UNIVERSITY OF GHANA, LEGON

DEPARTMENT OF COMPUTER ENGINEERING

SCHOOL OF ENGINEERING SCIENCES

COLLEGE OF BASIC AND APPLIED SCIENCES

FINAL YEAR THESIS

ON

DEVELOPMENT OF AN INTEGARTED WEARABLE DEVICE FOR REMOTE MONITORING OF PREGNANT WOMEN IN GHANA

PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE BACHELOR OF SCIENCE DEGREE IN COMPUTER ENGINEERING

OWOH EINSTEINA OFUNNE 10943874

ANANE GEORGE NYARKO 10947340

ADIKA NATHANIEL AGBESI JUNIOR 10957036

NAME OF SUPERVISOR: DR. GODFREY MILLS

NAME OF CO – SUPERVISOR: PROF. ELSIE EFFAH KAUFMANN

DATE OF SUBMISSION: SEPTEMBER 15, 2025.

DEVELOPMENT OF INTEGRATED WEARABLE DEVICE FOR REMOTE MONITORING OF PREGNANT WOMEN IN GHANA

By

Owoh Einsteina Ofunne	(10943874)
Anane George Nyarko	(10947340)
Adika Nathaniel Agbesi Junior	(10957036)

Submitted to the Department of Computer Engineering in Partial Fulfilment of the Requirements for the Degree of Bachelor of Science in Computer Engineering

University of Ghana

(September 15, 2025)

Name of Student: OWOH EINSTEINA OFUNNE

Signature of Student: ______________________________

Name of Student: ANANE GEORGE NYARKO

Signature of Student: ______________________________

Name of Student: ADIKA NATHANIEL AGBESI JUNIOR

Signature of Student: ______________________________

Name of Supervisor: DR. GODFREY A. MILLS

Signature of Supervisor: ______________________________

Name of Co-Supervisor: PROF. ELSIE EFFAH KAUFMANN

Signature of Co-Supervisor: ______________________________

Name of Head of Department: DR. PERCY OKAE

Signature of Head of Department: ______________________________

DECLARATION OF ORIGINALITY

Department of Computer Engineering

We certify that the content of this thesis document is our own work except where indicated by reference. We confirm that we have read and understood the guidelines on plagiarism in the Computer Engineering undergraduate thesis Handbook, including the University of Ghana's policy on plagiarism. We have acknowledged in the text of this document all sources used. We have also referenced all relevant texts, figures, data, and tables quoted from books, journals, articles, reports, Internet websites, and works of other people. We certify that we have not used the services of any professional person or agency to produce this thesis document. We have also not presented the work of any student present or past from other academic or research institutions. We understand that any false claim in respect of this thesis document will result in disciplinary action by the regulations of the University of Ghana.

Please complete the information below by hand and in BLOCK LETTERS.

Student Name: ....

Index Number: ....

Student Signature: .... Date.......................................

Student Name: ....

Index Number: ....

Student Signature: .... Date.......................................

Student Name: ....

Index Number: ....

Student Signature: .... Date.......................................

Name of Supervisor(s): ....

Supervisor’s Signature: .... Date.......................................

ABSTRACT

Preterm delivery (PTD) is a significant global issue involving babies born before 37 weeks of gestation. Traditional monitoring methods include manual uterus palpation and using Intrauterine Pressure Catheters (IUPC) devices. However, these methods face challenges, including invasive IUPC and infection risks, which can lead to miscarriages. Manual palpation often yields inaccurate results, making it difficult to differentiate between preterm and full-term deliveries. Preventing, detecting, and monitoring preterm conditions is crucial for deploying rapid intervention measures to preserve human life. The need for continuous, reliable, and accurate methods to detect and track labour conditions is a challenge. Standard methods for assessing PTD risk include maternal history, clinical examination, cervical length measurement, and biochemical markers. However, machine learning techniques can extract trends and patterns in data to make predictions, making them noninvasive. This thesis presents a monitoring device using a Random Forest Classifier to track and detect preterm conditions in patients.

Two devices were designed for monitoring and detection tasks: a patient monitoring device and a central controller device. The patient monitoring device captures contractions and extracts critical parameters (duration, frequency, and intensity), which are transmitted wirelessly to the controller device. The machine learning algorithm evaluates the patient's data to predict preterm conditions, classify them, and generate alerts. The model was developed and trained on a public dataset, Physionet, and tested on 517 samples. The model achieved 98% accuracy in training and validation.

The detection system, developed using pre-diagnosed patient data and actual data from volunteers, was tested to detect preterm and non-preterm conditions with 99.7% and 97.5% performance accuracies, respectively. This system is reliable for monitoring and tracking preterm delivery conditions, and its ability to continuously monitor patients non-invasively and generate alerts for

critical conditions will help in maternal healthcare delivery, especially in resource-constrained environments.

ACKNOWLEDGEMENT

We would like to thank the Almighty God for his unfailing grace and mercies for beginning our studies and project work with us and ending successfully with us. We would also like to express our deepest gratitude to our supervisors, Dr. Godfrey A. Mills and Prof. Elsie Effah Kaufman who have nurtured us with their mentorship, expertise, and patience throughout this journey. A big thank you for your helpful feedback and suggestions. Their comments helped us to improve our thesis significantly. This project would not have been possible without their support, encouragement, and commitment.

We would like to acknowledge all the project team members of Dr. Godfrey A. Mills, Mr. Kenneth Broni, and the teaching assistants, especially Douglas Kojo Gidimadjor and Ernest Dumenu who generously shared their time and insights to contribute to the progress of the project.

We would like to extend our heartfelt appreciation to the Google College Outreach for Machine Learning for their financial support for the machine learning training program, through which we acquired technical knowledge and skills, which helped with the machine learning development for the project. We also want to thank our respective families for their boundless love and understanding. Their unwavering support through prayers has been our anchor in both good and challenging times and we say, “Thank you”.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

WHO	-	World Health Organization
EHG	-	Electrohysterography
UC	-	Uterine Contraction
PTD	-	Pre-Term Delivery
IUPC	-	Intrauterine Pressure Catheter
AUC-ROC	-	Area Under Receiver Operating Characteristics TABLE OF CONTENT

CHAPTER 1 - INTRODUCTION

1.0 Introduction

This chapter presents an introduction to preterm delivery, how preterm conditions are monitored, the challenges of monitoring preterm conditions and the identification of the problem required for the solution. The chapter also presents the objectives of the work including the project scope and the significance of the project to society.

1.1 Background

Preterm delivery (PTD) is characterized by babies being born prematurely, which can result in underdeveloped organs and increased health risks. Recognizing the unique characteristics of PTD is vital for effective medical management and improving outcomes for premature infants. Human implantation of the embryo can result in a regular or abnormal pregnancy, which can be ended by spontaneous or planned abortion or delivery [1]. Preterm delivery occurs between twenty-four (24) and thirty-seven (37) weeks; term delivery occurs between thirty-seven (37) and forty-two (42) weeks; and post-term delivery occurs beyond forty-two (42) weeks [1]. Thirty-seven (37) to fortytwo (42) weeks is the average length of a pregnancy [1][2]. Preterm births involve underdeveloped organs, low birth weight, and potential long-term health challenges like respiratory issues. In contrast, Term births indicate full-term development, with infants experiencing fewer complications, leading to better overall health outcomes. The number of pregnancies that resulted in deliveries beyond twenty (20) weeks or in babies weighing more than five hundred (500) grams is referred to as parity [1].

Preterm delivery is a major global health problem, affecting about 15 million births yearly and accounting for more than 80% of neonatal deaths [3]. Preterm delivery can have severe and lifelong impacts on the child's physical, mental, and social development, and the emotional and economic well-being of the family and society [4]. Therefore, preventing, detecting, and managing preterm delivery is a crucial public health priority. One of the challenges in preventing preterm delivery is the need for more reliable and accurate methods to detect and track labour conditions. Currently, the most common methods to assess the risk of preterm delivery are based on maternal history, clinical examination, cervical length measurement, and biochemical markers. However, these methods have limited sensitivity and specificity and cannot provide real-time information on the dynamic changes in uterine activity and fetal status during labour [5]. Moreover, some of these methods are invasive and carry the risk of infections, injury, membrane rupture, and other complications [6].

Machine learning is a branch of artificial intelligence that enables computers to learn from data and make predictions or decisions without explicit programming. Machine learning techniques have been widely applied in various fields of medicine, such as diagnosis, prognosis, treatment, and prevention [7]. Machine learning techniques can also analyze complex and high-dimensional data from various sources, such as physiological signals, medical images, and electronic health records [8].

In this project, we designed and developed a monitoring device for detecting and tracking labour conditions for preterm delivery using machine learning techniques. The preterm detection system consists of two devices (a patient device and a controller device); the patient device has sensors (surface electrodes) and an embedded microcontroller system for the measurement of uterine contraction signals and wireless transmission of the data to the controller device. The controller device is the user (healthcare service provider) monitoring device and it has a microcontroller system which is integrated with a machine learning algorithm to process the patient data and predict the likelihood of preterm or term birth. The algorithm also further classifies the preterm condition into different categories or classes (Regular and Irregular). The device aims to solve the following problems:

1. lack of continuous and non-invasive monitoring of the frequency, duration, and intensity of uterine contractions;
2. difficulty in analyzing real-time data and accurately predicting preterm delivery;
3. absence of an alert mechanism that prompts users of any prevailing potential preterm labour conditions or complications to facilitate rapid response;

This project is an extension of a previous on the "*Development of uterine contraction monitoring system*" by Derek Kwaku Degbedzui [9] from the Department of Biomedical Engineering, School of Engineering Sciences, University of Ghana. The pioneering work was further continued by two students of the Department of Computer Engineering, Ernest Agyemang and Michael Warren [10]. While the work of Derek focused on the measurement of the duration and frequency of contraction signals to establish preterm conditions of patients, the work of Ernest and Michael focused on the measurement of the intensity in addition to the frequency and duration of the contraction signals to determine preterm conditions.

In this study, machine learning techniques have been integrated in addition to the development of the patient device and the controller device for the detection and classification of preterm delivery. This will help continuous monitoring of the patient to predict the occurrence of preterm delivery accurately through machine learning by detecting irregular uterine parameters that can indicate the onset or progression of labour. The solution will also help to provide early interventions by alerting the user and the healthcare service provider about the risk of preterm delivery or any signs of complications or distress. Furthermore, it can help to free health professionals from the stress of periodic check-ups on patients because of remote recordings of the device. Additionally, it can provide healthcare professionals with a comprehensive tool that aids and analyzes data for making informed decisions.

1.2 Problem statement

Detection of labour conditions of patients has become critical due to the challenges and impact it can pose on both the patients and the baby. To determine preterm conditions, different methods are employed. These range from manual approaches to the use of intelligent techniques.

Conventional methods of labor monitoring involve assessing uterine contractions and fetal wellbeing. Two common techniques are manual palpation, where healthcare providers feel the uterus to gauge contraction strength and frequency, and by using Intrauterine Pressure Catheters, which measure the strength of uterine contractions by placing the IUPC into the amniotic space during labour. The conventional methods for labour monitoring often need to catch up in terms of accuracy and real-time capabilities. Manual detection mechanisms, a mainstay in healthcare, are prone to errors and subjectivity, leading to inaccurate results. Moreover, these methods cannot provide continuous, periodic monitoring of uterine contractions, which is crucial for understanding the progression of labor and making informed decisions regarding delivery timing. Predicting whether a birth will be preterm or full-term remains to be discovered, with existing methods failing to provide reliable foresight.

Although invasive methods offer some insights into labour conditions, they introduce significant risks. These techniques, such as amniocentesis or fetal monitoring via electrodes, carry the potential for infections, injuries to the mother or fetus, and membrane ruptures. Consequently, there is a critical need for non-invasive alternatives that provide accurate and real-time information about labour conditions, particularly the risk of preterm delivery.

This research endeavor aims to bridge these gaps in labour monitoring by developing a novel device that harnesses the power of machine learning. By leveraging advanced algorithms and realtime data analysis, the proposed device seeks to provide more accurate and continuous monitoring of uterine contractions, thus improving the precision of labor condition assessment. Additionally, the device's machine learning capabilities will be harnessed to predict the likelihood of a preterm birth, enabling healthcare providers to take proactive measures in managing high-risk pregnancies.

In summary, this project addresses the shortcomings of existing labour monitoring methods by proposing an innovative, non-invasive device that employs machine learning techniques. This device holds the promise of revolutionizing the field of obstetrics by enhancing the accuracy and timeliness of labour condition assessment, ultimately contributing to better outcomes for both mothers and newborns, particularly in cases of preterm deliveries.

1.3 Project objectives

The objectives of this project revolve around the creation of devices bolstered by the integration of machine learning techniques to address the critical issues associated with labour condition monitoring, with a specific emphasis on distinguishing between term delivery and preterm delivery. The objectives of this work are in three holds:

- The primary objective entails designing and developing a comprehensive monitoring device that can effectively harness the power of machine learning algorithms to detect and classify whether a birth is likely to be term or preterm based on the collected data. This project component is crucial as it strives to improve the accuracy and reliability of pregnancy outcome predictions, ultimately contributing to better prenatal care and management.
- The second key objective focuses on creating a sophisticated hardware system capable of continuously and accurately measuring real-time parameters associated with uterine contractions. These parameters include contraction frequency, duration, and Intensity. By continuously monitoring these critical indicators, the device aims to provide a comprehensive and real-time assessment of labor conditions, empowering healthcare

providers with valuable information for making informed decisions regarding the timing and nature of medical interventions.

- Lastly, the project aims to enhance user engagement and safety by implementing an alert system within the device. This alert mechanism will proactively notify users, including healthcare professionals and expectant mothers, of the prevailing labor conditions through regular alarms. This timely notification system ensures that potential issues or complications during labor are promptly recognized and addressed, ultimately improving the mother's and newborn's overall safety and well-being.

By achieving these objectives, the research aims to significantly enhance the quality of healthcare services that are provided during pregnancy and childbirth, potentially reducing the risks associated with preterm deliveries and improving maternal and neonatal outcomes.

1.4 Scope of project

The scope of our project, focused on the "Device for Detection and Tracking of Labor Conditions for Pre-Term Delivery using Machine Learning Techniques," encompasses the development and deployment of a comprehensive monitoring and prediction system. This system is designed to enhance the care of pregnant women by continuously assessing uterine contractions and providing valuable insights to healthcare providers and users.

Our device, consisting of a patient monitor and a central system, is dedicated to non-invasively measuring uterine contraction parameters such as frequency, duration, and intensity through surface electrodes. These measurements are then wirelessly transmitted to the central system in real-time, facilitating immediate analysis using machine learning algorithms. This not only ensures accurate predictions of labor conditions but also allows for the classification of these conditions.

Moreover, the central system's capabilities extend beyond mere analysis. It diligently stores the collected data in a database for future reference and research purposes. One crucial feature of our system is its ability to provide timely alerts and feedback to healthcare providers and users through

buzzer beeps. This ensures early intervention in cases of irregular contractions, potentially preventing pre-term delivery complications.

However, it's essential to note that our project does not include the development of a web application for delivering results to healthcare professionals. While this feature is not within the current scope, it leaves room for future expansion and integration to improve accessibility and convenience for healthcare providers.

Additionally, our project's focus is primarily on the connection of patient monitoring devices to the central controller system. The potential scalability of the system to include more devices is not addressed in this scope, but it lays the foundation for future enhancements to accommodate a broader patient population.

1.5 Significance of project to development

The relevance of this project lies in its potential to significantly improve the field of obstetrics and maternal healthcare, particularly in the context of preterm deliveries, through the development of an advanced labor monitoring device enhanced with machine learning techniques. Some of the benefits of the work to society are:

- The project aims to detect irregular uterine parameters, such as frequency, duration, and intensity of contractions, which indicate labour onset and progression. This can alert healthcare professionals and pregnant women about the risk of preterm delivery, enabling early interventions like medication, cervical cerclage, or labor induction, depending on the situation and the fetus' gestational age.

- The project provides a data-driven tool for healthcare professionals to manage labor conditions. It uses machine learning techniques to analyze data and predict labor conditions as term or preterm. This enhances medical interventions' precision, ensuring safer and more effective care for expectant mothers and their newborns. The device is highly relevant in healthcare, providing valuable insights for informed decision-making.

- The project aims to reduce healthcare professionals' workload by remotely recording and transmitting data on uterine contraction parameters, eliminating the need for in-person check-ups. This reduces stress and workload, improves patient comfort, and saves time, resources, and staffing. It represents a significant advancement in modern healthcare by offering a more efficient and patient-centered approach to labour monitoring.

