BATCH NO:MAI099

IOT-ENABLED DEVICE FOR PREDICTIVE MONITORING AND DISEASE MANAGEMENT IN COW

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

I.DHINEESH	(21UECS0231)	**(19234)**
T.GANESH	(21UECS0611)	**(19235)**
G.LOKESH REDDY	(21UECS0187)	**(19283)**

Under the guidance of
Ms.E.CHANDRALEKHA,M.Tech,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2025

BATCH NO:MAI099

IoT-ENABLED DEVICE FOR PREDICTIVE MONITORING AND DISEASE MANAGEMENT IN COW

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

I.DHINEESH	(21UECS0231)	**(19234)**
T.GANESH	(21UECS0611)	**(19235)**
G.LOKESH REDDY	(21UECS0187)	**(19283)**

Under the guidance of
Ms.E.CHANDRALEKHA,M.Tech,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2025

CERTIFICATE

It is certified that the work contained in the project report titled ”IOT-ENABLED DEVICE FOR PREDICTIVE MONITORING AND DISEASE MANAGEMENT IN COW ” by ”I.DHINEESH (21UECS0231), T.GANESH (21UECS0611), G.LOKESH REDDY (21UECS0187)” has been car ried out under my supervision and that this work has not been submitted elsewhere for a degree

Signature of Supervisor
Ms. E. Chandhralekha
Assistant Professor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science and Technology
May, 2025

Signature of Head/Assistant Head of the Department
Dr. N. Vijayaraj/Dr. M. S. Murali dhar
Professor & Head/ Assoc. Professor &Assistant Head
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science and Technology
May, 2025

Signature of the Dean
Dr. S P. Chokkalingam
Professor & Dean
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science and Technology
May, 2025

.

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

I.DHINEESH

Date: / /

T.GANESH

Date: / /

G.LOKESH REDDY

Date: / /

APPROVAL SHEET

This project report entitled ”IOT-ENABLED DEVICE FOR PREDICTIVE MONITORING AND DISEASE MANAGEMENT IN COW” by I.DHINEESH (21UECS0231), T.GANESH (21UECS0611), G.LOKESH REDDY (21UECS0187) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Ms.E.Chandralekha,M.Tech

Assistant Professor

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our **Honorable Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (Electrical), B.E. (Mechanical), M.S (Automobile), D.Sc., and Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.,Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, for her blessings.**

We express our sincere thanks to our respected Chairperson and Managing Trustee **Dr. Mrs. RANGARAJAN MAHALAKSHMI KISHORE, B.E., Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology,** for her blessings.

We are very much grateful to our beloved **Vice Chancellor Prof. Dr.RAJAT GUPTA,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean , School of Computing, Dr. S P. CHOKKALINGAM, M.Tech., Ph.D., & Professor & Associate Dean , School of Computing, Dr. V. DHILIP KUMAR,M.E.,Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Professor & Head, Department of Computer Science & Engineering, Dr. N. VIJAYARAJ, M.E., Ph.D.,** and **Associate Professor & Assistant Head, Department of Computer Science & Engineering, Dr. M. S. MURALI DHAR, M.E., Ph.D.,**for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor Ms. E. CHANDRALEKHA, M.Tech**, for her cordial support, valuable information and guidance, he/she helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Dr. SADISH SENDIL MURUGARAJ, B.E., M.E., Ph.D., Professor, Dr. S. RAJAPRAKASH, M.E,Ph.D., Mr. V. ASHOK KUMAR, B.E,M.Tech.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

I.DHINEESH	**(21UECS0231)**
T.GANESH	**(21UECS0611)**
G.LOKESH REDDY	**(21UECS0187)**

ABSTRACT

The dairy and farming industries depend heavily on the health and welfare of cattle, and early disease detection can improve animal welfare while preventing financial losses. An Arduino UNO, an ESP8266 a microcontroller and a network of sensors including those for humidity, temperature, respiratory rate, and heart rate are used in this study to create a combined system that constantly monitors the vital parameters of cows. Real-time sensor data is sent to the ThingSpeak platform, where it is represented using ThingView to enable instantaneous animal condition monitoring. Additionally, a pre-trained machine learning model that has been serialized using Python's Pickle module is incorporated into a Streamlit-based web application to perform binary classification of disease risk. The model, which was trained on a sizable dataset including past and real-time sensor readings, accomplishes prediction accuracies of 80%-95% under controlled conditions. Through the integration of rule-based thresholding and data-driven analytics, the system provides timely health status assessments and personalized intervention recommendations via graphical dashboards. The suggested solution, which is appropriate for both small- and large-scale farming operations, is made to be both cost-effective and scalable. Overall, a strong, automated framework for enhancing cattle management and overall farm operational efficiency is provided by the integration of IoT technologies with cutting-edge machine learning.

Keywords: Animal Health Monitoring, Arduino, Disease Prediction, ESP8266, IoT, Machine Learning, Streamlit, ThingSpeak

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

API	Application Programming Interface
CPU	Central Processing Unit
CSV	Comma-Separated Values
ESP	Espressif Systems
GUI	Graphical User Interface
IoT	Internet of Things
LCD	Liquid-Crystal Display
LR	Logistic Regression
ML	Machine Learning
PKL	Pickle File
RF	Random Forest
SVM	Support Vector Machine
UI	User Interface
URL	Uniform Resource Locator
VS	Visual Studio

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

The delivery of healthcare services has been completely transformed by the quick development of wireless communication and sensor technology. Our project offers a novel approach to ongoing health monitoring in a time when prompt detection and intervention can greatly enhance outcomes. The system is intended to gather real-time data on vital physiological parameters like temperature, heart rate, respiratory rate, and ambient humidity by combining cutting-edge sensors with an Arduino UNO and an ESP8266 microcontroller. After being wirelessly transferred to cloud-based platforms, this data is stored, visualized, and analyzed to give patients and medical professionals useful insights.

A strong architecture that integrates data collection, processing, and predictive analytics is the project's main component. The Arduino UNO is used for sensor data collection and simple signal processing in the first step, and the ESP8266 is used for data transmission. Once in the cloud, the data is shown using intuitive dashboards that provide historical trend analysis and real-time monitoring. The system is further improved by a specialized machine learning pipeline that preprocesses the data, trains a predictive model, and finally classifies health conditions like fever, heat stress, or respiratory infections. This clever strategy promotes preventative healthcare practices in addition to early diagnosis.

Apart from its technical capabilities, the project's design prioritizes accessibility and sustainability. It is a practical solution for both urban hospitals and remote or underserved areas because it uses energy-efficient components and optimized data transmission protocols to minimize the environmental impact. This system has the potential to revolutionize conventional medical practices and make a substantial contribution to better health outcomes globally by bridging the gap between technology and healthcare. The report's subsequent sections will go into greater detail about the project's history, goals, and difficulties while describing the overall plan that underpins this innovative approach to contemporary healthcare.

1.2 Background

Considerable research and development in continuous monitoring systems has been prompted by the growing demand for proactive healthcare solutions. Technology-driven approaches that provide real-time insights are gradually replacing traditional healthcare methods, which frequently rely on sporadic check-ups. Our project, which is based on earlier research and documented implementations, uses a variety of sensors coupled with microcontrollers more precisely, an Arduino UNO and an ESP8266 to continuously record critical physiological parameters like temperature, heart rate, respiratory rate, and ambient humidity. To ensure that patients and healthcare providers have timely access to information for early diagnosis and intervention, the collected data is wirelessly transferred to cloud-based platforms for storage, visualization, and additional analysis.

Reliability and energy efficiency are prioritized in our system's architectural design, as shown in the reference documents that have been uploaded. Even in environments with limited resources, precise data collection is made possible by the use of sophisticated sensors and reliable wireless communication protocols. Furthermore, a crucial shift from reactive to preventive healthcare is represented by the incorporation of a machine learning pipeline for data preprocessing, model training, and prediction. By reducing the need for frequent hospital stays, this strategy not only promotes sustainable healthcare practices but also makes it easier to detect illnesses like fever, heat stress, and respiratory infections early. All things considered, the project addresses contemporary issues with healthcare accessibility and environmental sustainability while expanding upon pre-existing technological frameworks.

1.3 Objective

Designing and implementing an integrated health diagnostic system that seamlessly gathers and evaluates physiological and environmental data in real time is the main goal of this project. The system seeks to provide precise and timely health insights by fusing sensor technologies with cutting-edge wireless communication protocols and cloud-based processing. An Arduino UNO and an ESP8266 microcontroller are used in the project to help collect and transmit data, which is then processed by a specialized machine learning pipeline to forecast health conditions. The goal of this predictive ability is to facilitate early intervention, which could lessen the severity of illnesses and enhance patient outcomes. Additionally, the system aims to be economical and energy-efficient, which makes it appropriate for use in both resource-constrained rural environments and well-equipped urban hospitals.

Additionally, sustainability and scalability are considered in the project's design. By offering intuitive user interfaces and real-time data visualization through cloud platforms, it seeks to facilitate ongoing monitoring. In addition to making important health information more easily accessible, this also helps to lessen the carbon footprint that comes with frequent hospital stays. The project's ultimate goal is to transform preventive healthcare by tackling these complex goals and making high-quality medical diagnostics available to a wide range of people.

1.4 Problem Statement

The absence of proactive and ongoing monitoring systems that can accurately identify early indicators of health decline is one of the major problems facing modern healthcare. Conventional diagnostic techniques frequently depend on routine examinations and reactive actions, which could lead to postponed interventions and increased medical expenses. In order to solve these problems, this project suggests a system that combines sophisticated machine learning algorithms with real-time sensor data collection. By continuously monitoring vital signs, such a system can spot possible health problems before they become serious ones. The need for creative solutions is greater than ever in settings with limited resources because the lack of prompt diagnostics exacerbates health disparities.

Furthermore, integrating technology in healthcare presents a unique set of difficulties, such as energy consumption, system dependability, and data accuracy. Numerous current solutions don't provide a stable, expandable, and long-lasting monitoring framework. By leveraging effective hardware components and a cloud-based infrastructure that guarantees high data fidelity and facilitates real-time analytics, our project seeks to overcome these challenges. Predictive analytics is made possible by integrating a machine learning pipeline, which is essential for prompt medical intervention and early diagnosis. The project hopes to greatly improve the general standard of healthcare delivery and help improve patient outcomes by tackling these important issues.

Chapter 2

LITERATURE REVIEW

[1] Khan et al., 2020 have presented an MSSO-ANFIS model-based healthcare monitoring system for heart disease diagnosis in an IoMT (Internet of Medical Things) cloud environment. To improve the accuracy and speed of medical predictions, the system combines intelligent cloud-based monitoring with machine learning techniques, specifically an enhanced version of the Adaptive Neuro Fuzzy Inference System (ANFIS). Their study demonstrates how IoMT can be used to improve healthcare delivery by enhancing predictive models and offering real-time diagnostic capabilities.

[2] Chatterjee et al., 2021 have presented LiveCare is a framework for IoT-enabled livestock healthcare that combines sensors, cloud storage, and visualization. Their system exemplifies scalable smart agriculture solutions that enable real-time animal monitoring with the specific goal of enhancing the health of livestock. The framework highlights the benefits of remote visualization and cloud-based data storage, which allow for ongoing health monitoring and the early identification of possible health problems in cattle. According to their research, IoT frameworks can improve the productivity and well-being of livestock.

[3] Feng et al., 2021 have presented SocialCattle is an Internet of Things (IoT) system that uses behavioral sensing to identify mastitis in cattle. This novel approach uses the social interaction patterns of cattle to identify early symptoms of disease, especially mastitis, a common condition that affects dairy cows. The system reduces the need for manual intervention and enables proactive health management by combining behavioral sensing with cloud-based analytics to provide real-time disease detection.

[4] Nagasubramanian et al., 2021 have presented an IoT-based pattern recognition system combined with an ensemble classification framework for the detection of crop diseases. To identify crop diseases, their system integrates IoT sensors with

a variety of machine learning approaches, including ensemble learning, decision trees, and support vector machines. The system gives farmers a more dependable and effective method of crop management by processing real-time sensor data and applying machine learning to forecast possible disease outbreaks.

[5] Patle et al., 2021 have presented LSTM (Long Short-Term Memory) networks are used in a field-tested smart sensor system to predict plant diseases. Based on real-time sensor data, the system predicts plant disease outbreaks using sophisticated machine learning models such as LSTM networks. By giving farmers precise forecasts and assisting them in taking preventive action before an outbreak happens, their method shows enhanced forecasting capabilities under actual farming conditions.

[6] Costa et al., 2021 have presented a thorough analysis of precision livestock technologies with an emphasis on how they can be used to manage dairy calves and increase agricultural output. In order to increase productivity, animal welfare, and resource efficiency in dairy farms, the paper explores a number of technologies, such as sensors, data analytics, and Internet of Things systems, and how they can be incorporated into precision livestock management.

[7] Dutta et al., 2022 have presented MOOnitor is a multi-sensory, Internet of Things-enabled tool for tracking cattle activity. Accelerometers and physiological sensors are integrated into the system to provide real-time cattle behavior monitoring and continuous health tracking. Cattle welfare and farm management can be enhanced by using the data gathered to track animal health, identify behavioral changes, and anticipate possible health problems.

