VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590018, Karnataka

Report
on

"DIABETIES PREDICTION USING MACHINE LEARNING AND EMBEDDED SYSTEM"

Submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering
in
Computer Science & Engineering

Submitted by

USN	Name
1BI20CS070	Harsh Adhikari
1BI20CS047	Chinmay Lohomi
1BI20CS057	Dheeraj Katoch
1BI20CS189	Vivek Gupta

Under the Guidance of
Dr. Madhuri J
Assistant Professor
Department of CS&E,BIT
Bengaluru-560004

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BANGALORE INSTITUTE OF TECHNOLOGY

K.R. Road, V.V. Pura, Bengaluru-560 004

2023-24

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590018, Karnataka

BANGALORE INSTITUTE OF TECHNOLOGY

Bengaluru-560 004

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

This is to certify that the project work entitled **"DIABETIES PREDICTION USING MACHINE LEARNING AND EMBEDDED SYSTEM"** carried out by

USN	Name
1BI20CS070	**Harsh Adhikari**
1BI20CS047	**Chinmay Lohomi**
1BI20CS057	**Dheeraj Katoch**
1BI20CS189	**Vivek Gupta**

a bonafide students of VII semester B.E. for the partial fulfillment of the requirements for the Bachelor's Degree in Computer Science & Engineering of the **VISVESVARAYA TECHNOLOGICAL UNIVERSITY** during the academic year 2023-24. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said degree.

Dr. Madhuri J	**Dr. Girija J**	**Dr. Aswath M U**
Assistant Professor	Professor & Head Dept of CSE	Principal, BIT

External Viva

Name of the Examiners Signature with date

1.

2.

Similarity Report

DrillBit

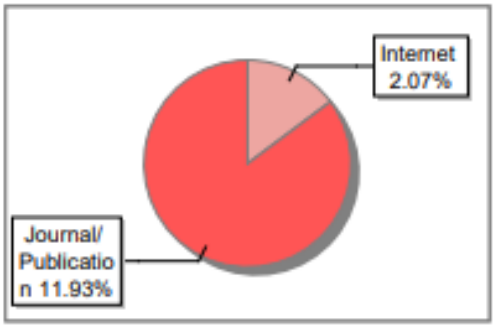
The Report is Generated by DrillBit Plagiarism Detection Software

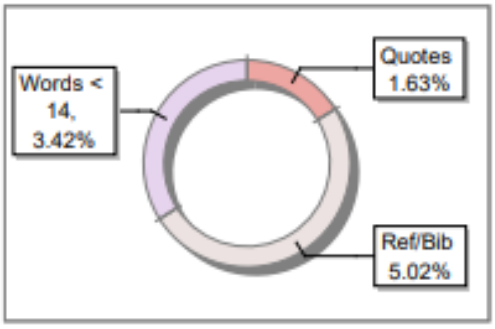
Submission Information

Author Name	Harsh Adhikari
Title	Harsh Adhikari
Paper/Submission ID	1753347
Submitted by	bitlibrary79@gmail.com
Submission Date	2024-05-06 11:31:19
Total Pages	33
Document type	Project Work

Result Information

Similarity **14 %**





Exclude Information

Quotes	Excluded	Language	English
References/Bibliography	Excluded	Student Papers	Yes
Sources: Less than 14 Words %	Excluded	Journals & publishers	Yes
Excluded Source	0 %	Internet or Web	Yes
Excluded Phrases	Excluded	Institution Repository	Yes

Database Selection

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

14	13	B
SIMILARITY %	MATCHED SOURCES	GRADE

A-Satisfactory (0-10%)
B-Upgrade (11-40%)
C-Poor (41-60%)
D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	engineering.futureuniversity.com	6	Publication
2	gnit.ac.in	2	Publication
3	dspace.daffodilvarsity.edu.bd 8080	1	Publication
4	qdoc.tips	1	Internet Data
5	dspace.nm-aist.ac.tz	1	Publication
7	www.atlantis-press.com	1	Publication
8	sjcit.ac.in	<1	Publication
9	www.linkedin.com	<1	Internet Data
10	engineering.futureuniversity.com	<1	Publication
11	qdoc.tips	<1	Internet Data
12	pdf4pro.com	<1	Internet Data
13	testsigma.com	<1	Internet Data
14	www.dx.doi.org	<1	Publication