1.6 Outline of thesis

The thesis is structured into five chapters. Chapter 1 introduces the topic, background, problem statement, research questions, objectives, significance, scope, and limitations of the project. Chapter 2 provides critical review of existing solutions for the detection of preterm conditions in literature to establish the justification for the project. The review focuses on the detection of preterm delivery, labour conditions, detection and tracking methods, and machine learning techniques. The chapter also presented the proposed project following the identification of the research gaps and opportunities for further research. Chapter 3 of the thesis describes the design and development process of the device for detecting and tracking labor conditions for pre-term delivery using machine learning techniques. It explains the requirements, specifications, architecture, components, modules, algorithms, tools, technologies, and techniques used or developed to create the device. Chapter 4 describes the implementation and testing process of the developed device. It explains how the device will be deployed, configured, integrated, and evaluated in an actual or simulated environment. It also presents and interprets the results of the testing and evaluation. The thesis concludes with Chapter 5, which summarizes the project's main findings, contributions, and limitations. It also provides suggestions for future research and practice based on the results.

CHAPTER 2 – LITERATURE REVIEW

2.0 Introduction

Maternal and fetal healthcare has witnessed remarkable advancements over the past few decades, with a growing focus on leveraging technology to enhance the well-being of mothers and their unborn children. In particular, preventing preterm birth, a leading cause of neonatal mortality and morbidity worldwide [11], has emerged as a critical challenge in obstetrics.

To address this pressing issue, there has been a burgeoning interest in developing innovative and non-invasive methods for the early detection and tracking of labour conditions leading to preterm delivery. The convergence of machine learning techniques, wearable devices, and comprehensive data analysis has opened new avenues for research in this domain. This chapter explores the existing literature, encompassing studies, methodologies, and technologies employed to pursue this crucial goal.

We aim to provide a thorough understanding of the current state of knowledge in the field, highlighting key breakthroughs, challenges, and emerging gaps. By examining the literature, we seek to uncover the pivotal role that machine learning techniques play in designing and implementing devices for monitoring labor conditions and predicting preterm birth risk.

2.1 Survey of existing solutions

1. A validation of electrohysterography for uterine activity monitoring during labor [12]

Jacod, et al. [12] conducted research and the study compared the monitoring of pregnant women using EHG and IUPC. 32 expectant women in labor were enlisted by the authors to record EHG and IUPC simultaneously for at least 30 minutes. They automated the detection of uterine contractions using a straightforward algorithm. The program determined the uterine peaks' amplitude and duration. The authors concluded that EHG has a sensitivity rate (true positive rate)

of 94.5% for identifying uterine contractions. The false negative rate, or uterine contractions that happened but went undetected, was not mentioned by the authors. Additionally, because pregnant women must be hospitalized to be observed, the study is not financially viable because it requires hospital equipment, medical personnel, and staff for monitoring. Additionally, it is performed at the time of birth, making it inappropriate to monitor pregnant women at any stage of pregnancy.

2. Home uterine monitoring for detecting preterm labor [13]

In 2017, Urquhart et al. [13] conducted a study on the impact of home uterine activity monitoring on the health of pregnant women at high risk of premature birth and premature newborns. The authors wanted to know if utilizing a home monitoring system would result in the same outcomes for the pregnant woman's and premature baby's health status as not using one throughout pregnancy. The authors compiled randomized control trials of home uterine activity monitoring for pregnant women at high risk of premature birth from 15 research. The authors studied the care of pregnant women with and without home uterine activity monitoring, as well as with and without patient education programs. The study was well planned and executed. The authors assessed the trials for inclusion and bias risks, as well as the data for accuracy. They gathered and evaluated information from 6008 patients. The researchers discovered that pregnant women who used home uterine activity monitoring were less likely to have a premature birth before the 34th week of pregnancy. The researchers also discovered that babies born preterm to a pregnant woman who utilized a home monitoring of the uterine activity system were less likely to require a stay in the neonatal intensive care unit. Pregnant women who used home uterine activity monitoring had more unscheduled prenatal visits.

3. Telemonitoring system oriented towards high-risk pregnant women [14]

Vermeulen-Giovagnoli et al. [14] suggested Nemo telemonitoring healthcare for home signals monitoring of fetal and maternal heart rates, uterine contractions, and fetal electrocardiogram. The system is made up of sensors from the Nemo Healthcare system, a web server, and a web application. The Nemo Healthcare system sensors gather the pregnant woman's vital signs in realtime and wirelessly transmit them to the web server through the Internet. The web tool then allows the doctor to monitor and diagnose the pregnant woman's health status. To simulate the signals, the system was tested using pre-defined data. Although it is a well-designed system, it necessitates the use of a server and a physician to monitor the findings, which can be costly and necessitates the physicians' time to monitor and evaluate the data.

4. Framework to monitor pregnant women with a high risk of premature labor using sensor networks [15]

Allahem et al [15] proposed a safe, simple, and low-cost system to monitor pregnant women who were at high risk of premature labour. The system consisted of a wireless body sensor network to non-invasively monitor the uterine contractions and trigger a warning via a smartphone if the readings were outside the normal thresholds. Based on the advantages of a Wireless Body Sensor Network and Electrohysterography, the author designed a proof-of-concept prototype and tested it for reliability, performance, and power consumption. The system consisted of a sensor and a smartphone that used the Android operating system. The sensor was implemented on the pregnant woman's abdomen as a non-invasive method to continuously measure EHG signals of the uterus and send them to the smartphone. The smartphone analyzed the received data and triggered a warning for readings outside the threshold. To test the application, the authors used a uterine contraction signal dataset of pregnant women in labor from the PhysioNet databases. The application used an average of 40% of the CPU power consumption and 26.64 MB of memory was allocated in terms of memory usage out of 1.80 GB of the device's memory.

5. Deep neural network for semi-automatic classification of term and preterm uterine recordings, Artificial Intelligence in Medicine [16]

Chen et al [16] proposed the use of a deep neural network (DNN) to classify uterine recordings. The DNN was trained on a large dataset of uterine recordings to learn the patterns that distinguish between term and preterm recordings. The authors also used a semi-automatic approach, where a human expert initially identified and marked the recordings as term or preterm, and then the DNN was used to classify the remaining unmarked recordings. The authors used a cross-validation method to validate the performance of the DNN. The dataset was divided into training, validation, and testing sets. The DNN was trained on the training set and the hyper-parameters were tuned on the validation set. The performance of the DNN was evaluated on the testing set. The results obtained from the testing showed that the proposed DNN achieved a high classification accuracy of 94.3% in distinguishing between term and preterm uterine recordings. The proposed semiautomatic approach reduced the time required for human expert labeling, thus making the classification process more efficient and scalable. In conclusion, the authors demonstrated the effectiveness of using a DNN for the semi-automatic classification of term and preterm uterine recordings. The proposed method achieved high accuracy and reduced the time required for human expert labeling. The proposed approach has the potential to be used in clinical practice for the diagnosis of preterm birth.

The limitations faced in this paper are as follows;

The semi-automatic approach used in the study still requires a human expert to initially label a portion of the recordings, which may introduce bias or errors in the classification process. Also, the study did not compare the performance of the proposed DNN with other existing methods for the classification of uterine recordings, which could provide more insight into the strengths and weaknesses of the proposed approach.

6. Automatic recognition of uterine contractions with electrogastrogram signals based on the zero-crossing rate [17]

Song et al [17] focused on the development of an automatic recognition system for uterine contractions using electrogastrogram (EHG) signals. The authors aimed to solve the problem of manually identifying and recognizing uterine contractions during labor. This process is usually time-consuming and requires a high level of expertise, leading to the need for an automatic system that can recognize contractions more efficiently and accurately. The authors used the zero-crossing rate (ZCR) to extract features from the EHG signals. The ZCR is a commonly used feature in signal processing, which measures the number of times the signal crosses the zero-axis per unit time. The authors also used a linear discriminant analysis (LDA) classifier to classify the extracted features and distinguish between contractions and non-contractions. The authors conducted experiments on a dataset of EHG signals collected from 54 pregnant women during labor. The dataset was split into training and testing sets, and the performance of the proposed system was evaluated using different metrics such as sensitivity, specificity, and accuracy. The proposed system achieved a high level of accuracy in recognizing uterine contractions, with an overall accuracy of 97.3%. The sensitivity and specificity were 97.8% and 96.8%, respectively, indicating a low rate of false positives and false negatives. The authors successfully developed an automatic recognition system for uterine contractions using EHG signals and the ZCR feature extraction technique. The proposed system achieved a high level of accuracy, demonstrating its potential to be used in clinical settings to improve the efficiency and accuracy of labor monitoring.

The system was evaluated on signals recorded during the first stage of labor, which is typically longer and less intense than the second stage of labor. The system used only the ZCR feature for contraction recognition which may not capture all relevant information from the EHG signals. The system relied on the assumption that EHG signals are of high quality and free from noise or artifacts.

7. Preterm Baby Birth Prediction using Machine Learning Techniques [18]

Begum et al [18] developed a system to predict pre-term babies. They intended to study and recognize the main factors corresponding to premature babies with the consultancy of specialized doctors initially. The main factors were the mother's weight before pregnancy, the mother's age, the number of previous pregnancies, the mother's BMI, cervical problems, etc. After that, the dataset was pre-processed and normalized. The dataset consisted of 3500 samples of data and there were 2 classes and 10 attributes to predict preterm birth. Four binary classifiers i.e., KNN, Decision Tree, SVM, and Naïve Bayes were trained and tested. The investigation result showed the effectiveness of the projected system with 99% accuracy for Decision Tree. KNN, SVM, and Naïve Bayes had an accuracy of 98%, 91%, and 92% respectively.

2.2 Summary of existing solutions

Table 2. 1 Literature review

Authors	Problems	Technique(s) Used	Performance	Limitations
Jacod B. C et al [12]	A validation of electrogastrography for uterine activity monitoring during labor	Electrohysterography	Sensitivity 94.5%	Because pregnant women must be hospitalized to be observed, the study is not financially viable because it requires hospital equipment, medical personnel, and staff for monitoring. It is performed at the time of birth making it inappropriate for monitoring pregnant women at any stage of pregnancy

Urquhart C. et al [13]	Home uterine monitoring for detecting preterm labor	Home uterine monitoring	N/A	The studies included in the meta-analysis were of varying quality. Some of the studies were welldesigned and wellconducted, while others were not. This makes it difficult to draw firm conclusions from the study.
Vermeulen et al [14]	Telemonitoring system oriented towards high-risk pregnant women	Telemonitoring system (Consisting of two mobile web applications)	The system is considered a specialized tool because it satisfies the requirements and suitable parameters that should be monitored in	It necessitates the use of a server and a physician to monitor the findings which can be costly. It requires an internet connection to be accessed by users.

			hypertensive, diabetic, and highrisk pregnant women according to the specialized literature.	
H. Allahem et al [15]	Framework to monitor pregnant women with a high risk of premature labor using sensor networks	Data Fusion (Smartphone and Sensor)	The system uses CPU Power of 40% and Memory of 26.64Mb meaning the application consumes battery charge efficiently	Other pregnant womene might have different thresholds set by the authors which may lead to wrong predictions
Lili Chen et al [16]	Deep Neural Network for Semi-Automatic Classification of Term and Preterm Uterine Recordings	Deep Neural Network	Accuracy 94.3%	The performance of DNN was not compared with other existing methods which could provide more insight into the strengths and weaknesses of the proposed approach
Xiaoxiao Song et al [17]	Automatic Recognition of Uterine Contractions with Electrohysterogram	Zero-Crossing Rate	Accuracy = 97.3% Sensitivity = 97.8% Specificity = 96.8%	Data quality Limited validation Limited features extracted

	Signals Based on the Zero-Crossing Rate			
Manoara Begum et al [18]	Preterm Baby Birth Prediction Using Machine Learning Techniques	K-Nearest Neighbour Logistic Regression Decision Tree Naïve Bayes	Decision Tree outperformed the other algorithms with the following results Accuracy = 99% Recall = 99% Precision = 99% AUC-ROC = 99%	Small (3500 samples) and imbalanced datasets (63.8% not premature and 36.2% premature)

2.3 Proposed solution

Our planned project entails designing and creating an automated system that will monitor and categorize uterine contractions, classify them as regular or irregular contractions, and alert the medical staff when some irregularities occur. The project aims to give healthcare professionals a simple and practical tool to keep track of labor development.

The functions of the proposed system include the following:

1. Develop and design a device that allows for continuous and non-invasive monitoring of the Frequency, Duration, and Intensity of uterine contractions.

2. Determine whether uterine contractions are regular or irregular.
3. Implement an alert mechanism that prompts users of potential preterm labor conditions or complications.
4. Develop an intelligent system that employs machine learning techniques to analyze realtime data and accurately predict PTD.

CHAPTER 3 – SYSTEM DESIGN AND DEVELOPMENT

3.0 Introduction

This chapter describes the design and development process of the device for detecting and tracking labor conditions for pre-term delivery using machine learning techniques. The device is intended to provide a non-invasive, accurate, and reliable method for monitoring the labor progress and predicting the risk of pre-term delivery in pregnant women. The device consists of two devices: a patient monitoring device and a central one. The patient device has a sensor that measures the uterine contractions' duration, frequency, and intensity and alerts the user through a buzzer for any irregular contractions. The central device is located at the primary nurses' table and wirelessly receives the recorded data from the patient-side device. The central device also has a processing module that applies machine learning algorithms to analyze the data and extract relevant features and patterns. The following sections explain the requirements, specifications, architecture, components, modules, algorithms, tools, technologies, and techniques used or developed to create the device.

3.1 System overview and functions

The developed system comprises two independent functional units: the patient device for detecting and monitoring labor contractions and the central device carrying the trained and tested machine learning algorithms for classifying uterine parameters resulting in term birth or preterm birth. Both devices have user interfaces to display contraction parameters and an alert system to call on timely interventions from health personnel.

The patient device comprises a signal acquisition subsystem, a signal processing subsystem, a microcontroller, a power supply, display and alert subsystems, and a transmitter. For the patient's device's proper functioning, its power supply is first turned on to allow the device and its subsystems to start working. The signal acquisition system in the form of electromyography

(EMG) surface electrodes are placed on specific demarcations on the belly of the pregnant woman. Signal processing occurs after contraction signals have been detected to remove unwanted parts of the signal, such as noise, and also to apply amplification for accurate feature extraction.

The central device also consists of a controller subsystem with machine learning algorithms, a power supply, display and alert subsystems, a receiver, a storage card, and a wireless communication module for data transmission between devices. After powering the central device, the patient device establishes a wireless connection with the central device to enhance communication in the system. With cutting-edge machine learning algorithms, the proposed system is intended to revolutionize the monitoring and early diagnosis of labor conditions, particularly in situations of preterm birth. It combines a variety of sensors and intelligent algorithms to give medical professionals and pregnant mothers real-time information and alerts. Figure 3.1 shows the two main blocks of our system: the patient monitor device and the central controller device.

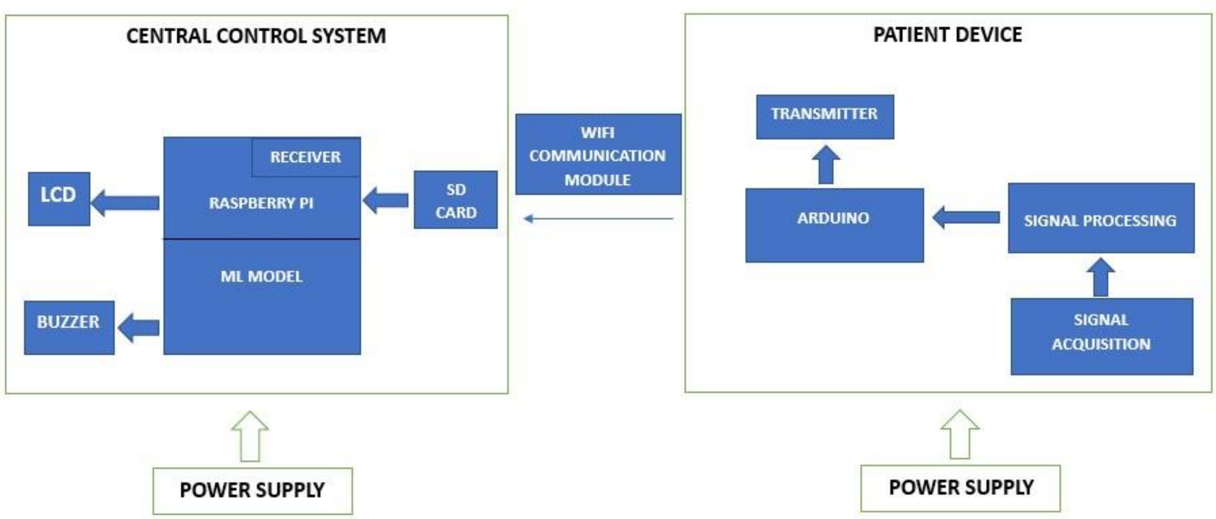

Figure 3. 1 Block diagram of the entire system

The operational logic of the system implemented in both patient and central devices to control the flow of activities is shown in Figure 3.2 and Figure 3.3. For the start of the operational logic in the patient device, the sensors attached to the microcontroller detect signals when the device is powered. Signals detected by the microcontroller by connected sensors then go through a processing stage to extract significant features of contraction signals.