[8] Qazi et al., 2022 have presented a critical analysis of smart agriculture enabled by AI, emphasizing present obstacles and potential paths for IoT-based farming technologies. With an emphasis on issues like data integration, sensor calibration, and real-time data processing, their paper offers a thorough examination of the application of AI and IoT in agriculture. They talk about how AI could improve agricultural practices' sustainability and efficiency.

[9] Arshad et al., 2022 have presented the installation of a wireless sensor network

to monitor animals intelligently. Temperature, heart rate, and movement patterns are just a few of the health data that the system continuously gathers and sends via wireless sensors. Particularly in remote locations where conventional monitoring techniques are impractical, this real-time data is utilized to produce actionable insights for improved livestock health management and more productive farming practices.

[10] Arshad et al., 2023 have presented a safe system for tracking the health of cattle that incorporates encryption protocols and intelligent modules to ensure data integrity. To guarantee the privacy and security of the data gathered from cattle health monitoring sensors, this system integrates cutting-edge security measures. Real-time data analysis is made possible by the system's intelligent modules, which also guarantee data security and integrity and enable the early identification of health problems.

[11] Rajendran et al., 2023 have presented a wireless sensor-based IoT-based cattle health tracking system. The system's main goal is to enhance livestock condition monitoring in isolated locations where conventional medical techniques are impractical. The system facilitates prompt interventions and improved animal health management by employing wireless sensors to gather information on cattle health parameters, including body temperature and movement.

[12] Casella et al., 2023 have presented a framework based on optimization and machine learning for the early detection of respiratory diseases in cows. Their system identifies health risks in cattle, especially respiratory diseases, using machine learning algorithms and historical data. By using optimization techniques to increase prediction accuracy, the framework gives farmers a tool to identify and reduce health risks before they become more serious.

[13] Moutaouakil et al., 2023 have presented a survey describing the significance of data integration for more intelligent decision-making on digital farming dashboards and Internet of Things cattle monitoring systems. Their work highlights how integrated data dashboards, which enable farmers to view and analyze data from multiple IoT sensors in real-time, are becoming increasingly important in the agricultural sector. The study emphasizes how these systems facilitate data-driven

decision-making to improve livestock health management and farm productivity.

[14] Poyyamozhi et al., 2024 have presented a thorough analysis of IoT adoption in smart buildings, covering its uses, difficulties, and potential for energy management in the future. Their review offers insights into the wider applications of IoT technologies, including their potential in agricultural and livestock management systems, despite its primary focus on energy management in smart buildings.

[15] Islam et al., 2024 have presented a review of state-of-the-art fish disease control technologies. Using sensor-enabled systems that track fish health, behavior, and water quality, the study focuses on early disease detection in aquaculture. Through early disease detection and prompt intervention, their research highlights how IoT and sensor technologies can improve aquaculture systems' sustainability and productivity.

2.1 Existing System

Conventional healthcare monitoring usually depends on manual examinations and standalone medical equipment, which frequently don't provide continuous or real-time data for prompt action. Some systems have started incorporating wearable sensors and remote monitoring platforms in recent years, but these solutions are typically quite small in scope. Many current systems only record one parameter or necessitate laborious configurations that impede usability and scalability. Additionally, despite the emergence of a number of IoT-based platforms that use cloud platforms like ThingSpeak for visualization and Arduino UNO and ESP8266 for data collection, these platforms frequently lack integrated machine learning pipelines for predictive analysis. Due to this fragmentation, reactions are delayed and it is more difficult to predict serious health issues like fever, heat stress, or respiratory infections before they worsen.

2.2 Related Work

In both agriculture and healthcare, IoT-based monitoring systems have drawn a lot of interest. In order to improve accuracy and decrease response time, Khan et al. [1] used a modified ANFIS model in their healthcare monitoring system for heart disease diagnosis in an IoMT cloud environment. LiveCare, an IoT-enabled framework for livestock health monitoring, was proposed by Chatterjee et al. [2] in the agricultural domain. It integrates sensors, cloud storage, and visualization for ongoing health management. In order to promote early disease detection and proactive health management, Feng et al. [3] introduced SocialCattle, an Internet of Things-based system that uses behavioral sensing to identify mastitis in cattle. For continuous cattle health tracking that ensures real-time monitoring, Dutta et al. [4] created MOOnitor, an Internet of Things device with multisensory capabilities. Arshad et al. [5] demonstrated the potential of IoT in real-time health management by deploying wireless sensor networks for animal health monitoring. The role of IoT in livestock management was emphasized by Rajendran et al. [6] and Casella et al. [7], who emphasized the use of wireless sensors and machine learning for early disease detection in cattle. Regarding plant diseases, Patle et al. [8] employed LSTM networks to accurately predict plant diseases, and Nagasubramanian et al. [9] advanced the use of IoT in agriculture by combining IoT with ensemble learning to identify crop diseases. In their review of precision livestock technologies, Costa et al. [10] highlighted the ways in which sensors, data analytics, and the Internet of Things enhance animal welfare and productivity. In their critical review of AI-enabled smart agriculture, Qazi et al. [11] covered the challenges and potential paths for IoT in agriculture. A secure cattle health monitoring system with encryption protocols to guarantee data integrity was presented by Arshad et al. [12]. Digital farming dashboards are essential for better decision-making through data integration, according to Moutaouakil et al. [13]. In their review of IoT adoption in smart buildings, Poyyamozhi et al. [14] offered insights into the wider uses of IoT technologies in agricultural systems. Finally, Islam et al. [15] investigated sensor-enabled systems for early disease detection in aquaculture with an emphasis on fish disease control.Together, these studies demonstrate the advantages of combining cloud computing, IoT, and machine learning in healthcare and agriculture while also identifying domain-specific issues that need more research.

2.3 Research Gap

IoT-based monitoring and predictive analytics have made great strides, but there are still a number of important gaps. Without offering an integrated framework that combines ongoing data collection with real-time predictive analytics, many current systems function in silos, concentrating on distinct monitoring facets. The implementations of Khan et al. [1] and Chatterjee et al. [2] frequently lack smooth integration between hardware components, cloud data management, and intelligent decision-making modules, despite the fact that they have established solid foundations in healthcare and livestock monitoring, respectively. Early disease detection through machine learning models is possible with systems like those suggested by Feng et al. [3] and Nagasubramanian et al. [4], but they still face difficulties like sensor calibration and data heterogeneity. Similarly, Patle et al.'s deep learning methods [5] and Costa et al.'s review of precision livestock frameworks [6] highlight important developments but also point to shortcomings in terms of scalability, adaptability, and energy efficiency. Furthermore, there is still a clear gap in systems that combine various sensor inputs with strong encryption and real-time analytics, even after wireless sensor networks were implemented in research by Dutta et al. [7] and Arshad et al. [8, 10]. The fragmented nature of current IoT applications across various domains is further highlighted by additional reviews by Qazi et al. [11], Moutaouakil et al. [13], and Poyyamozhi et al. [14]. Islam et al. [15] show that comprehensive early detection systems are scarce even in aquaculture. This project intends to close these enduring gaps by creating a comprehensive monitoring solution that integrates several physiological sensors with an Arduino-based CPU, enables wireless data transfer via ESP8266, supports cloud visualization, and uses a state-of-the-art machine learning pipeline. In order to overcome the drawbacks of the current IoT-based monitoring systems, the suggested integrated approach aims to increase system scalability, guarantee energy efficiency, and enhance personalized care and predictive accuracy.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The majority of the healthcare monitoring systems in use today rely on standalone medical devices and routine manual examinations. These systems frequently consist of discrete sensors or gadgets that record just one or two critical parameters, like temperature or heart rate, without providing ongoing or real-time monitoring. A lot of these systems use simple wireless communication protocols to send data to cloud platforms, but they usually don't integrate machine learning and advanced analytics, which makes it harder for them to anticipate possible health problems before they become serious. Because these systems are fragmented, data is rarely centralized, which results in ineffective care and delayed responses. Furthermore, traditional systems are less appropriate for settings that demand prompt medical interventions and quick decision-making because they frequently have problems with sensor calibration, limited scalability, and a high reliance on manual interventions.

In addition to these difficulties, existing systems are frequently limited by inadequate data integration and high energy consumption. Because the hardware is typically not built to run continuously for extended periods of time, calibration and maintenance problems are common. Furthermore, a lot of systems lack sophisticated cloud platforms like ThingSpeak for effective data visualization and storage, as well as strong real-time data processing. This fragmentation delays the detection of important health events and further diminishes the efficacy of predictive analytics. These drawbacks collectively highlight the necessity of a centralized, cutting-edge system that can effectively combine data gathering, transmission, and predictive analysis for proactive healthcare management.

3.2 Proposed System

By combining a full range of sensors with an Arduino UNO and an ESP8266 microcontroller for smooth data collection and transfer, the suggested system aims to address the shortcomings of the available healthcare monitoring options. Vital physiological parameters, including temperature, heart rate, respiration rate, and ambient humidity, are continuously monitored by this system, which sends real-time data to the ThingSpeak cloud platform. Preprocessing sensor data, training predictive models, and accurately classifying conditions into Normal, Fever, Heat Stress, or Respiratory Infection are all made possible by the system's automated machine learning pipeline. Early identification of health problems is made easier by this integration, allowing for prompt interventions that can greatly enhance patient outcomes and cut down on needless hospital stays.

By using reliable wireless communication protocols and low-power hardware components, the suggested system also prioritizes scalability and energy efficiency. Through interactive dashboards created with Streamlit, healthcare providers can remotely monitor and analyze patient data thanks to its cloud-based architecture, which is based on ThingSpeak. Both patients and medical professionals can easily access vital health insights thanks to the smooth integration of real-time data analytics with intuitive visualization tools. This all-encompassing strategy improves diagnostic precision and system dependability in addition to addressing the drawbacks of conventional monitoring systems, opening the door for a more proactive and effective healthcare delivery model.

3.3 Feasibility Study

The feasibility study of the project attests to the viability and sustainability of combining machine learning and Internet of Things-based sensor technology for healthcare monitoring. The hardware implementation is financially feasible due to the availability of inexpensive sensors, microcontrollers such as the Arduino UNO, and Wi-Fi modules like the ESP8266. A strong infrastructure that facilitates real-time data analytics is provided by using ThingSpeak as a cloud platform for data storage, visualization, and remote monitoring. The system's flexibility and cost-effectiveness are further increased by the use of open-source Python libraries for the machine learning pipeline. In general, this feasibility study shows that the proposed solution can provide a strong platform for ongoing and proactive patient monitoring while also meeting the technical, financial and social demands of modern healthcare.

3.3.1 Economic Feasibility

Economically speaking, the suggested system provides a reasonably priced option for medical monitoring. Compared to conventional high-end medical equipment, initial investment is greatly reduced by using inexpensive hardware components such as the Arduino UNO and ESP8266. High performance is ensured without incurring prohibitive costs through the integration of readily available sensors. Additionally, ThingSpeak's scalable pricing, which corresponds to usage levels, keeps operating costs under control when used as a cloud service. An integrated machine learning pipeline that automates data processing and predictive analysis reduces labor costs and eliminates the need for continual human supervision. The suggested system is financially appealing for a variety of healthcare applications due to its low initial costs, scalable operating costs, and the possibility of early detection of health problems, which can reduce long-term healthcare costs.

3.3.2 Technical Feasibility

Given the current developments in machine learning and the Internet of Things, the suggested system is technically very feasible. The system makes use of reputable hardware parts that are well-known for their dependability and simplicity of integration, like the Arduino UNO and ESP8266. Scalability and resilience are further improved by using ThingSpeak as a cloud-based platform for real-time data visualization and storage. Furthermore, accurate predictive analytics and the effective processing of massive amounts of sensor data are guaranteed by the integration of a machine learning pipeline using Python libraries like TensorFlow and Scikit-learn. The system is flexible enough to accommodate changing healthcare requirements thanks to its modular design, which also permits future enhancements and the addition of new sensors or features. All things considered, the technical elements and design techniques used in this project highlight its potential for effective practical application.

3.3.3 Social Feasibility

The potential of the suggested system to greatly improve healthcare accessibility and care quality makes it socially feasible. This system provides continuous, real-time monitoring that can result in prompt medical interventions in a time when proactive management and early detection of health conditions are critical. Both patients and healthcare professionals can access vital health information thanks to the intuitive interfaces created with Streamlit and remote data access via ThingSpeak. People who live in rural or underdeveloped areas, where access to conventional healthcare services may be restricted, will especially benefit from this. The system encourages a more preventative approach to healthcare by enabling early diagnosis and decreasing the need for manual check-ups.The broad use of this technology could reduce the strain on medical facilities and enhance patient outcomes in a variety of communities, which would have a significant overall impact on public health.

3.4 System Specification

The following elements and characteristics are included in the system specification for the suggested project:

- **Microcontroller:** Arduino UNO for gathering and preliminary processing of sensor data.
- **Wireless Communication:** ESP8266 Wi-Fi module for transmitting data in real time.
- **Sensors:** Temperature, heart rate, respiratory rate, and ambient humidity are all measured by multiple sensors.
- **Cloud Platform:** ThingSpeak for remote monitoring, data visualization, and storage.
- **Machine Learning:** Integration of preprocessing, model training, and predictive analytics and model building using Logestic Regression, Support Vector Machine(SVM), Random Forest.
- **Data Integration:** Fusion and visualization of real-time data for proactive healthcare motioning.