EXCLUDED PHRASES

ACKNOWLEDGEMENT

The successful completion of phase I of the project stands as a testament to the unwavering support and invaluable contributions of numerous individuals, each of whom played an indispensable role in its realization. We extend our deepest gratitude to the visionary Principal **Dr.Ashwath M U**, whose unwavering encouragement and provision of resources paved the way for our academic pursuits.

Our heartfelt thanks go to **Dr. Girija J, Professor and Head of Department of CS&E,** whose insightful guidance, encouragement, and belief in our abilities provided the foundation for this endeavor within the **Department of CS&E**. Their wisdom and continuous support served as a guiding beacon throughout this project's journey.

A project of this magnitude necessitated seamless coordination and meticulous planning. Hence, our heartfelt appreciation goes to our diligent project coordinators, **Dr. Maya B S** and **Prof. Neetha Das**, for their exceptional organizational skills, unwavering commitment, and tireless efforts in ensuring every aspect of this endeavor proceeded smoothly. A special mention to **Prof. Tejashwini P S** for her valuable feedback during the evaluation of report.

The depth and quality of this project owe much to the expertise and guidance of our esteemed project guide, **Dr. Madhuri J.** Their invaluable insights, patience, and mentorship empowered us to navigate complexities, refine methodologies, and achieve milestones that seemed daunting initially.

In essence, the culmination phase I of the project is a testament to the collaborative spirit and unwavering support of these remarkable individuals, whose contributions have been integral to its success.

1BI20CS070	**Harsh Adhikari**
1BI20CS047	**Chinmay Lohomi**
1BI20CS057	**Dheeraj Katoch**
1BI20CS189	**Vivek Gupta**

ABSTRACT

Diabetes mellitus (DM) poses a significant global health burden, with increasing prevalence rates and associated complications, including diabetic foot ulcers and amputations. Early detection of foot pressure abnormalities indicative of diabetic neuropathy is crucial for preventing these complications. This research presents an intelligent foot pressure analysis system integrating sensor technology and machine learning algorithms for early detection of diabetic foot complications.

Through quantitative assessment of foot pressure distribution and gait dynamics, the system identifies subtle alterations in foot pressure patterns associated with diabetic neuropathy. Preliminary results demonstrate the system's feasibility, with high accuracy, sensitivity, and specificity in discriminating between normal and abnormal foot pressure profiles.

Real-time monitoring capabilities enable timely intervention and preventive measures, promising improved outcomes for individuals with diabetes mellitus. Future research aims to validate and refine the system through clinical studies, with the ultimate goal of enhancing diabetic foot care practices and patient outcomes.

CONTENTS

LIST OF FIGURES

LIST OF Tables

CHAPTER 1

INTRODUCTION

Chapter 1

INTRODUCTION

1.1 Overview

This project pioneers a transformative approach in podiatric medicine by introducing an innovative embedded system designed for comprehensive gait analysis and posture correction. Traditional methods in podiatry face substantial drawbacks, including limited accessibility, cumbersome equipment, and restricted mobility, which hinder accurate assessments of foot-related issues. To counter these limitations, our proposed system leverages modern technology, utilizing discrete sensors, an Arduino Uno microcontroller, and a Bluetooth module to enable real-time pressure mapping of foot dynamics.

The system's objectives span various facets of podiatric care, aiming to address existing challenges. Its lightweight, portable, and wireless nature facilitates assessments in diverse healthcare settings, breaking free from the confinements of traditional laboratory-bound systems. By providing a cost-effective, user-friendly, and accessible solution, this system aims to democratize gait analysis, making it available in clinics, research labs, and sports facilities alike.