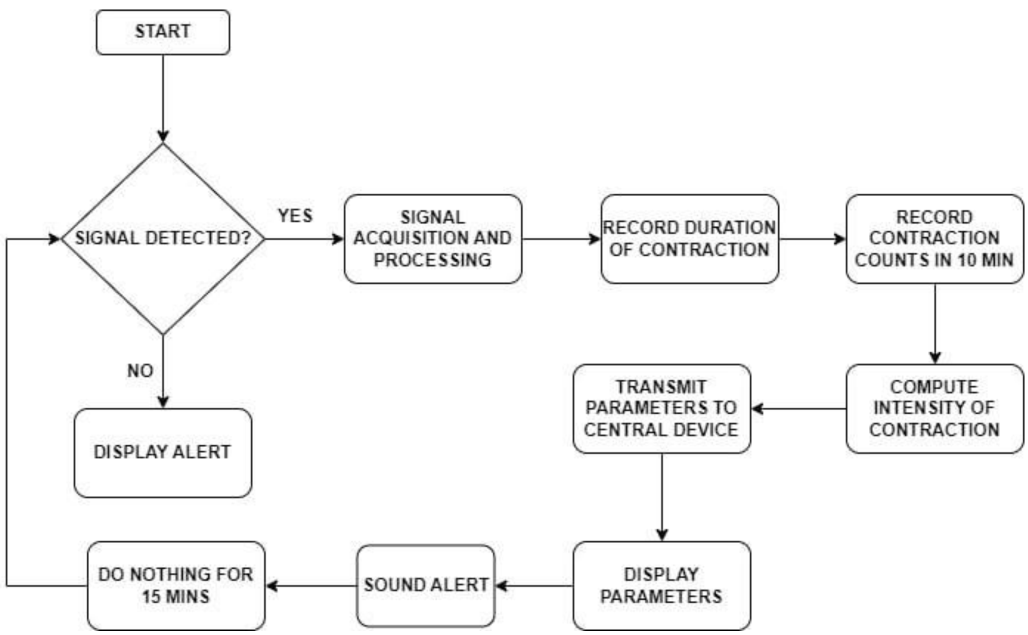

Figure 3. 2 Operational flow chart of patient device

Extracted features are computed to record uterine contractions' duration, frequency, and intensity. After contraction parameters are recorded and displayed successfully, the features extracted are transmitted to the central device wirelessly through the Wi-Fi module. A publish-subscribe protocol is adopted for the transmission of signal features. Messages received in the central device are stored in a database and serve as input to the machine learning model. Classification takes place to predict the occurrence of preterm or term birth.

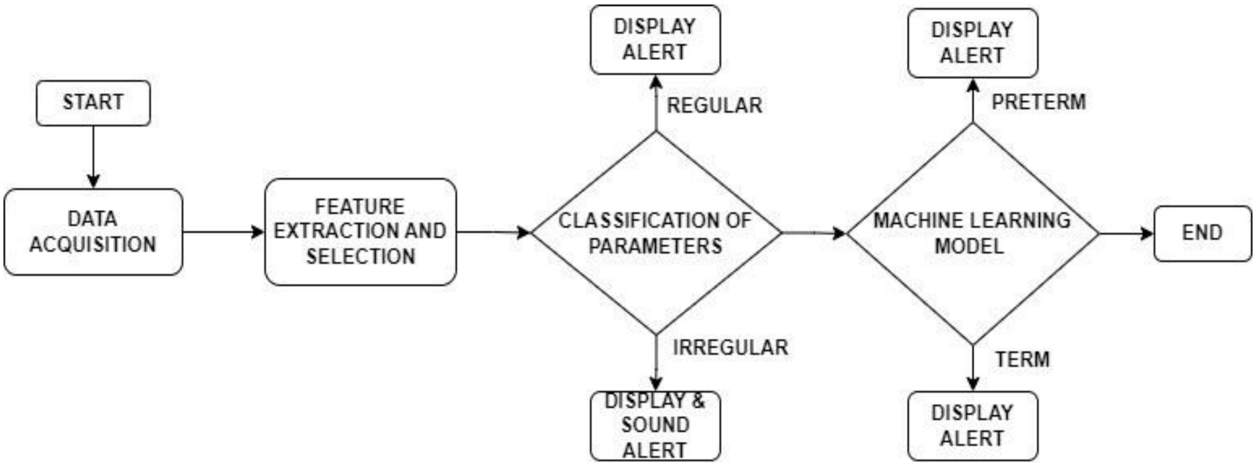

Figure 3. 3 Operational flow chart of the central device

3.2 Design requirement, analysis, and specifications

To develop the uterine contraction monitoring and detection system with unparalleled machine learning algorithms for preterm delivery, various requirements, and specifications were utilized to develop this project. The requirement analysis of the project was categorized into functional requirements and non-functional requirements and are examined as follows:

1. Functional requirements
   - Accurately measure and compute uterine contraction signals' contraction parameters (frequency, duration, and intensity).
   - Record and store contraction parameters from the contraction signals.
   - Transmit the data through wireless Wi-Fi communication from the patient device to the central or remote location for monitoring.
   - Alert health personnel for any irregularities in contractions.

- Timely predict the possibility of contractions leading to a term or preterm birth.

2. Non-functional requirements

The non-functional requirements listed below apply to both the patient device and the central device as a system:

- The system should be reliable and accurate.
- The system should be safe for the patient to use.
- The system should be tamper-proof.
- The system should comply with all applicable regulations to accommodate growing demand without reducing performance.
- The system should have a long-lasting power source.

3. Safety requirement

- The system must meet the WHO standards for classifying the parameters and conditions.
- The electrodes must be sterile at all times.
- The patient device must be designed to minimize the user's electrical shock risk.
- The central system must be designed to handle large amounts of data and process and analyze the data quickly and accurately.
- The device must safeguard the privacy and security of the data collected and stored.

4. Electrical requirement

- The wiring and components must be well connected to avoid system failure.
- The system must have enough or an accurate power supply.
- A low-voltage battery must power the patient's device to minimize the risk of electrical shock.

- The device must comply with the IEC 60601-1 standard 123, the international standard for medical electrical equipment and systems and its relevant parts.
- The device must be tested and verified to meet the electrical safety and performance requirements under normal and fault conditions.

3.3 Theoretical framework and assumptions

The proposed system for detecting and tracking labour conditions, particularly preterm birth, using machine learning techniques relies heavily on a theoretical framework and underlying assumptions, which are outlined in this section.

Theoretical Framework

The project is grounded in the principles of electrohysterography, a specialized medical diagnostic technique that involves the measurement and analysis of electrical signals associated with uterine contractions during pregnancy. EHG is considered a non-invasive and valuable tool for monitoring labor conditions and detecting irregularities. With the non-invasive methods, since the uterus is not readily accessible for direct recording [19], recording is done via surface electrodes or transducers to provide information on the frequency and duration of uterine contraction [20] and avoid the disadvantages of invasive measuring techniques. The non-invasive method eliminates threats of infection [21] and provides frequency and duration of uterine contraction [22]. The depolarization and repolarization processes of the myometrial cells within the uterine muscles generate EHG signals. During uterine contractions, these muscle cells exhibit electrical activity, producing low-amplitude electrical signals. The shape and pattern of EHG signals change throughout pregnancy, reflecting the progress of labor and any deviations from the ordinary course. The project also leverages the power of machine learning algorithms, mainly supervised learning techniques, for pattern recognition within EHG signals. It draws from established algorithms like Random Forest and Support Vector Machines for classification tasks.

Theoretical concepts from signal processing are integrated to preprocess and enhance raw EHG data. Techniques like filtering, feature extraction, and fast Fourier-frequency analysis contribute to accurate pattern identification.

Assumptions

Clinical Relevance: It is assumed that the patterns recognized by machine learning models align with clinically relevant markers for preterm labor, as established in the medical literature.

Supervised Learning Applicability: The applicability of supervised machine learning techniques to classify EHG patterns as indicators of preterm labor is assumed to be feasible. Sufficient labelled data for model training is also assumed to be available. The device assumes that the collected data is accurate and representative of the labour conditions of pregnant women.

Hardware Compatibility: The hardware components, including sensors and microcontrollers, are assumed to be compatible and function according to specifications.

User Acceptance: The system assumes user acceptance among medical professionals and expectant mothers, highlighting the importance of a user-friendly interface.

3.4 System design and development

This section aims to design and develop a device that can detect and track labor conditions for preterm delivery using machine learning techniques. The system design includes hardware systems such as patient monitor devices and central controller systems, wireless communication systems such as Wi-Fi and MQTT, and database development using Postgres SQL. The machine learning component of the system will be developed, trained, and tested to predict the likelihood of preterm delivery by monitoring the mother's uterine contractions. The system will provide an efficient and reliable way to monitor preterm labour conditions, which can help reduce the risk of preterm delivery.

3.4.1 Hardware System

A. Signal acquisition and processing unit

With existing works, EHG signals generated from electrical activities of the uterine smooth muscles are typically of low signal strength. These bioelectric signals measured from the skin's surface are mainly in the range of 0-2000µV (2mV). As a result, they were extending the knowledge of the type of amplifiers to ensure precise and accurate recording of uterine contractions by the microcontroller unit. In this project, surface electrodes served as input units to the system for detecting and acquiring contraction signals. The processing unit consisting of a relative band pass filter and instrumentation amplifier was simulated in Proteus. For the system, carefully selected 100µV and 80µV were used to represent the input signals. An instrumentation amplifier AD620 was chosen with a high accuracy of 40 ppm maximum nonlinearity, low offset voltage of 50 µV max, and offset drift of 0.6 µV/°C max [22]. Using the datasheet of an AD620, the gain was calculated using Equation 3.1.

$$A_V = 1 + \frac{49.99k}{Rg} \tag{3. 1}$$

where Rg is the gain setting resistor.

Standard resistor(R) and capacitor(C) values were selected for the design and simulations. A passive bandpass filter was designed consisting of both a low-pass RC filter and a high-pass RC filter. The cut-off frequencies (FC) were approximately 0.34Hz and 5Hz for the high pass filter and low pass filter using the formula;

1

$$Cut - off\ frequency(FC) = \frac{\quad}{2\pi RC} \tag{3.2}$$

For low pass filter, $fc = \frac{1}{2\pi(3.3k\Omega)(10\mu F)} = 4.82Hz$ (3. 3) For

high-pass filter, $fc = \frac{1}{2\pi(10k\Omega)(47\mu F)} = 0.34Hz$ (3. 4)

After passing through the filtration process, the signals are subjected to an inverting gain amplifier, LM741. The LM741 are general-purpose operational amplifiers with gain (G) calculation using Equation 3.5:

$$\frac{Rf}{Ri} = -\frac{10k\Omega}{1k\Omega} = -10 \tag{3.5}$$

B. Uterine contraction parameter evaluation

Parameter evaluation

In monitoring uterine contractions, essential factors play a crucial role in helping health personnel assess the progress of labor and ensure the well-being of both the mother and the baby. These factors are known as the contraction parameters. Parameters such as duration, frequency, and amplitude (Intensity) provide valuable insight into contractions' strength, frequency, and regularity. In this section, we will delve into these parameters, their significance, and how they are classified; 1. Duration(D) of a uterine contraction refers to the length of time that a contraction lasts from the beginning of the contraction until it subsides or ends. Monitoring the duration of contractions is essential because it indicates how long the uterine muscles are actively contracting. Prolonged contractions can lead to maternal exhaustion and fetal distress, making it a critical parameter to observe during labour. It is typically measured in seconds, starting when the contraction begins and stopping when it ends.

2. Frequency(F) of uterine contractions represents how often contractions occur within a specific timeframe, usually in ten minutes. It is measured as the time between the beginning of one contraction and the beginning of the next, as labour contractions occur at regular

intervals with increasing frequency. The frequency also provides insights into the stage of labour. The frequency is measured as contractions per unit of time in seconds.

3. In terms of labour, the Intensity (I) is the strength or force of the contractions above the basal tone (baseline). It measures how forcefully the uterine muscles are contracting. The Intensity of contractions is vital because it affects the efficiency of labour. Muscular contractions help dilate the cervix and push the baby through the birth canal. Monitoring amplitude ensures that contractions are powerful enough to progress labor, not overly strong to harm the baby or exhaust the mother. The Intensity (amplitude of contraction signal) is often measured in millivolts (mV) using an electromyography (EMG) sensor. These sensors detect the electrical activity of the uterine muscles, which correlates with contraction strength.

Table 3. 1 Contraction parameters

UC Parameters	**Normal uterine contraction parameter readings**

Duration	✞ Lasts 45 – 80 seconds ✞ Should not exceed 90 seconds
Frequency	✞ 2 – 5 contractions every 10 minute ✞ Should not occur more frequently than 1 minute
Intensity	✞ Strength of the contraction at its peak ✞ 50 – 80 millivolts (mV)

Feature extraction calculations

1. DURATION:

The period from the start of the occurrence of uterine contraction to the end of contraction. Width of the peak in the power spectrum.

$$R_T - S_T \tag{3. 6}$$

- Duration end time(R_T)
- Duration start time(S_T)

2. FREQUENCY:

This value estimates the dominant frequency or the frequency range within which the EHG signal is concentrated. Peak of the power spectrum.

$$f_{med} = i\frac{f_s}{N}, \sum_{i=0}^{i=N-1} P(i) \tag{3. 7}$$

where i is the i-th line of the power spectrum and fs is the sampling frequency.

3. INTENSITY:

The conventional method for evaluating signal amplitude changes is through the RMS. Given a time series of x(i); i=0,…,N-1, where N is the signal length. RMS was calculated as:

$$\text{RMS} = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1} x^2\,(i)} \qquad (3.8)$$

And the Intensity was then calculated as;

$$Intensity = (RMS)^2 \qquad (3.9)$$

RMS = Root Mean Square

The RMS value is a measure of the overall or total amplitude or power of the signal.

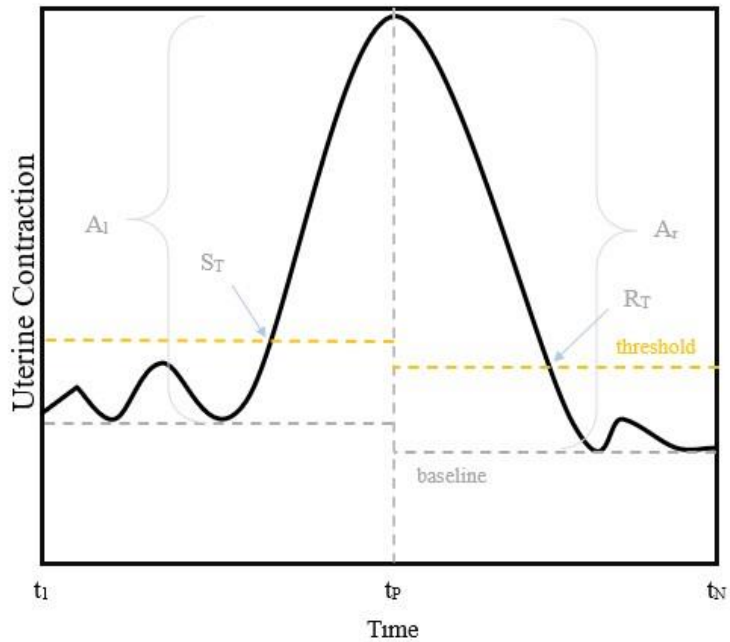

Figure 3. 4 Typical wave diagram of contraction signal

3.4.2 Machine Learning Development

Machine learning is a subfield of artificial intelligence (AI) consisting of an input, hidden, and output layer. The hidden layer usually varies depending on the machine learning algorithm being developed. For the development of our model, we followed the procedure in Figure 3.4 below;

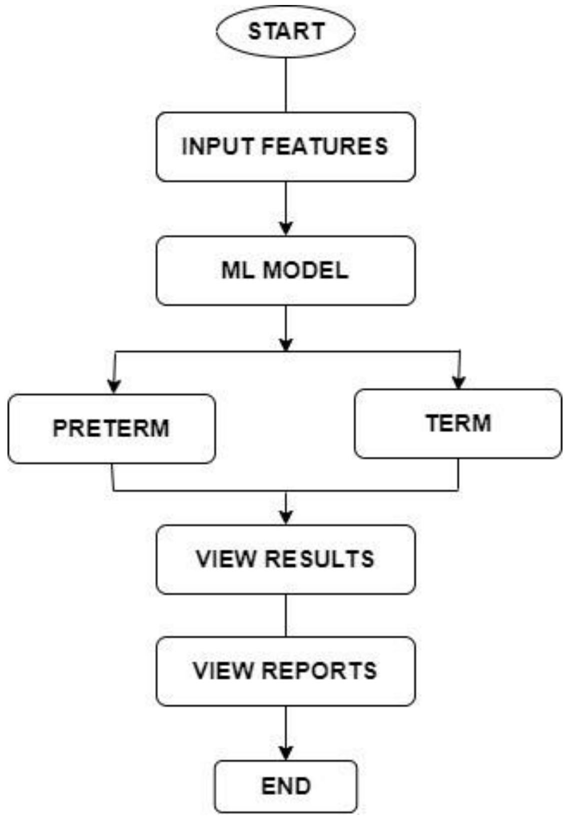

Figure 3. 5 Operational flow chart for machine learning model

Data processing

Data processing is a crucial component of machine learning (ML) that involves transforming, cleaning, and preparing raw data into a suitable format for ML model training and analysis. It plays a foundational role in the ML pipeline and significantly impacts the performance and accuracy of the resulting models. The EHG signals used in our study were from the open-access term-preterm EHG (TPEHG) database established in 2008 at the Faculty of Computer and Information Science, University of Ljubljana, Ljubljana [23]. The labelled dataset with a binary class system of “t” denoting true for preterm or premature birth and “f” denoting false for term birth or otherwise. The dataset was processed to extract essential features for classification. The true was assigned a binary "1," and the false was assigned a binary"0".