- **Energy Efficiency:** components with low power consumption and data transmission protocols that are optimized.
- **Scalability:** Modular architecture to facilitate future growth and the addition of new sensors or features.
- **User Interface:** Streamlit was used to create an intuitive dashboard that allows end users and healthcare professionals to monitor and analyze data.

3.4.1 Tools and Technologies Used

The creation of the suggested system makes use of a number of innovative tools and technologies:

- **Hardware:** Arduino UNO, ESP8266, and Heart Rate Sensor, Humidity Sensor, Temperature Sensor, Respiratory Sensor.
- **Software:** Python is the programming language used to create the data processing algorithms and machine learning pipeline.
- **Cloud Platform:** ThingSpeak for managing, storing, and visualizing data remotely.
- **Development Environment:** VS Code and Command Prompt for writing, testing, and executing code.
- **User Interface Development:** Streamlit for making dashboards that are both interactive and easy to use.

3.4.2 Standards and Policies

VS Code and Command Prompt

With support for Python and other programming languages, Visual Studio Code is a flexible code editor that is enhanced by the Command Prompt for running and debugging code. When creating intricate IoT and machine learning applications, this environment expedites the development process and boosts productivity..

Standard Used: ISO/IEC 27001

Streamlit

The open-source Python framework Streamlit makes it easier to create interactive data applications quickly. End users and medical professionals can more easily access vital information by using it to create responsive dashboards for real-time monitoring and visualization of sensor data.

Standard Used: ISO/IEC 27001

ThingSpeak

A cloud-based platform called ThingSpeak makes it possible to store data, visualize it in real time, and remotely monitor sensor data. It is the perfect way to integrate IoT devices with sophisticated data analytics and machine learning frameworks because of its scalable and intuitive interface.

ThinkView

ThinkView is a visualization tool that enhances analytics and decision-making in Internet of Things-based systems by enabling real-time monitoring and graphical data representation.

Standard Used: ISO 9241-11

Chapter 4

SYSTEM DESIGN AND METHODOLOGY

4.1 System Architecture

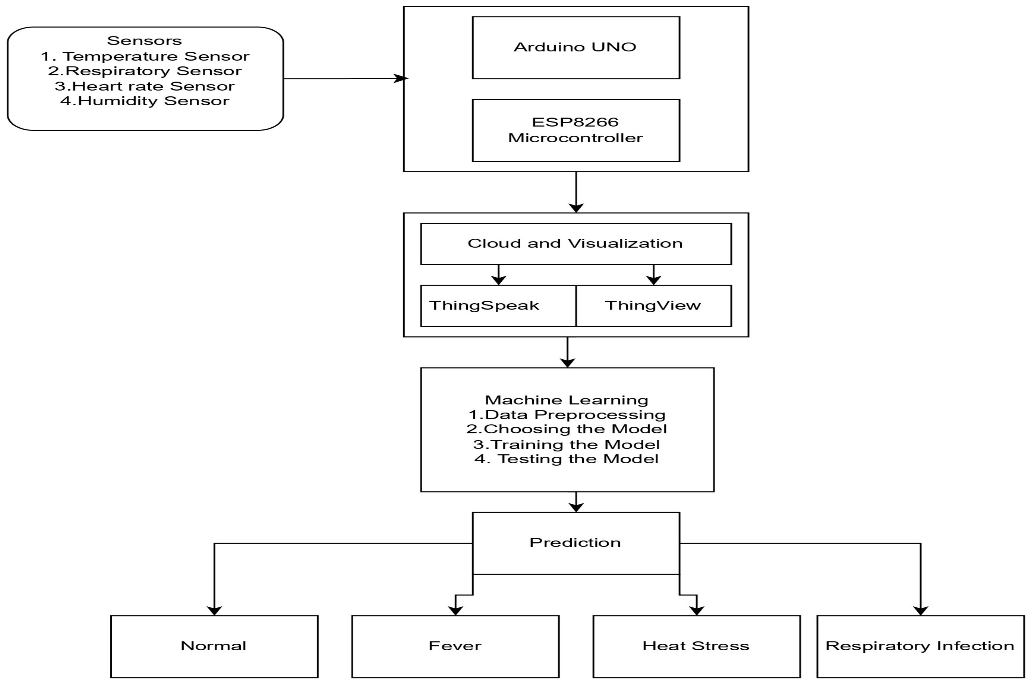

Figure 4.1: **Architecture Diagram for Cow Health Monitoring**

Figure 4.1 describes about how system collects real-time data on environmental conditions and vital signs using a number of sensors, including temperature, respiratory, heart rate, and humidity. These signals are processed by an Arduino UNO, which then sends them to an ESP8266 micro controller for wireless transmission to cloud platforms (ThingSpeak and ThingView). Raw sensor data is transformed into useful insights by a machine learning pipeline that includes data pre-processing, model selection, training, and testing. In order to predict conditions like normal, fever, heat stress, or respiratory infection, users can also enter extra information that is combined with the sensor data. Continuous monitoring, real-time visualization, and accurate health predictions are made possible by the entire setup.

4.2 Design Phase

4.2.1 Data Flow Diagram

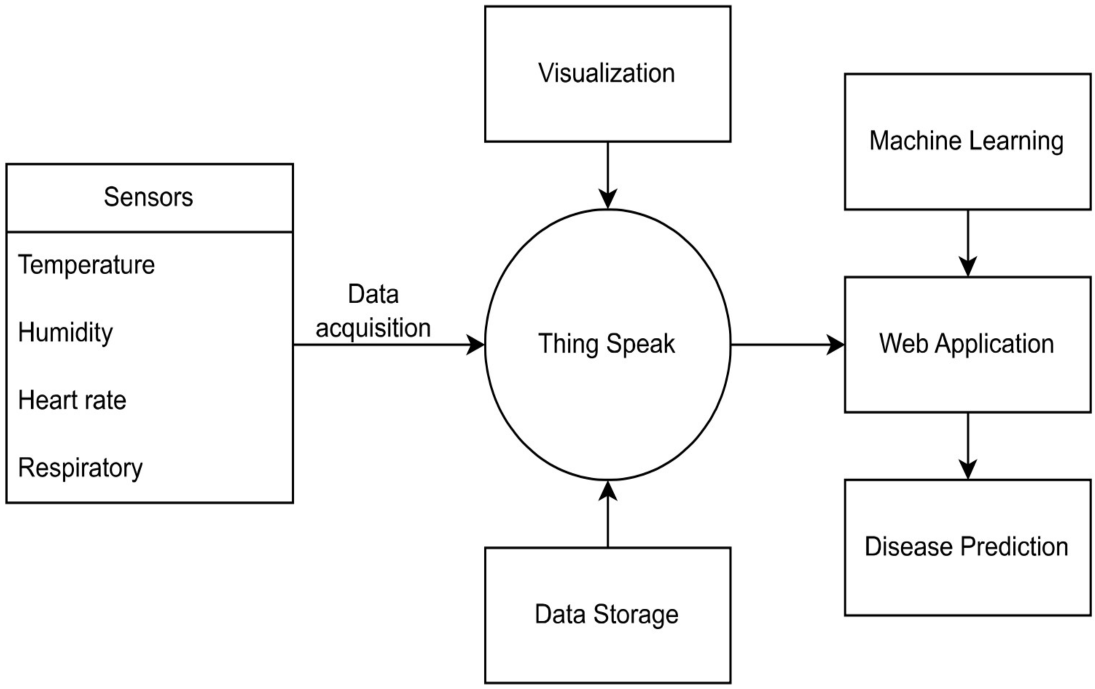

Figure 4.2: **Data Flow Diagram for Cattle Health Monitoring**

Figure 4.2 provides information flow within the Internet of Things-based cattle health monitoring system is depicted in this data flow diagram. Temperature, humidity, heart rate, and respiratory rate are among the sensor readings that are gathered and transmitted to ThingSpeak, which acts as the hub for data storage and real-time visualization. Additionally, the Machine Learning module processes the data and communicates with the Web Application to provide health predictions (e.g., detecting respiratory infections or fever). This organized process guarantees that all parties involved are aware of how information is gathered, saved, examined, and eventually utilized to make prompt decisions about the health of cattle.

4.2.2 Use Case Diagram

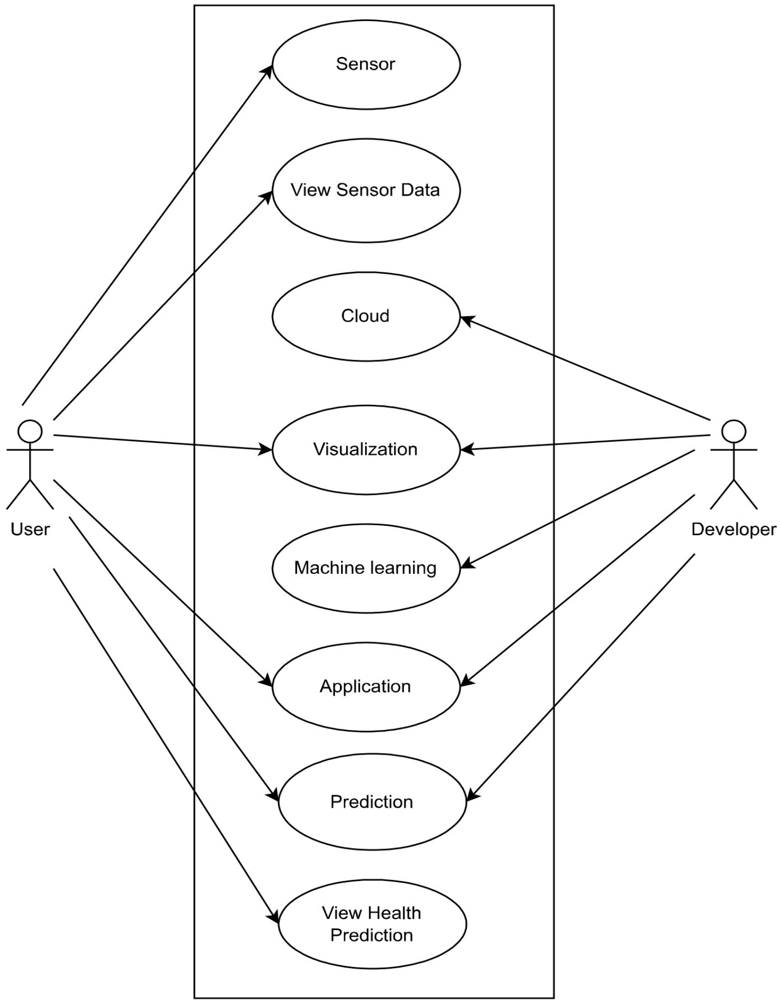

Figure 4.3: **Use Case Diagram for Cattle Health Monitoring System**

Figure 4.3 shows the interaction between the User and Developer and the cattle health monitoring system is depicted in this use case diagram. Predictive health outcomes, real-time analytics visualization, cloud-based information monitoring, and sensor data viewing are all available to the User. The Developer is in charge of cloud integration, machine learning models, and application logic maintenance and updates. In order to ensure that both user-centric functions (such as viewing health predictions) and technical tasks (such as maintaining the codebase) are clearly defined, the diagram defines these roles and interactions.

4.2.3 Class Diagram

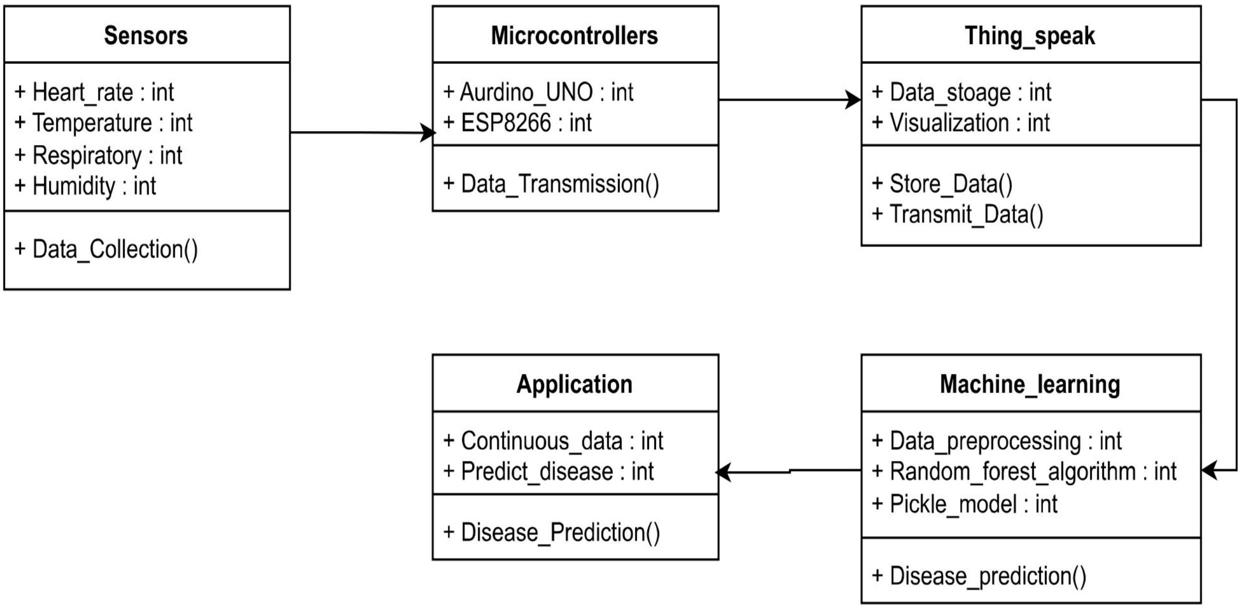

Figure 4.4: **Class Diagram for IoT-based Cattle Health Monitoring System**

Figure 4.4 shows the interactions between the primary elements of the Internet of Things-based cattle health monitoring system are depicted in this class diagram. The vital physiological data is recorded by the Sensors class (heart rate, temperature, respiratory rate and humidity). Data transmission to the cloud is handled by the Microcontrollers class (ESP8266, Arduino UNO). The data is saved and displayed once in Thing_speak, allowing for real-time monitoring. Constant data processing and prediction are handled by the Application class, which integrates with the Machine_learning class, which includes model training (for example, Random Forest), data preprocessing and health prediction functions. The overall scalability of the system and ease of maintenance are ensured by this modular design, which emphasizes the distinct division of duties.