Notably, the integration of machine learning models such as KNN, SVM, and Random Forest enhances the system's diagnostic capabilities, predicting potential foot-related diseases. This predictive analysis aspect stands as a valuable addition, enabling early intervention and personalized treatment plans based on comprehensive gait analysis data.

Overall, this embedded system emerges as a multifaceted solution, poised to revolutionize podiatric care. Its accessibility, accuracy in gait analysis, predictive diagnostic capabilities, and potential for biomechanical advancements position it as a pivotal tool in podiatric medicine, catering to diverse user needs and advancing research in the field.

1.2 Objectives

- **To Pressure Mapping:** Implement a discrete sensor system to accurately map foot pressure in real-time during various phases of gait, providing detailed data on pressure distribution.
- **Analysis of Foot Injuries:** Utilize the pressure data for comprehensive analysis, enabling the identification and assessment of various foot injuries, abnormalities, and orthopaedic conditions.
- **Research and Product Development:** Provide a reliable platform for researchers and product developers to study foot anatomy, develop innovative medical tools, and design footwear tailored to individual needs.
- **Biomechanical Insights:** Enable sports performance analysis by studying biomechanical data, enhancing understanding of lower limb function in athletes for performance optimization.
- **Disease Detection Using Machine Learning:** Implement machine learning models, including KNN, SVM, and Random Forest, to predict the likelihood of disease based on gathered gait analysis data, contributing to early diagnosis and intervention.

1.3 Purpose, Scope and Applicability

1.3.1 Purpose

- **The Enhance Accessibility and Mobility:** Develop a portable and wireless embedded system to overcome the limitations of existing gait analysis methods, allowing for assessments in various environments and making it accessible to a broader range of healthcare settings.
- **Facilitate Early Detection and Intervention:** Utilize pressure mapping and machine learning to predict potential foot-related diseases, promoting early detection and intervention for improved patient outcomes in podiatric care.

1.3.2 Scope

- **Clinical Applications:** The system's scope extends to clinical settings, enabling podiatrists to conduct comprehensive gait analyses for accurate diagnosis and treatment planning. Its lightweight, wireless nature facilitates mobility, allowing assessments in diverse healthcare

environments.

- **Research and Development:** The project's scope encompasses research institutions and product development teams, providing a valuable tool for studying foot biomechanics, developing innovative medical devices, and designing customized footwear, thereby contributing to advancements in podiatric medicine and related fields.

1.3.3 Applicability

- **Podiatric Medicine:** Applicable in podiatric clinics for precise gait analysis, aiding in the diagnosis, treatment, and monitoring of foot-related disorders and injuries.
- **Sports Science:** Relevant for sports performance analysis, helping coaches and athletes understand and optimize lower limb biomechanics for enhanced athletic performance and injury prevention.
- **Biomedical Engineering:** Applicable in the development of medical tools, orthotic devices, and footwear design, contributing to advancements in biomedical engineering by providing valuable foot pressure data for research and innovation.

- .

CHAPTER 2

LITERATURE SURVEY

Chapter 2

LITERATURE SURVEY

2.1 Introduction

A literature survey is a survey of sources like books, journals, articles related to a specific topic or a research question. The literature survey is important and should be done at the beginning of any project. Writing the literature survey shows the reader how our work relates to existing research and what new insights it will contribute. The reasons to conduct the literature survey:

• To familiarize with the current state of knowledge on the project topic.

• To ensure that our topic is not just repeating what others have already done.

• To identify gaps in knowledge and unresolved problems that the research can address.

• To develop the theoretical framework and methodology.

• To provide an overview of the key findings and debates on the topic.

• To present the knowledge in the form of a written report.

It requires critical analyses of the information gathered by identifying gaps in current knowledge, by showing limitations of the theory and points of view, and by formulating areas for further research and reviewing other related areas.

We have surveyed the literature related to diabetes , foot pressure analysis, machine learning , Hence worth here are the research papers.