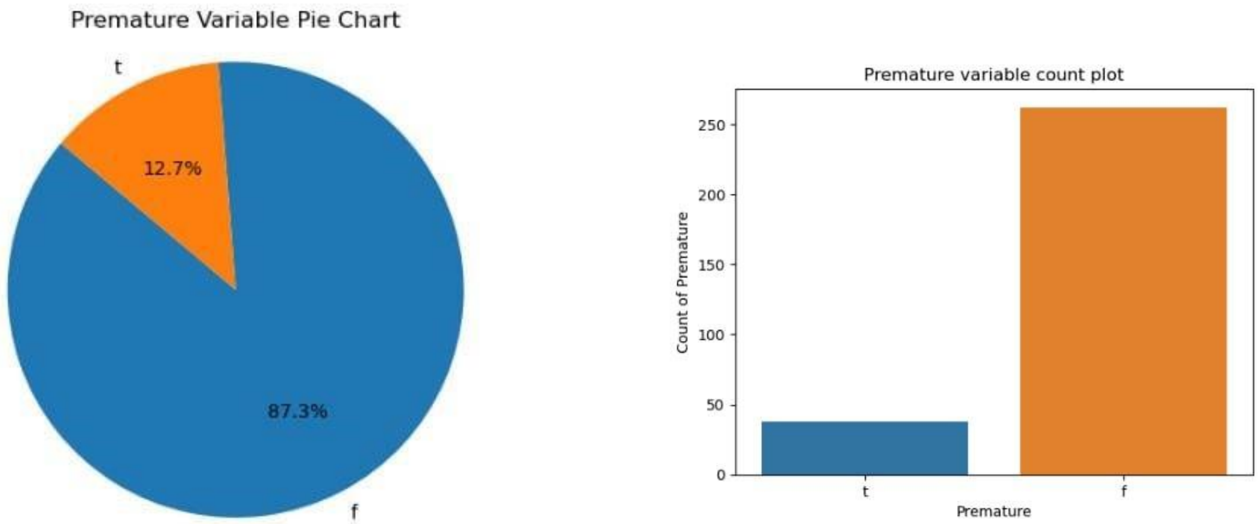

Figure 3. 6 Pie and Bar chart of the unbalanced dataset

Due to the imbalanced nature of the dataset, one class significantly outnumbers the other, causing an issue during classification. Techniques such as oversampling, undersampling, or using different evaluation metrics can address this problem.

The ADASYN technique was used to address the issue [24]. Adaptive Synthetic Sampling (ADASYN) is a sophisticated technique to tackle imbalanced datasets in machine learning, improving classification performance for underrepresented classes. ADASYN is designed to alleviate this issue by generating synthetic samples for the minority class during training. It first identifies which classes are underrepresented in the dataset. It does this by analyzing the distribution of samples in each class. Then, it concentrates on the minority class, where the imbalance issue is most critical. Through intelligent means, it generates synthetic samples for the

minority class. However, it does not create these samples uniformly. Instead, it assigns a higher weight to those minority samples that are more challenging to classify correctly. In other words, it generates more synthetic samples for instances in densely populated regions of the minority class.

The total number of records obtained after balancing the dataset is 517, with a preterm record of 255 and a term record of 262. This balance helps machine learning algorithms to learn the underlying patterns of both majority and minority classes more effectively.

Other oversampling methods:

1. Random Oversampling
2. Synthetic Minority Oversampling Technique (SMOTE)
3. Random Oversampling Examples (ROSE)
4. Generative Adversarial Network (GAN)

Model training and testing

Several machine learning algorithms were employed for training the predictive model. The features extracted from the training data were used as input variables, while the delivery outcome" label (categorized as term or preterm) served as the target variable. Hyperparameter tuning was performed to optimize model performance. 70% of the dataset was used to train the model, 10% was a validation set which helped tune hyperparameters and avoid overfitting, and a 20% test set was employed to evaluate model performance. Each algorithm was chosen for its potential to capture distinct patterns in the data. Hyperparameters such as; the number of estimators (500) which gives the number of decision trees in the forest, the minimum samples per leaf (8) describing the maximum number of samples needed to create a leaf node and a criterion (Gini) to measure the quality of node split. The selected algorithms were then trained on the training data, and

hyperparameters were fine-tuned using cross-validation techniques that helped optimize the model's performance. To mitigate the risk of overfitting and ensure the robustness of the model, k-fold cross-validation was applied, where k represents the number of folds. The trained algorithms were evaluated using the testing dataset. Common classification metrics, including accuracy, precision, recall, F1-score, and ROC curves, were used to assess the performance of the models in detecting labour conditions.

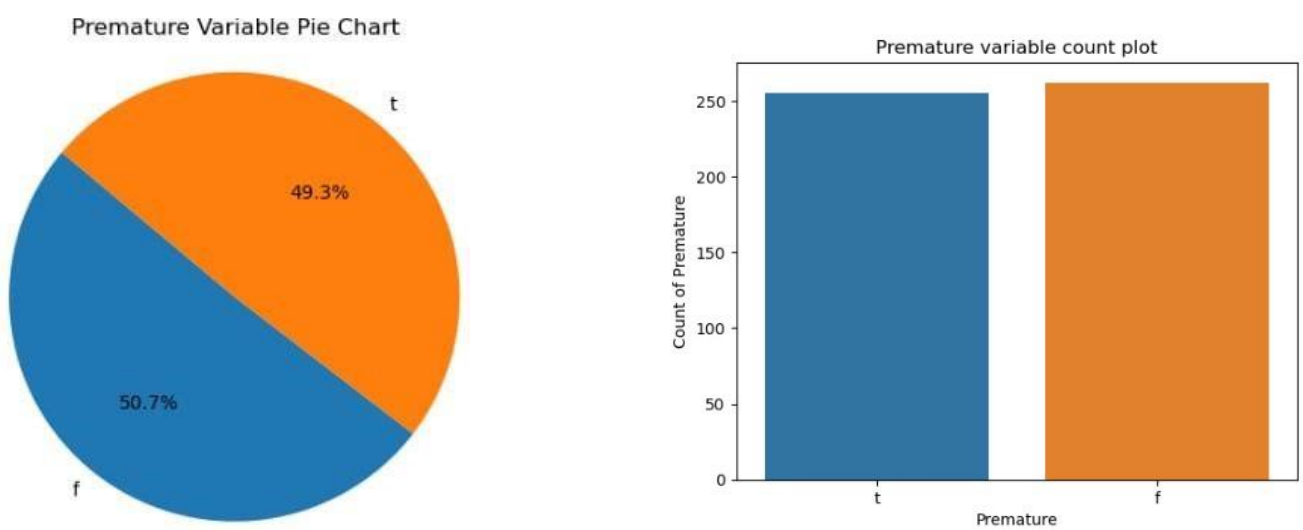

Figure 3. 7 Pie and Bar chart of the balanced dataset

Classification Algorithms/ Techniques considered

1. Logistic Regression
2. K-Nearest Neighbour

3. Support Vector Machine (SVM)
4. Decision Tree (DT)
5. Random Forest Classifier (RF)

The machine learning classification model was evaluated based on the metrics below;

1. Recall - Recall evaluates the ability of a classification model to identify all relevant instances in a dataset. It answers the question: "Out of all the actual positive cases, how many were correctly predicted as positive by the model?"

$$Recall = \frac{True\ Positives}{(True\ Positives + False\ Negatives)} \tag{3. 10}$$

2. Precision - Precision measures the ability of a model to correctly identify positive instances among the predicted positive cases. It answers the question: "Out of all the predicted positive cases, how many were truly positive?"

$$Precision = \frac{True\ Positives}{(True\ Positives + False\ Positives)} \tag{3. 11}$$

3. F1- score - The F1-Score measures the harmonic mean of precision and recall. Provides a balance between precision and recall, offering a single metric to evaluate the overall model performance.

$$F1 - score = \frac{2*(Precision \times Recall)}{(Precision + Recall)} \tag{3. 12}$$

4. Accuracy - Accuracy calculates the overall correctness of predictions made by a model. It answers the question: "Out of all the instances, how many were predicted correctly?"

$$Accuracy = \frac{True\ Positives + True\ Negatives}{(Total\ Instances)} \tag{3. 13}$$

5. AUC ROC curve - The ROC curve plots the true positive rate (recall) against the false positive rate at various thresholds, showing the model's ability to distinguish between classes. The AUC-ROC score quantifies the area under this curve. A higher AUC-ROC score (closer to 1) indicates better model performance in distinguishing between positive and negative cases. An AUC of 0.5 suggests random guessing.

Table 3. 2 Performance of Machine Learning Algorithms

Machine Learning Models	**Precision**	**Recall**	**F1 – Score**	**Accuracy**
Logistic Regression	0.82	0.78	0.80	0.81
K-Nearest Neighbour	0.87	0.85	0.86	0.86

Support Vector Machine	0.86	0.90	0.88	0.87
Decision Tree	0.80	0.82	0.80	0.80
Random Forest	0.98	0.98	0.98	0.98

Model selection

After a comprehensive evaluation of the model's performance concerning both accuracy and execution time, the decision was made to adopt the Random Forest classification algorithm. This choice was driven by the algorithm's impressive accuracy rate of 98% and its relatively efficient execution time of 144 seconds. In the context of machine learning models designed for medical systems, paramount importance is attributed to the accuracy and precision of the system due to critical applications. Table 3.2 provides a breakdown of the selected model's performance metrics with others.

Random Forest Classifier

The broad category of ensemble-based learning methodologies includes Random Forest Classifier. It leverages an ensemble of multiple decision trees to generate predictions or classifications. By combining the outputs of these trees, the random forest classifier delivers a consolidated and more accurate result. Higher accuracy is obtained and overfitting is avoided when the number of trees in the forest is large.

Classification Evaluation

The effectiveness of the RF for categorizing preterm and term deliveries was assessed using the ten-fold cross validation approach. The dataset was divided into 10 subsets at random, nine of which were used to train the RF and one for testing. Each of the ten subsets was utilized as test data once throughout the cross-validation approach, which was carried out ten times.

To assess how well the RF classification results performed, the accuracy, precision, recall, and F1-score from the ten-fold cross-validation were averaged. The optimum hyperparameters for the classification, such as the depth of the tree, the maximum depth of the tree, the minimum sample

split, and the criterion, were chosen using hyper-tuning techniques to get the best results. Along with calculating the accuracy of the classifier, the AUC-ROC was also computed.

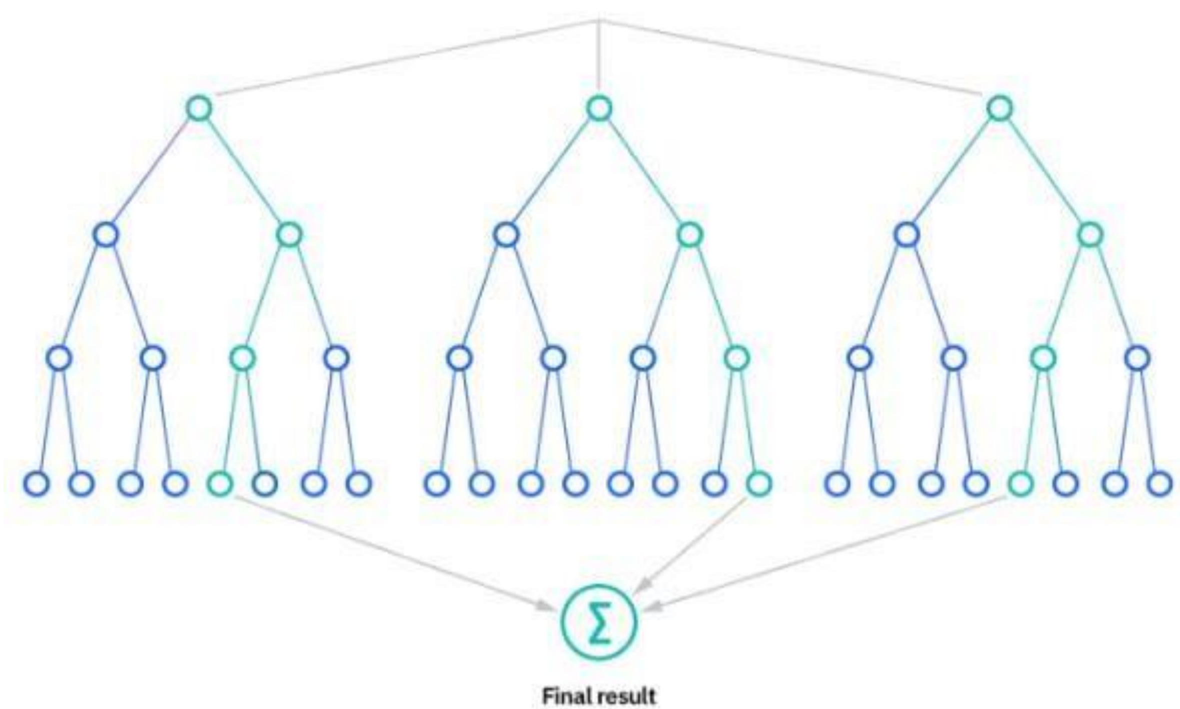

Figure 3. 8 Schematic diagram of Random Forest Classifier

	precision	recall	f1-score	support
0	1.00	0.96	0.98	55
1	0.96	1.00	0.98	49
accuracy			0.98	104
macro avg	0.98	0.98	0.98	104
weighted avg	0.98	0.98	0.98	104

Figure 3. 9 Performance Metrics

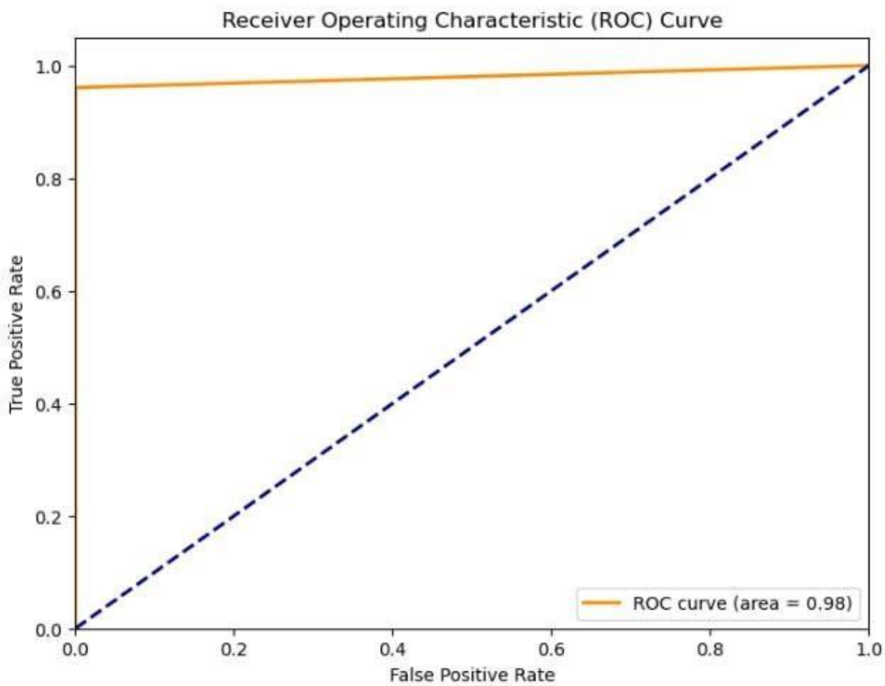

Figure 3. 10 ROC curve

3.4.3 Wireless Communication System

The wireless communication propagation employed in the development of this project is line-ofsight propagation (LOS). It is a fundamental concept in wireless communication systems, particularly in indoor environments. LOS propagation refers to the direct, unobstructed transmission path between a transmitter and a receiver. In indoor settings, devices must have a clear line of sight to establish efficient and reliable communication. Despite efforts to maintain LOS, indoor environments often involve reflective surfaces and materials that can cause signal reflections and absorption. These phenomena can lead to signal degradation, multipath interference, and loss. To mitigate LOS-related challenges in indoor propagation, we considered the strategic placement of devices and antennas by finding the transmission power of both devices as follows;

A. Microcontroller Power transmission

The relationship between the transmitted power ***Pt*** and the received power ***Pr*** is given by;

$$Pr = \frac{Pt \times Gt \times Gr \times \lambda^2}{(4\pi)^2 \times d^2 \times L} \quad (3.14)$$

Gt - transmitter antenna gain Gr

- receiver antenna gain

d - distance between transmitter and receiver

λ - wavelength of the signal

L - system loss

- Wireless transmitter module - Node MCU

Pt = 20 dBm = 0.1W

Gt = 2 dBi

- Wireless receiver module - Raspberry Pi Model 3B

Gr = 1 dBi

Sensitivity = -100 dBm Pr

= ?