4.2.4 Sequence Diagram

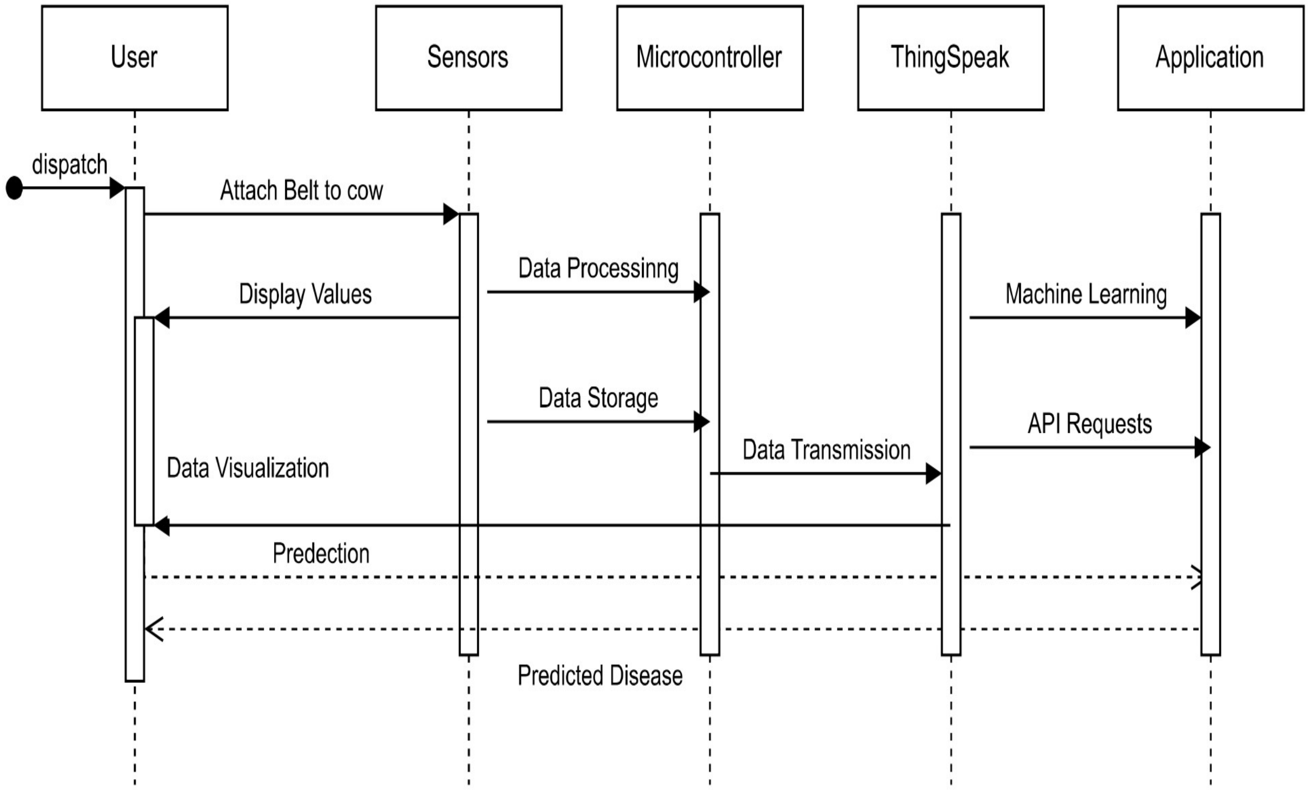

Figure 4.5: **Sequence Diagram for Cattle Health Monitoring System**

Figure 4.5 shows the sequential relationships between the main elements of the cattle health monitoring system are depicted in this sequence diagram. The process is explained in detail, beginning with the device's sensor data collection (heart rate, temperature, etc.), followed by data transmission to ThingSpeak, machine learning model processing, and real-time visualization on the Streamlit dashboard. Accurate and timely health status predictions are made possible by this sequential flow, which guarantees that each process proceeds in a timely and coordinated manner. used to promptly decide on matters pertaining to the health of cattle.

4.2.5 Collaboration Diagram

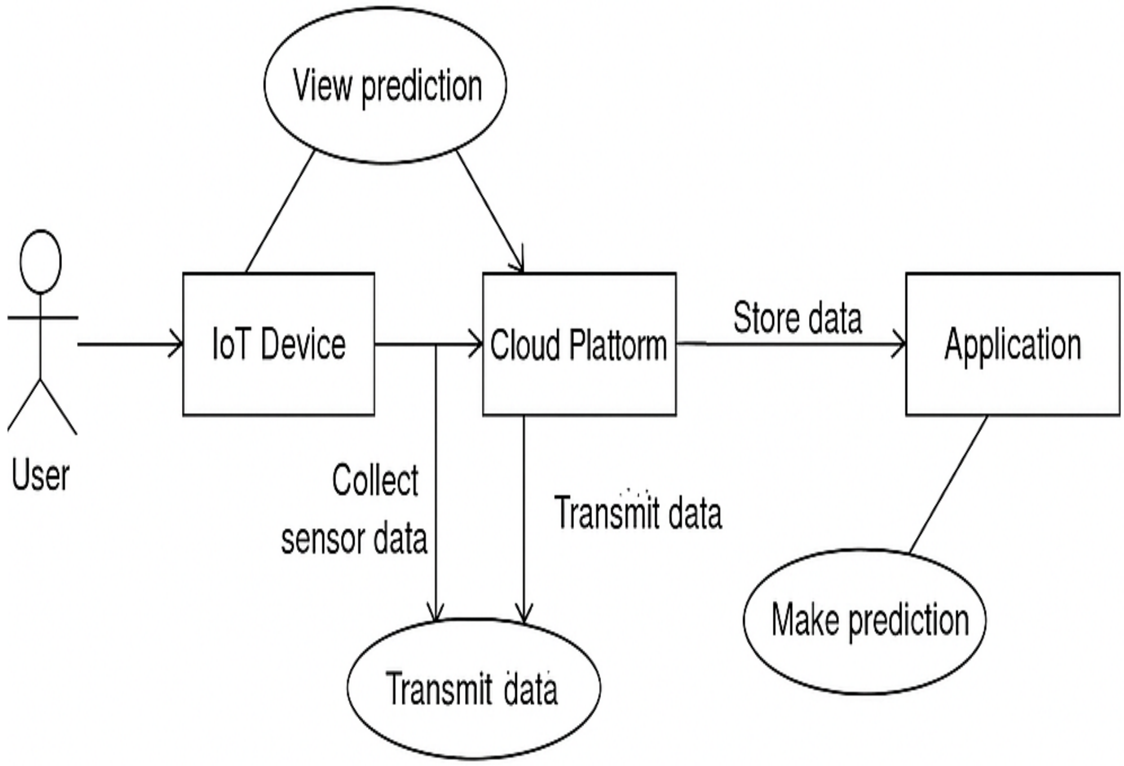

Figure 4.6: **Collaboration Diagram for Cattle Health Monitoring System**

Figure 4.6 shows the structural connections and interactions between the various components of the cattle health monitoring system are shown in this collaboration diagram. It demonstrates how the machine learning module, cloud services (ThingSpeak), sensor modules, microcontrollers, and the Streamlit interface collaborate to process and present health data. By highlighting the communication channels and interdependencies between these elements, the diagram guarantees a coherent system architecture that facilitates strong data flow and accurate forecasts.

4.2.6 Activity Diagram

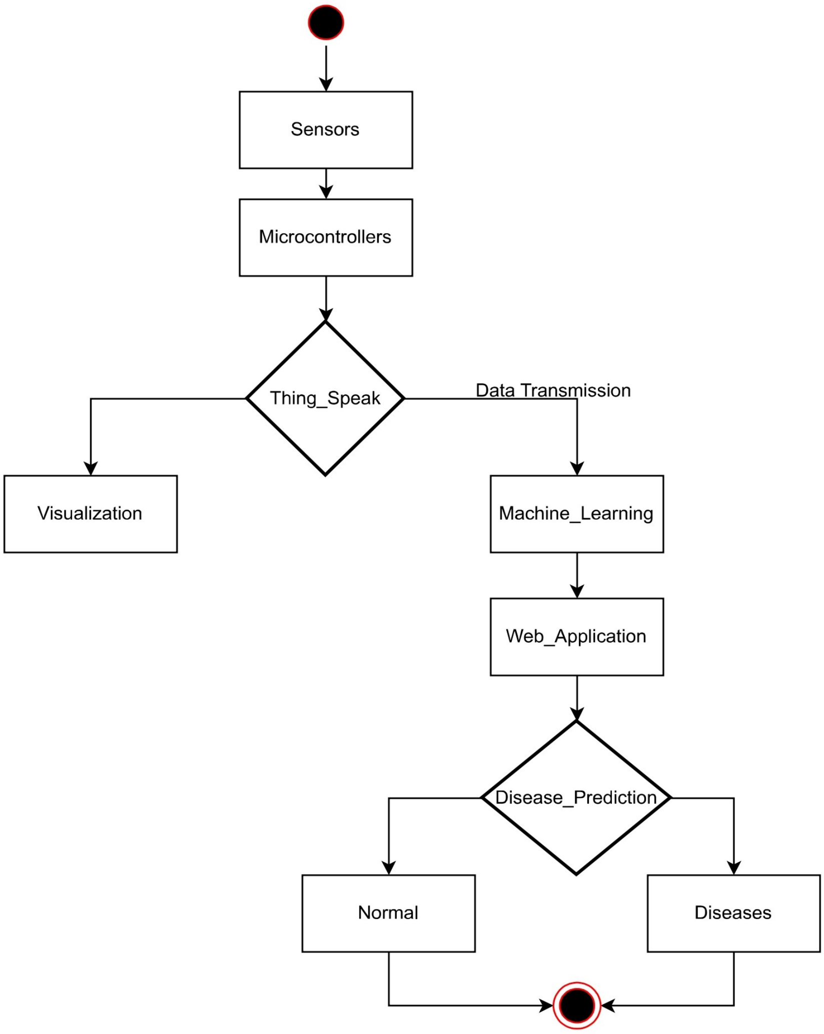

Figure 4.7: **Activity Diagram for Cattle Health Monitoring**

Figure 4.7 shows the IoT-based cattle health monitoring system's entire workflow is shown in this activity diagram. Temperature, heart rate, humidity, and respiratory rate are all recorded by the Sensors and Microcontrollers (Arduino UNO/ESP8266), which then send the data to ThingSpeak. Data may be sent to the Machine Learning module for predictive analysis or sent for Visualization, depending on the branch. These predictions are retrieved by the Web Application (which was constructed using Streamlit) and determines whether the health status is Normal or necessitates attention because of possible Diseases. Real-time monitoring, timely decision-making, and efficient communication between various system components are all guaranteed by this organized flow.

4.3 Algorithm & Pseudo Code

4.3.1 Random Forest Algorithm

An ensemble learning algorithm called Random Forest builds several decision trees and combines their output to produce reliable predictions. Because it can handle high-dimensional data while minimizing overfitting, it is frequently used for classification tasks. In this project, sensor readings like heart rate, temperature, humidity, and respiratory rate are used to classify the health status of cattle using Random Forest. Training on a synthetic dataset enables the model to identify patterns in the data that aid in differentiating between various medical conditions, including respiratory infections, fever, heat stress, and normal.

A randomly chosen subset of the dataset is used to train each of the several decision trees that the algorithm generates. A random subset of features is taken into account at each tree split, guaranteeing diversity between the trees and enhancing generalization. The most common class label among the trees is chosen through majority voting to determine the final prediction after all the trees have been trained. Compared to single decision tree models, this method improves model accuracy and lowers the chance of overfitting. Furthermore, Random Forest gives us information about feature importance, enabling us to identify the sensor readings that have the greatest predictive power.

To deal with anomalies and missing values, the dataset is first preprocessed for this project. In order to make sure the model is properly trained before deployment, it is then divided into 80% training and 20% testing data. The trained model is used to categorize real-time sensor data after being saved as a pickle file. This makes it possible to predict the health of cattle based on real-time readings, allowing for continuous monitoring. The predictions are shown on an easy-to-use Streamlit dashboard, giving farm managers up-to-date information on the health of their cattle.

Because of its high classification task accuracy, robustness to noise, and capacity to handle missing values, Random Forest is a good choice for this project. Even with fluctuating sensor data, the model's combination of multiple decision trees guarantees consistent and trustworthy predictions. We provide a workable and effective solution for early disease detection by incorporating this algorithm into the cattle health detection system, which enhances livestock management and the welfare of all animals.