2.2 Summary of papers

[1] Zihang You, et al., presents a wearable system for diabetic foot prevention, incorporating shoe-pads with FSR402 sensors. These sensors collect plantar pressure data, transmitted wirelessly via Bluetooth to LPC1768. MATLAB processes data to generate real-time hot maps, providing valuable insights for diabetic patients to prevent foot ulcers by monitoring pressure variations.

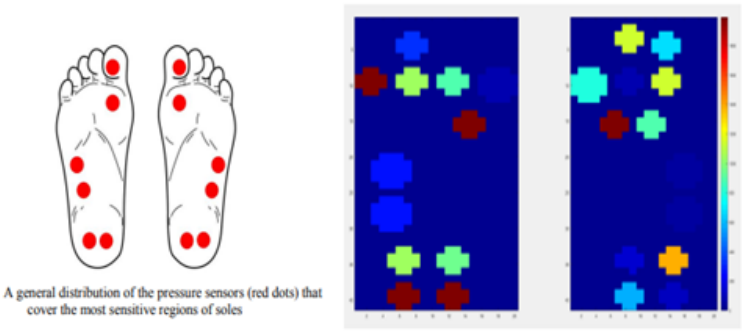


Fig 2.1 : Distribution of foot Pressure

The paper uses Pressure points as shown in Fig2.1 which is exemplified as hot map generated in MATLAB, illustrating the plantar pressure distribution in a graphical format, aiding in real-time monitoring for diabetic foot prevention.

Drawbacks:

- **Training Static Placement:** The sensor placement is fixed, potentially missing variations in pressure from dynamic foot movements.
- **User Discomfort:** Wearable nature may cause discomfort, affecting user adherence.
- **Bluetooth Reliance:** Real-time monitoring depends on Bluetooth connectivity, which may pose reliability challenges.

[2] Wang C, et al.,Employed Gaitboter, a portable system, to analyze gait in type 2 diabetic patients. Significant differences in walking speed and specific gait parameters were observed between patients with and without peripheral neuropathy, emphasizing the system's potential in assessing gait abnormalities in diabetes and guiding clinical interventions.

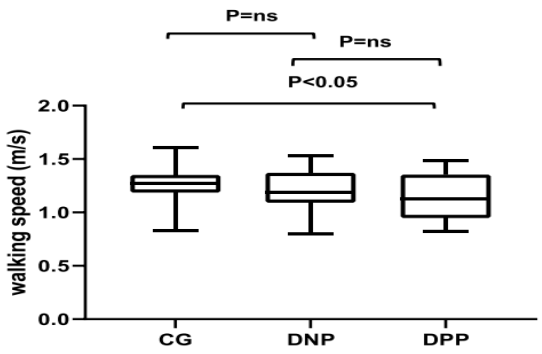


Fig 2.2: Mean of walking speed (m/s) in different groups.

Abbreviations: DPP, diabetic patients with peripheral neuropathy; DNP, diabetic patients without peripheral neuropathy; CG, control group

The paper as shown in Fig 2.2 illustrates the mean landing angles in diabetic patients with and without peripheral neuropathy, alongside a control group. Differences may highlight gait alterations associated with diabetes and neuropathy.

Drawbacks:

- **Limited Sample Size:** The study included a small number of participants, potentially limiting the generalizability of findings to a broader population.
- **Participant Homogeneity:** Lack of diversity in age and diabetes duration may affect the applicability of results to a wider range of patients.
- **Cross-Sectional Design:** The study's cross-sectional nature prevents establishing causal relationships between gait parameters and diabetes or neuropathy.
- **Equipment Dependency:** Findings heavily rely on the Gaitboter system; the exclusive use of this technology may limit comparisons with other gait analysis methods.
- **Short Observation Period:** Gait analysis was conducted during a brief walk, providing a snapshot rather than a comprehensive assessment of daily life activities.