Carrier frequency (Fc) = 2.4GHz, C = $3.0 \times 10^{8} m/s$

$\lambda = \frac{C}{fc}$, c is the speed of light

$\frac{3.0 \times 10^{8}}{45 \times 10^{9}}$ 2. $= = 0.122\ m$

Taking d = 30m, calculating Pr and comparing to the maximum Pr value

$$\text{Pr} = 0\frac{1 \times 2 \times 1 \times 0.122}{(4\pi)^2 \times 30^2 \times 1}.^2 = 209.5 \times 10^{-10}W = 209.5 \times 10^{-7}mW$$

$$\text{Pr}(dBm) = 10\log\left(209.5 \times 10^{-7}\right) = -46.78\ dBm$$

B. Free Space Path Loss

This model measures path loss as a function of T-R separation when the receiver and transmitter are under the LOS range in a free-space environment.

$$PL = -10\log\left(\frac{Pt}{Pr}\right) \quad (3.\ 15)$$

Pt is the transmitter antenna power

Pr is the receiver antenna power

Power Loss(PL) for 30m distance range:

$$PL = -10 \log \left(\frac{Pt}{Pr}\right) = -10 \log \left(\frac{0.1}{209.5 \times 10^{-10} W}\right) = -66.788 \text{ dB}$$

C. Message Queuing Telemetry Transport (MQTT)

MQTT is a lightweight and efficient messaging protocol for machine-to-machine IoT (Internet of Things) applications and remote monitoring systems. The MQTT protocol employs a publishsubscribe messaging pattern, allowing devices to publish data on specific topics and subscribe to receive data from those topics. This approach enables efficient and targeted data exchange between devices. It relies on a broker that manages the distribution of messages. The broker plays a central role in routing messages to their intended destinations. With our project, the central device serves as the broker, server, and subscriber.

On the other hand, the patient device publishes messages that are the recorded contraction parameters, to the same topic to which the central is subscribed. MQTT also supports Quality of Service (QoS) levels, allowing for different degrees of message reliability. The reason for using

the MQTT protocol was due to its high scalability, accommodating a growing number of devices and data streams.

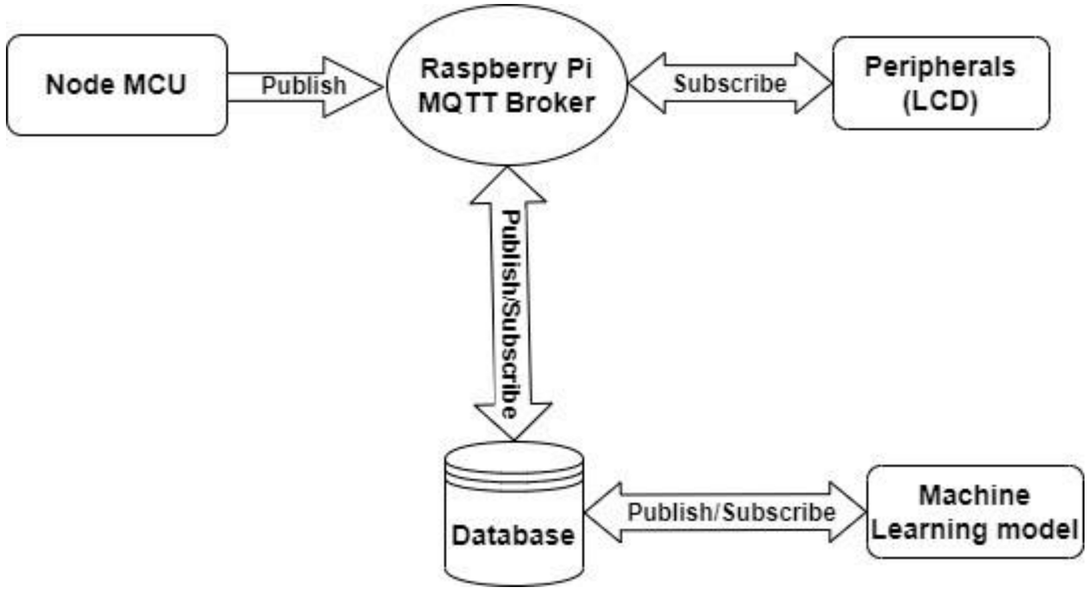

Figure 3. 11 Architectural model for MQTT protocol

3.4.4 Database Development

The database for our central device was deployed on the Raspberry Pi using a PostgreSQL database. PostgreSQL is an open-source, powerful, and highly extensible relational database management system. It is known for its reliability, performance, and support for complex data types. A script running on the Raspberry Pi was used to create a database if it does not exist or connect to it if it already exists. Structured Query Language (SQL) queries were embedded in the script to retrieve, filter, and analyze the contraction parameter readings from the patient-side device through the MQTT broker.

3.5 System modeling and simulation

A. Signal processing simulation

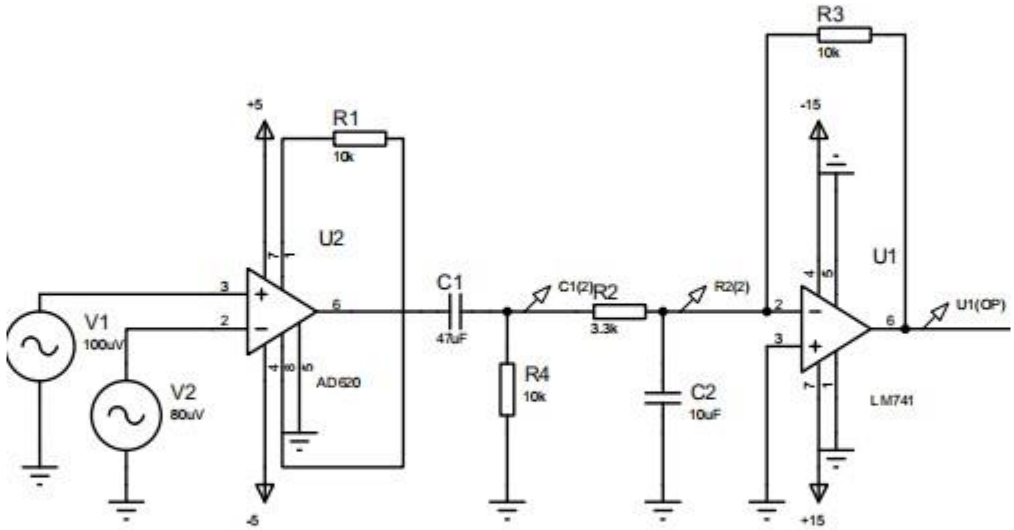

Figure 3. 12 Diagram of signal processing unit

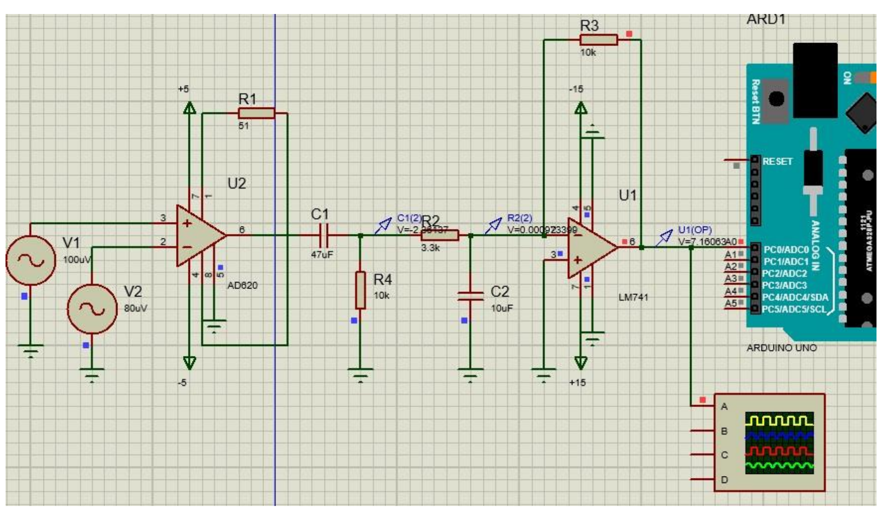

Figure 3. 13 Simulation of signal processing unit

B. Patient monitoring device simulation

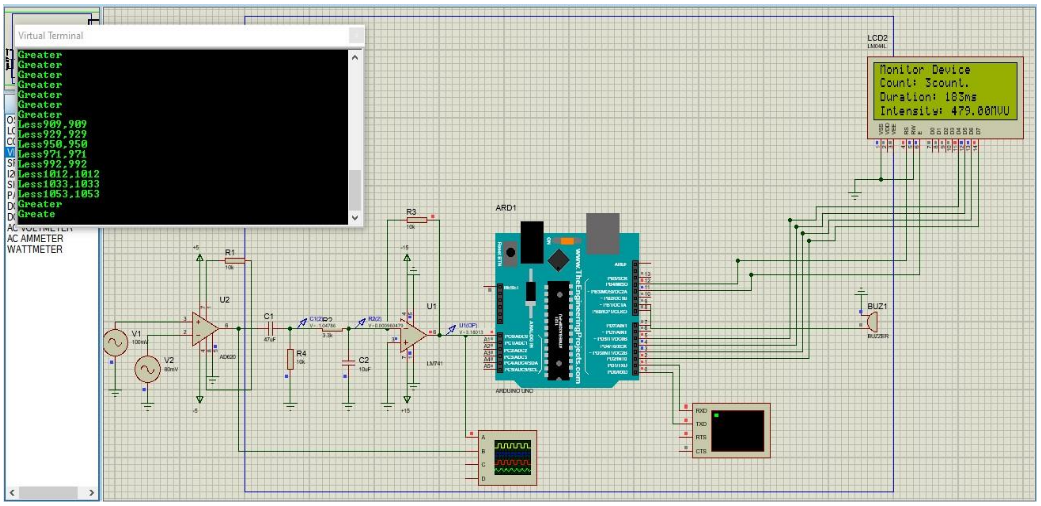

Figure 3. 14 Simulation of Patient monitoring device

C. Central device simulation

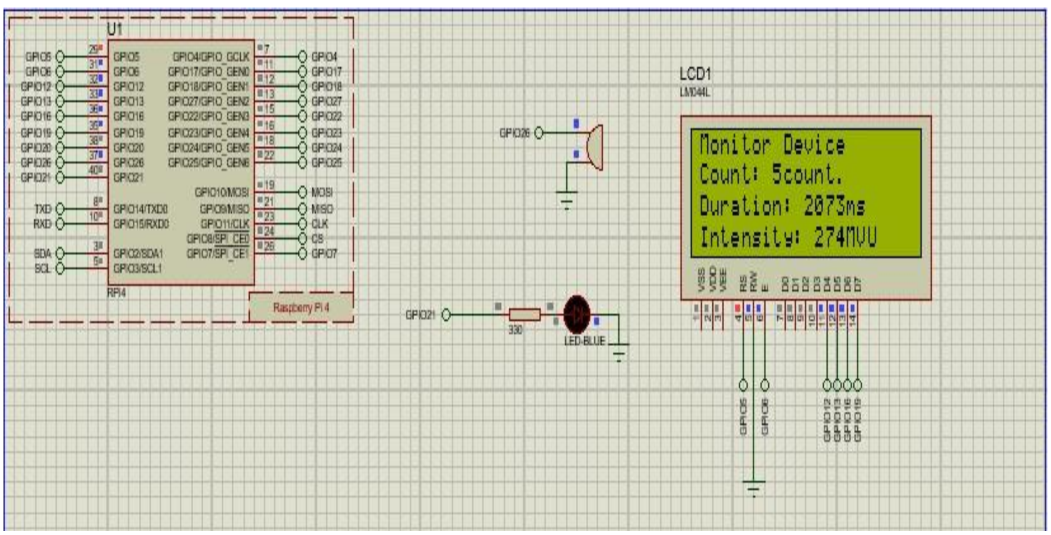

Figure 3. 15 Simulation of Central device

3.6 Development tools and material requirements

A. Hardware system components

1. NodeMCU (ESP8266) Module

The Node Microcontroller Unit (NodeMCU) is an open-source Internet of Things (IoT) platform based on the popular ESP8266 WiFi networking module manufactured by Espressif Systems. It combines the capabilities of a microcontroller with CPU and RAM, with built-in WiFi, making it an excellent choice for creating IoT projects and prototypes.

Figure 3. 16 Image of Node MCU

2. Raspberry Pi

The Raspberry Pi Foundation created the adaptable single-board computer known as the Raspberry Pi 3 Model B. It was introduced in February 2016 and is the third edition of the Raspberry Pi series. It has a 1.2 GHz 64-bit quad-core ARM Cortex-A53 CPU, inbuilt 802.11n Wi-Fi, Bluetooth, and USB boot capabilities. The Raspberry Pi is a popular option for IoT applications, as well as numerous computer and maker projects, thanks to its General-Purpose Input/Output (GPIO) ports. The board is small, reasonably priced, and, considering its size, has outstanding hardware capabilities.

Figure 3. 17 Image of a Raspberry Pi 3B

3. EMG Muscle Sensor Module

The EMG Muscle Sensor Module V3.0 is an electronic device designed for the measurement, filtering, amplification, and monitoring of electromyography (EMG) signals produced by muscle contractions. This compact and versatile module is commonly used in medical, research, and related projects to capture and analyze muscle activity. That is, it produces an analog output signal that can easily be read by a microcontroller, enabling muscle-controlled interfaces. It also provides a valuable tool for understanding muscle behavior and can be integrated into various applications.

Figure 3. 18 Image of an EMG sensor

4. Liquid-Crystal Display (I2C 20x4) Module

The I2C 20x4 LCD is a liquid crystal display (LCD) module equipped with an Inter-Integrated Circuit (I2C) interface, designed to provide a convenient and efficient way to display information in various electronic projects. It features a 20x4 character configuration, meaning it can display up to 20 characters per line and has four lines. This LCD module communicates with other devices, commonly in microcontroller-based projects, IoT devices, and embedded systems via the I2C protocol by simplifying the wiring and reducing the number of pins required for interfacing, to provide visual feedback and information display.

Figure 3. 19 Image of an I2C LCD

5. Active Buzzer Module

An active buzzer is an electronic audio signalling device that generates sound when an electrical current is applied to it. They can be controlled by varying the voltage or frequency of the input signal. This allows for different tones and volumes to be produced, making them versatile for various applications. Active buzzers are widely used in devices where an audible alert or sound notification is required. Unlike passive buzzers, active buzzers are often designed to be energy-efficient, making them suitable for battery-operated devices where power conservation is important and also have an internal oscillator circuit, so they can produce sound without any external signal source.

Figure 3. 20 Image of an Active buzzer

Other hardware components

- Breadboard
- 9V batteries
- Jumper wires
- EMG Surface Electrodes
- Micro Secure Digital (SD) card

B. Software system components

1. Arduino IDE
2. Proteus
3. Raspberry Pi Command Line
4. Jupiter Notebook
5. Visual Studio Code

CHAPTER 4 – DESIGN IMPLEMENTATION AND TESTING

4.0 Introduction

This chapter will discuss how our system was developed representing how our subsystems were implemented and tested for their various functionalities. The results are verified to ensure the objectives of the project have been met. Tests conducted on the system to evaluate its ability to meet functional and non-functional requirements and specifications mentioned in Chapter 3 are outlined. The results of these tests are also discussed in this section

4.1 Design implementation process

The implementation process for the various sections of the uterine contraction detection system is described in this section. The entire system was implemented in two separate parts. The following subsections explain the process.