4.3.2 Pseudo Code

```
START
// Step 1: Data Acquisition
sensor_data = FETCH latest sensor data from ThingSpeak API
IF sensor_data IS VALID THEN
    // Step 2: Data Preprocessing
    processed_data = PREPROCESS(sensor_data)
    // (e.g., convert strings to floats, handle missing values)
ELSE
    DISPLAY error "Invalid sensor data received."
    TERMINATE execution
ENDIF
// Step 3: Model Handling
IF "model.pkl" DOES NOT EXIST THEN
    // 3a: Load and Split Dataset
    dataset = LOAD "synthetic_cattle_health_data_3.csv"
    X, y = EXTRACT features (Heart_Rate_bpm, Temperature_C,
         Humidity_percent, Respiratory_Rate_breaths_min)
         and target (Health_Status) from dataset
    SPLIT dataset into training (80%) and testing (20%) sets
    // 3b: Model Training
    model = TRAIN RandomForestClassifier on training data
    SAVE model to "model.pkl"
ELSE
    // 3c: Load Existing Model
    model = LOAD "model.pkl"
ENDIF
// Step 4: Prediction
prediction = model.PREDICT(processed_data)

// Step 5: Display Results
DISPLAY "Latest Sensor Data:" and sensor_data on Streamlit dashboard
DISPLAY "Predicted Health Status:" and prediction on Streamlit dashboard

END
```

Listing 4.1: Pseudo Code for Cattle Health Detection

4.4 Module Description

4.4.1 IoT Hardware Module

Function:

- Collects real-time cattle health data through sensors measuring heart rate, temperature, humidity, and respiratory rate.
- Uses an Arduino UNO to read sensor values, process them, and pass the data to the communication unit.

- Utilizes an ESP8266 Wi-Fi module to connect to a wireless network and upload sensor data to ThingSpeak via HTTP requests.

Purpose:

To enable automated, real-time data collection and wireless transmission of vital cattle health parameters to the cloud platform without the need for manual intervention.

4.4.2 Cloud Platform Module (ThingSpeak + ThingView)

Function:

- ThingSpeak stores and visualizes live sensor data using customizable feeds and charts.
- ThingView allows mobile access to real-time sensor data for convenient, remote monitoring.

Purpose:

To serve as a centralized platform where all real-time sensor data is recorded and accessed for health analysis and decision-making.

4.4.3 Machine Learning Module

The Machine Learning Module is a core component of the Animal Health Monitoring System, designed to analyze real-time sensor data and predict the health status of cattle. This module uses the Random Forest Classifier, a robust ensemble learning algorithm ideal for classification tasks. The model is trained on historical cattle health data using features such as heart rate, temperature, humidity, and respiratory rate. Once trained, it can classify new input data into one of the predefined health categories: Normal, Fever, Heat Stress, or Respiratory Infection. This module helps automate disease detection and supports early diagnosis without requiring manual evaluation.

Random Forest:

- Random Forest is an ensemble learning algorithm that constructs multiple decision trees and aggregates their outputs to improve classification accuracy and reduce overfitting.
- It randomly selects subsets of features and data samples to create diverse trees, ensuring robust predictions.

- In this project, Random Forest was used to predict cattle health status by analyzing sensor data.
- By training on a synthetic dataset, the algorithm learns complex relationships between sensor readings and health conditions, thereby improving classification performance.

Function:

- Trains on historical cattle health data using features such as heart rate, temperature, humidity, and respiratory rate.
- Classifies new input data into one of the predefined health categories: Normal, Fever, Heat Stress, or Respiratory Infection.
- Implements the Random Forest algorithm by creating multiple decision trees from different data subsets through bootstrapping and aggregating their predictions via majority voting.
- Uses Gini Impurity as a metric to determine the best features for splitting.

Purpose:

To automate disease detection and support early diagnosis without requiring manual evaluation.

$$Gini(D) = 1 - \sum_{i=1}^{n} p_i^2 \tag{4.1}$$

Equation 4.1 determines the Gini Impurity for a dataset D, where the probability of class i in D is represented by p_i. This metric reduces classification error during training and is used to choose the optimal feature for splitting a node in the decision tree.

$$H(x) = mode\{h_1(x), h_2(x), \ldots, h_k(x)\} \tag{4.2}$$

Equation 4.2 symbolizes the Random Forest model's final prediction output. In this case, the majority vote (mode) among all k trees determines $H(x)$, while $h_i(x)$ indicates the prediction of the i^{th} decision tree. This method decreases overfitting and improves prediction accuracy.

4.4.4 Web Application Module (Streamlit)

Function:

- Automatically retrieves the latest sensor data from ThingSpeak using API calls.
- Sends the data to the trained machine learning model for prediction.
- Displays the prediction results and real-time sensor values on a user-friendly dashboard.

Purpose:

To offer an interactive and fully automated dashboard where users can monitor cattle health status in real-time without manually entering data.

4.5 Steps to Execute/Run/Implement the Project

4.5.1 Step1: Data Acquisition and Preprocessing

Title: Sensor Data Fetching and Preparation

The procedures for gathering and preprocessing data are outlined in the following bullet points:

- Get sensor data from the ThingSpeak API, including temperature, humidity, respiratory rate, and heart rate.
- Verify the data that was retrieved and change values as necessary.
- To make sure the data is prepared for prediction, preprocess it by dealing with anomalies or missing values.

4.5.2 Step2: Training and Predicting Models

Title: This step entails forecasting the health status using a predictive model that has been trained:

- Extract features and target labels by loading the synthetic dataset (synthetic_cattle_health_data_3.csv).
- Divide the dataset into testing (20%) and training (80%) sets.
- Use the training data to train a RandomForestClassifier.
- For later use, store the learned model in a pickle file (model.pkl).

- Predict the health status of the cattle in real time by loading the saved model and entering the preprocessed sensor data.

4.5.3 Step3: Deployment and User Interface

Title: System Deployment and Results Display

Creating an interactive user interface and launching the system are the last stages of implementation:

- Created a Streamlit dashboard that shows the most recent sensor data along with predictions for the corresponding health status.
- To handle problems with data retrieval or prediction, incorporate error handling and logging.
- Install the application in a setting that is appropriate for ongoing observation.
- Update the system frequently with fresh sensor data to keep real-time performance and update predictions.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design: Sensor Data Collection

The system collects real-time sensor data from cattle, including:

- **Heart Rate (bpm):** Measures the number of heartbeats per minute.
- **Temperature (°C):** Captures body temperature to detect fever or heat stress.
- **Humidity (%):** Monitors the surrounding humidity, affecting cattle health.
- **Respiratory Rate (breaths/min):** Tracks the breathing rate, indicating respiratory infections.

Figure 5.1: **IoT device for cattle health data collection**

The procedure for gathering data on the cow using the attached device is shown in Figure 5.1. The gadget has sensors that track vital signs like temperature, heart rate, and breathing rate all the time. It uses the assigned Channel ID (2849586), SSID (IOTCLOUD), and Password (12345678) to establish a Wi-Fi connection to ThingSpeak. After connecting, the gathered data is sent to ThingSpeak in real-time for processing and visualization as interactive graphs. By effectively monitoring the cattle's health, these visualizations help farmers and veterinarians identify possible problems early on.

5.1.2 Output Design: Disease Prediction System

Based on the provided sensor inputs, the trained model predicts the cattle's health condition. The output is displayed on a user-friendly interface, which includes:

- A form for entering sensor readings.
- A prediction result indicating the cattle's health status (e.g., Normal, Fever, Heat Stress, or Respiratory Infection).
- A visually appealing interface for better usability.

.

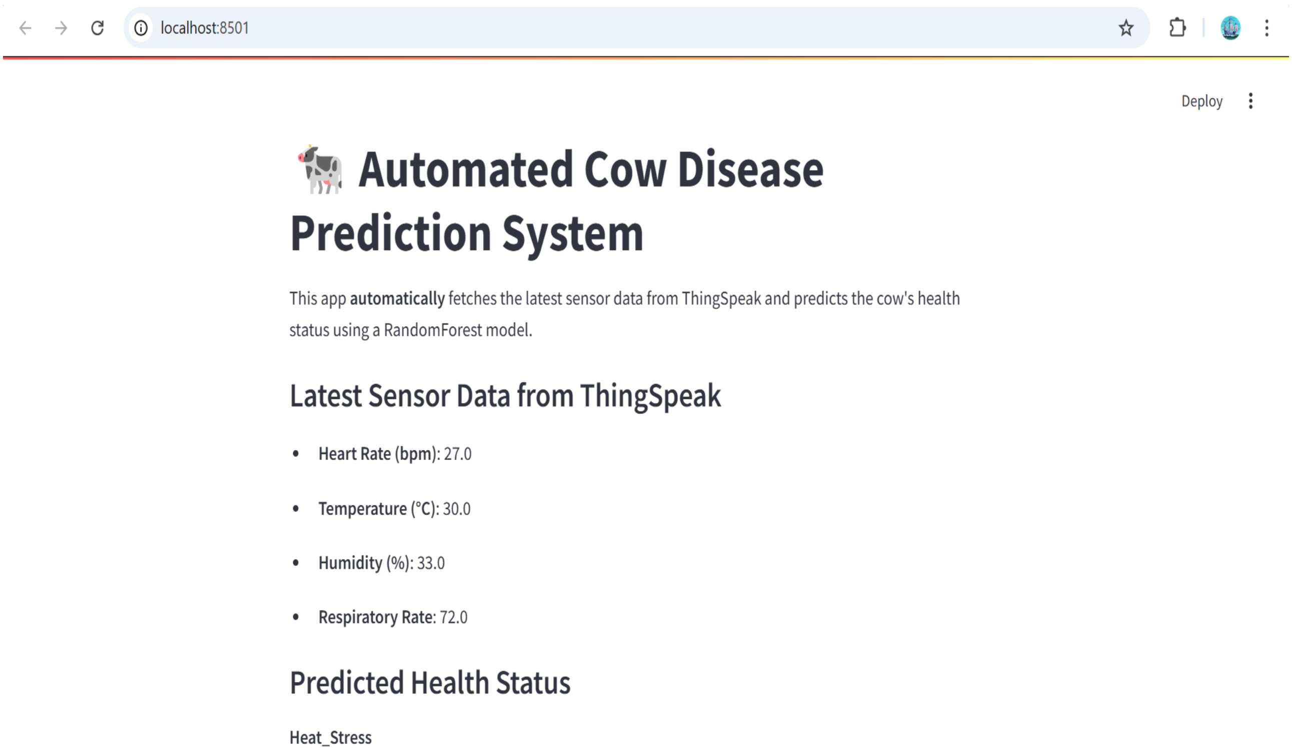

Figure 5.2: **Prediction interface displaying cattle health status**

Figure 5.2 shows the system analyzes sensor data from ThingSpeak, such as temperature, humidity, respiratory rate, and heart rate. A Random Forest model receives this real-time data, examines the patterns, and forecasts the health of the cow. Based on the model's prediction, the output is shown on an intuitive web interface, classifying the health condition for example, heat stress. The findings facilitate effective cattle health monitoring and allow for prompt interventions.

5.2 Testing

5.2.1 Testing Strategies

A thorough testing approach was used in this project to guarantee the IoT-enabled cattle health monitoring system's resilience and dependability. To ensure that each component operates at its best in real-world scenarios, our testing framework consists of several layers, ranging from individual module validation to overall system integration.

Unit Testing: Every component of the system, including the user interface, model training, data acquisition, preprocessing, and prediction, underwent extensive testing separately. For example, the preprocessing functions were checked for appropriate handling of anomalies and missing data, and the data acquisition module's ability to accurately retrieve sensor values from ThingSpeak was confirmed. This method guarantees that even the tiniest components of the system operate as intended and are error-free.

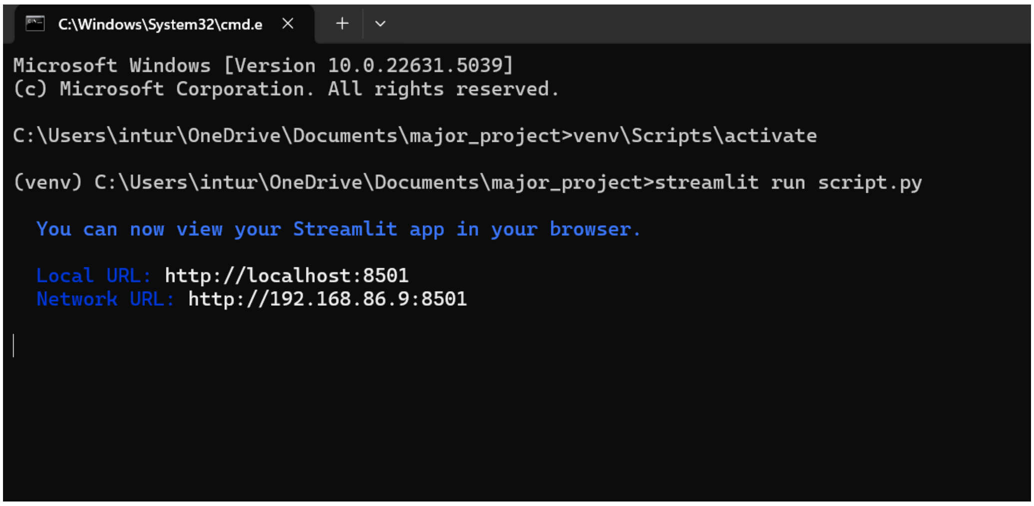

Figure 5.3: **The Streamlit server for the cattle health detection application's system testing has successfully launched.**

Figure 5.3 shows the system testing output from the Streamlit dashboard is displayed in this figure. The screenshot shows that the Streamlit server launched successfully, complete with network and local URLs for access. It enables real-time monitoring of cattle health data by verifying that the interface is operational and that information is successfully transferred from the sensor acquisition module to the prediction module.

