CHAPTER 1

INTRODUCTION

Animal husbandry encompasses the comprehensive management and nurturing of domesticated animals, including cattle, pigs, chickens, and sheep, with the primary goal of yielding food, fiber, and various other products essential for human consumption and industrial purposes. It entails a diverse array of practices ranging from breeding, feeding, and housing to healthcare interventions, all meticulously orchestrated to enhance the productivity, well-being, and welfare of the animals involved. Within the realm of animal husbandry, the meticulous care of dairy cattle, in particular, holds paramount importance. Recognized as the most proficient source of protein among all farm livestock, dairy cattle demand meticulous attention to ensure optimal condition, thereby guaranteeing the superior quality of dairy products like milk. Central to this endeavor is the vigilant monitoring of cattle health, a multifaceted endeavor encompassing regular check-ups, preventive measures such as vaccination protocols, nuanced adjustments in feed and nutrition, and timely interventions to address any emerging health concerns. The significance of cattle health monitoring transcends mere productivity concerns; it serves as a cornerstone for safeguarding the welfare of the animals under human stewardship. Timely identification and treatment of health issues not only mitigate suffering but also forestall the onset and spread of diseases and parasites within the herd, safeguarding the livelihoods of farmers and the sustainability of dairy operations. In essence, the holistic approach to healthcare in dairy cattle husbandry underscores a symbiotic relationship between human stewardship, animal welfare, and agricultural sustainability, epitomizing the intricate balance inherent in the practice of animal husbandry.

1.1 MOTIVATION FOR WORK

The integration of AI and IoT technologies into livestock health monitoring represents a significant advancement in modern agriculture, aligning with broader societal shifts towards more sustainable and ethical food production systems. By harnessing the power of data analytics and real-time monitoring capabilities, this innovative approach offers unprecedented insights into the health and well-being of farm animals, enabling proactive interventions and preventive measures.Furthermore, the adoption of such cutting-edge systems not only enhances animal welfare but also strengthens the economic viability of livestock operations. The early detection of health issues and disease outbreaks can significantly reduce veterinary expenses, minimize productivity losses, and safeguard market reputation, thereby bolstering the resilience of farmers against unforeseen challenges.Moreover, the societal benefits of smart livestock health monitoring extend beyond the confines of individual farms. In a world grappling with food security concerns and environmental degradation, the optimization of livestock productivity assumes heightened significance. Livestock farming plays a pivotal role in meeting the growing global demand for animal products, and ensuring the health and well-being of livestock is essential for maintaining a stable food supply chain.Additionally, by embracing sustainable farming practices facilitated by AI and IoT technologies, such as precision agriculture and resource optimization, the system contributes to mitigating the environmental impact of agriculture.In summary, the development and deployment of AI and IoT based smart livestock health monitoring systems signify a paradigm shift in agricultural practices, embodying a harmonious convergence of ethical,economic, and environmental imperatives. By prioritizing the welfare of farm animals, optimizing productivity, and advancing sustainable farming practices, these technologies pave the way for a more resilient, humane, and environmentally conscious agricultural future.

1.2 PROBLEM STATEMENT

Livestock farming operates within a dynamic landscape fraught with challenges, where ensuring the health and welfare of animals stands as a paramount concern alongside optimizing production efficiency and minimizing environmental footprints. Yet, the conventional approaches to monitoring and managing livestock often rely on manual observations, a process riddled with labor-intensive tasks, time-consuming efforts, and susceptibility to human errors. Compounding these challenges are external variables such as erratic weather patterns, sudden disease outbreaks, and environmental stressors, which wield significant influence over the well-being and productivity of livestock, demanding prompt interventions and astute management strategies. In light of these complexities, the call for innovative solutions reverberates louder than ever, urging the integration of cutting-edge technologies to augment the care and productivity of livestock. These solutions must transcend traditional boundaries, offering real-time monitoring capabilities that encompass a holistic spectrum of parameters from nuanced animal behaviors and vital health indicators to prevailing environmental conditions and intricate feed consumption patterns. Moreover, they should wield the power of predictive analytics, harnessing data-driven insights to preemptively identify and mitigate potential health risks, optimize feeding regimens for maximal efficiency, and refine overall farm management practices to unprecedented levels of precision and efficacy. Through the fusion of advanced technologies and visionary approaches, these solutions hold the promise of revolutionizing the livestock farming paradigm, fostering sustainability, resilience, and elevated standards of animal well-being.

1.3 OBJECTIVES

1. Develop a robust infrastructure.
2. Design and implement a reliable IoT system to collect real-time data from sensors attached to livestock, including vital signs such as temperature, heart rate, and activity level.
3. implement Machine Learning Algorithms: Utilize machine learning techniques to analyze the collected data and build predictive models that can detect early signs of health issues in livestock, such as diseases or stress.
4. Real-time Monitoring and Alerting: Create a user-friendly dashboard or mobile application that provides real-time monitoring of livestock health parameters.
5. Implement an alerting system to notify farmers or caretakers of any abnormal health conditions detected by the machine learning models.
6. Improve Livestock Management: Develop recommendations and insights based on the collected data to help farmers make informed decisions regarding the well-being and management of their livestock.

CHAPTER 2
LITERATURE SURVEY

2.1 LITERATURE SURVEY

A considerable volume of research has delved into AI and IoT based smart livestock health monitoring systems, reflecting an emerging trend in precision agriculture and animal welfare management. Numerous studies have explored the integration of advanced technologies to enhance livestock health monitoring, leveraging AI and IoT for real-time data collection, analysis, and decision support.

● In a study by [1], Brown and Davis (2021) investigated the use of IoT and AI technologies for early disease detection in cattle. The study explores how AI algorithms can be applied to analyze sensor data and identify patterns that indicate emerging health issues. This approach offers a powerful tool for early disease detection, allowing farmers to take preventive measures before health problems become critical. The researchers highlight the benefits of integrating AI with IoT data, as AI's predictive capabilities can enhance the accuracy of disease detection. By analyzing large volumes of data, AI algorithms can identify subtle changes in cattle behavior or health parameters, providing valuable insights for farmers. The study also discusses the potential for AI-based systems to automate certain tasks, such as data analysis and alert generation, reducing the workload on farmers. This automation allows farmers to focus on other critical aspects of farm management, improving overall efficiency. The researchers suggest that this approach can lead to improved livestock health and reduced veterinary costs by enabling early intervention and disease prevention.

● In a study by [2], Chatterjee et al. (2021) proposed "LiveCare," an IoT based healthcare framework for livestock in smart agriculture. This framework integrates various IoT technologies such as sensors, wireless communication, and cloud-based data storage to create a real-time monitoring system for cattle health. By collecting and analyzing data from multiple sensors, LiveCare enables farmers to monitor critical health parameters such as temperature, heart rate, and activity levels. The system's real-time capabilities allow farmers to take immediate action when abnormalities are detected, reducing the risk of severe health issues or loss of livestock. LiveCare also provides a user-friendly interface, enabling farmers to access data from their smartphones or other mobile devices, ensuring they can stay informed regardless of their location. This IoT framework is highly adaptable and can be tailored to different farm sizes and types, making it a versatile solution for smart agriculture. The researchers emphasize that this framework can contribute to improved animal welfare, reduced operational costs, and enhanced livestock productivity.

● In a study by [3], Chaudhry et al. (2020) introduced a comprehensive system that combines IoT and Machine Learning (ML) for livestock monitoring. This innovative approach uses a network of IoT sensors to collect data on cattle behavior, health, and environmental conditions. The data is then processed through ML algorithms to detect irregularities and predict potential health issues. The system is designed to learn from patterns and adapt to changing conditions, making it more effective at identifying early signs of disease or distress. By implementing this system, farmers can receive alerts and insights that help them take proactive measures to prevent disease outbreaks and other health-related problems. The integration of ML also enables the system to improve its accuracy over time, offering more reliable predictions. The authors highlight that this approach can lead to significant improvements in livestock management, reducing costs associated with reactive treatments and minimizing animal suffering. Additionally, the system's scalability allows it to be implemented in various agricultural settings, from small family farms to large commercial operations.

● In a study by [4], Cho (2021) presented an analysis and design of a cattle management system based on IoT. This system is designed to improve the efficiency and accuracy of cattle management by leveraging IoT technology to collect and analyze data on cattle health and behavior. The system incorporates various sensors to monitor vital health parameters, activity levels, and environment. The collected data is transmitted wirelessly to a central system, where it is processed and analyzed to detect potential health issues. Cho's design emphasizes scalability, allowing the system to be implemented in various farm environments, The study also explores the challenges involved in designing IoT based systems for cattle management, such as connectivity issues and data security. To address these challenges, the system uses robust communication protocols and encryption to ensure data integrity and privacy. Additionally, the system is designed with user-friendly interfaces, enabling farmers to interact with the system and access information with ease. This comprehensive approach to cattle management provides farmers with the tools they need to improve livestock health and productivity, contributing to more sustainable agricultural practices.

● In a study by [5], Darvesh et al. (2023) introduced a system that leverages both IoT and AI for smart cattle health monitoring. The system is designed to collect a range of health data from cattle, such as body temperature, activity levels, and other physiological parameters, using IoT based sensors. This data is then processed through AI algorithms to identify patterns and detect anomalies that could indicate health problems. The system's AI component enables it to learn from the data and improve its accuracy over time, making it more effective at predicting potential health risks. Darvesh et al. also emphasize the importance of user-friendly interfaces, providing farmers with easy access to real-time data on their cattle's health. This feature allows farmers to make informed decisions quickly and take preventive measures when necessary. Additionally, the system can be integrated with other farm management software, providing a more comprehensive approach to livestock management. The researchers suggest that this integrated system can help reduce disease outbreaks and improve overall livestock productivity by allowing for early detection and rapid response to health issues..

● In a study by [6]. In this study, Dutta et al. (2019) developed "Monitor," an IoT based multi-sensory intelligent device for cattle activity monitoring. This device incorporates various sensors to collect data on cattle activity, environmental conditions, and other health-related parameters. The goal is to provide a comprehensive view of the cattle's health and behavior, allowing farmers to detect anomalies and take appropriate action. The sensors measure temperature, humidity, and activity levels, providing insights into the cattle's comfort and well-being. The data collected by "Monitor" is transmitted wirelessly to a central system, where it can be analyzed and stored for future reference. This real-time monitoring capability enables farmers to respond quickly to any signs of distress or illness, reducing the risk of severe health issues. The researchers also emphasize the importance of data visualization and user-friendly interfaces, allowing farmers to access and interpret the information easily. This multi-sensory approach to cattle monitoring has the potential to improve livestock management and enhance animal welfare by providing a more detailed understanding of the cattle's environment and behavior.

● In a study by [7], Gupta and Mehta (2020) created an IoT based system for real-time cattle health monitoring, aiming to improve the overall health and productivity of livestock. The system incorporates various sensors to track vital health parameters, such as body temperature and heart rate, as well as behavioral patterns. This real-time monitoring capability enables farmers to detect and respond to health issues promptly, reducing the risk of serious complications. The system also provides historical data, allowing farmers to analyze trends and identify recurring patterns that may indicate underlying health problems. The researchers emphasize the importance of real-time monitoring in reducing the response time to health emergencies, ultimately leading to better livestock management and increased productivity. Additionally, the system is designed to be scalable, making it suitable for different farm sizes and configurations. This flexibility allows farmers to implement the system according to their specific needs and budget constraints. problems. The researchers emphasize the importance of real-time monitoring in reducing the response time to health emergencies ,The authors also suggest that this approach can help reduce livestock mortality rates and improve the overall efficiency of farm operations.

.

- In a study by [8], Habeeb et al. (2018) examined temperature-humidity indices as indicators for heat stress in farm animals. This study is crucial in the context of IoT based monitoring systems, as it provides a method for assessing environmental stress factors that can impact livestock health. By monitoring temperature and humidity levels, farmers can identify conditions that may lead to heat stress in cattle. This understanding allows them to implement measures to mitigate stress, such as providing additional shade, ventilation, or water resources. The authors discuss the impact of heat stress on cattle production and reproduction, highlighting the importance of maintaining optimal conditions for livestock health. The study also explores the relationship between heat stress and other health issues, emphasizing the need for proactive monitoring and management. By incorporating temperature-humidity indices into IoT based systems, farmers can gain insights into the environmental factors affecting their livestock, leading to improved animal welfare and productivity. This research underscores the value of environmental monitoring in smart agriculture and its role in supporting sustainable farming practices.

● In a study by [9],Kumar et al. (2021) developed a smart gas leakage detection system with IoT for industrial safety. Although the focus of this study is on industrial safety, it has relevance in the context of agricultural environments where hazardous gases could pose risks to livestock and farm operations. The system uses IoT sensors to detect gas leaks and provides real-time alerts to the concerned authorities, helping prevent accidents and ensuring a safer environment. This approach to safety monitoring highlights the versatility of IoT technology in different settings, including agriculture. By implementing similar IoT based safety systems in agricultural operations, farmers can ensure a safer environment for their cattle and workers. The authors point out that integrating gas detection systems with other IoT based farm management tools can contribute to a more comprehensive safety strategy, allowing farmers to manage risks effectively. This study underscores the importance of safety monitoring in smart agriculture and its potential impact on livestock health and productivity.

● In a study by [10], Kumar et al. (2019) designed an IoT based smart collar for real-time cattle monitoring in smart farming. The smart collar is equipped with sensors that collect data on cattle activity, location, and health parameters. This information is transmitted wirelessly to a central monitoring system, allowing farmers to track the health and behavior of their cattle in real-time. The smart collar's design is lightweight and comfortable, ensuring it does not interfere with the cattle's daily activities. The researchers emphasize that this technology can help detect unusual behavior patterns that may indicate health issues or stress. By monitoring cattle in real-time, farmers can respond quickly to potential problems, reducing the risk of severe health complications. The study also explores the integration of the smart collar with other IoT based systems, providing a comprehensive solution for livestock monitoring. This approach allows farmers to gain a detailed understanding of their cattle's health and behavior, leading to improved livestock management and animal welfare.

● In a study by [11], Li et al. (2019) explored the use of IoT in smart farming, focusing on livestock monitoring and disease detection. This study highlights the benefits of IoT technology in creating a connected environment within agricultural settings. By leveraging a network of sensors, farmers can collect real-time data on various health and environmental factors, providing a comprehensive view of the farm. The IoT based system allows for seamless data flow, facilitating early detection of diseases and enabling proactive management of livestock health. The authors emphasize that such systems can help reduce the risk of disease outbreaks and improve the overall efficiency of farm operations. Additionally, the study discusses the potential for integrating IoT data with other farm management systems, providing a holistic approach to agricultural management. This integration can lead to better resource utilization and cost savings, as farmers can make more informed decisions based on accurate and timely data. The researchers also highlight the importance of user-friendly interfaces, ensuring that farmers can easily interact with the system and access the information they need.