A. Patient Device

The Patient Device was designed, simulated, and developed. The device consists of Node MCU which acts as the processor to which other sensor systems are connected. The Electromyography (EMG) sensor which is used to detect muscle activities (uterine contraction) is connected to the Node MCU. The EMG sensor has surface electrodes that take signals from the uterine contractions and send them to the Node MCU for processing which includes filtering and amplification. The Node MCU has a WIFI module that can be used to connect to other devices wirelessly. After the processing of the data from the surface electrodes, the results of the parameters from the features are displayed on the LCD. The Patient Device contains a buzzer that only gives an alert when a patient has irregular contractions as shown in Figure 4.1.

The code for the development of the Patient Device was written with the aid of Arduino IDE which has a serial monitor for data visualization. Figure 4.2 shows the output signal visualization from the Patient Device in the serial monitor

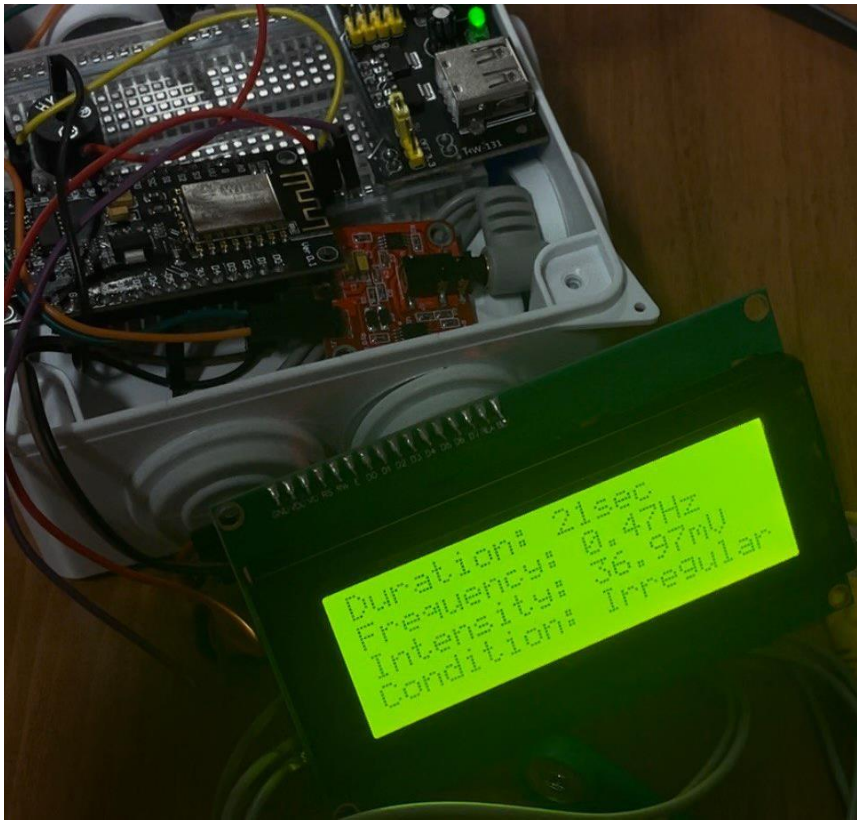

Figure 4. 1 Image of an I2C LCD displaying readings

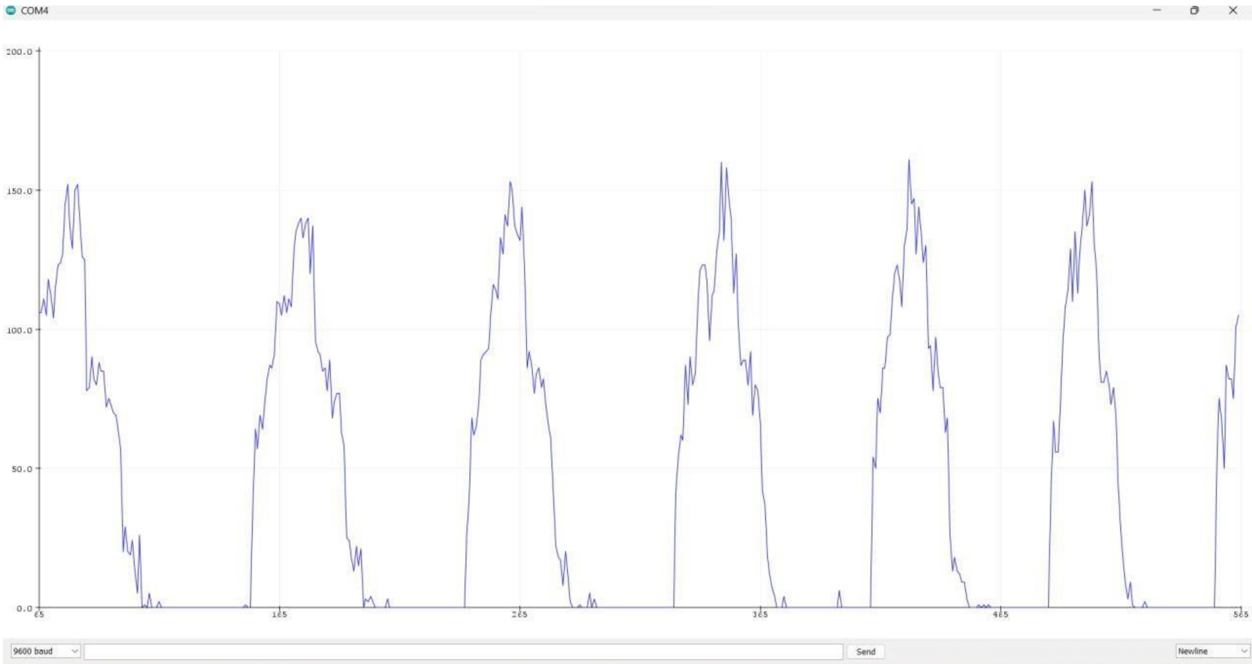

Figure 4. 2 Signal waveform from the EMG sensor

B. The Central Device

The Central Device consists of an LCD, a Raspberry Pi, and a power supply. The Patient Device connects wirelessly to the Central Device. Processed signals from the Patient Device are sent to the Central Device through an MQTT protocol. The Central Device has a machine learning model that is trained to detect whether a person is at risk of having a preterm birth or not and thus alert the health personnel by displaying it on the LCD. The Central Device also has a database built into it which stores the data from the Patient Device. It is from the database that the Central Device takes the data from the Patient Device for the machine learning model prediction. The system was built to meet the requirements and specifications in Chapter 3.

Figure 4. 3 Image of readings on the central device

4.2 Testing of design and results

To fully test the functionality of the system, we performed two variations of tests;

I. Test 1 – The source of data for the machine learning model prediction was the signal acquisition sensor (EMG surface electrodes)

The system was designed and developed to measure uterine contractions in pregnant women. To test the system on pregnant women requires ethical clearance which takes a long time to process and hence was not secured at the time of testing. Since there weren't any test subjects, we tested the system on ourselves as volunteers. Signals from contractions on a different part of the body were considered. The EMG sensor can measure muscle activities of the human body. The system was tested on the heart to check for cardiac activities, the muscle of the lower arm, and the belly of each of the volunteers.

For our first volunteer, the surface electrodes were placed on the chest to acquire signals. The surface electrodes used consisted of only three anodes which are of different colours. The yellowcoloured electrode was placed on the heart and the red-coloured electrode on the right chest. The green-coloured electrode serves as a reference electrode which was placed about 6 cm directly below the red-coloured electrode as shown in Figure 4.4.

The signals were pre-processed by the patient device and the contraction parameters were calculated and displayed on the LCD including the pregnancy condition whether irregular or regular. The processed signals from the patient device were sent to the central device from which features were extracted and machine learning model analysis were carried out. The central device consists of Raspberry Pi which has the machine learning model.

The whole process was repeated to measure contractions of the muscles of the lower arm and the belly for the first volunteer. The electrodes were positioned as seen in Figure 4.5 for the muscles of the lower arm and the belly.

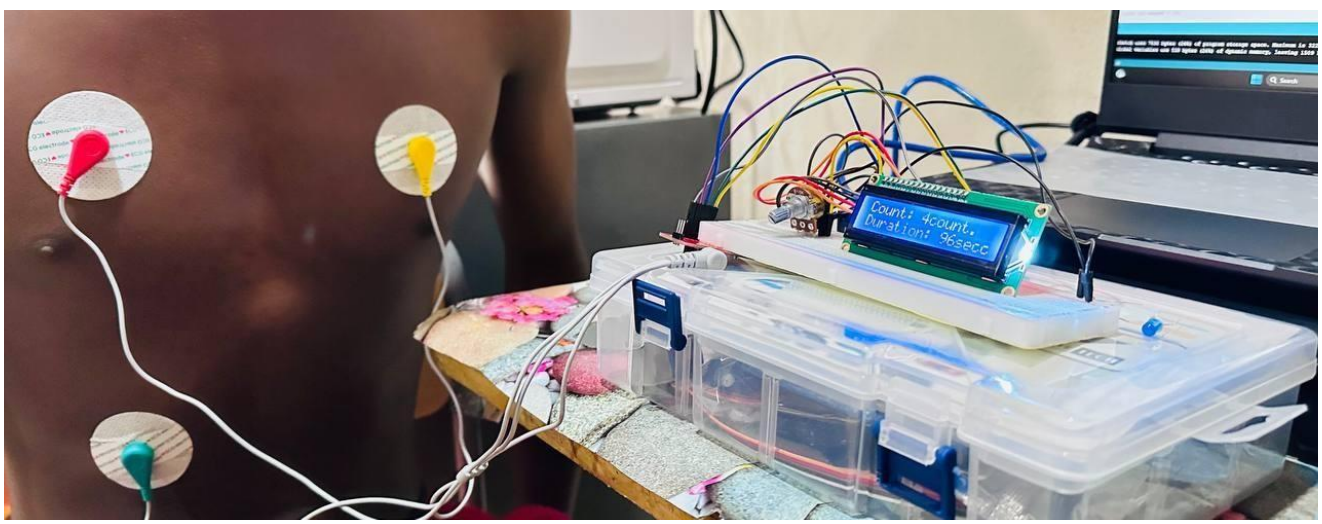

Figure 4. 4 Image of signal acquisition from the chest

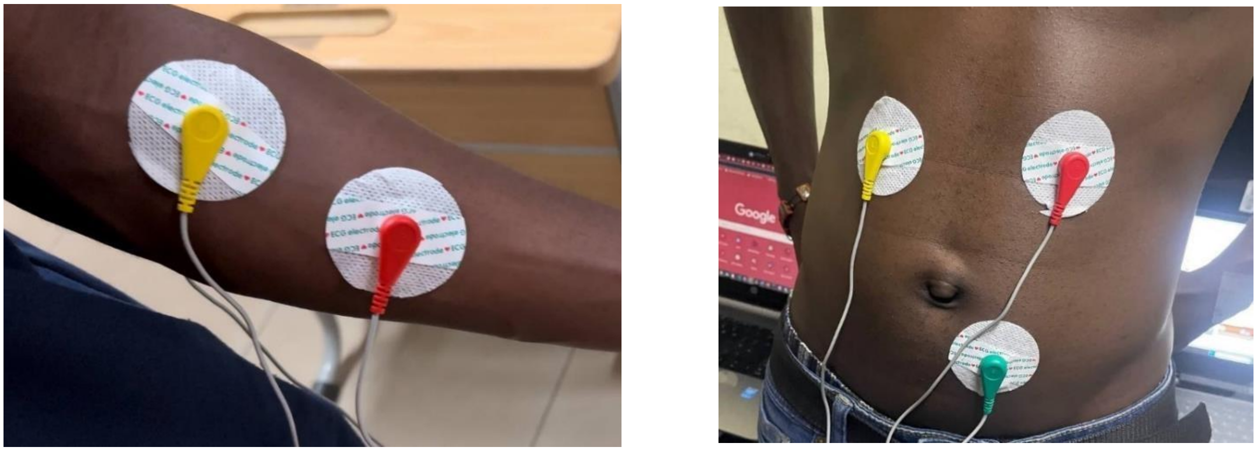

Figure 4. 5 Image of signal acquisition from the arm and the stomach

For the second volunteer, the same setup from the first volunteer was used and signals were acquired for the machine learning model analysis. The table below shows the results of the tests done on the two volunteers.

Table 4. 1 Test results from the two volunteers

	Volunteer 1			Volunteer 2		
Parameters measured	Chest	Arm	Stomach	Chest	Arm	Stomach
Duration (secs)	21.00	24.00	26.00	23.00	20.00	25.00
Intensity (mV)	36.97	50.69	28.84	10.64	34.28	28.86

Frequency (Hz)	0.47	0.42	0.39	0.43	0.49	0.39
Condition	Irregular	Regular	Irregular	Regular	Irregular	Irregular
ML Prediction	Preterm	Term	Preterm	Preterm	Preterm	Preterm

Figure 4. 6 Test results from the arm of volunteer 1

Figure 4. 7 Test results from the chest of volunteer 1

Figure 4. 8 Test results from the stomach of volunteer 1

II. Test 2 – The source of data for the machine learning model prediction is a dataset from an open repository (Physionet)

The dataset from Physionet (open repository) was pre-processed, trained, and validated to accurately predict the risk of a patient having preterm delivery. After the training of the machine learning model, we achieved high accuracy for the prediction. We randomly selected five (5) data which consisted of two (2) term records and three (3) preterm records from the database and tested them as if they were real-time signals directly from the EMG sensor. The Central device which has the machine learning model was able to accurately predict the delivery outcome for most of the data randomly picked.

Table 4. 2 Results from existing data on five patients

Patients	P1	P2	P3	P4	P5
Duration (secs)	20.00	17.00	25.00	25.00	23.00
Intensity (mV)	34.24	18.12	170.07	70.89	10.64

Frequency (Hz)	0.49	0.59	0.40	0.40	0.43
Condition	Irregular	Irregular	Irregular	Regular	Irregular
ML Prediction	Preterm	Term	Term	Preterm	Term
Prediction Probability	0.87	0.91	0.93	0.69	0.76
Accuracy	1.00	1.00	1.00	1.00	0

	Input Data	True Labels	Predictions	Accuracy
0	[5.8511, 0.4944, 0.3313, 0.643]	1	1	1.0
1	[4.2565, 0.5865, 1.1936, 0.768]	0	0	1.0
2	[13.0412, 0.4006, 0.3478, 0.391]	0	0	1.0
3	[8.4198, 0.3981, 0.2938, 0.564]	1	1	1.0
4	[3.2623, 0.4333, 0.3625, 0.396]	1	0	0.0

Figure 4. 9 Result of the existing data showing the predictions against the true labels

4.3 Discussion of results and analysis

From the test experiment using the real data from volunteers, the results in Table 4.1 shows the tests that were carried out for the first case of testing (test 1). The volunteer test subject who was suspected to have the risk of preterm delivery was predicted by the system to have preterm delivery from the test done on the chest and stomach. This is because the system was designed to measure uterine contractions in pregnant women. The tests were done by setting the electrodes of the EMG sensor on the belly and on the bicep of the lower arm. For regular uterine contractions, there is a range of values. Any value outside the range is said to be having irregular contractions. Since a person at risk of having preterm birth also has a higher probability of having irregular contractions, the probability that the system will predict for almost all of the volunteer subjects who are not pregnant women and hence not having uterine contractions is very high thus the system predicting preterm delivery for our volunteer test subject. However, the system was able to predict term delivery when signals were picked by the electrodes on the arm.

From the test experiment using the pre-existing data from the physionet database in Table 4.2, five samples were selected randomly and uploaded into the system which included two terms and three preterm conditions. The model was able to make four correct predictions however the model predicted wrongly for patient 5 with the probability prediction accuracy of 76%. This was highly to be expected as the data were from the dataset that the model was familiar with. The prediction probability in the table represents the random forest prediction probability from the ensemble of decisions in the model whilst the accuracy represents the probability of the predicted outcome to the true label from the dataset as seen in Figure 4.9.

Overall, the system test results show the system meets most requirements including the most important one, which is the correct prediction of labour conditions for preterm delivery.