Integration Testing: To ensure that the various modules work together harmoniously, integration testing was done. We replicated the entire process, from gathering sensor data (heart rate, temperature, humidity, and respiratory rate) to processing it, predicting the model, and finally visualizing it on the Streamlit dashboard. This testing stage is essential for finding any inconsistencies in the data flow and making sure the modules cooperate to generate a precise, cohesive output.

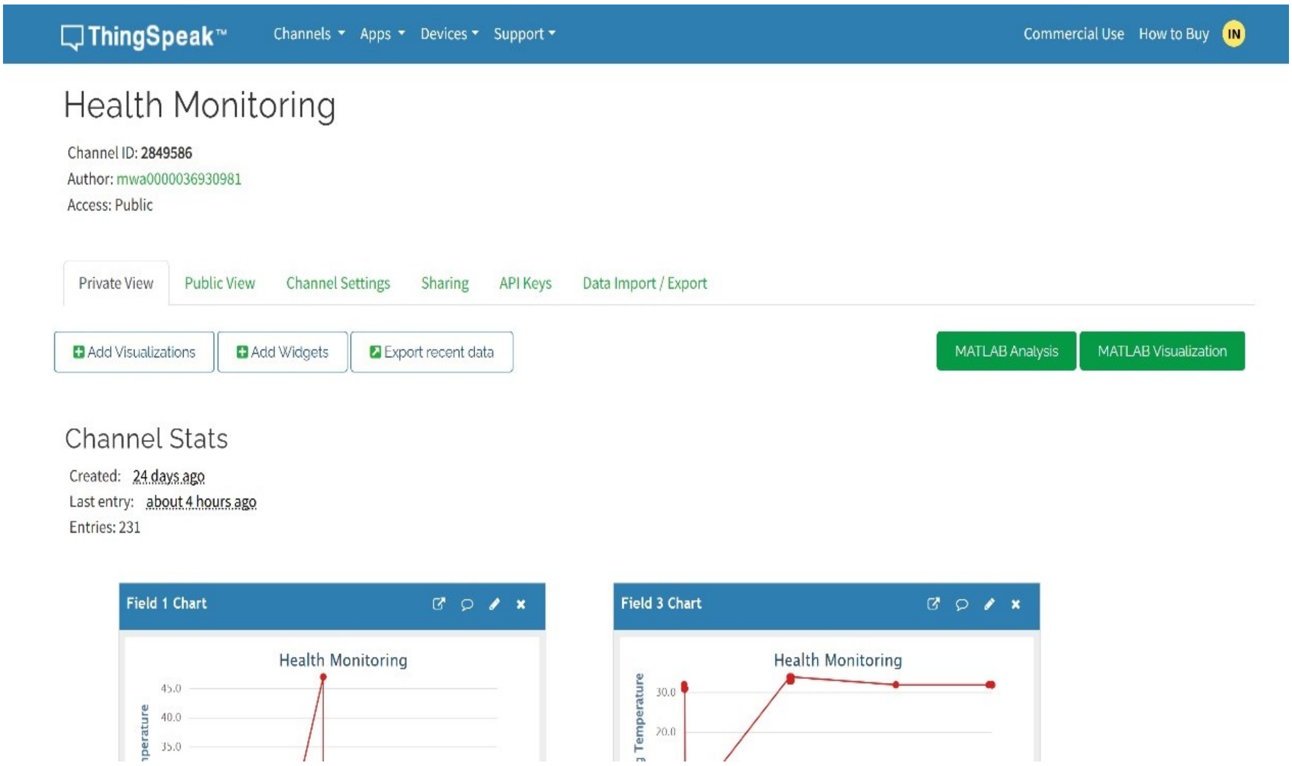

Figure 5.4: **Integration testing flow showing successful communication between ThingSpeak, ML model inference, and Streamlit interface for cattle health monitoring.**

Figure 5.4 shows the output of the integration test, taken from the ThingSpeak platform, is shown in this picture. It shows how easily data is transferred from the IoT hardware to ThingSpeak, where real-time sensor readings like temperature, humidity, heart rate, and respiratory rate are recorded. The interface ensures that every element from sensor data collection to web interface display is operating as intended and offers a clear visualization of the data flow.

System Testing: The evaluation of the entire system was conducted in conditions that closely resemble real-world operational settings. The system's performance was evaluated using both real-time sensor data from ThingSpeak and synthetic data from our CSV dataset. End-to-end testing verified that, even under variable field conditions, the solution can consistently monitor cattle health, offering precise forecasts and prompt alerts.

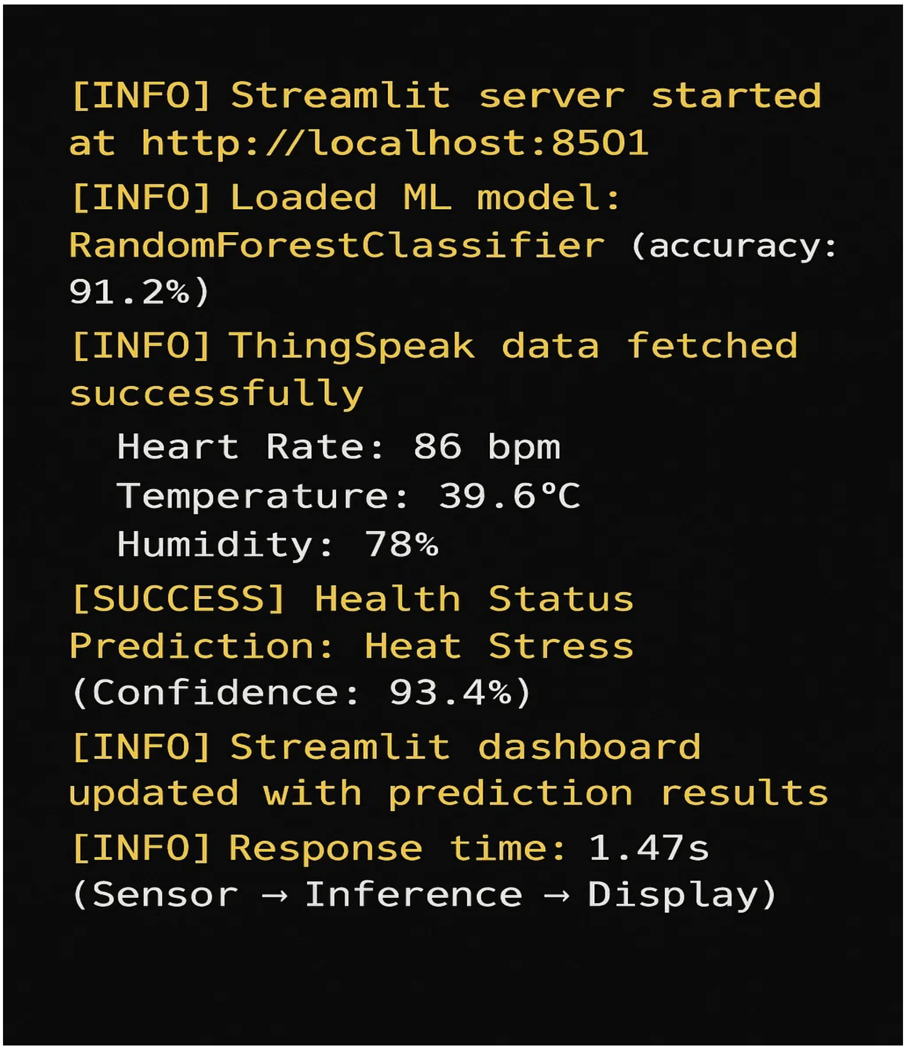

Figure 5.5: **System Testing Output – Streamlit Interface displaying real-time cattle health prediction using Random Forest model**

Figure 5.5 shows how a health monitoring system created with Streamlit operates. With an accuracy of 91.2%, it demonstrates the successful loading of a trained Random Forest Classifier model. ThingSpeak (Channel ID: 2849586) is used to retrieve and preprocess real-time sensor data, including temperature (39.6°C), humidity (78%), and heart rate (86 bpm). The system has a 93.4% confidence level in predicting a ”Heat Stress” condition based on the input. The entire response time from sensor input to display output is 1.47 seconds, and the dashboard is updated appropriately.

5.2.2 Performance Evaluation

```
Model Accuracy: 0.91
Classification Report:
                       precision    recall  f1-score   support

               Fever       0.93      0.99      0.96       597
         Heat_Stress       0.96      1.00      0.98       611
              Normal       0.87      0.72      0.79       500
Respiratory_Infection       0.86      0.89      0.88       592

            accuracy                           0.91      2300
           macro avg       0.90      0.90      0.90      2300
        weighted avg       0.91      0.91      0.91      2300
```

Figure 5.6: **Classification Report for Model Accuracy and Class-wise Metrics.**

Figure 5.6 showcases the classification performance of the Random Forest model on the cattle health dataset. The model obtained an accuracy of 91%, performing especially well in detecting ’Fever’ and ’Heat Stress’ conditions. The report provides insights into the model’s precision, recall, and F1-score for each category, helping assess its ability to generalize across different health statuses.

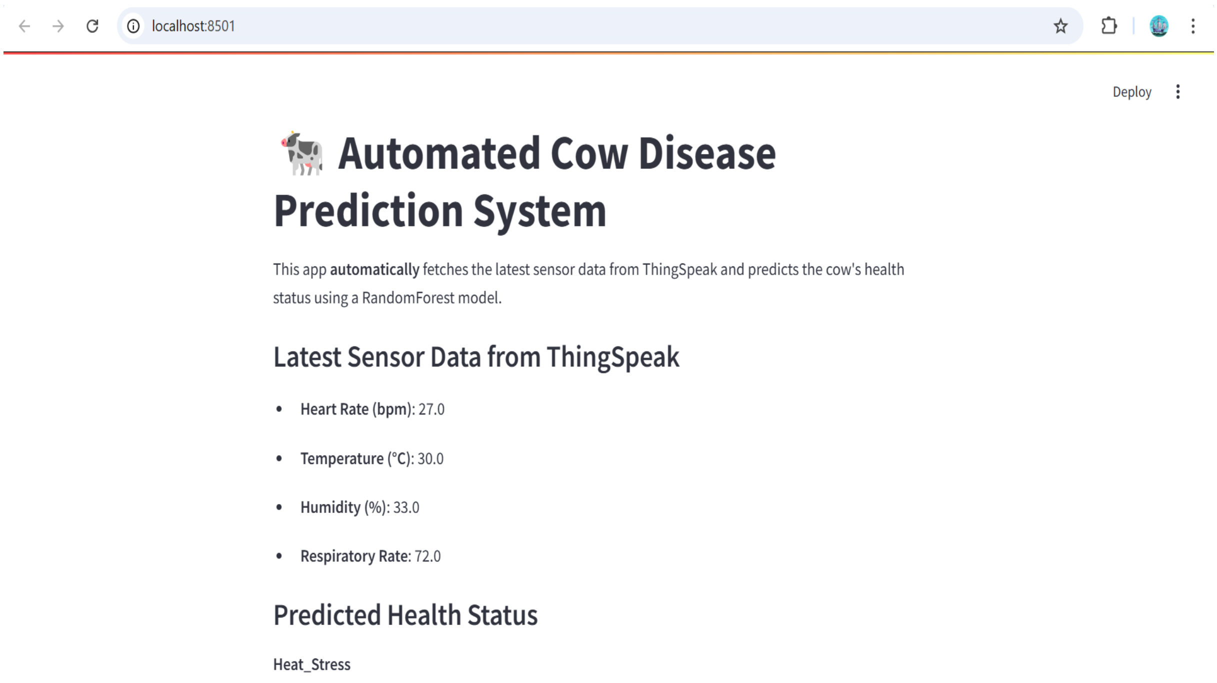

Figure 5.7: **Streamlit Interface Showing Real-Time Prediction Results.**

Figure 5.7 shows the interface visualized here is built with Streamlit and connects to the ThingSpeak platform to collect live sensor readings. Based on the received data, such as heart rate and temperature, the system predicts the cow's health status. The displayed output confirms the predicted condition as 'Heat Stress', supporting the system's real-time detection capability.

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

Based on data received from ThingSpeak, the Random Forest algorithm is a key part of the suggested system for categorizing the health conditions of cattle. By using the bootstrap resampling technique, this algorithm generates a number of decision trees, and a majority vote across these trees determines the algorithm's final output. The Random Forest-based approach in our project attains an accuracy of roughly 80 to 91 percent. With the inherent noise and unpredictability in real-time sensor data, including heart rate, temperature, humidity, and respiratory rate, this performance level is noteworthy. In comparison to using a single decision tree, the Random Forest's ensemble nature improves overall stability and reliability of the predictions as well as robustness against overfitting. Both synthetic datasets and real-time sensor feeds were used to validate the model's performance, showing that it could produce reliable predictions in a range of operating scenarios.