● In a study by [12], Nootyaskool and Ounsrimung (2020) developed a smart collar to predict cow behavior, focusing on identifying patterns that could indicate stress or health issues. The collar is equipped with IoT sensors that track the cow's movements, providing a continuous stream of data on its activity levels and behavior patterns. This data can be analyzed to detect unusual behavior that may signal health problems or stress. The smart collar is designed to be comfortable for the cow, ensuring it does not interfere with its daily activities. By monitoring behavior, farmers can identify signs of discomfort or distress early on, allowing them to take corrective action before the situation escalates. The researchers also point out that the smart collar can be integrated with other IoT systems to provide a more comprehensive view of the cattle's health and well-being. This approach to cattle monitoring is part of a broader trend toward precision agriculture, where technology is used to optimize farming practices and improve animal welfare. The study demonstrates that wearable technology can play a crucial role in advancing smart agriculture and supporting farmers in their efforts to maintain healthy livestock.

- In a study by [13], Rao and Singh (2022) described the design and implementation of an IoT based livestock health monitoring system. This system integrates various sensors to monitor cattle health, providing farmers with real-time data on vital parameters such as temperature, heart rate, and activity levels. The system's design is intended to be robust and scalable, allowing it to be used in different farming environments. The authors discuss the challenges involved in implementing IoT based systems, including connectivity issues, data security, and power management. To address these challenges, the system uses wireless communication technologies that ensure reliable data transmission and low power consumption. Additionally. The study also explores the potential for integrating the IoT based system with other farm management tools, enabling a comprehensive approach to livestock health monitoring. By providing real-time alerts and insights, this system helps farmers make informed decisions and improve the overall health and productivity of their livestock.

- In a study by [14], Smith and Johnson (2020) proposed an AI-based approach for cattle health monitoring in smart agriculture. By integrating AI with IoT data, the system can predict potential health risks and suggest preventive measures. The use of AI allows the system to analyze large volumes of data, identifying patterns and correlations that might not be immediately apparent to human observers. The AI-based system can also be used to develop custom health management plans for individual cattle, taking into account their unique health profiles and behavior patterns. This level of customization helps farmers optimize their livestock management practices, leading to improved productivity and animal welfare. The authors also point out that AI-based systems can reduce the workload on farmers by automating certain tasks, such as data analysis and alert generation. This automation allows farmers to focus on other critical aspects of farm management, contributing to more efficient and sustainable agricultural.

● In a study by [15] , Syarif et al. (2019) introduced an IoT based health monitoring system for dairy cows with a focus on early disease detection. The system utilizes a network of sensors placed on dairy cows to collect vital health data such as body temperature, heart rate, and activity patterns. This data is transmitted wirelessly to a central monitoring system, The researchers highlight that early detection can significantly reduce veterinary costs and increase dairy productivity by minimizing the downtime caused by illness. This system also provides farmers with real-time alerts when critical health thresholds are crossed, allowing them to act swiftly to address potential issues. Additionally, the system can be integrated with other farm management software, providing a holistic view of dairy farm operations. This integration helps farmers make more informed decisions and optimize their livestock management practices.

Furthermore, to prolong the lifecycle of cattle and sustain the quality of dairy products, regular monitoring of cattle's health is essential. This proactive approach not only ensures the early detection and management of potential health issues but also contributes to the overall welfare and productivity of the herd. However, the challenge arises in the practical implementation of such monitoring, particularly on large farms where the sheer scale of operations can make day-to-day observation of individual animals challenging.. As a result, there is a pressing need for technological solutions that can automate and streamline the process of cattle health monitoring, providing real-time insights and alerts to farm managers and veterinarians. By leveraging advancements in sensor technology, data analytics, and connectivity, these solutions have the potential to revolutionize livestock management practices, enabling early intervention, optimizing resource allocation, and ultimately improving the health outcomes and sustainability of dairy farming operations.

This paper investigates the existing based technology and provide a comparison of the features offered by these systems and their limitations. This is the main motivation behind this paper to analyze the different research methods concerning cattle health monitoring and present a solution that will monitor the cattle in real time with the help of IOT along with ML and cloud based technology, and bring some new features. This paper presents the Following IOT sensors.

Sensors

Our Project incorporates skin temperature sensors to monitor livestock health in real-time, aiding in the early detection of fever or stress indicators. These sensors provide crucial data for proactive veterinary interventions and herd management strategies.Our Project also utilizes heart rate sensors to continuously monitor the cardiovascular health of livestock, enabling early detection of irregularities and facilitating prompt veterinary attention when needed. These sensors play a pivotal role in optimizing animal welfare and farm productivity through proactive health management.Our Project integrates GPS sensors to track the movement and location of livestock, providing farmers with valuable insights into grazing patterns, herd behavior, and potential escape incidents. These sensors enhance farm management efficiency and facilitate optimal utilization of grazing resources.Additionally, our project leverages the capabilities of advanced data analytics and machine learning algorithms to analyze the vast amount of data collected from these sensors, enabling predictive modeling and actionable insights for improved livestock management practices.

CHAPTER 3
REQUIREMENT SPECIFICATION

3.1 HARDWARE REQUIREMENTS

The hardware requirements for Livestock wellbeing systems in sewage vary depending on the specific implementation and functionalities of the system. However, here are the hardware components typically used in the system.

3.1.1 NODE MCU ESP8266

NodeMCU is an open source LUA based firmware developed for ESP8266 wifi chip. By exploring functionality with ESP8266 chip, NodeMCU firmware comes with ESP8266 Development board/kit i.e. NodeMCU Development board.

Since NodeMCU is open source platform, their hardware design is open for edit/modify/build. NodeMCU Dev Kit/board consist of ESP8266 wifi enabled chip. The ESP8266 is a low-cost WiFi chip developed by Espressif Systems with TCP/IP protocol. For more information about ESP8266, you can refer ESP8266 WiFi Module.There is Version2 (V2) available for NodeMCU Dev Kit i.e. NodeMCU Development Board v1.0 (Version2), which usually comes in black colored PCB.

NodeMCU Dev Kit has Arduino like Analog (i.e. A0) and Digital (D0-D8) pins on its board. It supports serial communication protocols i.e. UART, SPI, I2C etc. Using such serial protocols we can connect it with serial devices like I2C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules, touch screen displays, SD cards etc.

NodeMCU Development board is featured with wifi capability, analog pin, digital pins and serial communication protocols.To get start with using NodeMCU for IoT applications first we need to know about how to write/download NodeMCU firmware in NodeMCU Development Boards. And before that where this NodeMCU firmware will get as per our requirement.There is online NodeMCU custom builds available using which we can easily get our custom NodeMCU firmware as per our requirement.

After setting up ESP8266 with Node-MCU firmware, let’s see the IDE (Integrated Development Environment) required for development of NodeMCU. NodeMCU with ESPlorer IDE Lua scripts are generally used to code the NodeMCU. Lua is an open source, lightweight, embeddable scripting language built on top of C programming language. NodeMCU with Arduino IDE Here is another way of developing NodeMCU with a well-known IDE i.e. Arduino IDE. We can also develop applications on NodeMCU using Arduino development environment. This makes easy for Arduino developers than learning new language and IDE for NodeMCU.

Difference in using ESPlorer and Arduino IDE

Well, there is a programming language difference we can say while developing application for NodeMCU using ESPlorer IDE and Arduino IDE.We need to code in C\C++ programming language if we are using Arduino IDE for developing NodeMCU applications and Lua language if we are using ESPlorer IDE. Basically, NodeMCU is Lua Interpreter, so it can understand Lua script easily. When we write Lua scripts for NodeMCU and send/upload it to NodeMCU, then they will get executes sequentially. It will not build binary firmware file of code for NodeMCU to write. It will send Lua script as it is to NodeMCU to get execute. In Arduino IDE when we write and compile code, ESP8266 toolchain in background creates binary firmware file of code we wrote.

And when we upload it to NodeMCU then it will flash all SNSCT NODE MCU GAUTAMI.A,AP/ECE NodeMCU firmware with newly generated binary firmware code. In fact, it writes the complete firmware. That's the reason why NodeMCU not accept further Lua scripts/code after it is getting flashed by Arduino IDE. After getting flashed by Arduino sketch/code it will be no more Lua interpreter and we got error if we try to upload Lua scripts. To again start with Lua script, we need to flash it with NodeMCU firmware. Since Arduino IDE compile and upload/writes complete firmware, it takes more time than ESPlorer IDE.

Nodemcu Development Kit/Board

NodeMCU Development Kit/Board consist of ESP8266 wifi chip. ESP8266 chip has GPIO pins, serial communication protocol, etc. features on it. ESP8266 is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. For more information about ESP8266, you can refer ESP8266 WiFi Module. The features of ESP8266 are extracted on NodeMCU Development board. NodeMCU (LUA based firmware) with Development board/kit that consist of ESP8266 (wifi enabled chip) chip combines NodeMCU Development board which make it stand-alone device in IoT applications. Let's see 1st version of NodeMCU Dev Kit and its pinout as shown in below images.

GPIO (General Purpose Input Output) Pins

NodeMCU has general purpose input output pins on its board as shown in above pinout diagram. We can make it digital high/low and control things like LED or switch on it. Also, we can generate PWM signal on these GPIO pins.

ADC (Analog to Digital Converter) channel (A0)

NodeMCU has one ADC channel/pin on its board.

SPI (Serial Peripheral Interface) Pins

NodeMCU based ESP8266 has Hardware SPI (HSPI) with four pins available for SPI communication. It also has SPI pins for Quad-SPI communication. With this SPI interface, we can connect any SPI enabled device with NodeMCU and make communication possible with it.

I2C (Inter-Integrated Circuit) Pins

NodeMCU has I2C functionality support on ESP8266 GPIO pins. Due to internal functionality on ESP-12E we cannot use all its GPIOs for I2C functionality. So, do tests before using any GPIO for I2C applications.

UART (Universal Asynchronous Receiver Transmitter) Pins

NodeMCU based ESP8266 has two UART interfaces, UART0 and UART1. Since UART0 (RXD0 & TXD0) is used to upload firmware/codes to board, we can't use them in applications while uploading firmware/codes.

Difference in between 1st and 2nd version NodeMCU Board

We can make difference in 1st and 2nd version of NodeMCU Development board by their boards design and ESP modules on it.

- In 1st version of NodeMCU Dev Kit v0.9, CH341SER USB to Serial converter is used whereas in 2nd version of NodeMCU Dev Kit v1.0, CP2102 USB to Serial converter is used.

- 1st version uses ESP-12 and 2nd version uses ESP-12E (Enhanced version).

- Extra 6 pins (MTDO, MTDI, SD_3, MTMS, MTCK, SD_2) brought out on ESP-12E version of ESP-12 modules as shown in below figure.3.1.

NodeMCU Dev Kit in Markets

NodeMCU hardware is open source, allowing anyone to freely edit, modify, and produce their own versions of NodeMCU development boards. As a result, we often see a variety of NodeMCU Dev boards from different manufacturers such as Amica, DOIT, Lolin, and D1 mini/Wemos available in the market. Among these, Amica is a prominent producer of NodeMCU ESP8266 Development Boards, particularly their v1.0 (Version 2) models, which adhere to designed hardware specifications. Notably, Amica has been a key contributor to the NodeMCU ecosystem, consistently delivering reliable and high-quality boards that have gained widespread adoption among makers, hobbyists, and professionals alike. The availability of these open-source hardware options has significantly contributed to the accessibility and democratization of IoT development, enabling innovators to bring their ideas to life with ease and flexibility.

Nodemcu Hardware Specifications

We can see the NodeMCU Dev Kit v0.9 hardware specifications and design from below link that is open for all. NodeMCU Dev Kit v1.0 hardware specifications and design is given in below link NodeMCU DevKit v1.0

The ADC block in both version uses resistor divider network (220K and 100K) to scale ESP8266 ADC input voltage range of 0-1V to 0-3.3V. Since the input analog voltage range for ADC pin of ESP8266 is 0–1.0V (while reading external analog voltage), NodeMCU Dev boards uses this resistor divider network to scale it up to 0-3.3V.

ESP8266 Development Boards Also, we can see ESP8266 development boards/breakouts that are different in design than above NodeMCU Development boards and we can use them with NodeMCU firmware for IoT applications. Below are those alternative boards for NodeMCU with different size, pinouts and specifications that are available in market.

Here, we can say that there is no unique NodeMCU Development board design in market. If we came across their official boards then we can realize that Amica boards are looks like an official version whereas others not (since they are applications wise designed). Amica provided some points regarding improving their development boards on their twitter page.

Figure 3.1 Nodemcu esp8266

In Summary, NodeMCU is an open-source firmware and development board based on the ESP8266 Wi-Fi module. It allows for easy prototyping of IoT projects by providing built-in Wi-Fi connectivity and support for the Lua scripting language. NodeMCU is widely used for developing smart home devices, sensor networks, and other IoT applications due to its low cost, versatility, and robust community support.NodeMCU also features a user-friendly development environment with a range of compatible libraries and tools, making it accessible to both beginners and experienced developers for rapid prototyping and deployment of IoT solutions.

3.1.1 PULSE SENSOR

This circuit is designed to measure the heart beat rate. The heart beat rate is measured by IR transmitter and receiver.

Infrared transmitter is one type of LED which emits infrared rays generally called as IR Transmitter. Similarly IR Receiver is used to receive the IR rays transmitted by the IR transmitter. One important point is both IR transmitter and receiver should be placed straight line to each other. The IR transmitter and receiver are placed in the pulse rate sensor. When you want measure the pulse rate, the pulse rate sensor has to be clipped in the finger. The IR receiver is connected to the Vcc through the resistor which acts as potential divider. The potential divider output is connected to amplifier section.

