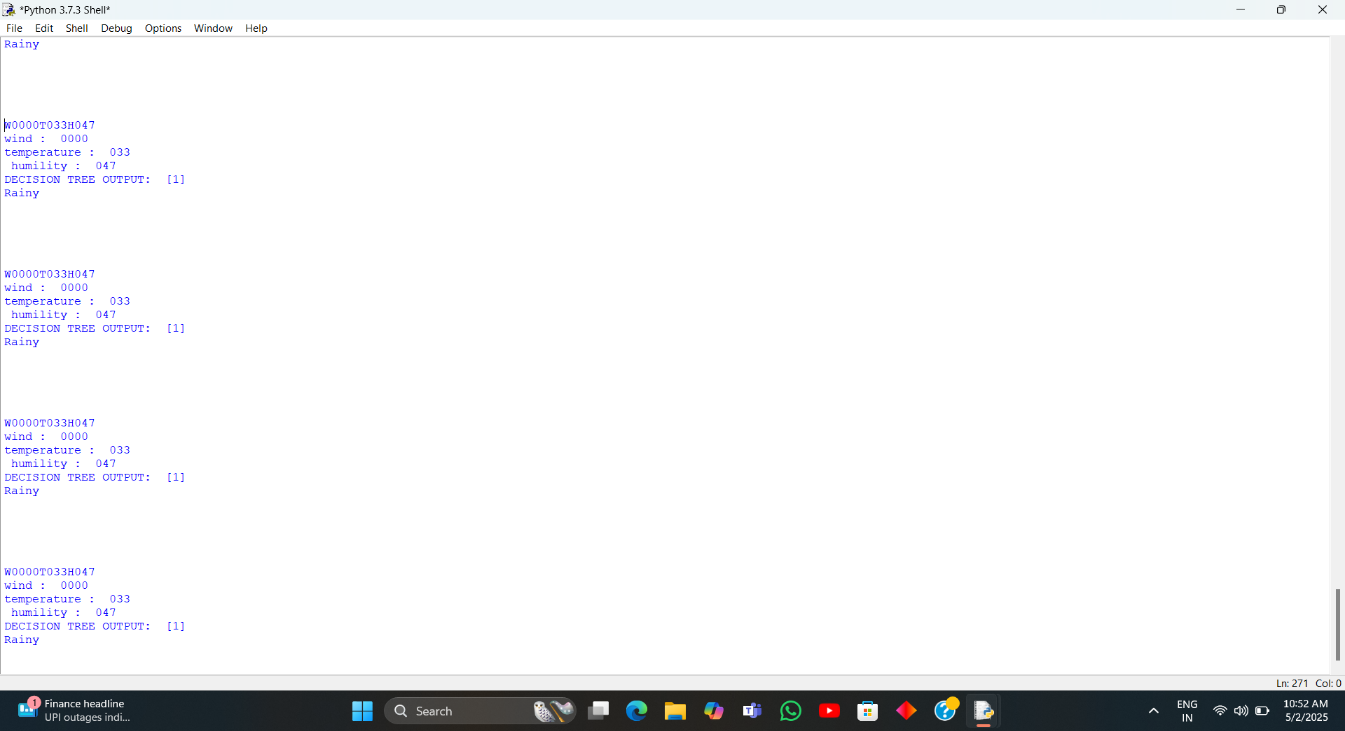
### **7.1 RESULTS AND DISCUSSION:**

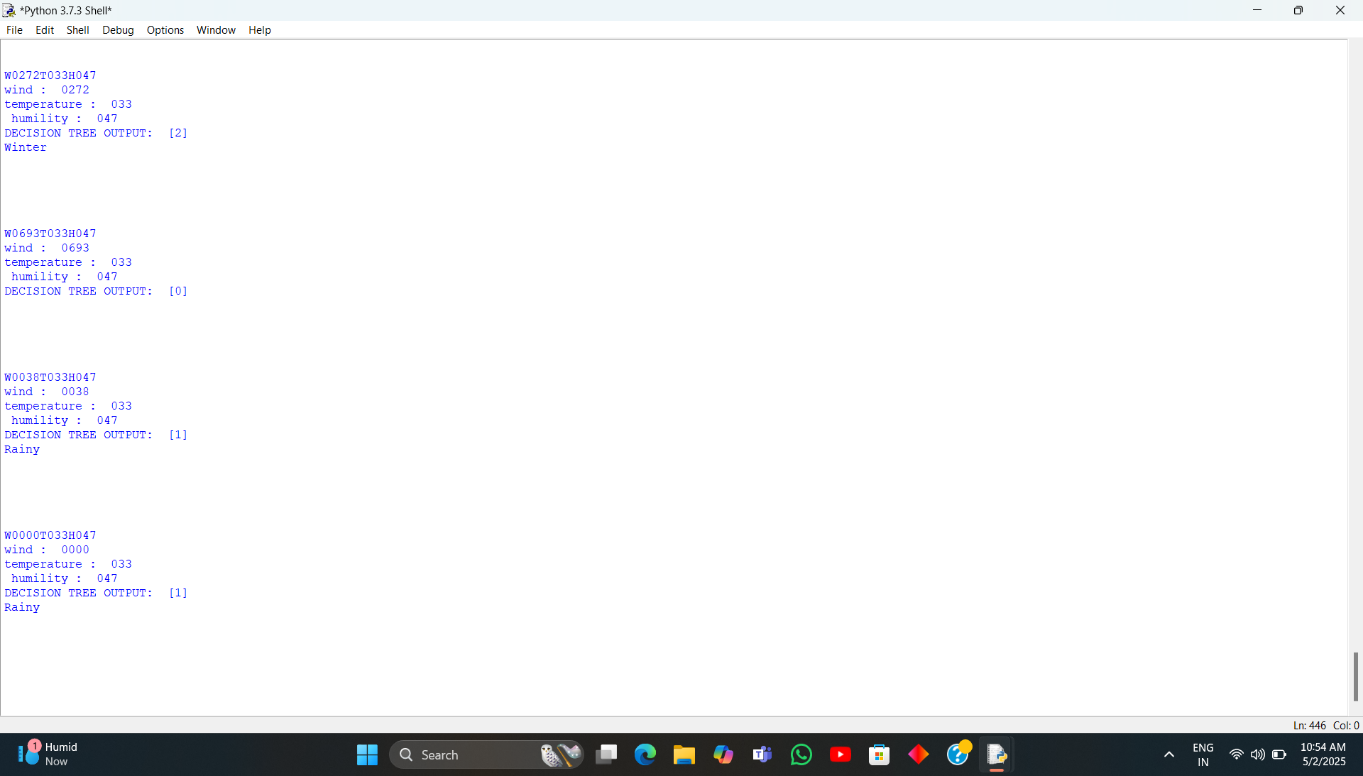
The proposed machine learning-based energy demand forecasting system showed promising results in accurately predicting energy consumption across different sectors, including residential, industrial, and commercial areas. By integrating real-time environmental data such as temperature, humidity, and wind speed, the system was able to provide dynamic forecasts that adapted to changing conditions. During testing, the system accurately predicted energy demand during peak hours, such as hot summer days when cooling demand increases, and during cold winters when heating demand is high. This was achieved through the system’s ability to identify patterns in historical usage data and environmental factors, improving prediction accuracy compared to traditional energy management systems.

Furthermore, the system demonstrated the potential for optimizing resource allocation. By forecasting peak demand periods, energy managers could proactively adjust energy distribution to prevent overloading and reduce the likelihood of power outages. The real-time data provided by the system also helped identify opportunities for energy savings, particularly during off-peak hours when energy consumption was lower. This proactive approach to energy management proved to be more efficient than conventional reactive methods.