[3] Imed Bouchrika, et al., presents a model-based method for human motion analysis, focusing on joint extraction using elliptic Fourier descriptors. Heel strikes aid gait analysis. The approach is tested on diverse datasets, achieving accurate joint localization for indoor and outdoor scenarios. Recognition results demonstrate its effectiveness, with future work focusing on generalized deployment

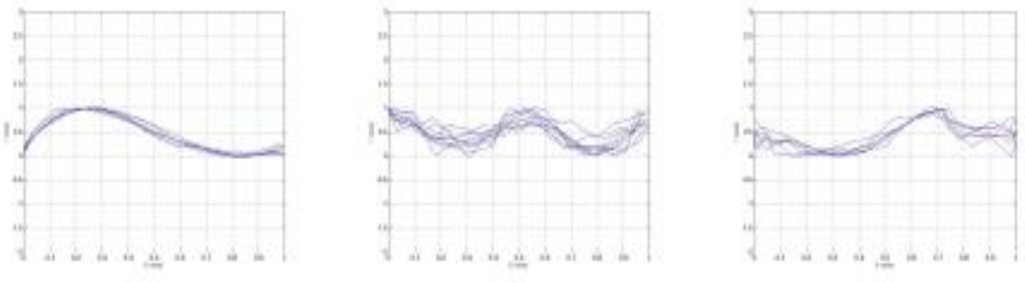


Fig 2.3: Motion Analysis of the Joints: (a) Ankle, (b) Hip , (c)Knee

Fig 2.3 illustrates the motion analysis of ankle, knee, and hip joints during a gait cycle. The cyclic graphs depict joint trajectories between successive heel strikes. These model templates guide the evidence-gathering process, aiding precise joint extraction throughout the paper's methodology.

Drawbacks:

- **Occlusion Challenges:** The model-based method faces difficulties in accurately localizing joints during occlusion, limiting its performance in scenarios where body parts are obscured.
- **Sensitivity to Clothing Types:** Variations in clothing types can impact joint extraction accuracy, potentially leading to errors in motion analysis.
- **Segmentation Errors:** Errors in the segmentation process can affect the reliability of joint localization, especially in complex environments or varied recording conditions.
- **Viewpoint Dependency:** Different viewpoints may pose challenges, affecting the algorithm's ability to consistently extract joints across diverse camera perspectives.
- **Computational Load:** The proposed Hough Transform method may have a high computational load, potentially limiting real-time applications or scalability in large datasets.

[4] Bochen Li, et al., introduces a foot plantar pressure measurement system using a flexible force-sensitive sensor. It undergoes calibration, comparing results with a commercial system. Clinical trials reveal significant gait parameter differences between knee joint injury subjects and controls, confirming the system's clinical value for assessing dynamic plantar pressure and gait characteristics.

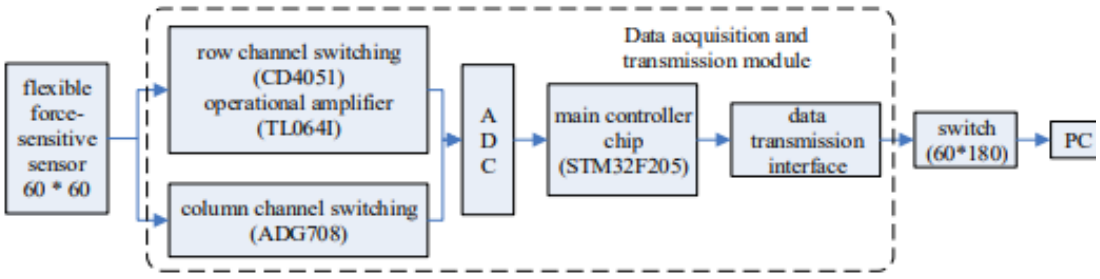


Fig 2.4 : The flow chart of data acquisition and data transmission.

The Fig 2.4 illustrates the flow chart of data acquisition and transmission in the foot plantar pressure measurement system. It outlines the hardware components, including the flexible force-sensitive sensor, main controller chip, and data transmission interface, depicting the process of data collection, integration, and transmission for subsequent analysis.