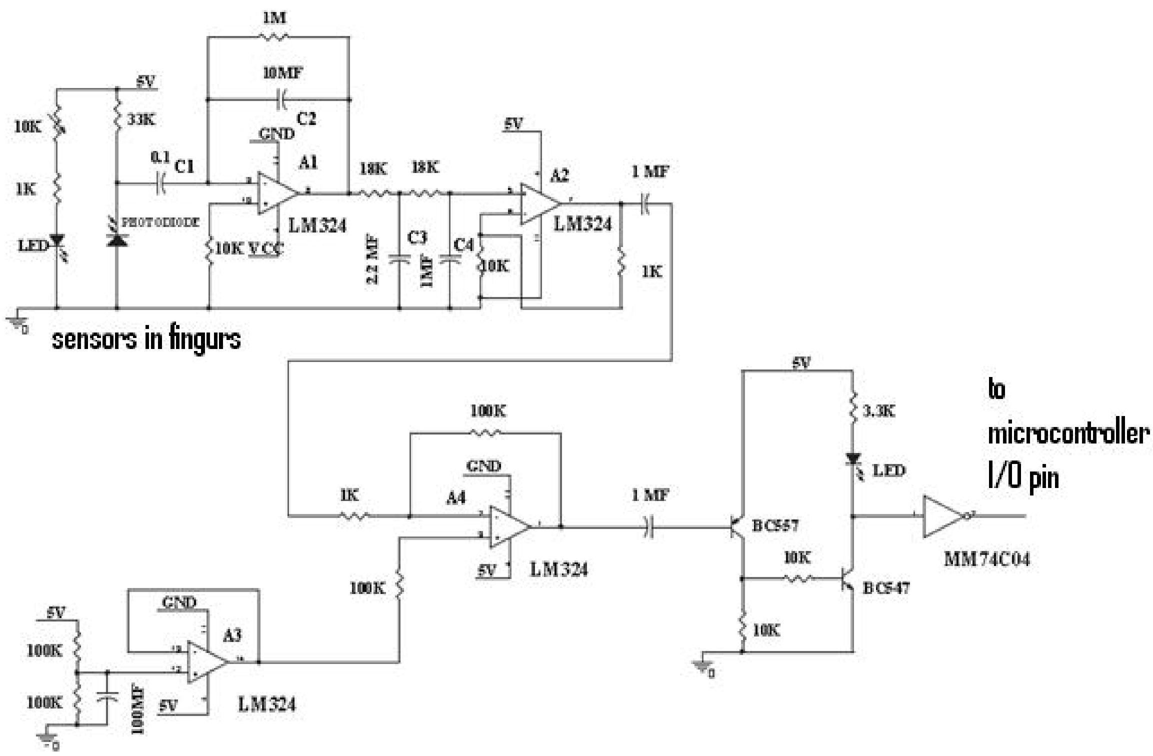


Figure 3.2 Pulse Sensor circuit

When supply is ON the IR transmitter passes the rays to the receiver. Depending on the blood flow, the IR rays are interrupted. Due to that IR receiver conduction is interrupted so variable pulse signals are generated in the potential divider point which is given to low pass filter through the capacitor C1. The coupling capacitor C1 is used to block the DC component because the capacitor reactance is depends on the frequency. For DC component the frequency is zero so the reactance is infinity now capacitor acts as open circuit for DC component

The amplifier section is constructed by the LM 324 quad operational amplifier. It consists of four independent, high gains and internally frequency compensated operational amplifiers named as A1, A2, A3 and A4 amplifiers. The varying pulse from the potential divider is amplified by the A1 amplifier. In this amplifier act as a low pass filter and the capacitor C2 is connected in parallel with feedback resistor to filter the any DC component in the output signal. If any spikes or noise in the output signals, they are further filtered by the C3 and C4 capacitors. After filtration the signal is again amplified by the A2 amplifier

Then amplified signal is given to inverting input terminal of c amplifier. The amplifier is constructed by the A4 amplifier in which the off set voltage is given to non inverting input terminal. The offset voltage is generated by the A3 amplifier. Then the amplifier amplifies the signal and delivered its output.

Then the signal is given to base of the BC 557(PNP) and BC547 (NPN) switching transistors in order to convert the TTL voltage 0 to 5v level. Finally the TTL output is given to 74C04 inverter to invert the pulse in digital form. Then the final square wave signal is given to microcontroller or other interfacing circuit in order to monitor the heart rate.

Heart Rate

The pulse rate is a measurement of the heart rate, or the number of times the heart beats per minute. As the heart pushes blood through the arteries, the arteries expand and contract with the flow of the blood. Taking a pulse not only measures the heart rate, but also can indicate the following:

- Heart rhythm
- Strength of the pulse

The normal pulse for healthy adults ranges from 60 to 100 beats per minute. The pulse rate may fluctuate and increase with exercise, illness, injury, and emotions. Females ages 12 and older, in general, tend to have faster heart rates than do males. Athletes, who do a lot of cardiovascular conditioning, may have heart rates near 40 beats per minute and experience no problems.

Heart rate is a term used to describe the frequency of the cardiac cycle. It is considered one of the four vital signs. Usually it is calculated as the number of contractions (heart beats) of the heart in one minute and expressed as "beats per minute" (bpm). See "Heart" for information on embryofetal heart rates. The heart beats up to 120 times per minute in childhood. When resting, the adult human heart beats at about 70 bpm (males) and 75 bpm (females), but this rate varies among people. However, the reference range is normally between 60 bpm (if less termed bradycardia) and 100 bpm (if greater, termed tachycardia). Resting heart rates can be significantly lower in athletes. The infant/neonatal rate of heartbeat is around 130-150 bpm, the toddler's about 100–130 bpm, the older child's about 90–110 bpm, and the adolescent's about 80–100 bpm.

The pulse is the most straightforward way of measuring the heart rate, but it can be deceptive when some heart beats do not have much cardiac output. In these cases (as happens in some arrhythmias), the heart rate may be considerably higher than the pulse rate.

Then amplified signal is given to inverting input terminal of c amplifier. The amplifier is constructed by the A4 amplifier in which the off set voltage is given to non inverting input terminal. The offset voltage is generated by the A3 amplifier. Then the amplifier amplifies the signal and delivered its output.

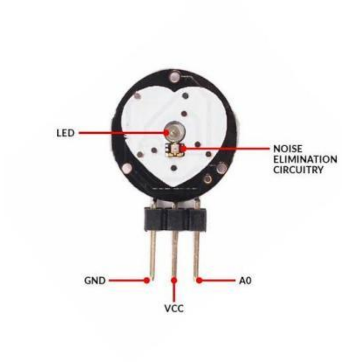


Figure 3.3 Pulse Sensor

The pulse, commonly measured at the radial artery on the wrist or the carotid artery in the neck, serves as a straightforward indicator of heart rate, providing a glimpse into the rhythm of cardiac contractions. However, this method can occasionally yield deceptive results, particularly when certain heartbeats fail to generate sufficient cardiac output. Such instances, often observed in individuals with arrhythmias, underscore the limitations of relying solely on pulse measurements for assessing heart rate.The heart rates across different age groups offer further insight into the variability of normal cardiac rhythms. In infants and neonates, the heart beats at a rapid pace, typically ranging between 130 to 150 beats per minute (bpm), reflecting the heightened metabolic demands and physiological adaptation during early development. As children transition into toddlerhood, the heart rate gradually decreases to a range of 100 to 130 bpm, mirroring the maturation of the cardiovascular system. Similarly, older children exhibit heart rates averaging between 90 to 110 bpm, while adolescents typically have heart rates ranging from 80 to 100 bpm.

3.1.2 ACCELEROMETER

An accelerometer is an electromechanical device that will measure acceleration forces. These forces may be static, like the constant force of gravity pulling at your feet, or they could be dynamic - caused by moving or vibrating the accelerometer.

An accelerometer is a device that measures the vibration, or acceleration of motion of a structure. The force caused by vibration or a change in motion (acceleration) causes the mass to "squeeze" the piezoelectric material which produces an electrical charge that is proportional to the force exerted upon it. Since the charge is proportional to the force, and the mass is a constant, then the charge is also proportional to the acceleration.

There are two types of piezoelectric accelerometers (vibration sensors). The first type is a "high impedance" charge output accelerometer. In this type of accelerometer the piezoelectric crystal produces an electrical charge which is connected directly to the measurement instruments. The charge output requires special accommodations and instrumentation most commonly found in research facilities. This type of accelerometer is also used in high temperature applications (>120C) where low impedance models cannot be used.

Triaxial accelerometers measure the vibration in three axes X, Y and Z. They have three crystals positioned so that each one reacts to vibration in a different axis. The output has three signals, each representing the vibration for one of the three axes. The ACC301 has lightweight titanium construction and 10 mV/g output with a dynamic range of +/-500 g's over a range of 3 to 10 kHz.

The second type of accelerometer is a low impedance output accelerometer. A low impedance accelerometer has a charge accelerometer as its front end but has a tiny built-in micro-circuit and FET transistor that converts that charge into a low impedance voltage that can easily interface with standard instrumentation. This type of accelerometer is commonly used in industry. An accelerometer power supply like the ACC-PS1, provides the proper power to the microcircuit 18 to 24 V @ 2 mA constant current and removes the DC bias level, they typically produces a zero based output signal up to +/- 5V depending upon the mV/g rating of the accelerometer. All OMEGA(R) accelerometers are this low impedance type.

Features

- Selectable Sensitivity (1.5g/2g/4g/6g)
- Low Current Consumption: 500 µA
- Sleep Mode: 3 µA
- Low Voltage Operation: 2.2 V – 3.6 V
- 6mm x 6mm x 1.45mm QFN
- High Sensitivity (800 mV/g @ 1.5g)
- Fast Turn On Time
- Integral Signal Conditioning with Low Pass Filter
- Robust Design, High Shocks Survivability
- Pb-Free Terminations
- Environmentally Preferred Package
- Low Cost

Applications

• HDD MP3 Player: Freefall Detection

• Laptop PC: Freefall Detection, Anti-Theft

• Cell Phone: Image Stability, Text Scroll, Motion Dialing, E-Compass

• Pedometer: Motion Sensing

• PDA: Text Scroll

• Gaming: Tilt and Motion Sensing, Event Recorder

• Robotics: Motion Sensing

Accelerometers can be used to measure vehicle acceleration. They allow for performance evaluation of both the engine/drive train and the braking systems. Useful numbers like 0-60mph, 60-0mph and 1/4 mile times can all be found using accelerometers.

Accelerometers can be used to measure vibration on cars, machines, buildings, process control systems and safety installations. They can also be used to measure seismic activity, inclination, machine vibration, dynamic distance and speed with or without the influence of gravity. Applications for accelerometers that measure gravity, wherein an accelerometer is specifically configured for use in are called gravimeters.

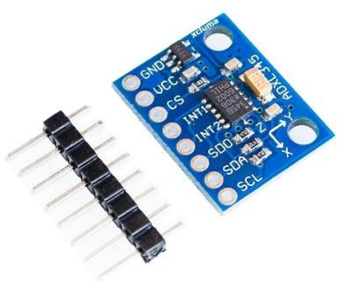


Figure 3.4 Accelerometer

3.1.3 TEMPERATURE SENSOR DHT-11

Temperature sensors are crucial components in a variety of systems and industries. Their primary function is to measure temperature or heat energy, yielding data that can inform or automate responses within a larger system. Understanding these versatile devices is vital in many fields The DHT11 is a low-cost, digital temperature and humidity sensor widely used in electronics projects and IoT applications. It can accurately measure temperature in the range of 0°C to 50°C with a resolution of 1°C and humidity in the range of 20% to 80% with a resolution of 1%. The DHT11 sensor communicates with microcontrollers via a single-wire digital protocol, making it easy to integrate into various projects for environmental monitoring, weather stations, HVAC systems, and more. The DHT11's affordability, combined with its simplicity of use and reliable performance, has made it a popular choice among hobbyists, educators, and professionals alike, driving innovation in diverse fields ranging from home automation to industrial process control.Its compact size and low power consumption further enhance its versatility, allowing for seamless integration into battery-operated devices and applications where space is limited.

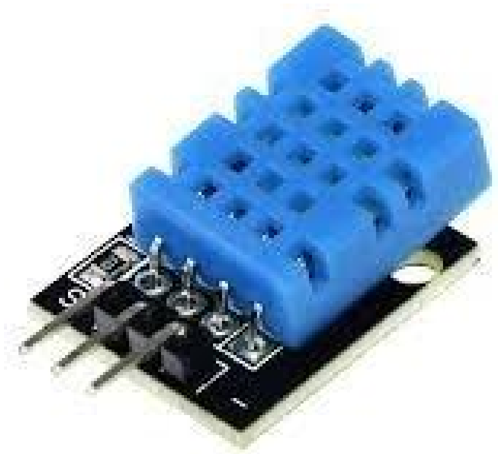


Figure 3.5 Temperature Sensor DHT-11

Schematic diagram

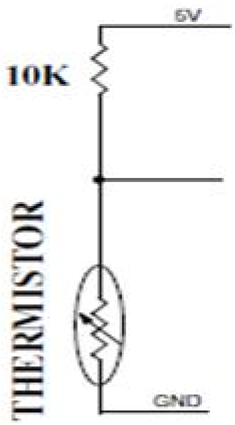


Figure 3.6 Temperature Sensor-Schematic Diagram

Schematic Explanation

In this circuit the thermistor is used to measure the temperature. Thermistor is nothing but temperature sensitive resistor. There are two type of thermistor available such as positive temperature co-efficient and negative temperature co- efficient. Here we are using negative temperature co-efficient in which the resistance value is decreased when the temperature is increased.If the R1 and R2 value is equal means the output is half of the Vcc supply. In this circuit output is a variable one. So the output is depending upon the R2 resistance value.

$$V_{out} = V_{in} \frac{R_2}{(R_1 + R_2)}$$

Resistance value will be varied depend upon the Temperature level. Temperature varied means the resistance value also varied. If resistance value increased means output also increased. The resistance value and output is a directly proportional one. Then the final voltage is given to ADC for convert the analog signal to digital signal. Then the corresponding digital signal is taken to process in microcontroller.

The ADC value will increase if the temperature increased. We can measure the temperature only with the help of any controller or processor.

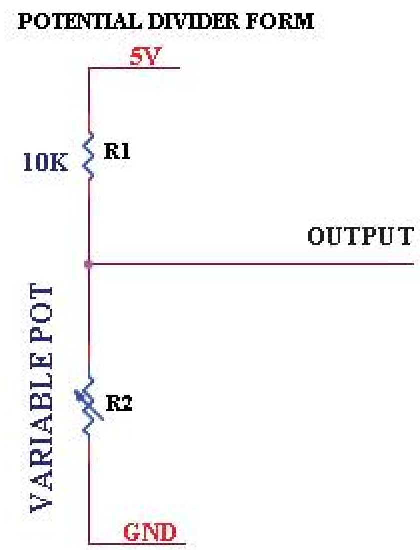


Figure 3.7 Potential Divider Diagram

Thermistor

A thermistor is a type of resistor used to measure temperature changes, relying on the change in its resistance with changing temperature. Thermistor is a combination of the words thermal and resistor. If we assume that the relationship between resistance and temperature is linear (i.e. we make a first-order approximation), then we can say that:

$\Delta R = k\Delta T$

Where

ΔR = change in resistance

ΔT = change in temperature

k = first-order temperature coefficient of resistance

These three values determine the performance of the thermistor on how well it acts as wheather a resistor or as a thermistor.