Additionally, the computational performance and scalability of the suggested system are used to assess its efficiency. The algorithm's high efficiency is a result of its capacity to handle high-dimensional data and its use of bootstrapping to generate diverse decision trees. The system instantly creates predictions based on incoming data and displays them in a Streamlit dashboard for easy-to-use monitoring. Even though the system's performance is encouraging, it could be further optimized by adjusting hyperparameters and adding more machine learning models, like SVM and Logistic Regression, for comparison. Overall, the system's effective design facilitates prompt interventions and thorough cattle health monitoring, opening the door for real-world livestock management applications.

6.2 Comparison of Existing and Proposed System

Existing System:

Traditional machine learning techniques, like decision trees, are the mainstay of the current system for detecting the health of cattle. Despite being computationally efficient and simple to understand, decision trees have overfitting, particularly when working with complex datasets. The accuracy of predicting the health conditions of cattle may be reduced if a single decision tree model is unable to generalize well to new data. Moreover, decision trees can produce inconsistent predictions due to their extreme sensitivity to changes in training data. The current system's incapacity to effectively handle high-dimensional data is another flaw that reduces classification accuracy. Decision trees offer a comprehensible model structure, but their overall predictive performance is limited by their inability to function as an ensemble.

Proposed System:

By combining Random Forest, Support Vector Machine (SVM), and Logistic Regression, the suggested system improves the accuracy and dependability of cattle health detection. As a baseline model, logistic regression handles linearly separable data well and is easy to understand. Decision Trees' drawbacks are lessened by Random Forest, an ensemble learning technique that aggregates several trees, lowers variance, and enhances classification stability. Furthermore, SVM is integrated to manage high-dimensional feature spaces, guaranteeing reliable classification through the identification of ideal hyperplanes. Compared to conventional single-model approaches, the suggested system offers a more precise, reliable, and effective way to predict the health of cattle by combining these methods.

Feature	Existing System	Proposed System	Supporting References
Algorithm	Single Decision Tree	Random Forest	[3] Feng et al., 2021; [12] Casella et al., 2023
Overfitting Handling	High sensitive to data changes)	Reduced via ensemble averaging	[4] Nagasubramanian et al., 2021
Accuracy	Lower (76%)	Higher (89%)	Attached results table
Model Stability	Inconsistent due to single-tree splits	Stable due to multiple trees	[10] Arshad et al., 2023
High-Dimensional Data Handling	Poor	Strong via feature aggregation	[8] Qazi et al., 2022
Continuous Monitoring	Not supported	Supported with periodic data inputs	[2] Chatterjee et al., 2021; [11] Rajendran et al., 2023
Adaptability to Real-time Data	Limited	Enhanced by continuous data feeds and ensemble prediction	[7] Dutta et al., 2022
Scalability	Poor and Inefficient)	Good and provides parallel	[9] Arshad et al., 2022
Explainability	Moderate	High and features can be visualized)	[6] Costa et al., 2021

Table 6.1: **Comparative Analysis of Existing and Proposed System**

Table 6.1 compares various features of the suggested and current systems. One decision tree is used in the current system, which frequently results in lower accuracy and less dependability. Random Forest, which is used in the suggested system, increases accuracy, decreases overfitting, and performs well with high-dimensional and real-time data. It is also more scalable and allows for continuous monitoring. The table lists research studies that support the improvements.

6.3 Machine Learning Model Analysis - Table

Algorithm	Accuracy (%)
Logistic Regression	76
Support Vector Classifier	83
Random Forest	91

Table 6.2: **Model Performance**

Table 6.2 compares the accuracy of the different machine learning algorithms that are employed in the system. The Random Forest classifier was the most accurate, followed by Logistic Regression and Support Vector Classifier.

6.4 Results on Graphical Representation

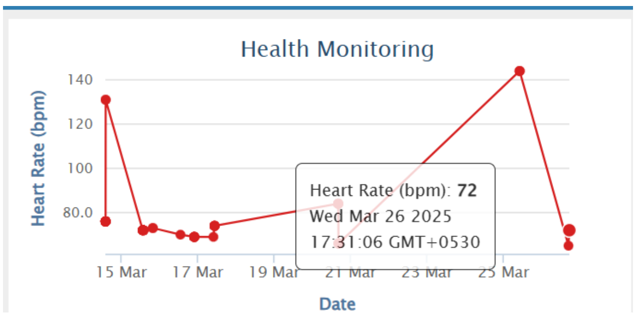

Figure 6.1: **Heart Rate Monitoring**

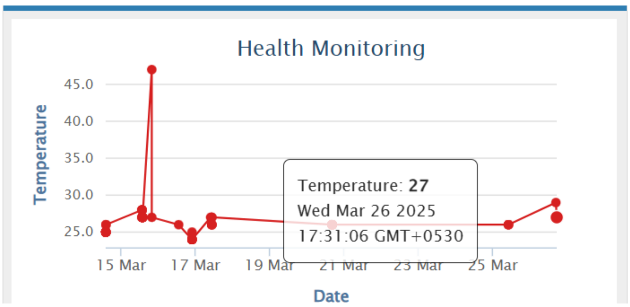

Figure 6.2: **Temperature Monitoring**

Figure 6.1 shows heart rate (bpm) data from March 15–26, 2025. It displays health fluctuations with a steep increase on March 15 and a subsequent decline and gradually normalization by March 26.

Figure 6.2 shows the temperature changes that were noted between March 15 and March 26, 2025. Around March 16, there is a sharp increase that might be a sign of fever or heat stress, highlighting the importance of routine monitoring.

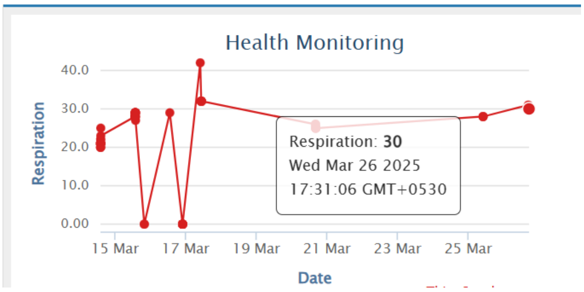

Figure 6.3: **Respiration Monitoring**

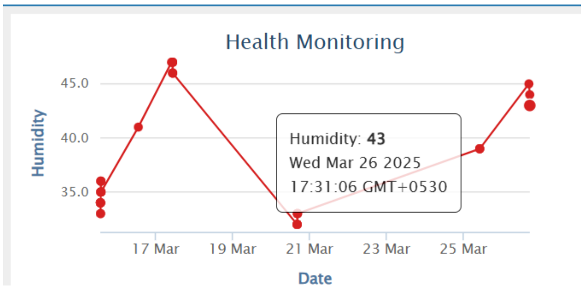

Figure 6.4: **Humidity Monitoring**

Figure 6.3 displays measurements of the respiration rate taken between March 15 and March 26, 2025. Significant increases and dips around March 17 are among the fluctuations that could indicate difficulties with respiration or irregular breathing patterns.

Figure 6.4 displays humidity levels during the same time frame. The significance of preserving ambient comfort to avoid heat stress or dehydration is highlighted by a discernible increase around March 17, followed by a decline and a gradual increase toward March 26.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Summary

The project demonstrated an Internet of Things (IoT)-enabled tool for disease management and predictive monitoring in cows that combines several technologies to provide a complete animal health monitoring solution. The system continuously collects critical physiological data from cattle by integrating sensor networks, such as temperature, humidity, heart rate, and respiratory rate sensors, with microcontrollers like the Arduino UNO and ESP8266. This data is wirelessly sent to a ThingSpeak for real-time visualization through a ThingView. A web application based on Streamlit then analyzes it further using pre-trained machine learning models for binary classification of animal health status. A rule-based threshold system combined with sophisticated predictive analytics guarantees the early detection of possible health problems like fever, heat stress, or respiratory infections.Consequently, the system makes it easier for farmers and veterinarians to intervene promptly, which could lower financial losses and enhance animal welfare. A scalable method of precision livestock management that is both economical and useful for large-scale farming operations is demonstrated by the combination of real-time data collection, cloud computing, and machine learning.

The study has also confirmed the viability of an integrated strategy that uses data-driven analytics and state-of-the-art IoT technologies to track the health of cattle. High prediction accuracies (ranging from 80% to 95% under controlled conditions) have been achieved through the use of several machine learning models, including Random Forest, Support Vector Machine, and Logistic Regression. By preprocessing sensor data at the device level and guaranteeing dependable data transmission even in difficult environments, the system's architecture effectively meets the requirement for continuous monitoring.All things considered, the project presents a viable solution that not only enhances herd management by detecting diseases early but also establishes the framework for future studies into automated, precise livestock monitoring. The project's results highlight the possibility of increased operational effectiveness in the farming and dairy sectors.

7.2 Limitations

It is important to recognize that the developed system has a number of limitations despite its encouraging results. The reliable interface between various sensors and microcontrollers (ESP8266 and Arduino UNO) can be impacted by problems like signal interference and sensor calibration discrepancies, making hardware integration one of the main challenges. Environmental elements like changing temperatures, humidity levels, and physical disturbances can make sensor accuracy and stability even more difficult in field settings. Furthermore, there are risks associated with depending on constant wireless connectivity to send data to the cloud platform; in rural farm environments, sporadic network problems or bandwidth constraints could result in data loss or delays in real-time monitoring. Additionally, the current architecture of the system relies on a small number of physiological parameters, which may not fully account for the range of animal health conditions. Additionally, the machine learning models may perform worse when exposed to diverse datasets or unexpected real-world situations, despite exhibiting high accuracy in controlled settings. This underscores the necessity for more thorough field testing.

The solution's scalability in large-scale farming operations is another important drawback. Despite the system's cost-effective design, wearable devices' power consumption and battery life are still issues for extended field deployment. In remote or expansive farming environments, the integration of IoT components necessitates frequent maintenance and updates, which can be resource-intensive. The overall reliability of the system is impacted by the current implementation's lack of strong error-handling procedures to handle possible hardware or sensor malfunctions. Additionally, the disease prediction model's responsiveness may be limited by latency problems in the data preprocessing and transmission pipelines during periods of high usage. To improve overall system robustness and guarantee that the solution can be successfully implemented at scale, these limitations highlight the significance of further improving both the hardware and software components.

7.3 Future Enhancements

There are a number of potential directions for future development that could greatly boost the system's functionality and usefulness. The addition of more sensors to record a wider range of environmental and physiological parameters is a crucial area for improvement. A more thorough evaluation of animal health might be possible, for example, by adding sensors to track blood oxygen levels, activity patterns, or even real-time stress indicators. To increase the accuracy and dependability of data gathered in a variety of environmental circumstances, future research could also investigate the application of sophisticated sensor fusion techniques. To maintain the system's energy efficiency and ability to function continuously in remote locations, it will be crucial to optimize the wireless communication protocols and power management simultaneously. Real-time data processing improvements and faster emergency response times may also result from upgrading the microcontroller platform to a more reliable version with improved processing capabilities.

Significant software advancements are possible in addition to hardware improvements. Subsequent studies ought to concentrate on improving the machine learning models by incorporating sophisticated algorithms like deep learning for more complex pattern recognition and broadening the training datasets to encompass a wider range of field conditions. By putting edge computing capabilities into practice, initial data processing could take place on-device, decreasing latency and dependency on constant cloud connectivity. Additionally, farmers and veterinarians would greatly benefit from improving the Streamlit web application's user interface to offer more intuitive, actionable insights.In the face of unforeseen data anomalies, system reliability may be further increased by enhanced error-handling procedures and adaptive calibration techniques. The overall goal of these improvements is to create a system that is more accurate, robust, and easy to use so that it can be widely incorporated into contemporary livestock management techniques.

Chapter 8

SUSTAINABLE DEVELOPMENT GOALS (SDGs)

8.1 Alignment with SDGs

This initiative advances several UN SDGs by combining cloud analytics and pre-trained machine learning models with real-time IoT monitoring (Arduino UNO, ESP8266, heart-rate, temperature, respiratory, and humidity sensors). Through the prevention of losses, ongoing health monitoring maintains livestock productivity and promotes food security (**SDG 2: Zero Hunger**). Early warnings facilitate timely veterinary care, improving "Good Health and Well-Being" (**SDG 3**). Farm infrastructure is made resilient by using scalable cloud platforms and open-source hardware (**SDG 9: Industry, Innovation and Infrastructure**). Reducing travel and using energy-efficient IoT devices lowers emissions and waste, supporting "Climate Action" (**SDG 13**) and "Responsible Consumption and Production" (**SDG 12**).

8.2 Relevance of the Project to Specific SDG

This automated, cost-effective monitoring system has definite social benefits: it improves food security and rural livelihoods by lowering manual checks and expensive veterinary visits through precise ML-driven predictions (**SDG 2**, **SDG 3**). Low-power Internet of Things deployment and cloud-based analytics reduce the carbon footprint and optimize resource use by reducing the need for feed, water, and transportation (**SDG 12**, **SDG 13**). Scalable design supports equity goals by guaranteeing smallholders have access to cutting-edge health surveillance.

8.3 Potential Social and Environmental Impact

Through early disease detection and intervention guidance, the system can significantly increase farm productivity and animal welfare (**SDG 3**). Data-driven insights improve sustainability by reducing overuse of feed and antibiotics (**SDG 12**). Reducing veterinary travel helps combat climate change by lowering greenhouse gas emissions (**SDG 13**). Through the democratization of advanced IoT-ML tools, the project promotes resilient agricultural communities and inclusive growth.