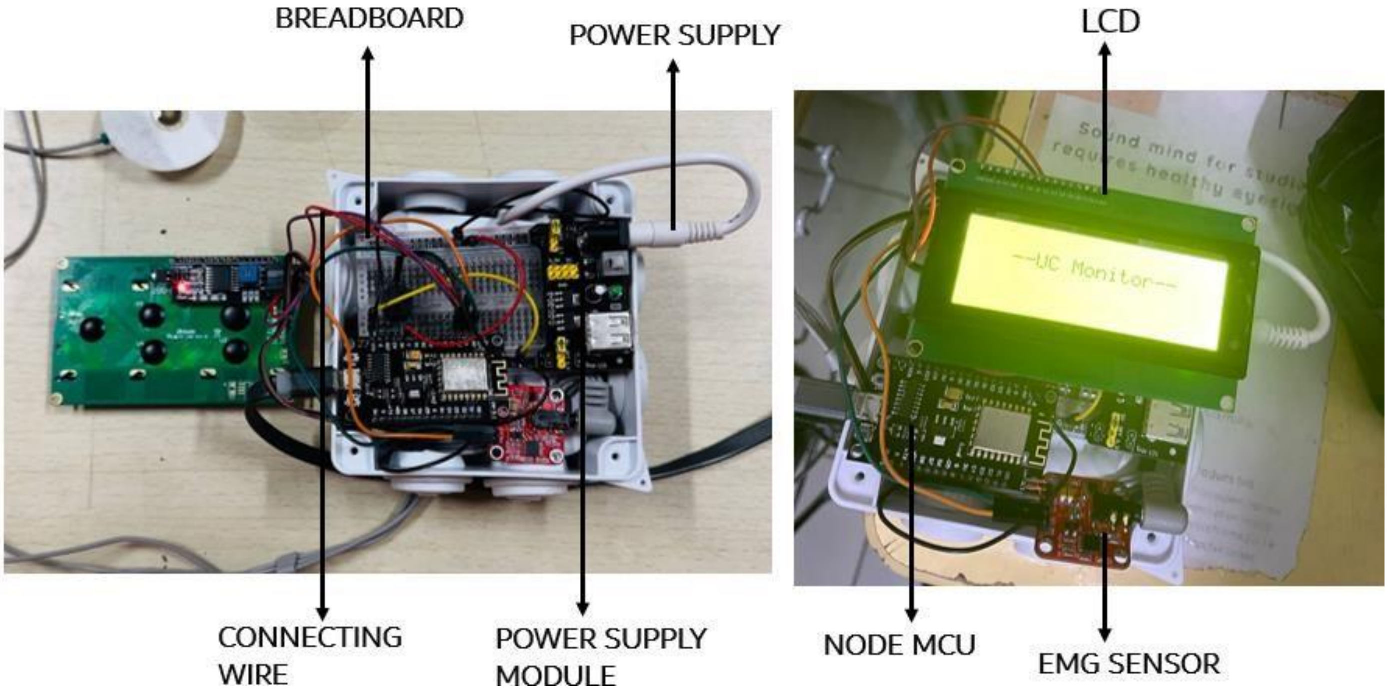

Figure 4. 10 Labelled part of the patient device

4.4 Comparative analysis and evaluation

To a large extent, the system performed as expected. The central device successfully received and classified the processed signals from the patient device using the detection model. The accuracy, precision, F1-Score, recall, and AUC-ROC of the model were the criteria used to assess its performance. On test data, the model's performance is compared to that of other models from the literature in the table below.

Table 4. 3 Comparative analysis of existing solutions

Model author(s)	**Algorithm**	**Accuracy**	**Precision**	**F1- Score**	**Recall**	**AUC-ROC**

Project Team	Random Forest Classifier	98%	98%	98%	98%	98%
Lili Chen et al [14]	Deep Neural Network	94.30%	N/A	N/A	N/A	N/A
X. Song et al [16]	ZeroCrossing Rate	97.30%	N/A	N/A	N/A	N/A
Manoara Begum et al [17]	Decision Tree	99%	99%	N/A	99%	96%
Jin Peng et al [25]	Random Forest Classifier (Early)	93%	N/A	N/A	N/A	N/A
	Late	92%	N/A	N/A	N/A	N/A

Comparing the models in the above table in terms of accuracy, precision, and recall, Manoara Begum et al had the best result but failed to provide metrics on F1- Score. Our mode was benchmarked against that of Jin Peng et al since they used the same type of algorithm. Our model outperforms theirs in terms of accuracy.

4.5 Limitations and Constraints

The system was designed to measure uterine contractions and predict the outcome from the results of the contraction parameters. However, since there were no volunteers (pregnant women) to test for the accuracy of the system using the EMG sensor, the device was tested on the biceps of the lower arm. This is a limitation of the system since we could not tell whether the system would work very well in measuring uterine contractions and making predictions on that.

The dataset for the machine learning model classification was of limited number even after balancing. For an efficient model, a large data is needed to train the model. We trained and tested our machine-learning model using the balanced dataset. This is to prevent incorrect prediction in real-life testing on pregnant women should we have generated a larger dataset which is bad practice in the field of health.

The system also a uses wireless transmission medium to send signals (data) from the patient device to the central device. One major problem in wireless propagation is attenuation and obstacles. We assumed a distance of about 30 m and were able to successfully receive data. Our system was tested with very few obstacles which can affect the network performance when used in clinical settings where there are a lot of people and other obstacles.

CHAPTER 5 - CONCLUSION AND RECOMMENDATION

5.0 Introduction

This chapter covers the project that designed and developed a device for detecting and tracking uterine contractions for preterm delivery using machine learning. It presents the main findings, conclusions, contributions, observations, challenges, and project recommendations. It also discusses the potential implications and applications of the device for improving the quality of care and outcomes for pregnant women at risk of preterm delivery. It also proposes future research directions and improvements for the device.

5.1 Major findings for the project

1. The device could accurately and timely detect and track labour conditions for preterm delivery.
2. The machine learning algorithms could learn the patterns of uterine contractions and predict preterm delivery with high accuracy.
3. The device is a promising new tool for improving maternal care and reducing the incidence of preterm birth.

The device accurately detects and tracks labour conditions for preterm delivery, using machine learning algorithms to predict with high accuracy. It is easy to use and comfortable, making it a promising tool for improving maternal care and reducing preterm birth incidence. Early warning of preterm delivery can prevent premature births and associated health risks, while also educating pregnant women and healthcare providers about preterm birth. The device is still under development, but the findings of this project are encouraging. The device can make a real difference in pregnant women's and their babies' lives.

5.2 Conclusion

This project aimed to address the challenges of pre-term delivery (PTD) challenges and monitor labour conditions using machine learning techniques. The device, developed for this purpose, has successfully achieved its objectives, enabling early intervention and reducing the risk of severe health consequences for pregnant women and newborns. Its non-invasive and continuous monitoring capabilities capture nuanced variations in uterine contractions, providing healthcare professionals with timely insights into labour progression.

The second objective, establishing a robust risk assessment model based on machine learning algorithms for PTD prediction, has yielded remarkable results. Integrating machine learning into the device's core functionality has enabled accurate and timely predictions of PTD, empowering healthcare professionals with the data needed to make informed decisions regarding labour management. This represents a groundbreaking advancement in obstetric care, enhancing the precision of pre-term delivery prediction and improving the safety and well-being of expectant mothers and their newborns.

In conclusion, the innovative concept proposed in this project has been successfully implemented and exceeded expectations in its capacity to enhance maternal care. The device's ability to provide effective UC monitoring and machine learning-driven PTD prediction capabilities marks a significant leap forward in obstetric care, reducing maternal and neonatal healthcare risks and providing valuable tools for improved decision-making.

5.3 Contribution to knowledge and society

The project introduces an innovative device that improves labour condition monitoring in obstetric care. The device uses surface electrodes for real-time measurement of uterine contraction parameters and machine learning algorithms to provide healthcare professionals with unprecedented insights into labour progression. This technology reduces the risk of severe health consequences for mothers and newborns. It validates the device's potential for improving obstetric care and lays the groundwork for further labour monitoring and prediction studies.

The societal impact of this project is balanced, as it provides expectant mothers and healthcare providers with a tool for continuous, non-invasive monitoring and early detection of preterm delivery risks. This reduces the stress of periodic in-person check-ups for pregnant women and healthcare professionals, promoting better mental and emotional well-being. The broader economic and social benefits of improved maternal health and reduced preterm deliveries extend beyond individual healthcare. The project's development underscores its significant contribution to obstetric care and societal needs.

5.4 Observations and Challenges

The labour monitoring device was developed to be user-friendly, portable, and comfortable for pregnant women, ensuring non-invasiveness and minimal discomfort. However, it faced challenges in data quality, transmission, and security, including noise, artifacts, and interference in uterine contraction signals. The device also had to ensure a stable wireless connection between the patient device and the central system, which could be affected by environmental factors or malicious attacks.

Additionally, the device had to protect user data privacy and confidentiality, which could be vulnerable to unauthorized access or misuse. The accuracy and reliability of machine learning algorithms for timely preterm delivery prediction required rigorous validation and testing, which encountered technical obstacles.

Despite these challenges, the project persevered, providing valuable insights for future refinements and realizing the device's full potential in transforming obstetric care.

5.5 Recommendations

As we advance, it is imperative to consider several key recommendations to enhance further the effectiveness and impact of the labour monitoring device:

1. Mobile applications can further be developed for more accessibility of the patient's records or results anytime needed.
2. The central controller system should be modified to accommodate many patient monitor devices.
3. The device should be further evaluated in clinical trials.
4. The device should be integrated with other healthcare systems to facilitate data exchange.
5. The device should be used to educate pregnant women and healthcare providers about preterm birth.

These recommendations are based on the project's findings and the device's potential to improve maternal care and reduce the incidence of preterm birth. By following these recommendations, the device can be made more effective and accessible, and its potential to benefit pregnant women and their babies can be maximized.

References

[1] T. Callahan and A. Caughey, *Blueprints obstetrics and gynecology*, 15th ed. Edinburgh Churchill Livingstone, 2013, pp. 435–451.

[2] D. M. Fraser and M. A. Cooper, *Myles Textbook for Midwives*, 15th ed. London: Churchill Livingstone, 2009.

[3] World Health Organization. Preterm birth. https://www.who.int/news-room/factsheets/detail/preterm-birth. Accessed 3 Sep 2023.

[4] Liu L, Oza S, Hogan D, et al. Global, regional, and national causes of child mortality in 2000–13, with projections to inform post-2015 priorities: an updated systematic analysis. Lancet. 2015;385(9966):430-440. doi:10.1016/S0140-6736(14)61698-6

[5] Romero R, Dey SK, Fisher SJ. Preterm labor: one syndrome.

[6] Saccone G, Berghella V. Transvaginal ultrasound cervical length for prediction of spontaneous labor at term: a systematic review and meta-analysis. Am J Obstet Gynecol. 2015;212(1):18-31. doi:10.1016/j.ajog.2014.06.054

[7] Deo RC. Machine learning in medicine. Circulation. 2015;132(20):1920-1930. doi:10.1161/CIRCULATIONAHA.115.001593

[8] Topol EJ. High-performance medicine: the convergence of human and artificial intelligence. Nat Med. 2019;25(1):44-56. doi:10.1038/s41591-018-0300-7

[9] Degbedzui, Derek Kweku & Mills, Godfrey & Effah Kaufmann, Elsie. (2016). Development of uterine contraction monitoring system for resource-limited settings. Biomedical Engineering: Applications, Basis and Communications. 28. 1650045. 10.4015/S1016237216500459

[10] Agyemang, Ernest & Frimpong, M. Warren, Mills,A. Mills & Effah Kaufmann.(2021). Classification and Detection System. Computer Engineering.

[11] Chawanpaiboon, S., Vogel, J.P., Moller, A.B., Lumbiganon, P., Petzold, M., Hogan, D., Landoulsi, S., Jampathong, N., Kongwattanakul, K., Laopaiboon, M. and Lewis, C., 2019. Global, regional, and national estimates of levels of preterm birth in 2014: a systematic review and modeling analysis. *The Lancet global health*, *7*(1), pp.e37-e46.

[12] Jacod, B. C., Graatsma, E. M., Van Hagen, E., & Visser, G. H. (2010). A validation of electrohysterography for uterine activity monitoring during labour. *The Journal of MaternalFetal & Neonatal Medicine*, *23*(1), 17-22.

[13] Urquhart Christine, Rosemary Currell, Francoise Harlow, and Liz Callow. "Home uterine monitoring for detecting preterm labour." *Cochrane Database of Systematic Reviews* 2 (2017).

[14] Vermeulen-Giovagnoli B, Peters C, Mischi M, van Pul C, Cottaar EJ, Oei SG. The development of an obstetric tele-monitoring system. In2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC) 2015 Aug 25 (pp. 177-180). IEEE.

[15] H. Allahem and S. Sampalli, "Framework to monitor pregnant women with a high risk of premature labour using sensor networks," 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, 2017, pp. 1178-1181, doi:10.23919/INM.2017.7987458.

[16] Lili Chen, Huoyao Xu, Deep neural network for semi-automatic classification of term and preterm uterine recordings, Artificial Intelligence in Medicine, Volume 105, 2020, 101861, ISSN

[17] Song X, Qiao X, Hao D, Yang L, Zhou X, Xu Y, Zheng D. Automatic recognition of uterine contractions with electrohysterogram signals based on the zero-crossing rate. Scientific Reports. 2021 Jan 21;11(1):1956.0933-3657, https://doi.org/10.1016/j.artmed.2020.101861.
(https://www.sciencedirect.com/science/article/pii/S0933365719302568)

[18] Begum, M., Redoy, R.M. and Anty, A.D., 2021, February. Preterm Baby Birth Prediction using Machine Learning Techniques. In *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)* (pp. 50-54). IEEE.

[19] D. Malinouskas and G. Hojaiban, “Wireless optical patient monitoring apparatus,” *US Pat. 5,865,733*, pp. 1–18, 1999.

[20] D. M. Disabito, G. R. Allen, R. Hubbard, James, J. C. Naphy, and G. A. Thomas, “Intrauterine Pressure Catheter System,” US Patent 52793081994.

[21] Analog Devices Inc., “Low Cost Low Power Instrumentation Amplifier,” 2011. [Online]. Available: www.analog.com. [Accessed: 21-Jan-2015].

[22] N. S. C. (2000), "LM741 Operational Amplifier Operational Amplifier," 2000. [Online]. Available: www.national.com. [Accessed: 21-Jan-2015].

[23] G. Fele-Žorž, G. Kavšek, Ž. Novak-Antolič, "A comparison of various linear and nonlinear signal processing techniques to separate uterine EMG records of term and pre-term delivery groups," Med Biol Eng Comput, 46 (9) (2008), pp. 911-922

[24] F. Jager, S. Libensek, K. Gersak, *et al.* "Characterization and automatic classification of preterm and term uterine records", PLoS One, 13 (8) (2018), Article e0202125

[25] Jin Peng, Dongmei Hao, Lin Yang, Mengqing Du, Xiaoxiao Song, Hongqing Jiang, Yunhan Zhang, Dingchang Zheng, "Evaluation of electrohysterogram measured from different gestational weeks for recognizing preterm delivery: a preliminary study using random Forest", Biocybernetics and Biomedical Engineering, Volume 40, Issue 1,2020, Pages 352-362, ISSN 0208-5216, https://doi.org/10.1016/j.bbe.2019.12.003.