Thermistors can be classified into two types depending on the sign of k. If k is positive, the resistance increases with increasing temperature, and the device is called a positive temperature coefficient (PTC) thermistor, Posistor. If k is negative, the resistance decreases with increasing temperature, and the device is called a negative temperature coefficient (NTC) thermistor. Resistors that are not thermistors are designed to have the smallest possible k, so that their resistance remains almost constant over a wide temperature range.The thermistor symbol represents a specialized type of resistor whose electrical resistance changes with temperature variations. Thermistors are widely used in temperature sensing and control applications due to their sensitivity and reliability. They come in two main types: negative temperature coefficient (NTC) thermistors, where resistance decreases as temperature rises, and positive temperature coefficient (PTC) thermistors, where resistance increases with temperature. The thermistor symbol serves as a visual representation of this versatile component, which finds applications in diverse fields such as automotive, electronics, HVAC (heating, ventilation, and air conditioning), and industrial process control.

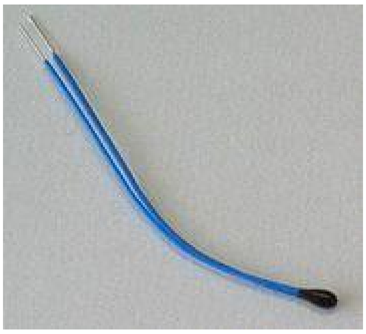


Figure 3.8 Thermistor Diagram

Symbol:

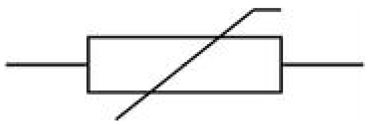


Figure 3.9 Thermistor Symbol

3.1.4 GPS MODULE WITH PATCH ANTENNA

A GPS receiver determines its own location by measuring the time it takes for a signal to arrive at its location from at least four satellites. Because radio waves travel at a constant speed, the receiver can use the time measurements to calculate its distance from each satellite. Using multiple satellites makes the GPS data more accurate.

A GPS receiver operated by a user on Earth measures the time it takes radio signals to travel from four or more satellites to its location, calculates the distance to each satellite, and from this calculation determines the user's longitude, latitude, and altitude.

This is a third generation POT (Patch Antenna On Top) GPS module. This POT GPS receiver providing a solution that high position and speed accuracy performances as well as high sensitivity and tracking capabilities in urban conditions & provides standard NMEA0183 strings in "raw" mode for any microcontroller.

For a GPS device to work correctly, it must first establish a connection to the required number of satellites. This process can take anywhere from a few seconds to a few minutes, depending on the strength of the receiver. For example, a car's GPS unit will typically establish a GPS connection faster than the receiver in a watch or smartphone. Most GPS devices also use some type of location caching to speed up GPS detection. By memorizing its previous location, a GPS device can quickly determine what satellites will be available the next time it scans for a GPS signal.

The module provides current time, date, latitude, longitude, speed, altitude and travel direction / heading among other data, and can be used in a host of applications, including navigation, tracking systems, fleet management, mapping.

This is a standalone GPS Module and requires no external components except power supply decoupling capacitors. It is built with internal RTC Back up battery. It can be directly connected to Microcontroller's USART. The module is having option for connecting external active antenna if necessary

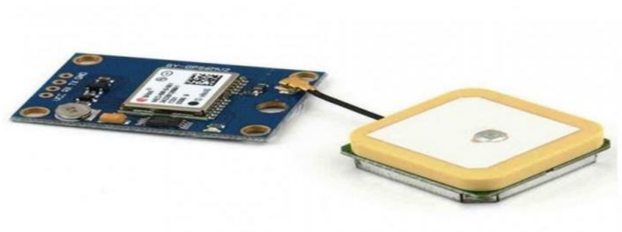


Figure 3.10 GPS Module

The GPS chipsets inside the module are designed by MediaTek Inc., which is the world's leading digital media solution provider and largest fab-less IC company in Taiwan. This process can take anywhere from a few seconds to a few minutes, depending on the strength of the receiver. For example, a car's GPS unit will typically establish a GPS connection faster than the receiver in a watch or smartphone. This is a third generation POT (Patch Antenna On Top) GPS module. This POT GPS receiver providing a solution that high position and speed accuracy performances as well as high sensitivity and tracking capabilities in urban conditions & provides standard NMEA0183 strings in “raw” mode for any microcontroller. The module can support up to 51 channels. The GPS solution enables small form factor devices. They deliver major advancements in GPS performances, accuracy, integration, computing power and flexibility. They are designed to simplify the embedded system integration process.

In Summary,A GPS module is a compact electronic device that receives signals from satellites orbiting the Earth to determine its precise geographical location, as well as velocity and time information. It typically consists of a GPS receiver chip, an antenna, and supporting circuitry housed within a small enclosure. Upon receiving signals from multiple satellites, the GPS module utilizes trilateration algorithms to calculate its exact position on the Earth's surface. GPS modules are widely used in various applications, including navigation systems for vehicles, aircraft, and ships, as well as in outdoor recreational devices like handheld GPS units and fitness trackers. They provide accurate positioning information in real-time, enabling users to navigate efficiently and safely, regardless of their location on the planet.

Specifications

- Supply: 3.3V, 45mA
- Chipset: MTK MT3318
- Antenna: High gain GPS patch antenna from Cirocomm
- Data output: CMOS UART interface at 3.3V
- Protocol: NMEA-0183@9600bps (Default) at update rate of 1 second.
- Protocol message support: GGA, GSA, RMC, VTG
- No. of Satellite simultaneously tracked: 51
- Tracking Sensitivity: On-module antenna : -157 dBm
- Position Accuracy : <3 m
- Max. Update Rate?5Hz (Default: 1 Hz)
- Time to First Fix (Open sky and stationary position)
- Obscuration recovery: 0.1 second average
- Hot start: <1 seconds average
- Warm start: <34 seconds average
- Cold start: <36 seconds average
- Size: 26mm x 26mm x11.7mm

3.2 SOFTWARE REQUIREMENTS

The Livestock health monitoring system using IoT and Machine Learning requires several software components to function effectively. For an AI and IoT based smart livestock health monitoring system, the software requirements can vary depending on the specific functionalities and components of the system. Here's a general outline of the software requirements

3.2.1.LIBRARIES USED

Numpy (Np)

NumPy is used for numerical operations and data manipulation, which is essential for processing sensor data collected from livestock. It allows for efficient handling of multidimensional arrays, which can represent various sensor readings such as temperature, pulse, and accelerometer data.

Pandas (Pd)

Pandas is used for data manipulation and analysis, particularly for working with structured data like the dataset containing livestock health data. It enables tasks such as data cleaning, preprocessing, and organizing sensor data into a DataFrame for further analysis.

Matplotlib.Pyplot (Plt)

Matplotlib is used for data visualization, allowing the visualization of sensor data trends and patterns. Plots and charts generated with Matplotlib can help farmers and veterinarians visualize the health status of livestock based on sensor readings like temperature, pulse, and accelerometer data

Seaborn (Sns)

Seaborn enhances data visualization capabilities by providing high-level functions for creating attractive and informative statistical graphics. It can be used to create advanced visualizations of livestock health data, such as heatmaps or violin plots, to gain insights into the distribution and relationships between different health parameters

Warnings

The warnings module is used to suppress warnings related to data processing, ensuring a cleaner output. While not directly related to livestock health monitoring, it contributes to the code's readability and maintenance by avoiding unnecessary warnings during execution

Sklearn

The model selection module from scikit-learn is used for splitting the dataset into training and testing sets. This is crucial for evaluating the performance of the Gaussian Naive Bayes classifier and ensuring that the model generalizes well to unseen data, including new livestock health observations

The preprocessing module from scikit-learn provides functions for scaling and preprocessing the sensor data before feeding it into the machine learning model. This ensures that the input features have similar scales and distributions, which can improve the stability and performance of the Gaussian Naive Bayes classifier.

The naive bayes module from scikit-learn contains implementations of Naive Bayes classifiers, including GaussianNB. In the smart livestock health monitoring system, Gaussian Naive Bayes is used to classify livestock health status based on readings, such as temperature, pulse, and accelerometer data.

The metrics module from scikit-learn provides functions for evaluating the performance of the Gaussian Naive Bayes classifier. It allows for the computation of classification metrics such as precision, recall, and F1-score, which are important for assessing the accuracy and reliability of the health monitoring system.

Urllib

The urllib module is used to retrieve sensor data from external sources, such as IoT platforms or APIs. In the context of the smart livestock health monitoring system, urllib may be used to fetch real-time sensor data from IoT devices installed on the farm or from external weather APIs to incorporate environmental data into the health monitoring process.

These libraries collectively play a crucial role in various aspects of the smart livestock health monitoring system, including data processing, visualization, machine learning model training and evaluation, and data retrieval from external sources.

3.2.2 IoT PLATFORM

An IoT platform is essential for managing IoT devices, handling data ingestion, facilitating device communication, and ensuring security. These platforms provide centralized control and management capabilities, enabling seamless integration of diverse IoT devices into a unified system. Popular IoT platforms include AWS IoT, Google Cloud IoT, Microsoft Azure IoT, Blynk IoT, as well as open-source platforms like Eclipse IoT, offering flexibility and scalability to accommodate various project needs.

3.2.3 LEARNING TOOLS

Software tools for data analytics and machine learning are crucial for processing and analyzing collected data to detect patterns, anomalies, and trends. These tools empower developers and data scientists to extract meaningful insights from raw data, enabling informed decision-making and proactive intervention. Commonly used Python libraries for this purpose include TensorFlow, scikit-learn, Pandas, and NumPy, providing a robust foundation for implementing advanced analytics algorithms and models.

3.2.4 REAL-TIME DATA PROCESSING

Real-time data processing requires software frameworks capable of handling data streams from IoT devices and sensors efficiently. These frameworks offer scalable and fault-tolerant solutions for processing massive volumes of data in real-time, ensuring timely insights and responses to dynamic environmental changes. Popular choices for real-time data processing include Apache Kafka, Apache Spark Streaming, and Apache Flink, known for their high throughput, low latency, and ease of integration with IoT systems..

3.2.5 WEB DEVELOPMENT FRAMEWORK

A web development framework is necessary for creating user interfaces and dashboards for data visualization and monitoring. These frameworks provide developers with a set of tools and libraries for building responsive and interactive web applications, simplifying the development process and accelerating time to market. Options include Django, Flask, React, Angular, or Vue.js, each offering distinct features and capabilities to suit different project requirements and developer preferences.

3.2.6 DATABASE MANAGEMENT SYSTEM (DBMS):

A robust Database Management System (DBMS) is essential for storing and managing the vast amounts of data generated by the system. It serves as the foundation for efficient data organization, retrieval, and manipulation, ensuring reliability and consistency in handling diverse data types. Options include relational databases like MySQL or PostgreSQL, known for their structured data storage and powerful querying capabilities. Alternatively, NoSQL databases like MongoDB offer flexibility and scalability, particularly suitable for handling unstructured or semi-structured data and accommodating dynamic schema changes. The choice between relational and NoSQL databases depends on factors such as data structure, scalability requirements, and the specific needs of the application or project.

3.2.7.INTEGRATION AND API'S:

The system may require integration with external APIs or services to enhance its functionality, such as accessing weather data APIs for environmental monitoring, connecting with veterinary services for health analysis, or integrating with farm management software for streamlined operations. To facilitate seamless integration, software tools for API integration and web services development become essential. These tools empower developers to establish connections with external systems, exchange data efficiently, and leverage additional functionalities to enhance the system's capabilities. By leveraging APIs and web services, the system can access a wide range of resources and services, enabling comprehensive data analysis, decision-making, and automation within the agricultural ecosystem.

3.2.8 ALERT NOTIFICATION:

In addition to its core functionality, a robust agricultural monitoring system may require seamless integration with external APIs or services to enhance its capabilities. These integrations could encompass a wide range of functionalities, such as accessing weather data APIs for real-time meteorological insights, connecting to veterinary services for health monitoring and intervention, or interfacing with farm management software for streamlined data management and analysis. However, the effectiveness of such integrations hinges on the system's ability to keep stakeholders promptly informed about critical events or conditions. To fulfill this need, Twilio emerges as a reliable communication platform, offering versatile options for sending notifications via SMS, voice calls, or emails in real-time. Leveraging Twilio's services enables the agricultural monitoring system to ensure timely alert delivery to relevant parties, facilitating prompt responses to emerging issues or opportunities within the farming operation. Moreover, the integration process with Twilio necessitates the utilization of software tools for API integration and web services development to ensure seamless communication between the monitoring system and Twilio's platform. These tools also enable customization of alert notifications according to specific requirements, ensuring that stakeholders receive relevant and actionable information tailored to their needs. By combining the functionalities of external APIs with Twilio's communication capabilities, the agricultural monitoring system can enhance its efficiency, effectiveness, and overall value proposition for farmers and stakeholders alike.

CHAPTER 4
PROPOSED MODEL

4.1 INTRODUCTION

In this study, we developed a new general health monitoring system for dairy cows, this paper investigates the existing research methods and those research methods where some are only IOT based, some are only ML based and some are IOT along with ML and the majority of research method uses 3 main IOT devices:

• Body temperature.

• Heart rate.

• Accelerometer

Which was already mentioned. This IOT device can provide valuable insights into our system such as heart rate, activity monitoring, heat stress, the surrounding temperature, and sleep tracking, heat stress, and sleep tracking is the idea that we came up with as a new feature in general health monitoring of cows. And where AI is used to predict the status of cattle's health, continuously observing the real time health status will alert the user if cattle suffer from a health issue, and a mobile app is developed to observe data visualization

4.2 MONITORING SYSTEM

IOT devices such as skin temperature, heart rate, and accelerometer, these devices allow you to monitor the health of cows by these five parameters.

• Heart rate.

• Activity level.

• Heat stress.

• Surrounding temperature.

• Sleep tracking

Monitor Heartbeats

Heart rate monitoring is a critical aspect of cattle health management, as it provides valuable insights into the cardiovascular well-being of the animals. In our system, heart rate is measured using a pulse sensor, which captures the rhythmic contractions of the heart and provides readings in beats per minute (BPM) format. By continuously monitoring heart rate data, farmers can detect abnormalities and signs of stress or illness early on, enabling prompt intervention and preventive measures to ensure the overall health and welfare of the cattle.