The final output of we are getting the data by integrating real-time environmental data such as temperature, humidity, and wind speed. It will show three type of data, they are rainy, winter and summer season data. The ranges of these factors are given in the following.

* 0 to 300 - Rainy Season
* 300 to 500 – Winter Season
* 500 to above – Summer Season





**7.2 CONCLUSION**

The proposed system represents a significant advancement in the field of energy demand forecasting by intelligently integrating machine learning algorithms, real-time environmental data, and Internet of Things (IoT) technologies. This combination allows the system to analyse complex patterns of energy consumption with greater precision than traditional forecasting methods.

Traditional models often rely heavily on historical usage data alone, which can limit their adaptability to sudden changes such as unexpected weather shifts or unusual consumption patterns. In contrast, the proposed system dynamically incorporates live environmental factors—including temperature, humidity, and wind speed—captured through IoT sensors, enabling it to make accurate, context-aware predictions about future energy needs.

One of the standout advantages of this approach is its enhanced accuracy. By using machine learning, the system continually learns from new incoming data, refining its predictions over time and significantly reducing errors compared to static, rule-based methods. Furthermore, because predictions are updated in real-time, energy providers can respond more quickly to changes in demand, preventing wastage during low-usage periods and ensuring adequate supply during peak times.

The system’s cost-effectiveness is another important advantage. By optimizing energy generation and distribution based on precise demand forecasts, utility companies and industries can lower operational costs, reduce energy losses, and delay or avoid expensive infrastructure upgrades. Additionally, better resource allocation ensures that the right amount of energy is produced and distributed where and when it is needed, supporting grid stability and preventing blackouts.

Finally, the system empowers energy managers by providing them with detailed insights, analytics dashboards, and decision-support tools. With access to timely, accurate forecasts, managers can make proactive, informed decisions regarding energy purchasing, load balancing, maintenance scheduling, and more—ultimately building a more sustainable, reliable, and resilient energy future for communities and industries alike.

**APPENDIX 1**

**SAMPLE CODE**

import pandas as pd #for reading dataset

import numpy as np # array handling functions

from time import sleepdataset = pd.read\_csv("Book1.csv")#reading dataset

x = dataset.iloc[:,:-1].values #locating inputs

y = dataset.iloc[:,-1].values #locating outputs

from sklearn.preprocessing import LabelEncoder

labelencoder\_y = LabelEncoder()

y= labelencoder\_y.fit\_transform(y)

#printing X and Y

print("x=",x)

print("y=",y)

from sklearn.model\_selection import train\_test\_split # for splitting dataset

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x ,y, test\_size = 0.25 ,random\_state = 0)

#printing the spliited dataset

print("x\_train =",x\_train)

print("x\_test=",x\_test)

print("y\_train=",y\_train)

print("y\_test=",y\_test)

#importing algorithm

from sklearn.tree import DecisionTreeClassifier

classifier=DecisionTreeClassifier()

classifier.fit(x\_train,y\_train)#trainig Algorithm

y\_pred=classifier.predict(x\_test) #testing model

print("y\_pred",y\_pred) # predicted output

try:

import serial

ser = serial.Serial('COM4',baudrate=9600,timeout=0.3)

ser.flushInput()

A=1

B=1

while True:

a=ser.readline().decode('ascii') # reading serial data

print(a)

b=a

for letter in b:

if(letter =='W'):

D1 =b[1]+b[2]+b[3]+b[4]

print("wind : ",D1)

a1 =int(D1)

if(letter =='T'):

D2 =b[6]+b[7]+b[8]

print("temperature : ",D2)

a2 =int(D2)

if(letter =='H'):

D3 =b[10]+b[11]+b[12]

print(" humility : ",D3)

a3 =int(D3)

##PREDICTED OUTPUT

OUTPUT = classifier.predict([[a1,a2,a3]])

print('DECISION TREE OUTPUT: ',OUTPUT)

if OUTPUT ==1:

print("Rainy")

elif OUTPUT==2:

print("Winter")

elif OUTPUT==3:

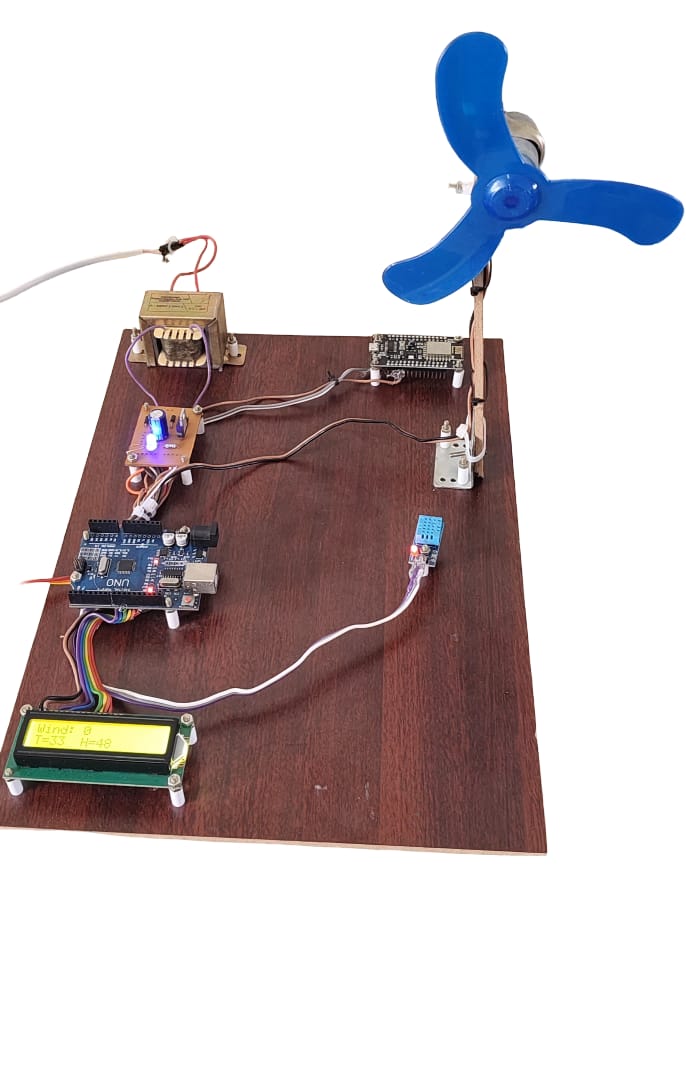
print("Summer")

except Exception as e:

print(e)