Appendices

Appendix A -- Code for the patient device

```
#include <Arduino.h> const char* MQTT_PUB_DBS_READINGS
                                      = "devices/device_id/readings";
#include <arduinoFFT.h>
                                                     WiFiClient espClient;
#include <LCD_I2C.h>
                                      PubSubClient client(espClient);
#include <Filters.h>
                                      // Set the LCD address to 0x27 for a 16 chars
#include <ESP8266WiFi.h> and 2 line display

#include <PubSubClient.h> LCD_I2C lcd(0x27, 20, 4);

const char* ssid = "RaspberryPi";      // const int SAMPLING_FREQUENCY = 20; wifi
ssid
                                                    const int SAMPLES = 4;
```

const char* password = "Raspberrypi@3B";
// wifi password const float V_REF = 5.0;

const char* mqttServer = "10.42.0.1"; // IP
adress Raspberry Pi const int EHG_PIN = A0;

const int mqttPort = 1883; const float

const char* mqttUser = "utcon"; // if you
don't have MQTT Username, no need input

const char* mqttPassword = "UtContraction";
// if you don't have MQTT Password, no need
const int MEASUREMENT_PERIOD_MS =
in milliseconds

const char* MQTT_PUB_LCD_READINGS
"lcd/readings"; 0.3; // Lower cutoff frequency

// Reference
voltage of the Arduino's ADC (usually 5V)

CONTRACTION_THRESHOLD =
50.0; // Adjust this threshold to detect
contractions

const int MIN_CONTRACTION_DURATION
= 60; // Minimum duration (in seconds) to be
considered a valid contraction input
// Readings MQTT Topics 60000; // 1 minutes

const float BANDPASS_LOWER_FREQ = =
of the bandpass
filter in Hz

if (client.connect("ESP8266Client", mqttUser, mqttPassword)) {

const float BANDPASS_UPPER_FREQ = 5.0; // Upper cutoff frequency of the bandpass filter in Hz double vReal[SAMPLES]; // Real part of the FFT result double vImag[SAMPLES]; // Imaginary part of the FFT result

double featureValues[5]; // Array to store the calculated feature values double duration, medianFrequency, rmsValue, peakFrequency, sampleEntropy;

arduinoFFT FFT; // Create an instance of the ArduinoFFT library

unsigned long startTime; // Variable to store the start time of the 10-minute measurement period const size_t numMessages = 5; // Number of features to be sent via MQTT unsigned long contractionStartTime = 0; unsigned long contractionEndTime = 0; bool isContraction = false; bool initialDisplay = true; // Flag for initial display

void setup() {

Serial.begin(115200);

lcd.begin(); // Initialize the LCD

lcd.backlight(); lcd.setCursor(0, 1); lcd.print(" --UC Monitor-- "); delay(10000);

pinMode(D5, OUTPUT);

// WiFi connection to the Raspberry Pi

WiFi.begin(ssid, password); while (WiFi.status() != WL_CONNECTED) {

delay(500);

Serial.println("Connecting to WiFi..");}

Serial.println("Connected to the WiFi network");

client.setServer(mqttServer, mqttPort);

while (!client.connected()) {

Serial.println("Connecting to MQTT...");
Serial.println("Connected to MQTT broker");

} else {

Serial.print("MQTT connection failed, state: ");

Serial.print(client.state());

delay(2000);}}} void

reconnect() { while

(!client.connected()) {

Serial.println("Attempting MQTT connection..."); if

(client.connect("ESP8266Client", mqttUser, mqttPassword)) {

Serial.println("Connected to MQTT broker");

Serial.println(" Retrying in 5 seconds...");
while (WiFi.status() != WL_CONNECTED)
{

delay(500);

Serial.print("Reconnecting to the wifi...");

}

Serial.println("Connected to the WiFi Network");} void loop() {

if (!WiFi.isConnected()) {

connectToWiFi(); } if

(!client.connected()) {

reconnect();}

// Check if the 1-minute measurement period has elapsed if (millis() - startTime >= MEASUREMENT_PERIOD_MS) {

delay(5000);}}} for (int i = 0; i < SAMPLES; i++) { void connectToWiFi() {

analogValue = analogRead(EHG_PIN);

WiFi.begin(ssid, password);

} else {

Serial.print("MQTT connection failed, state: ");

Serial.print(client.state());

// Read analog EHG signal

int analogValue; double

voltage;

voltage = (analogValue / 1023.0) * V_REF; // Convert analog value to voltage (05V range)

// Apply bandpass filter

if (i > 1) {

vReal[i] = (2 * vReal[i - 1] - vReal[i - 2] + voltage - 2 * vReal[i - 1] + vReal[i - 2]) / 32.0;

} else { vReal[i] =

voltage; }

```
vImag[i] = 0; // Set imaginary part to 0
```

```
(since we are dealing with real-valued signal)      //Display initila values of 0.00 if
peakFrequency = calculatePeakFrequency();           (initialDisplay) {   // Initialize the LCD
rmsValue = calculateRMS();                          lcd.begin();   lcd.backlight();
sampleEntropy = calculateSampleEntropy();           digitalWrite(D5, HIGH);
    delayMicroseconds(1000); // Sampling delay}

  // Perform FFT

  FFT.Windowing(vReal, SAMPLES,
FFT_WIN_TYP_HAMMING, FFT_FORWARD);

  FFT.Compute(vReal, vImag, SAMPLES,
FFT_FORWARD);

  FFT.ComplexToMagnitude(vReal,       vImag,
SAMPLES);

    //   Calculate   feature   values
medianFrequency                  =
calculateMedianFrequency();
lcd.clear();   lcd.setCursor(0,   0);
lcd.print("Duration:    0    sec");
lcd.setCursor(0,                 1);
lcd.print("Frequency:   0.00   Hz");
lcd.setCursor(0,                 2);
lcd.print("Intensity:   0.00   mV");
lcd.setCursor(0,                 3);
lcd.print("Condition: N / A");


    delay(50000);
```

delay(500); }} }   // Check for irregular contractions

initialDisplay = false; // Update the flag after initial display

// Detect contractions   rmsValue =

calculateRMS();

if (rmsValue > CONTRACTION_THRESHOLD) {

if (!isContraction) {   // Start of a new

contraction   contractionStartTime =

millis();   isContraction = true;   if

(rmsValue < 40.99 || rmsValue > 80) {

// Irregular contraction (below threshold and above threshold)

lcd.setCursor(0, 3);

lcd.print("Condition: Irregular");   for (int

i = 0; i < 30; i++) { // Beep the buzzer 30

times for irregular   digitalWrite(D5,

HIGH);   delay(500);

digitalWrite(D5, LOW);

} else {   if (!client.connected()) {

reconnect(); }   if (client.connected()) {

if (isContraction) {   //

End of the contraction

contractionEndTime =

millis();   isContraction

= false;

// Calculate the duration of the contraction

unsigned long contractionDuration = contractionEndTime - contractionStartTime;

// Ignore contractions with duration shorter than the minimum duration threshold

if (contractionDuration >= (MIN_CONTRACTION_DURATIO N * 1000)) {

duration = contractionDuration / 1000.0; // Store the contraction duration in seconds

} else {   duration = 0.0; // If the contraction duration is below the threshold, set it to 0 } }   if (rmsValue <= 50.00 || rmsValue > 80.00) {   lcd.setCursor(0, 3);

lcd.print("Condition: Irregular");   for (int i = 0; i < 60; i++) { // Beep the buzzer 5 times

digitalWrite(D5, HIGH);   delay(1000);

digitalWrite(D5, LOW);   delay(1000); }

} else if (rmsValue > 50.00 && rmsValue <= 80.00) { // Check for regular contraction

lcd.setCursor(0, 3);   lcd.print("Condition: Regular");   digitalWrite(D5, HIGH);

}

delay(60000); // Adjust 1 minute as needed } }

// Convert float values to strings and then to byte arrays

String readingsMessage = String(duration)+ "," + String(rmsValue) + "," + String(medianFrequency)+ "," + String(peakFrequency) + ","
String message = String(duration) + "," + String(medianFrequency)+ "," + String(rmsValue) ;

// Publish all readings to the database topic   if

+String(sampleEntropy) ;

(client.connected()) {

if (client.publish(MQTT_PUB_DBS_READINGS, (const uint8_t*)readingsMessage.c_str(), readingsMessage.length(), true)) {

} else {

// try

reconnecting

reconnect();

}

// Publish selected readings to LCD topics

if (client.publish(MQTT_PUB_LCD_READINGS, (const uint8_t*)message.c_str(), message.length(), true)) {

}

// Reset the start time to the

current time   startTime =

millis();   // Wait for 1 minute

delay(60000);   displayValues();

// Wait for a moment before the next

reading delay(1000); }}}}

double calculateMedianFrequency() {

// Calculate the median frequency by finding the frequency that separates

// the lower half from the upper half of the

power spectrum. double medianFrequencyValue

= 0.0; double cumulativeSum = 0.0;

// Compute the cumulative sum of the magnitudes

for (int i = 0; i < SAMPLES; i++) {

cumulativeSum += vReal[i]; } double halfPower

= cumulativeSum * 0.5; // Half of the total power

// Find the frequency index that corresponds to the

cumulativeSum = 0.0;

for (int i = 0; i < SAMPLES; i++) {

cumulativeSum += vReal[i]; SAMPLES);

if (cumulativeSum >= halfPower) { return rmsValue; }

half power

medianFrequencyValue = (i *
SAMPLING_FREQUENCY) / SAMPLES;

break; }} return

medianFrequencyValue; }

double

calculatePeakFrequency() {

//Function to calculate peak

frequency double

peakFrequencyValue = 0.0;

double maxAmplitude = 0.0;

for (int i = 0; i < SAMPLES;

i++) { if (vReal[i] >

maxAmplitude) {

maxAmplitude = vReal[i];

for (int i = 0; i < SAMPLES; i++) {
sumSquared += vReal[i] * vReal[i]; }

double rmsValue = sqrt(sumSquared /

peakFrequencyValue = (i *
SAMPLING_FREQUENCY) /
SAMPLES; }
} return

peakFrequencyValue; } double

```
calculateRMS() {   //Function to calculate
Rms Value   double sumSquared = 0.0;
double calculateSampleEntropy() {    //Function
to calculate Sample Entropy   int m = 2; //
Embedding dimension   double r = 0.2; //
Tolerance (percentage of RMS amplitude)
  // Calculate RMS amplitude of the signal   double
rmsValue = calculateRMS();   int countMatches =
0;   for (int i = 0; i < SAMPLES - m; i++) {     for
(int j = i + 1; j < SAMPLES - m; j++) {       double
distance = 0.0;       for (int k = 0; k < m; k++) {
distance += pow(vReal[i + k] - vReal[j + k], 2);}
distance = sqrt(distance);       if (distance < r *
rmsValue) {         countMatches++; }}}
  double sampleEntropyValue =
log((double)countMatches / ((SAMPLES - m) *
(SAMPLES - m - 1)));   return
sampleEntropyValue;}
  lcd.clear();   lcd.setCursor(0, 0);
lcd.print("Duration: " + String(duration, 0) +
"sec");
lcd.setCursor(0,
1);
lcd.print("Freque
ncy: " +
String(medianFrequency, 2) + "Hz");
lcd.setCursor(0, 2);
  lcd.print("Intensity:          "          +
String(pow(rmsValue,2), 2) + "mV");
}
```

```
void displayValues(){  //Display to the LCD
```

Appendix B -- Code for the MQTT Protocol subscription

import Conn as dbconnection

Subscribing in on_connect() means that if we lose the connection and

import MLmodel as mL

import paho.mqtt.client as

mqtt

broker_username = "utcon" broker_password

= "UtContraction"

The callback for when the client receives a

CONNACK response from the server. def

on_connect(client, userdata, flags, rc): if

rc == 0: print("Connected to MQTT

Broker!")

else:

print("Failed to connect, return code
%d\n", rc) print("Connected with result code

"+str(rc))

reconnect then subscriptions will be renewed. device_ids = ["device1", "device2"] # Actual device IDs lwt_topics = [f"devices/{device_id}/lwt" for device_id in device_ids] for topic in lwt_topics: client.subscribe(topic)

subscribe to topic having readings client.subscribe("devices/device_id/readings")

Callback function to handle MQTT message records def on_record(msg): global topic global payload global qos topic = msg.topic payload = msg.payload.decode('utf-8') qos = msg.qos

The callback for when a PUBLISH message client.username_pw_set(broker_username,
is received from the server. broker_password)

def on_message(client, userdata, msg):

on_record(msg) client.loop_forever()

duration = payload.split(",",1)[0].strip()

rms = payload.split(",",2)[1].strip() fmed

= payload.split(",",3)[2].strip() fpeak =

payload.split(",",4)[3].strip() sampleEn =

payload.split(",",5)[4].strip()

dbconnection.main()

dbconnection.write_instance((duration,rms,fmed,fpeak,sampleEn))

mL.ML_predict((rms,fmed,fpeak,sampleEn))

client.connect("localhost", 1883, 60)

```
    print(topic+" "+" ".join(payload.split(",")))

client = mqtt.Client()

client.on_connect = on_connect

client.on_message = on_message
```


Appendix C -- Code for the Database Connection on Central Device

```
import psycopg2 as psycopg

from psycopg2 import sql

import secrets_file


def dbconnect():

   # function to connect to PostfreSQL

database     conn = psycopg.connect(

host = secrets_file.dbhost,

dbname = secrets_file.dbname,

user = secrets_file.dbuser,

password = secrets_file.dbpass,

port = secrets_file.dbport )     cur =

conn.cursor()

   cur.execute('DROP TABLE IF EXISTS
UCparameters')     create_script = """
CREATE TABLE IF NOT EXISTS
ContractionParameters (

                    Duration    NUMERIC,

                    RMS          NUMERIC,

Fmed       NUMERIC,

                    Fpeak      NUMERIC,

                    SampEn     NUMERIC   )
"""

cur.execute(create_script)

conn.commit()

cur.close()     conn.close()

return conn


def dbexist():     # check if

database exists

   try:

      conn = psycopg.connect(

host = secrets_file.dbhost,

dbname = secrets_file.dbname,

user = secrets_file.dbuser,

password = secrets_file.dbpass,

port = secrets_file.dbport  )

conn.close()       print('Database

exists.')       return True     # returns
```

boolean except

psycopg.OperationalError:

print('Database does not exist.')

return False

def dbcreate(conn): # creates database

conn = psycopg.connect(user =

secrets_file.dbuser, password =

secrets_file.dbpass, host =

secrets_file.dbhost, port =

secrets_file.dbport,) conn.autocommit

= True cursor = conn.cursor()

create_db = sql.SQL("CREATE DATABASE {};").format(sql.Identifier(secrets_file.dbname))

try:

cursor.execute(create_db)

print(f"Database '{secrets_file.dbname}'

created successfully!") except

psycopg.Error as e:

print(f"Error creating database: {e}")

finally:

cursor.close()

conn.close()

def main(): conn =

psycopg.connect(host =

secrets_file.dbhost, user =

secrets_file.dbuser,

password = secrets_file.dbpass,

port = secrets_file.dbport) if

dbexist():

```
    return dbconnect()
else:
```

```
return dbcreate(conn)     conn.close()
```

```
    return

if __name__ == "__main__":

    main()


def write_instance(data_to_write):         conn =

psycopg.connect(

            host = secrets_file.dbhost,

dbname = secrets_file.dbname,              user =

secrets_file.dbuser,              password =

secrets_file.dbpass,              port =

secrets_file.dbport )

    cur = conn.cursor()


    sql = "INSERT INTO
ContractionParameters (Duration, RMS,
Fmed, Fpeak, SampEn) VALUES  (%s, %s,
%s, %s, %s)"

    cur.execute(sql, data_to_write )

    conn.commit()

    cur.close()
```

Appendix C – Machine Learning Code

```
import numpy as np import

pandas as pd

from sklearn.model_selection import
    train_test_split from
    sklearn.ensemble import
    RandomForestClassifier

# Load your own dataset from CSV file

dataset = pd.read_csv('balanced_dataset -
    Copy - Copy2.csv')

# Assuming your dataset has features columns
  named 'feature1', 'feature2', ..., and a target
  column named 'target'

X = dataset.drop(columns=['Premature']) y

    = dataset['Premature']

#dataset = pd.DataFrame(dataset)

#mapping = {'t': 1, 'f': 0}

#dataset['Premature'] =
dataset['Premature'].map(mapping)
```

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

#feature_names = X.columns.tolist()

# Initial model training initial_model =

RandomForestClassifier()

initial_model.fit(X_train, y_train) def

collect_feedback(instance, prediction):

return 1 if input(f"Is prediction
{prediction:.2f} (prob) correct for instance
{instance}? (1/0): ") == "1" else 0

# Simulated feedback loop

for instance, true_label in

zip(X_test.values, y_test.values):

prediction_probs =

initial_model.predict_proba([instance])[0]

max_prob = max(prediction_probs)

predicted_label =
np.argmax(prediction_probs)

print(f"Instance: {instance} - True Label:
{true_label} - Predicted Label:
{predicted_label} - Max Prob:

{max_prob:.2f}")    if

max_prob < 0.98:

feedback = collect_feedback(instance,

prediction_probs[predicted_label])

if feedback == 0:

corrected_label = int(input("Enter the
correct label for this instance (0/1): "))

y_train = np.append(y_train,

corrected_label)

X_train = np.vstack((X_train,
instance))

# Retrain the model with updated data

updated_model =
RandomForestClassifier()

updated_model.fit(X_train, y_train)

print("Model updated based on
feedback.")

# Evaluate the final model
```

```
print(f"Final model accuracy: {accuracy}")

y_pred = updated_model.predict(X_test)

y_pred from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)

from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)

from sklearn.metrics import classification_report
print(classification_report(y_pred, y_test))

from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
y_test_numeric = label_encoder.fit_transform(y_test)
y_pred_numeric = label_encoder.transform(y_pred)

from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt fpr, tpr, thresholds = roc_curve(y_test_numeric, y_pred_numeric)
roc_auc = auc(fpr, tpr) plt.figure(figsize=(8, 6)) plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (area = {roc_auc:.2f})') plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0]) plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
```

accuracy = updated_model.score(X_test, y_test)

plt.ylabel('True Positive Rate')

```
plt.title('Receiver Operating Characteristic
(ROC) Curve')  plt.legend(loc="lower right")
plt.show()

features = dataset.columns importances =
updated_model.feature_importances_
indices = np.argsort(importances)

plt.title("Feature Importances")
plt.barh(range(len(indices)),
importances[indices], color = 'b', align =
'center')
plt.yticks(range(len(indices)), [features[i] for i
in indices])
plt.xlabel('Relative Importance') plt.show()

import joblib
joblib.dump(updated_model,
'random_model.joblib')
```