Monitor Activity Level

Tracking the activity level of cattle is essential for assessing their overall fitness, mobility, and well-being. In our monitoring system, activity level is measured using an MPU6050 sensor, which captures motion and orientation data in three axes: "X", "Y", and "Z". By analyzing changes in movement patterns and activity levels over time, farmers can gain insights into the behavioral patterns of the cattle, identify signs of distress or discomfort, and adjust management practices accordingly to optimize comfort and productivity.

Monitor Heat Stress

Heat stress is a significant concern for cattle, especially during periods of high temperatures and humidity. Our monitoring system incorporates an MLX90614 sensor, which measures ambient temperature in degrees Celsius and also provides humidity readings in the vicinity of the cattle. By continuously monitoring temperature and humidity levels, farmers can assess the risk of heat stress and take proactive measures to mitigate its effects, such as providing shade.

Monitor Sleep Check

Sleep tracking is an emerging area of interest in livestock management, as adequate rest is essential for the overall health and productivity of cattle. In our system, sleep tracking is facilitated by a combination of pulse sensor data and activity level monitoring. By analyzing fluctuations in heart rate and activity levels during resting periods, farmers can infer sleep patterns and quality, identify potential sleep disturbances or discomfort, and implement measures to optimize sleeping conditions for the cattle, thereby promoting better health and performance outcomes.

Machine Learning

Machine learning serves as a cornerstone in our comprehensive approach to cattle health management, encompassing both 'AI healthcare' and 'environmental monitoring.' In 'AI healthcare,' our system analyzes a myriad of physiological and behavioral data collected from the cattle to assess their overall health status. Leveraging advanced machine learning algorithms, such as logistic regression, support vector machines, and neural networks, we can derive actionable insights regarding each animal's well-being, identifying potential health issues or abnormalities early on

On the other hand, 'environmental monitoring' focuses on predicting cattle health status based on a thorough analysis of environmental factors. This approach recognizes the significant influence of external elements, such as air quality, temperature, and humidity, on the health and well-being of the herd. By integrating data from sensors placed throughout the barn or grazing area, our system captures real-time environmental metrics and utilizes machine learning algorithms to correlate these factors with cattle health outcomes..

To predict the overall health status of the cattle in 'environmental monitoring,' we employ the Random Forest algorithm. This versatile ensemble learning technique combines multiple decision trees to enhance prediction accuracy and generalization performance. By training the model on a diverse dataset comprising environmental variables and historical health records, our system can effectively forecast the health status of the herd, alerting farmers to potential risks or issues before they escalate.

The results of these predictions are then seamlessly integrated into a user-friendly mobile app, providing farmers and barn managers with instant access to critical insights and actionable recommendations. Armed with this information, stakeholders can make informed decisions regarding herd management practices, preventive healthcare measures, and resource allocation, ultimately optimizing the health, productivity, and profitability of their livestock operations.

Web Interface

In addition to leveraging advanced machine learning techniques for cattle health monitoring, our project also includes the development of a user-friendly web interface. This interface serves as a centralized platform for farmers and barn managers to access, visualize, and analyze the wealth of data collected by our monitoring system.

Alert System

Whenever the model predicts that the cattle is unheathy it will send a alert notification to the farmer with a message containing the information from the sensors and stating “The cattle needs immediate attention” for sending the alert notification we have used the service of Twilio unique authentication number and set it for alert notification.

4.3 CIRCUIT DIAGRAM

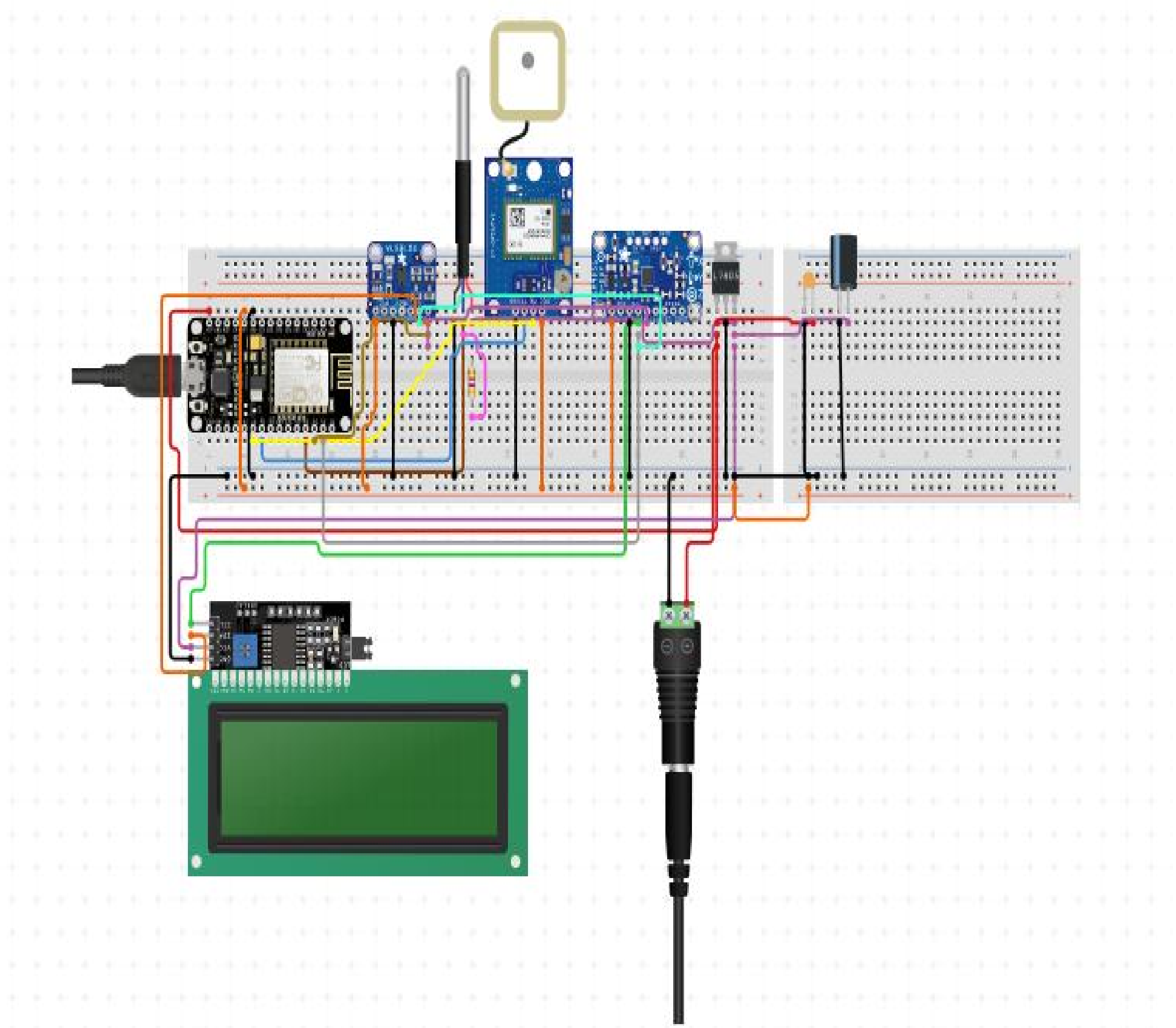


Figure 4.1 Circuit Diagram

4.4 BLOCK DIAGRAM

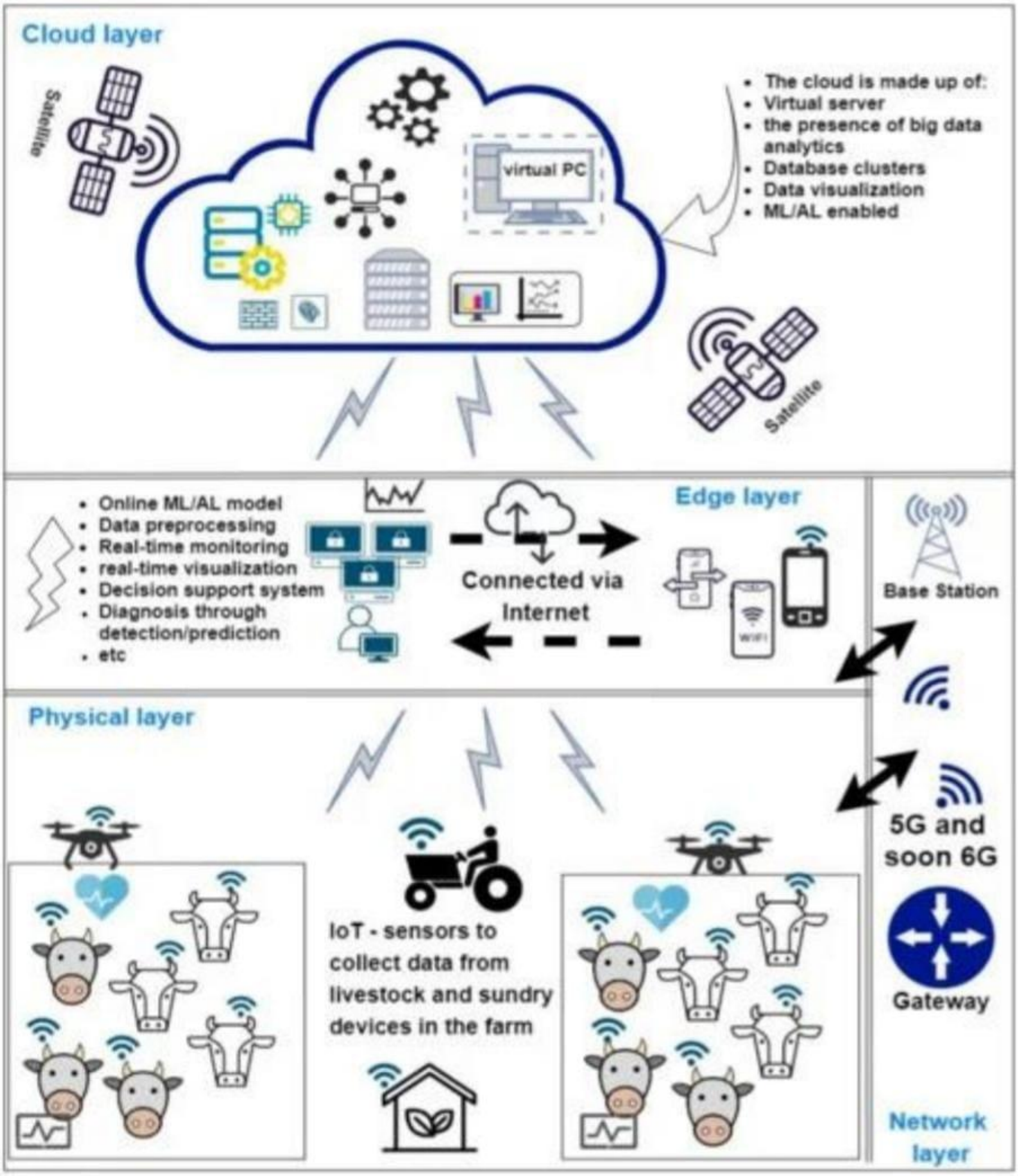


Figure 4.2 System Architecture

4.5 GAUSSIAN NAVIE BAYES ALGORITHM

Gaussian Naive Bayes is a probabilistic classification algorithm based on Bayes' Theorem with the assumption of independence between features. It is particularly useful for classification tasks when dealing with continuous-valued features. The term "Gaussian" refers to the assumption that the likelihood of the features follows a Gaussian distribution, also known as a normal distribution.

The algorithm is called "Naive" because it assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. In other words, it assumes that the features are conditionally independent given the class label. While this assumption may not always hold true in real-world datasets, Gaussian Naive Bayes often performs surprisingly well, especially in situations where the independence assumption is not violated to a significant degreeIn this project we have used this Naive Bayes Classifier to train a model which uses the give raw input and predict whether the particular cattle is healthy or unhealthy.

One of the key advantages of Gaussian Naive Bayes is its simplicity and computational efficiency, making it suitable for large datasets with many features. It requires minimal tuning of hyperparameters and is robust to noisy data. Additionally, it can handle multi-class classification problems with ease, making it a versatile choice for a wide range of applications.

The Naive Bayes algorithm is a popular machine learning algorithm commonly used for classification tasks, including in the context of smart livestock health monitoring systems. Here's a brief explanation of how the Naive Bayes algorithm works:Bayes' Theorem: The Naive Bayes algorithm is based on Bayes' theorem, which describes the probability of a hypothesis given the evidence. Mathematically, it can be expressed as:

$$P(h \mid d)=P(d)P(d \mid h)\cdot P(h)$$

Where,

(P(h|d)) is the probability of hypothesis (h) given the data (d).

(P(d|h)) is the probability of the data (d) given the hypothesis (h).

(P(h)) is the prior probability of hypothesis (h).

(P(d)) is the probability of observing the data (d)

Naive Assumption: The "naive" assumption in Naive Bayes is that the features are conditionally independent given the class label. This means that the presence of one feature does not affect the presence of another feature. While this assumption may not hold true in all cases, it simplifies the computation and often works well in practice.

The Gaussian Naive Bayes classifier assumes that the continuous features follow a Gaussian (normal) distribution. The formula for calculating the likelihood $P(Xi \mid C=c)P(Xi \mid C=c)$ of observing a value XiXi for feature ii given class cc is:

$$Cpredicted = argmaxc \in classes \quad P(C=c \mid X)$$

Using Bayes' theorem, this can be expressed as:

$$Cpredicted = argmaxc \in classes \quad P(X)P(X \mid C=c)\cdot P(C=c)$$

Where,

(P(X|C=c)) is the likelihood of observing the instance (X) given class (c)

(P(C=c)) is the prior probability of class (c)

(P(X)) is the evidence, which acts as a normalization factor.

Model Training: To train a Naive Bayes classifier, we need to estimate the likelihood ($P(X|C=c)$) and the prior probabilities ($P(C=c)$) from the training data. Depending on the type of features (e.g., categorical, numerical), different distributions (e.g., Gaussian, Multinomial, Bernoulli) can be used to model the likelihood.

Prediction: Once the model is trained, we can use it to predict the class labels of new instances by computing the posterior probabilities using Bayes' theorem and selecting the class label with the highest probability.

Overall, the Naive Bayes algorithm is a simple yet powerful probabilistic classifier that is particularly well-suited for text classification and other tasks with high-dimensional feature spaces. Its efficiency, ease of implementation, and ability to handle large datasets make it a popular choice for a wide range of classification tasks, including in smart livestock health monitoring systems.