**APPENDIX 2**

**PROJECT PHOTOGRAPH**



**CHAPTER 8**

**REFERENCES**

1. A Machine Learning Model for Occupancy Rates and Demand Forecasting in the Hospitality Industry William Caicedo-Torres and Fabi´an Payares Department of Computer Science - Universidad Tecnol´ogica de Bol´ıvar,Parque Industrial Tecnol´ogico Carlos V´elez Pombo, Km 1 V´ıa Turbaco, Cartagena,Colombia [wcaicedo@unitecnologica.edu.co](mailto:wcaicedo@unitecnologica.edu.co).
2. ENHANCING ENERGY EFFICIENCY WITH AI: A REVIEW OF MACHINE LEARNING MODELS IN ELECTRICITY DEMAND FORECASTING Adebayo Olusegun Aderibigbe1, Emmanuel Chigozie Ani2, Peter Efosa Ohenhen, Nzubechukwu Chukwudum Ohalete4, & Donald Obinna Daraojimba5 1Data Scientists Network (DSN), Lagos, Nigeria2Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, USA3Department of Mechanical Engineering, University of Nebraska-Lincoln, USA 4Kennesaw State University, GA, USA 5Ahmadu Bello University, Zaria, Nigeria
3. Machine Learning for Demand Forecasting in Manufacturing Sai Mani Krishna Sistla1, Gowrisankar Krishnamoorthy. Jawaharbabu Jeyaraman3, Bhargav Kumar Konidena41Soothsayer Analytics, USA2HCL America, USA3TransUnion, USA 4 State Farm, USA.
4. Machine Learning Forecasting of Daily Delivery Positions: A Modern Take on Industrial Workforce Planning Lukas Hans1, Patrick Eichenseer2, and Matthias Groß1Institute for Data Science, Engineering, and Analytics, TH Köln Steinmüllerallee 1, 51643 Gummers batch, Germany E-Mails: lukas.hans@th-koeln.de, matthias.gross2@th-koeln.de2Head of Operational Excellence, DEHN SE Umelsdorfer Str. 8, 92280 Utzenhofen, GermanyE-Mail: patrick.eichenseer@dehn.de
5. An overview of energy demand forecasting methods published in 2005-2015Iman Ghalehkhondabi", Ehsan Ardjmand¹, Gary R. Weckman', and William A. Young II Department of Industrial and Systems Engineering, Russ College of Engineering and Technology, Ohio University, Athens, OH 45701, USA Management, College of Business, Ohio University, Athens, OH 45701, USAE-mails: ig060113@ohio.edu ("Corresponding), e.ardjmand@gmail.com, weckmang@ohio.edu,youngw1 @ohio.edu.
6. Energy Demand Forecasting for Türkiye: Comparison Between Traditional Machine Learning Algorithms and Ensemble Learning Algorithms Ahmet Tezcan Tekih, Cem Sar Department of Management Engineering. Istanbul Technical University, Türkiye Department of Industrial Engineering. Istanbul Technical University. Türkiyetekina@itu.edu.tr. [saric21@itu.edu.tr](mailto:saric21@itu.edu.tr).
7. An Intelligent Hybrid Machine Learning Model for Sustainable Forecasting of Home Energy Demand and Electricity PriceBanafshe Parizad, HdssanRanjbarzadeh, Alijamali1.30 and Hamid Khayyam 1,School of Engineering, RMIT University, Melbourne 3000, Australia: 54023374@student.mit.edu.au (B.P alijamali@knu.ac.kr (A1)2 School of Engineering, Deakin University, Geelong 3217, Australia; [hranjbar@deakin.edu.au](mailto:hranjbar@deakin.edu.au) Department of Artificial Intelligence, School of Electronics Engineering, Kyungpook National University, Daegu 372 37224, Republic of Korna Correspondenor [hamid.khayyam@rmit.edu.au](mailto:hamid.khayyam@rmit.edu.au).
8. Machine Learning-Driven Demand Forecasting: A Comparative Analysis of Advanced Techniques and Real-Time Integration Satish Anchuri Walmart, USA.
9. Energy Demand Forecasting for Hybrid Microgrid Systems Using Machine Learning ModelsTahir Aja Zarma", Emmanuel Ali2, Ahmadu Adamu Galadimal, Tologon Karataev1, Suleiman Usman Hussein1.3. Adekunle Akanni Adeleke 4Department of Electrical Electronics Engineering, Nile University of Nigeria, Abuja, Nigeria Department of Computer Engineering, Nile University of Nigeria, Abuja, Nigeria "National Space Research and Development Agency, Abuja, Nigeria "Department of Mechanical Engineering, Nile University of Nigeria, Abuja, NigeriaReceived 05 August 2024, received in revised form 07 October 2014; accepted 11 October 2024DOI: https://doi.org/10.46604/peti2024,14098 .
10. Short-term water demand forecasting using data centric machine learning approachesGuoxuan Liu, Dragan Savic Da,b,c and Guangtao Fua,"Exeter, Enter Ex140F, U"Corresponding author. E-mail gheracik01. 0000-0001-4833-8352:06. 0000-0003-456-7-9041