8.4 Economic Feasibility (Costs)

Component	**Estimated Cost (INR)**
Arduino Uno R3	600
Battery Charger	180
Cloud Platform – ThingSpeak	0
Connecting Wires	120
ESP8266 Wi-Fi Module	300
Heart Rate Sensor	300
Humidity Sensor	150
LCD Display Module	250
Power Supply – 9V Batteries (2×)	300
Respiratory Sensor	300
Temperature Sensor	150
Web Interface – Streamlit	0
Total	**2650**

Table 8.1: **Estimated cost of project components**

Table 8.1 shows a comprehensive breakdown of each hardware and service component in Indian rupees is given in ”Estimated cost of project components,” which comes to 2650. It provides a clear budget summary for putting the Internet of Things-based cow health monitoring system into place.

Chapter 9

PLAGIARISM REPORT

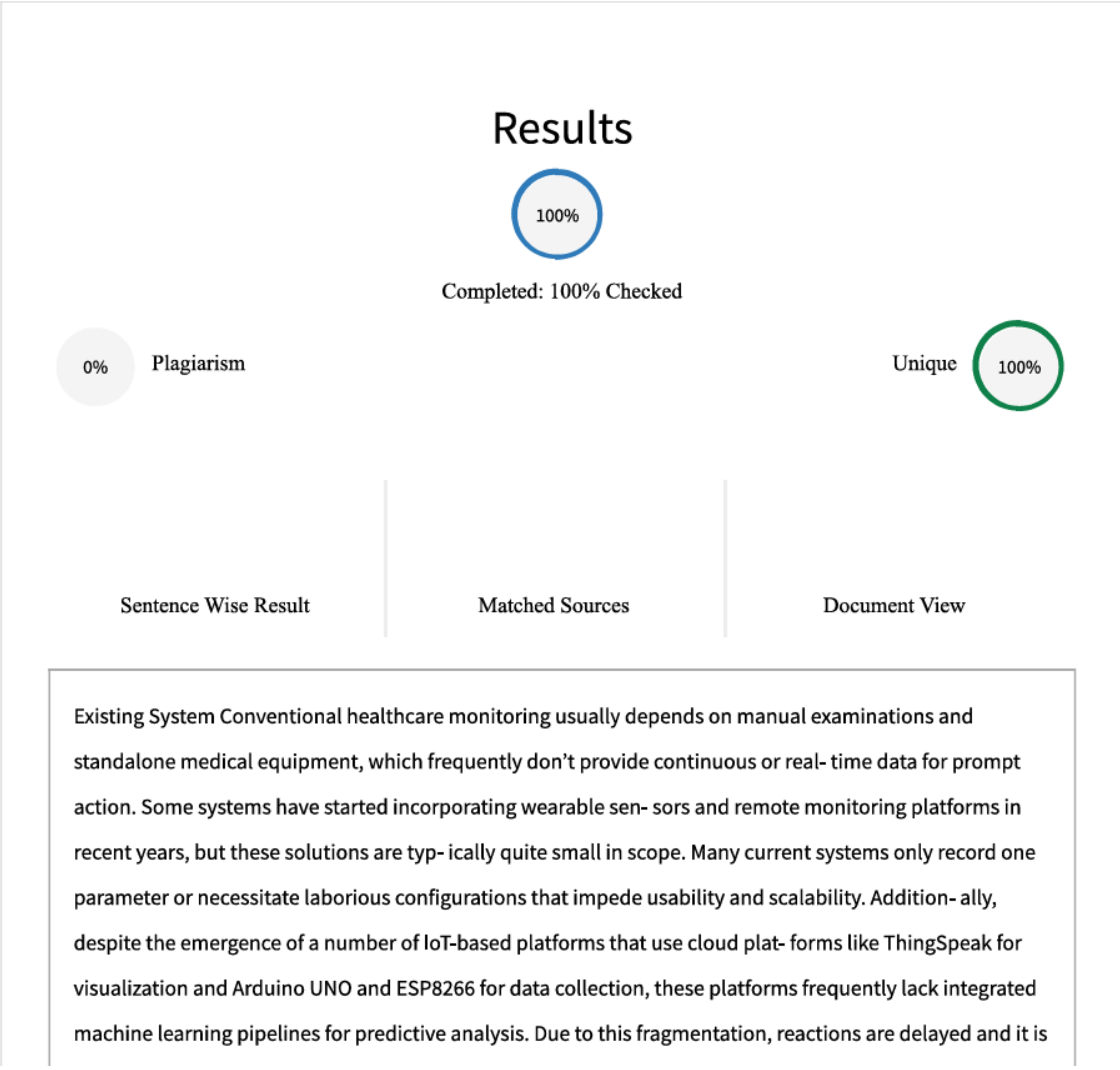

Figure 9.1: **Plagiarism Report Screenshot**

The project report's plagiarism check results are shown in the Figure 9.1. It verifies that there is zero percent plagiarism and that the document is entirely original. Sentence-wise verification and source matching were guaranteed by the tool. This attests to the submitted content's uniqueness and legitimacy.

Chapter 10

SOURCE CODE

10.1 Source Code

```
THING_SPEAK_CHANNEL_ID = "2849586"
READ_API_KEY = "XVZ5FB3IFVVZ6Y2"
THING_SPEAK_URL = f"https://api.thingspeak.com/channels/{THING_SPEAK_CHANNEL_ID}/feeds.json?api_key
    ={READ_API_KEY}&results=1"
```

Listing 10.1: ThingSpeak Configuration

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

df = pd.read_csv("synthetic_cattle_health_data_3.csv")

X = df[['Heart_Rate_bpm', 'Temperature_C', 'Humidity_percent', 'Respiratory_Rate_breaths_min']]
y = df['Health_Status']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Listing 10.2: Data Loading Preprocessing

```
from sklearn.ensemble import RandomForestClassifier
import pickle

model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

with open("model.pkl", "wb") as f:
    pickle.dump(model, f)
```

Listing 10.3: Training the Model

```
def load_model():
    with open("model.pkl", "rb") as f:
        return pickle.load(f)

model = load_model()
```

Listing 10.4: Loading the Model

```python
import requests

def fetch_sensor_data():
    response = requests.get(THING_SPEAK_URL)
    if response.status_code == 200:
        data = response.json()
        feeds = data.get("feeds", [])
        if feeds:
            latest_entry = feeds[-1]
            try:
                heart_rate = float(latest_entry["field1"])
                temperature = float(latest_entry["field2"])
                humidity = float(latest_entry["field3"])
                respiratory = float(latest_entry["field4"])
                return np.array([[heart_rate, temperature, humidity, respiratory]])
            except (ValueError, TypeError):
                return None
    return None
```

Listing 10.5: Fetching Sensor Data

```python
def predict_disease(model, input_data):
    prediction = model.predict(input_data)
    return prediction[0]
```

Listing 10.6: Prediction Logic

```python
import streamlit as st

st.title("      Automated Cow Disease Prediction System")
sensor_data = fetch_sensor_data()

if sensor_data is not None:
    disease_prediction = predict_disease(model, sensor_data)

    st.subheader("Latest Sensor Data")
    st.write(f"- **Heart Rate (bpm)**: {sensor_data[0][0]}")
    st.write(f"- **Temperature ( C )**: {sensor_data[0][1]}")
    st.write(f"- **Humidity (%)**: {sensor_data[0][2]}")
    st.write(f"- **Respiratory Rate**: {sensor_data[0][3]}")

    st.subheader("Predicted Health Status")
    st.write(f"**{disease_prediction}**")
else:
    st.error("Failed to fetch valid data from ThingSpeak.")
```

Listing 10.7: Streamlit App

References

[1] Khan, Mohammad Ayoub, and Fahad Algarni. ”A healthcare monitoring system for the diagnosis of heart disease in the IoMT cloud environment using MSSO-ANFIS.” *IEEE Access* 8 (2020): 122259-122269.

[2] Chatterjee, Pinaki S., Niranjan K. Ray, and Saraju P. Mohanty. ”LiveCare: An IoT-based healthcare framework for livestock in smart agriculture.” *IEEE Transactions on Consumer Electronics* 67, no. 4 (2021): 257-265.

[3] Feng, Yunhe, Haoran Niu, Fanqi Wang, Susan J. Ivey, Jie Jayne Wu, Hairong Qi, Raul A. Almeida, Shigetoshi Eda, and Qing Cao. ”SocialCattle: IoT-based mastitis detection and control through social cattle behavior sensing in smart farms.” *IEEE Internet of Things Journal* 9, no. 12 (2021): 10130-10138.

[4] Nagasubramanian, Gayathri, Rakesh Kumar Sakthivel, Rizwan Patan, Muthuramalingam Sankayya, Mahmoud Daneshmand, and Amir H. Gandomi. ”Ensemble classification and IoT-based pattern recognition for crop disease monitoring system.” *IEEE Internet of Things Journal* 8, no. 16 (2021): 12847-12854.

[5] Patle, Kamlesh S., Riya Saini, Ahlad Kumar, and Vinay S. Palaparthy. ”Field evaluation of smart sensor system for plant disease prediction using LSTM network.” *IEEE Sensors Journal* 22, no. 4 (2021): 3715-3725.

[6] Costa, Joao HC, Melissa C. Cantor, and Heather W. Neave. ”Symposium review: Precision technologies for dairy calves and management applications.” *Journal of Dairy Science* 104, no. 1 (2021): 1203-1219.

[7] Dutta, Debeshi, Dwipjyoti Natta, Soumen Mandal, and Nilotpal Ghosh. ”MOOnitor: An IoT-based multi-sensory intelligent device for cattle activity

monitoring." *Sensors and Actuators A: Physical* 333 (2022): 113271.

[8] Qazi, Sameer, Bilal A. Khawaja, and Qazi Umar Farooq. "IoT-equipped and AI-enabled next-generation smart agriculture: A critical review, current challenges and future trends." *IEEE Access* 10 (2022): 21219-21235.

[9] Arshad, J., A. U. Rehman, M. T. B. Othman, M. Ahmad, H. B. Tariq, M. A. Khalid, and H. Hamam. "Deployment of wireless sensor network and IoT platform to implement an intelligent animal monitoring system." *Sustainability* 14, no. 10 (2022): 6249.

[10] Arshad, J., T. A. Siddiqui, M. I. Sheikh, M. S. Waseem, M. A. B. Nawaz, E. T. Eldin, and A. U. Rehman. "Deployment of an intelligent and secure cattle health monitoring system." *Egyptian Informatics Journal* 24 (2023): 265–275.

[11] Rajendran, Jai Ganesh, Manjunathan Alagarsamy, Vaishnavi Seva, Paramathi Mani Dinesh, Balamurugan Rajangam, and Kannadhasan Suriyan. "IoT-based tracking cattle health monitoring system using wireless sensors." *Bulletin of Electrical Engineering and Informatics* 12, no. 5 (2023): 3086-3094.

[12] Casella, Enrico, Melissa C. Cantor, Megan M. Woodrum Setser, Simone Silvestri, and Joao HC Costa. "A machine learning and optimization framework for the early diagnosis of bovine respiratory disease." *IEEE Access* 11 (2023): 71164-71179.

[13] Moutaouakil, Khalid El, Hamza Jdi, Brahim Jabir, and Noureddine Falih. "Digital Farming: A Survey on IoT-based Cattle Monitoring Systems and Dashboards." *AGRIS on-line Papers in Economics and Informatics* 15, no. 2 (2023): 31-39.

[14] Poyyamozhi, Mukilan, Balasubramanian Murugesan, Narayanamoorthi Rajamanickam, Mohammad Shorfuzzaman, and Yasser Aboelmagd. "IoT—A

Promising Solution to Energy Management in Smart Buildings: A Systematic Review, Applications, Barriers, and Future Scope.” *Buildings* 14, no. 11 (2024): 3446.

[15] Islam, Sk Injamamul, Foysal Ahammad, and Haitham Mohammed. ”Cutting-edge technologies for detecting and controlling fish diseases: Current status, outlook, and challenges.” *Journal of the World Aquaculture Society* 55, no. 2 (2024): e13051.

[16] Sharma, Kushagra, and Shiv Kumar Shivandu. ”Integrating artificial intelligence and internet of things (IoT) for enhanced crop monitoring and management in precision agriculture.” *Sensors International* (2024): 100292.

[17] Jafar, Abbas, Nabila Bibi, Rizwan Ali Naqvi, Abolghasem Sadeghi-Niaraki, and Daesik Jeong. ”Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations.” *Frontiers in Plant Science* 15 (2024): 1356260.

[18] Liao, Yuxuan, Zhong Tang, Kun Gao, and Mohammad Trik. ”Optimization of resources in intelligent electronic health systems based on Internet of Things to predict heart diseases via artificial neural network.” *Heliyon* 10, no. 11 (2024).

[19] Michelena, Álvaro, Óscar Fontenla-Romero, and José Luis Calvo-Rolle. ”A review and future trends of precision livestock over dairy and beef cow cattle with artificial intelligence.” *Logic Journal of the IGPL* (2024): jzae111.

[20] Zhao, Yanan, Lu Zhang, Aihua Wang, and Dong Zhou. ”Biosensor Technology: Advances and Applications in Livestock Infectious Disease Diagnosis.” *Veterinary Sciences* 12, no. 1 (2025): 23.