Additionally, the Gaussian Naive Bayes algorithm is known for its robust performance even with relatively small training datasets, making it advantageous in scenarios where data availability may be limited, such as in certain agricultural settings. Its simplicity also lends itself well to applications where interpretability and transparency are crucial, allowing stakeholders to understand the reasoning behind classification decisions. Furthermore, Naive Bayes classifiers are inherently resistant to overfitting, which can be a concern in complex machine learning models, ensuring reliable performance in real-world environments. In the context of smart livestock health monitoring systems, Naive Bayes algorithms offer the potential to efficiently analyze large volumes of data from various sensors and sources, aiding in the early detection of health issues and facilitating timely interventions to optimize animal welfare and farm productivity.

CHAPTER 5

EXPERIMENTAL ANALYSIS AND RESULTS

5.1 SOURCE CODE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, confusion_matrix
df = pd.read_csv('animalhealth1000.csv')
df
df.isnull().sum()
plt.figure(figsize=(7,6))
sns.scatterplot(x="Temperature",y="Pulse",data=df)
plt.show()
plt.figure(figsize=(8,4))
sns.histplot(x="Temperature",y="Pulse",data=df)
plt.show()
rcParams['figure.figsize'] = 8,6
```

```
plt.bar(df['label'].unique(), df['label'].value_counts(), color = ['red', 'green'])
plt.xticks([2, 1])
plt.xlabel('Target Classes')
plt.ylabel('Count')
plt.title('Count of each Target Class')
plt.show()
y = df['label']
X = df.drop(['label'], axis = 1)
ss = StandardScaler()
scaled = ss.fit_transform(X)
X_scaled = pd.DataFrame(scaled,columns=X.columns)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size = 0.2,
    random_state = 0,stratify=y)
print("X_train :",X_train.shape)
print("X_test :",X_test.shape)
print("y_train :",y_train.shape)
print("y_test :",y_test.shape)
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
confusionmatrix = confusion_matrix(y_test,y_pred)
confusionmatrix
report = classification_report(y_test,y_pred)
print(report)
import urllib
import requests
```

```
import re
link="https://blynk.cloud/external/api/get?token=qSUBh7wQMRujpFumxO4cTjEQ6sW
    OpojN&V1"
f = requests.get(link)
f1=f.text
f1="".join(c for c in f1 if c.isalnum())
link2="https://blynk.cloud/external/api/get?token=qSUBh7wQMRujpFumxO4cTjEQ6s
    WOpojN&V2"
f = requests.get(link2)
f2=f.text
link3="https://blynk.cloud/external/api/get?token=qSUBh7wQMRujpFumxO4cTjEQ6s
    WOpojN&V3"
f = requests.get(link3)
f3=f.text
link4="https://blynk.cloud/external/api/get?token=qSUBh7wQMRujpFumxO4cTjEQ6s
    WOpojN&V4"
f = requests.get(link4)
f4=f.text
print(f1)
print(f2)
def spliter(line):
 dim = ['A','H','AQ','CO','G','PIR','PRE']
remove = ['P','T: ','D']
  word = ""+word+""
 line = line.replace(word," ")
 return line.split(" ")
```

```
result1 = spliter(f1)
result2 = spliter(f2)
result3 = spliter(f3)
result4 = spliter(f4)
print('Temperature : ',result2[0])
print('Accelerometer : ',result1[1])
print('Latitude : ', result3[0])
print('Longitude : ', result4[0])
print("Pulse value:", result1[0])
sensordata=[[float(result2[0]),float(result1[0]),float(result1[1]),float(f3),float(f4)]]
print(sensordata)
y_pred = gnb.predict(sensordata)
healthy_pulse_threshold = 40
unhealthy_pulse_threshold = 100
if float(result1[0]) <= healthy_pulse_threshold or float(result1[0]) >=
        unhealthy_pulse_threshold:
    print('Pulse rate indicates unhealthy condition')
elif y_pred == 1:
    print('Not Healthy')
    res1 = '1'
elif y_pred == 2:
    print("Healthy")
    res1="2"
```

FLASK

```
from flask import Flask, render_template, request
import joblib
from twilio.rest import Client
app = Flask(__name__)
with open('cattle.pkl', 'rb') as file:
    model = joblib.load(file)
healthy_pulse_threshold = 40
unhealthy_pulse_threshold = 90
account_sid = ""#enter your account sid from twilio here
auth_token ="" #enter your authentication token from twilio here
twilio_phone_number ="" #give your twilio phone number here
client = Client(account_sid, auth_token)
def send_notification(status, temperature, pulse, accelerometer, latitude, longitude):
    try:
        message = client.messages.create(
            body=f"Notification: {status}\nTemperature: {temperature}\nPulse:
        {pulse}\nAccelerometer: {accelerometer}\nLatitude: {latitude}\nLongitude:
        {longitude}",
            from_=twilio_phone_number,
            to="+91" #enter your phone number to which alert notification will be sent )
        print("Notification sent successfully:", message.sid)
        return True
    except Exception as e:
        print("Error sending notification:", e)
        return False
```

```
@app.route('/')
def home():
    return render_template('doc.html')
@app.route('/predict', methods=['POST'])
def predict():
    temperature = float(request.form['temperature'])
    pulse = float(request.form['pulse'])
    accelerometer = float(request.form['accelerometer'])
    latitude = float(request.form['latitude'])
    longitude = float(request.form['longitude'])
    prediction = model.predict([[temperature, pulse, accelerometer, latit]]
    if pulse <= healthy_pulse_threshold or pulse >= unhealthy_p:
        status = 'Pulse is critical cattle needs immediate attention.'
    elif prediction == 1:
        status = 'Not Healthy'
    else:
        status = 'cattle is Healthy
    notification_sent = False
    if status != 'cattle is Healthy':
        notification_sent = send_notification(status, temperature, pulse, accelerometer,
        latitude, longitude)
    return render_template('doc.html', status=status, notification_sent=notification_sent)
if __name__ == '__main__':
 app.run()
```

5.2 OUTPUT

The following figure depicts the prototype of our project which contains various sensors such as Pulse sensor, Temperature sensor, GPS sensor and a Accelerometer connected to a Node MCU.

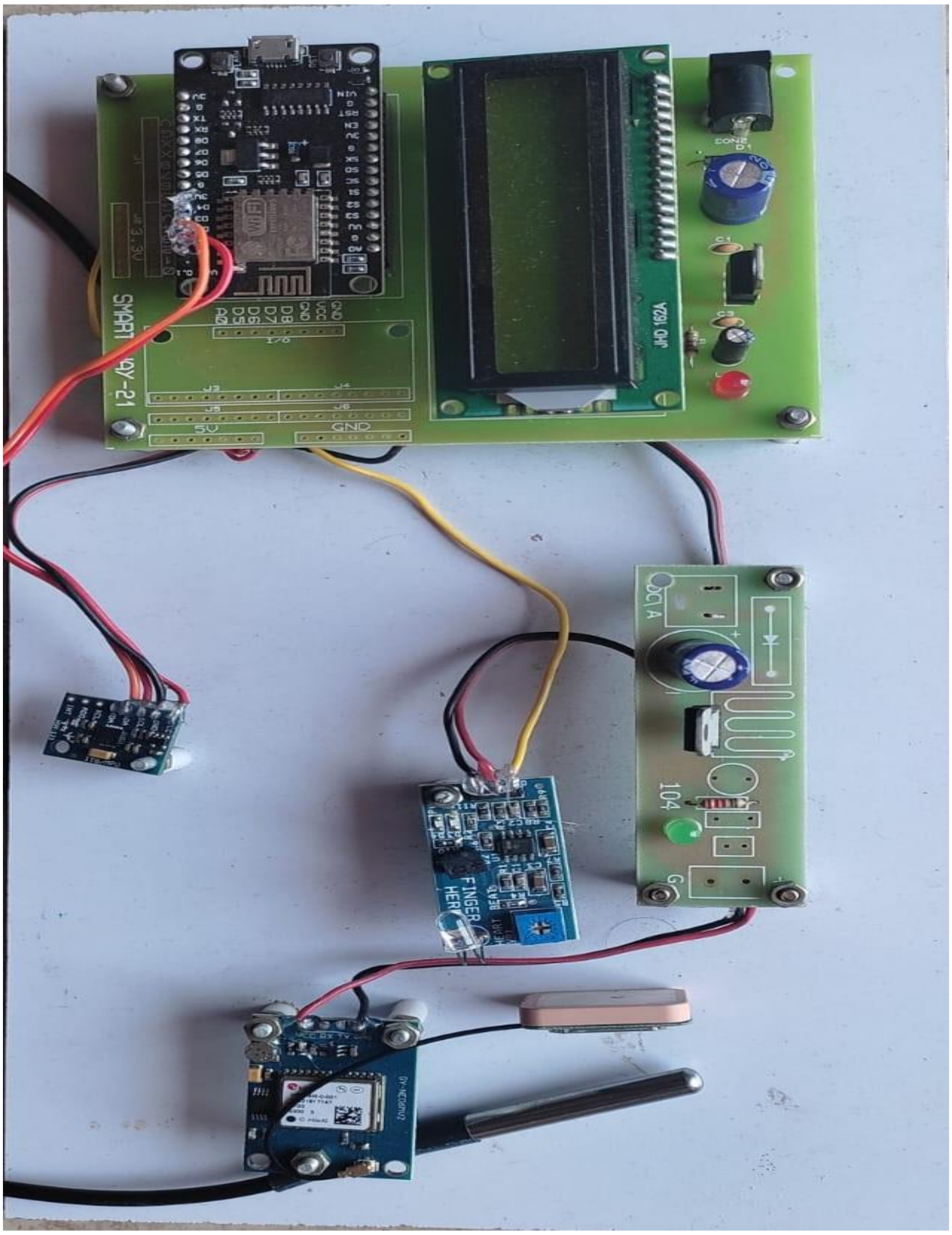


Figure 5.1 Prototype

The below figure shows the working of our prototype which shows the reading of the Heart rate(HB) , Accelerometer(A), and Temperature(T) in a LCD display .

Figure 5.2 LCD Display

In this below figure we have the user interface which receives the live data from the device and uses it to predict whether the cattle is healthy or not. If the model predicts it is unhealthy it will send an alert notification to the user regarding the health and need for quick attention through the message sent to user.

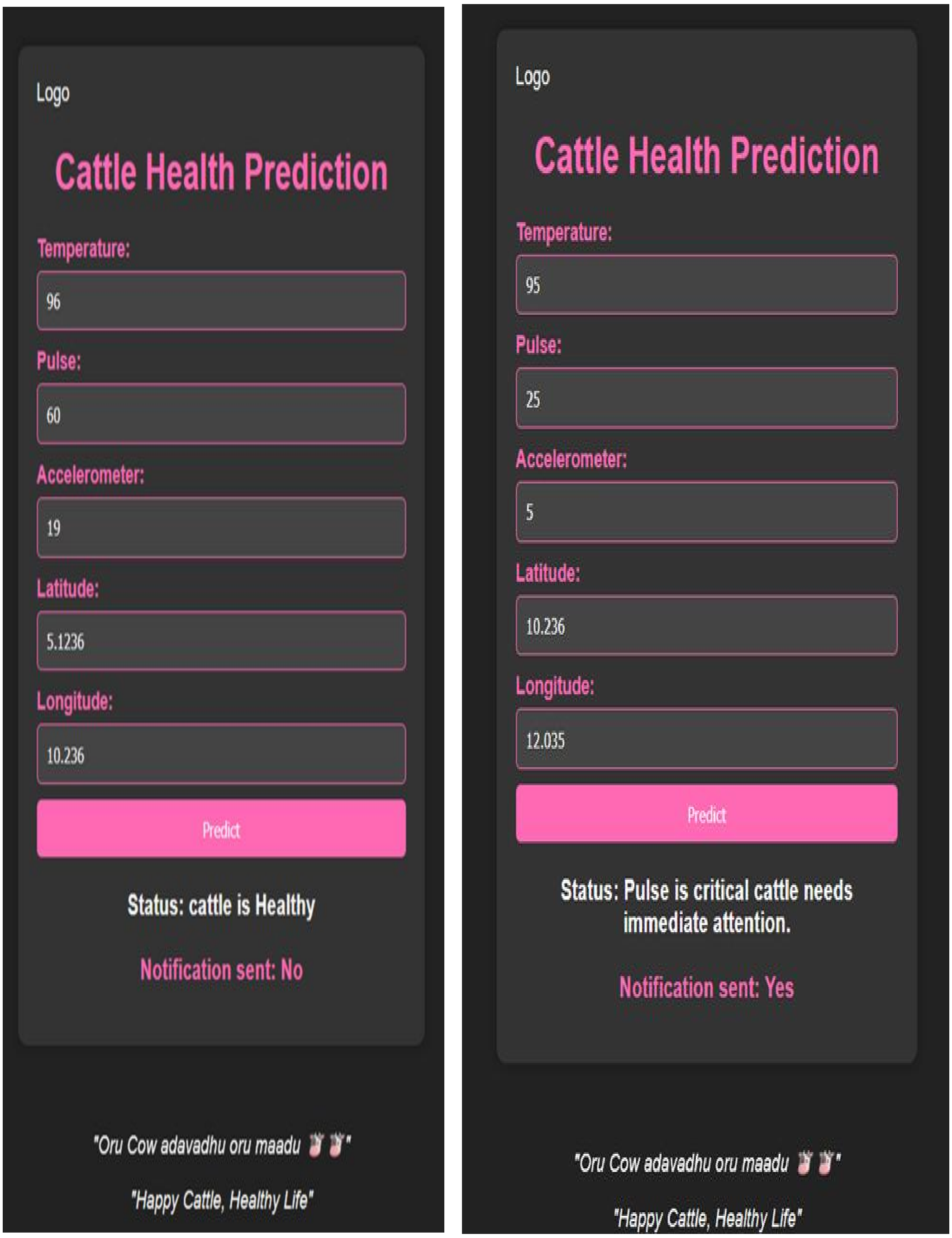


Figure 5.3 Web User Interface

In Figure 5.4 we can see that the user have received a Notification regarding the health of the cattle with the sensor readings and suggests that the cattle needs immediate attention.

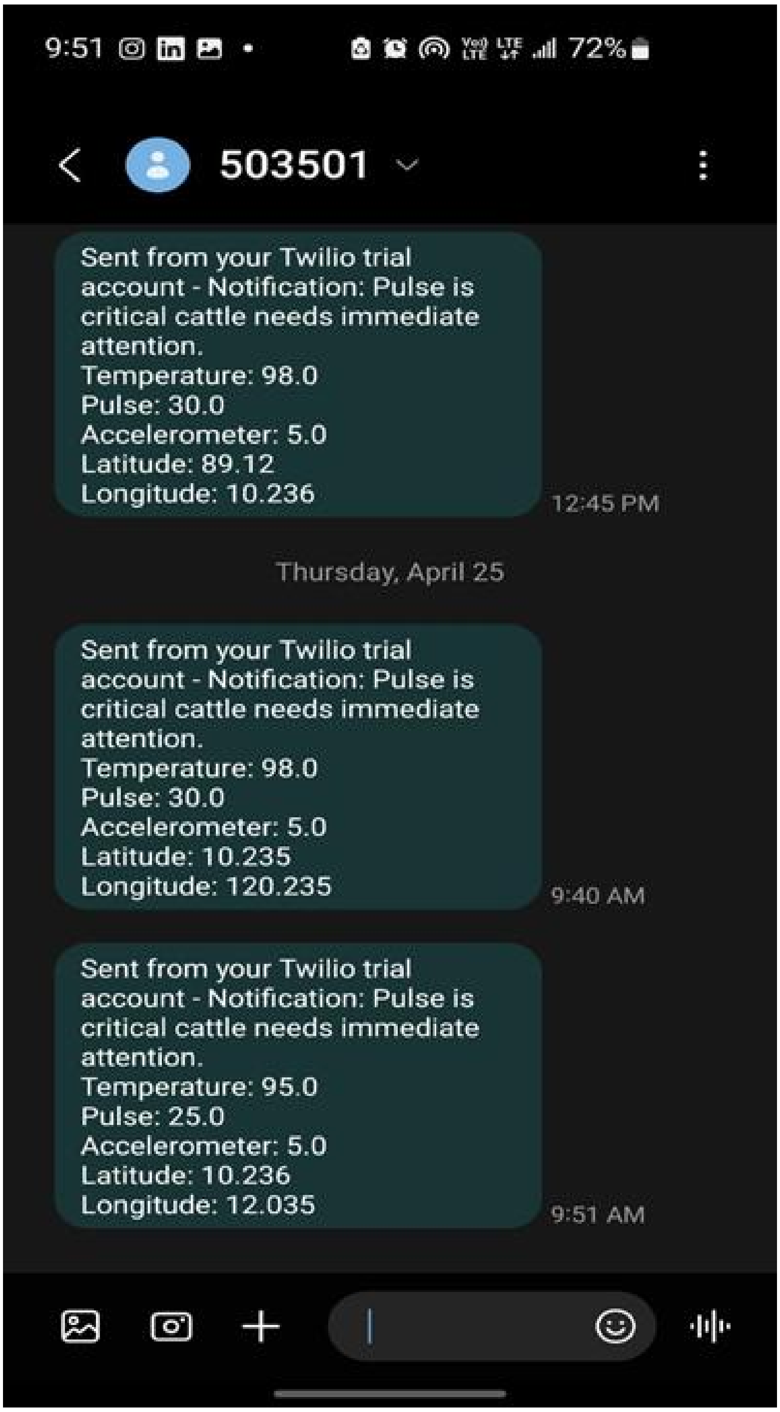


Figure 5.4 Alerting Notification

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

The implementation of IoT sensors and machine learning allows for continuous monitoring of vital parameters, enabling early detection of health issues and timely intervention, ultimately improving the overall welfare of the livestock as well as the production of dairy sources.The automated data collection and analysis reduce the need for manual monitoring, saving time and resources for farmers

6.2 FUTURE WORK

The aspects studied in this research not only present tangible solutions for the daily challenges faced by farmers and barn managers but also lay the foundation for future advancements in cattle monitoring systems. As we look ahead, our focus is on pioneering innovative approaches that not only address existing limitations but also set new standards for farm management and livestock welfare.

Advanced Technology Integration

Building upon the successes of current monitoring systems, our future endeavors will center on the integration of even more advanced technologies. This includes exploring the potential of emerging technologies such as blockchain and edge computing to enhance data security, scalability, and real-time decision-making capabilities. By incorporating these cutting-edge technologies, we aim to create a robust and future-proof monitoring system that can adapt to the evolving needs of modern agriculture.Furthermore, our commitment to innovation extends to leveraging artificial intelligence and machine learning algorithms to analyze.

Enhanced Disease Prediction and Prevention

Disease prevention is a cornerstone of effective livestock management. In our future research, we will delve deeper into the development of predictive models that can accurately forecast disease outbreaks and identify potential risk factors. By analyzing large datasets of historical health data alongside environmental variables, we aim to refine our algorithms to provide earlier and more accurate disease predictions. Additionally, we will explore the integration of precision vaccination and medication strategies based on real-time health monitoring data to further mitigate disease risks.

Holistic Monitoring and Management Solutions

Recognizing the interconnected nature of various aspects of cattle health and well-being, our future work will focus on developing holistic monitoring and management solutions. This entails expanding beyond traditional health metrics to include factors such as stress levels, social interactions, and environmental impact. By incorporating multi-modal sensor data and advanced analytics techniques, we aim to provide farmers with comprehensive insights into their cattle's overall welfare and facilitate informed decision-making across all aspects of farm management.

User-Centric Design and Implementation

In line with our commitment to practical applicability, future efforts will prioritize user-centric design principles to ensure the seamless integration of monitoring systems into existing farm workflows. This includes conducting user surveys and usability studies to understand the specific needs and preferences of farmers and barn managers. By incorporating feedback from end-users throughout the design and implementation we aim to create intuitive and user-friendly monitoring solutions that empower farmers to effectively leverage technology.

Scalability and Interoperability

As the scale of agricultural operations continues to grow, it becomes imperative to design monitoring systems that can seamlessly scale to accommodate large herds and multi-farm environments. In our future research, we will focus on enhancing the scalability and interoperability of cattle monitoring systems to support the diverse needs of farmers with varying herd sizes and management practices. This includes developing standardized data exchange protocols and interoperable APIs to enable seamless integration with existing farm management software and IoT platforms. By prioritizing scalability and interoperability, we aim to ensure that our monitoring solutions can adapt to the evolving needs of modern agriculture and facilitate collaboration and data sharing across the agricultural ecosystem.

In conclusion, the future work outlined above represents an exciting opportunity to push the boundaries of cattle monitoring technology and contribute to the ongoing transformation of agriculture towards more sustainable and efficient practices. Through collaborative research efforts and a commitment to innovation, we are confident in our ability to drive positive change and empower farmers with the tools they need to thrive in an increasingly complex and dynamic agricultural landscape.

APPENDIX I

SOURCE CODE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, confusion_matrix
df = pd.read_csv('animalhealth1000.csv')
df
df.isnull().sum()
plt.figure(figsize=(7,6))
sns.scatterplot(x="Temperature",y="Pulse",data=df)
plt.show()
plt.figure(figsize=(8,4))
sns.histplot(x="Temperature",y="Pulse",data=df)
plt.show()
rcParams['figure.figsize'] = 8,6
plt.bar(df['label'].unique(), df['label'].value_counts(), color = ['red', 'green'])
plt.xticks([2, 1])
plt.xlabel('Target Classes')
plt.ylabel('Count')
```

```
plt.title('Count of each Target Class')
plt.show()
y = df['label']
X = df.drop(['label'], axis = 1)
ss = StandardScaler()
scaled = ss.fit_transform(X)
X_scaled = pd.DataFrame(scaled,columns=X.columns)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size = 0.2,
random_state = 0,stratify=y)
print("X_train :",X_train.shape)
print("X_test :",X_test.shape)
print("y_train :",y_train.shape)
print("y_test :",y_test.shape)
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
confusionmatrix = confusion_matrix(y_test,y_pred)
confusionmatrix
report = classification_report(y_test,y_pred)
print(report)
import urllib
import requests
import re
link="https://blynk.cloud/external/api/get?token=qSUBh7wQMRujpFumxO4cTjEQ
6sWOpojN&V1"
f = requests.get(link)
f1=f.text
f1="".join(c for c in f1 if c.isalnum())
link2="https://blynk.cloud/external/api/get?token=qSUBh7wQMRujpFumxO4cTjE
```

```
Q6sWOpojN&V2"
f = requests.get(link2)
f2=f.text
link3="https://blynk.cloud/external/api/get?token=qSUBh7wQMRujpFumxO4cTjE
Q6sWOpojN&V3"
f = requests.get(link3)
f3=f.text
link4="https://blynk.cloud/external/api/get?token=qSUBh7wQMRujpFumxO4cTjE
Q6sWOpojN&V4"
f = requests.get(link4)
f4=f.text
print(f1)
print(f2)
def spliter(line):
 dim = ['A','H','AQ','CO','G','PIR','PRE']
remove = ['P','T: ','D']
  word = ""+word+""
 line = line.replace(word," ")
 return line.split(" ")
result1 = spliter(f1)
result2 = spliter(f2)
result3 = spliter(f3)
result4 = spliter(f4)
print('Temperature : ',result2[0])
print('Accelerometer : ',result1[1])
print('Latitude : ', result3[0])
print('Longitude : ', result4[0])
print("Pulse value:", result1[0])
sensordata=[[float(result2[0]),float(result1[0]),float(result1[1]),float(f3),float(f4)]]
```

```
print(sensordata)
y_pred = gnb.predict(sensordata)
healthy_pulse_threshold = 40
unhealthy_pulse_threshold = 100
if float(result1[0]) <= healthy_pulse_threshold or float(result1[0]) >=
unhealthy_pulse_threshold:
    print('Pulse rate indicates unhealthy condition')
elif y_pred == 1:
    print('Not Healthy')
    res1 = '1'
elif y_pred == 2:
    print("Healthy")
    res1="2"
```

FLASK

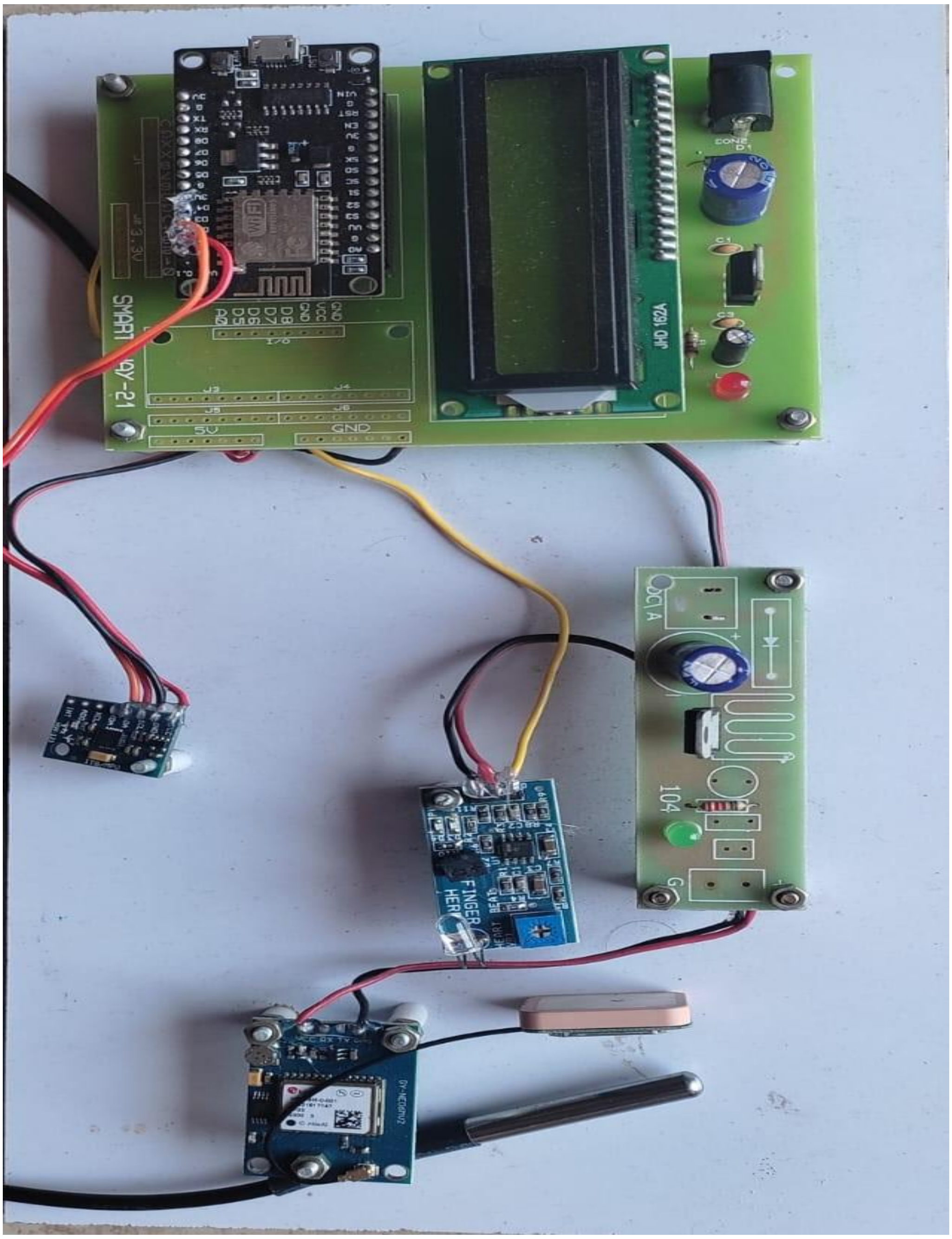
```
from flask import Flask, render_template, request
import joblib
from twilio.rest import Client
app = Flask(__name__)
with open('cattle.pkl', 'rb') as file:
    model = joblib.load(file)
healthy_pulse_threshold = 40
unhealthy_pulse_threshold = 90
account_sid = ""#enter your account sid from twilio here
auth_token ="" #enter your authentication token from twilio here
twilio_phone_number ="" #give your twilio phone number here
client = Client(account_sid, auth_token)
def send_notification(status, temperature, pulse, accelerometer, latitude, longitude):
```

```
 try:
        message = client.messages.create(
            body=f"Notification: {status}\nTemperature: {temperature}\nPulse:
{pulse}\nAccelerometer: {accelerometer}\nLatitude: {latitude}\nLongitude:
{longitude}",
            from_=twilio_phone_number,
            to="+91" #enter the mobile number to receive alert notification )
        print("Notification sent successfully:", message.sid)
        return True
    except Exception as e:
        print("Error sending notification:", e)
        return False
@app.route('/')
def home():
    return render_template('doc.html')
@app.route('/predict', methods=['POST'])
def predict():
    temperature = float(request.form['temperature'])
    pulse = float(request.form['pulse'])
    accelerometer = float(request.form['accelerometer'])
    latitude = float(request.form['latitude'])
    longitude = float(request.form['longitude'])
    prediction = model.predict([[temperature, pulse, accelerometer, latit]]
    if pulse <= healthy_pulse_threshold or pulse >= unhealthy_p:
        status = 'Pulse is critical cattle needs immediate attention.'
    elif prediction == 1:
        status = 'Not Healthy'
    else:
        status = 'cattle is Healthy
```

```
    notification_sent = False
    if status != 'cattle is Healthy':
        notification_sent = send_notification(status, temperature, pulse, accelerometer,
latitude, longitude)
    return render_template('doc.html', status=status,
notification_sent=notification_sent)
if __name__ == '__main__':
 app.run()
```

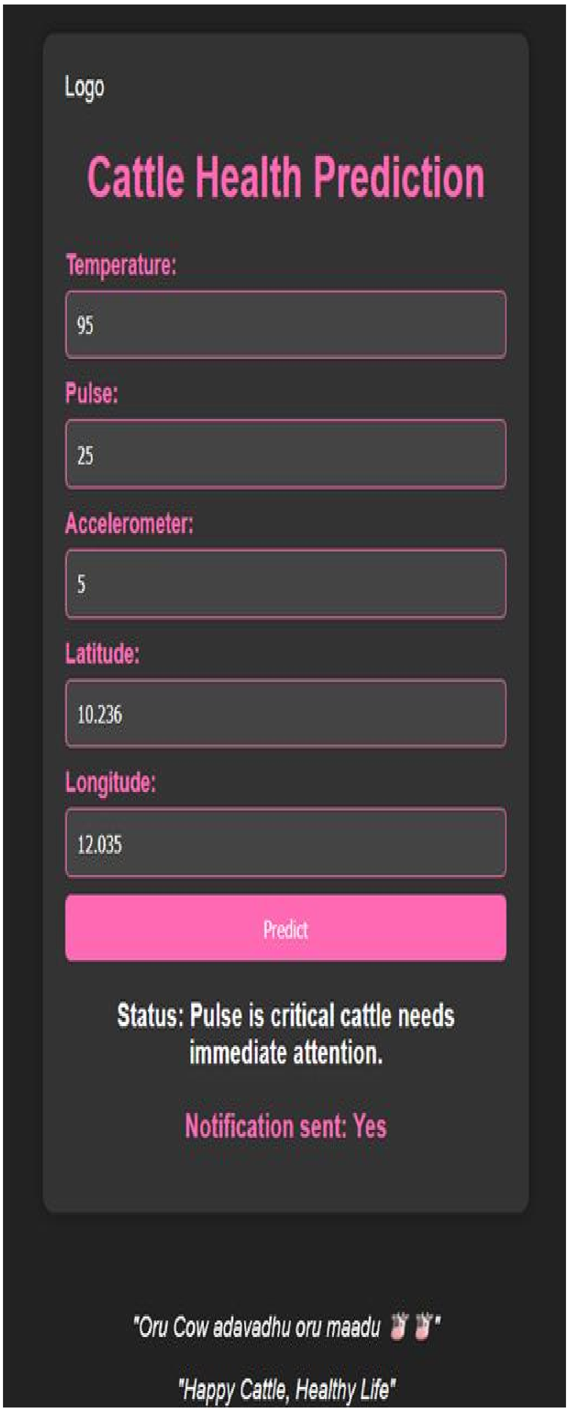
APPENDIX II

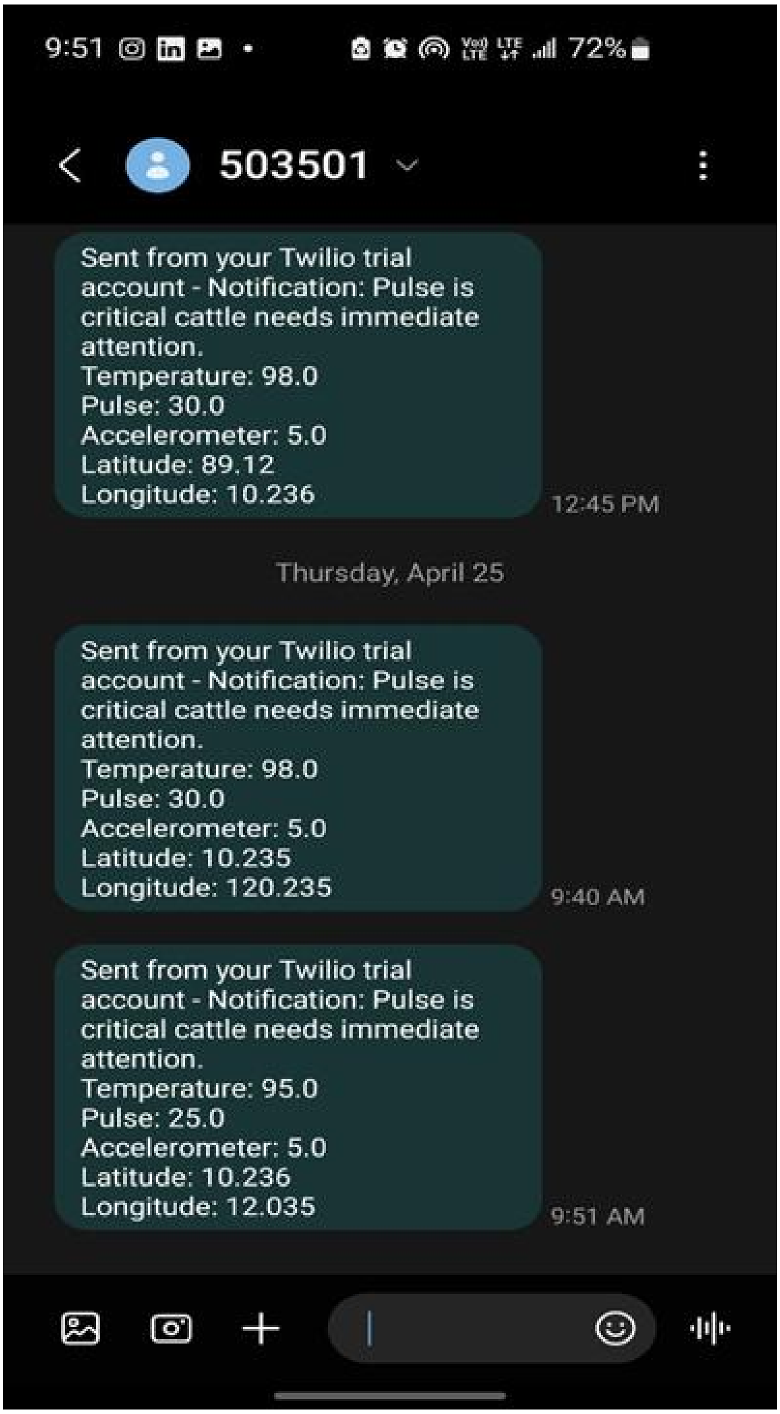
OUTPUT SNIPPETS




HB :069 A:005
T:095.22
JHD 162A
GND
5V
3.3V


Logo
Cattle Health Prediction
Temperature:
95
Pulse:
25
Accelerometer:
5
Latitude:
10.236
Longitude:
12.035
Predict
Status: Pulse is critical cattle needs immediate attention.
Notification sent: Yes
"Oru Cow adavadhu oru maadu"
"Happy Cattle, Healthy Life"


Logo
Cattle Health Prediction
Temperature:
95
Pulse:
25
Accelerometer:
5
Latitude:
10.236
Longitude:
12.035
Predict
Status: Pulse is critical cattle needs immediate attention.
Notification sent: Yes
"Oru Cow adavadhu oru maadu"
"Happy Cattle, Healthy Life"


9:51
72%
503501
Sent from your Twilio trial account - Notification: Pulse is critical cattle needs immediate attention.
Temperature: 98.0
Pulse: 30.0
Accelerometer: 5.0
Latitude: 89.12
Longitude: 10.236
12:45 PM
Thursday, April 25
Sent from your Twilio trial account - Notification: Pulse is critical cattle needs immediate attention.
Temperature: 98.0
Pulse: 30.0
Accelerometer: 5.0
Latitude: 10.235
Longitude: 120.235
9:40 AM
Sent from your Twilio trial account - Notification: Pulse is critical cattle needs immediate attention.
Temperature: 95.0
Pulse: 25.0
Accelerometer: 5.0
Latitude: 10.236
Longitude: 12.035
9:51 AM

REFERENCES

1. Brown, J., & Davis, K. (2021). 'IoT and AI technologies for early disease detection in cattle',Journal of Agricultural and Food Information,Vol.22,No.2,pp.57-64.

2. Chatterjee, P. S., Ray, N. K., & Mohanty, S. P. (2021). 'LiveCare: An IoT based healthcare framework for livestock in smart agriculture', IEEE Transactions on Consumer Electronics, Vol.67, pp.257-265.

3. Chaudhry, A. A., Mumtaz, R., Zaidi, S. M. H., Tahir, M. A., et al. (2020). 'Internet of Things (IOT) and Machine Learning (ML) enabled livestock monitoring', In 2020 IEEE 17th International Conference on Smart Communities: Improving quality of life using ICT, IOT, and AI (HONET) ,pp. 151-155.

4. Cho, B. H. (2021). 'Analysis and design of cattle management system based on IoT', The Journal of the Institute of Internet, Broadcasting and Communication,Vol.21,No.2,pp.125-130.

5. Darvesh, K., Khande, N., Avhad, S., & Khemchandani, M. (2023). 'IOT and AI Based Smart Cattle Health Monitoring', Journal of Veterinary Science & Technology, Vol.14,No.15.,pp. 234-289.

6. Dutta, D., Natta, D., Mandal, S., & Ghosh, N., et al. (2019). ‘Monitor: An IoT based multi-sensory intelligent device for cattle activity monitoring.’, Sensors and Actuators,pp. 234-255.

7. Gupta, R., & Mehta, R. (2020). ‘IoT based real-time cattle health monitoring system’, International Journal of Engineering Research and Technology, Vol.9,No.12, pp.123-128.

8. Habeeb, A. A., Gad, A. E., & Atta, M. A. (2018). ‘Temperature humidity indices as indicators to heat stress of climatic conditions with relation to production and reproduction of farm animals’, International Journal of Biotechnology and Recent Advances, Vol1, 35-50.

9. Kumar, P., et al. (2021). ‘A Smart Gas Leakage Detection System with IoT for Industrial Safety’, International Journal of Innovative Research in Science, Engineering, and Technology, Vol.10,No.1, pp.190-198.

10. Kumar, S., et al. (2019). ‘IoT based smart collar for real-time cattle monitoring in smart farming’, Journal of Internet Technology, Vol.20,No.5, pp.1451-1457.

11. Li, S., Xu, H., & Liu, J. (2019). ‘IoT based smart farming: Agricultural livestock monitoring and disease detection.’, In 2019 IEEE Global Conference on Internet of Things GCIoT, pp. 235-240.

12. Nootyaskool, S., & Ounsrimung, P. (2020). ‘Smart collar design to predict cow behavior’, In 2020 17th International Joint Conference on Computer Science and Software Engineering (JCSSE), pp. 92-97.

13. Rao, V., & Singh, S. (2022). ‘Design and implementation of IoT based livestock health monitoring system.’, 2022 IEEE Conference on Smart Agriculture and Livestock Management (SALM), pp. 101-106.

14. Smith, T., & Johnson, L. (2020). ‘An AI-based approach for cattle health monitoring in smart agriculture’, International Journal of Computational Intelligence Systems , Vol.13, No.3,pp.193-199.

15. Syarif, I., Ahsan, A. S., Harun Al Rasyid, M. U., & Pratama, Y. P., et al. (2019). ‘Health monitoring and early diseases detection on dairy cow based on internet of things and intelligent system’, In 2019 International Electronics Symposium IES ,pp. 183-188.

International Conference On Electronics, Communications and Advanced Processing Systems

ICECAPS 2024

C.ABDUL HAKEEM COLLEGE OF ENGINEERING AND TECHNOLOGY

Hakeem Nagar ,Melvisharam -632509 Ranipet Dt.T.N.
(Approved by AICTE,Affiliated to Anna University,Chennai)

CERTIFICATE OF PARTICIPATION

This is to certify that

Saravana Kumar J

Nehru Institute Of Engineering And Technology

Presented a paper titled

AI And IOT Based Smart Livestock Health Monitoring System

in International Conference on Electronics, Communication and Advanced Processing Systems ICECAPS'24 on 25th April 2024

Organized by

Department of Electronics and Communication Engineering

Convenor | Chair | HOD | Principal

International Conference On Electronics,Communications and Advanced Processing Systems

ICECAPS 2024

C.ABDUL HAKEEM COLLEGE OF ENGINEERING AND TECHNOLOGY
Hakeem Nagar ,Melvisharam -632509 Ranipet Dt.T.N.
(Approved by AICTE,Affiliated to Anna University,Chennai)

CERTIFICATE OF PARTICIPATION

This is to certify that

Arsath Mohammed F

Nehru Institute Of Engineering And Technology

Presented a paper titled

AI and IOT Based Smart Livestock Health Monitoring System

in International Conference on Electronics, Communication and Advanced Processing Systems ICECAPS'24 on 25th April 2024

Organized by

Department of Electronics and Communication Engineering

Convenor	Chair	HOD	Principal

International Conference On Electronics, Communications and Advanced Processing Systems

ICECAPS 2024

C.ABDUL HAKEEM COLLEGE OF ENGINEERING AND TECHNOLOGY

Hakeem Nagar, Melvisharam -632509 Ranipet Dt.T.N.

(Approved by AICTE, Affiliated to Anna University, Chennai)

CERTIFICATE OF PARTICIPATION

This is to certify that

Haribalan C

Nehru Institute Of Engineering And Technology

Presented a paper titled

AI And IOT Based Smart Livestock Health Monitoring System

in International Conference on Electronics, Communication and Advanced Processing Systems ICECAPS'24 on 25th April 2024

Organized by

Department of Electronics and Communication Engineering

T. S. Convenor | Chair | HOD | Principal

International Conference On Electronics, Communications and Advanced Processing Systems

ICECAPS 2024

C.ABDUL HAKEEM COLLEGE OF ENGINEERING AND TECHNOLOGY

Hakeem Nagar ,Melvisharam -632509 Ranipet Dt.T.N.
(Approved by AICTE,Affiliated to Anna University,Chennai)

CERTIFICATE OF PARTICIPATION

This is to certify that

Sunoj Ch

Nehru Institute Of Engineering And Technology

Presented a paper titled

AI And IOT Based Smart Livestock Health Monitoring System

in International Conference on Electronics, Communication and Advanced Processing Systems ICECAPS'24 on 25th April 2024

Organized by

Department of Electronics and Communication Engineering

Convenor	Chair	HOD	Principal