Drawbacks:

- **Limited Focus:** The study primarily concentrates on knee joint injuries, potentially limiting the generalizability of findings to other medical conditions.
- **Equipment Specificity:** Results heavily rely on the developed system, and generalizing to other commercially available systems may be challenging.
- **Restricted Demographics:** The study includes a specific age group and may not fully represent the diversity of the population.
- **Single Calibration Method:** The calibration method using an air pressure calibration station may have limitations in capturing real-world complexities.

[5] Alvaro Muro-de-la-Herran, et al., compares Non-Wearable Sensor (NWS) and Wearable Sensor (WS) systems for gait analysis. NWS offers in-depth, simultaneous parameter analysis in controlled environments. WS, though less accurate, allows continuous monitoring during daily activities, promoting autonomy. Considerations for future research involve sensor improvements, energy efficiency, miniaturization, and signal processing enhancements.

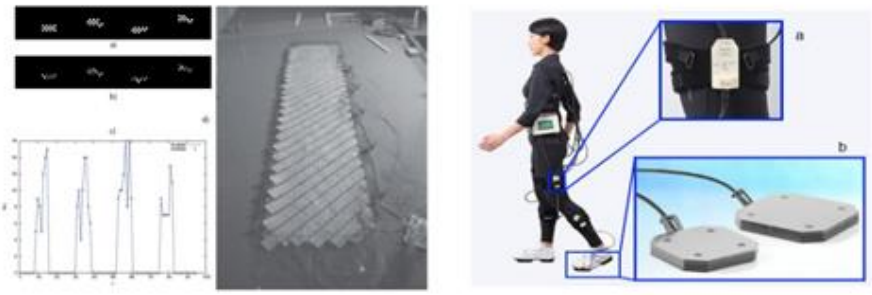


Fig 2.5 : Wearable and Non-Wearable Systems

The paper illustrates a comparison between Non-Wearable Sensor (NWS) and Wearable Sensor (WS) systems based on factors like power consumption and accuracy. Fig 2.5 categorizes gait analysis methods, emphasizing applications, accuracy, price, and ease of use, aiding in system selection for specific user profiles.

Drawbacks:

- **Limited Real-Life Monitoring:** Wearable sensor systems face challenges in monitoring real-life gait outside instrumented environments.
- **Power Consumption:** WS systems have restrictions due to limited battery duration, impacting continuous monitoring during daily activities.

- **Analysis Complexity:** Complex algorithms are needed for parameter estimation from inertial sensors, posing challenges in data interpretation and analysis.

[6] Sen Qiu, et al., implemented a wearable gait analysis system using shoe-integrated inertial sensors for stroke patients. Gait parameters were analyzed, revealing significant deviations from normal values. The system offers objective and patient-specific evaluation for rehabilitation, guiding tailored therapies. Future work involves clinical applications, database creation, and physiological analysis integration.

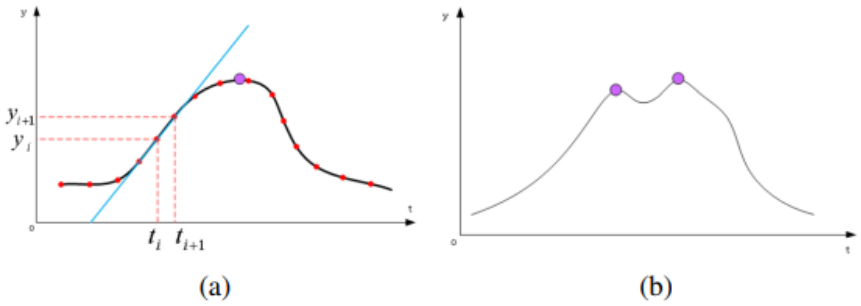


Fig 2.6 : Slope (a) Conventional slope method (b) signal fluctuation

The conventional slope method calculates the slope of the signal curve, often used in gait analysis. In Fig 2.6, it could refer to determining gait phases. Signal fluctuation involves variations in the signal, highlighting changes in gait dynamics, potentially indicating abnormalities or asymmetries.

Drawbacks:

- **The Limited Joint Kinematics:** The system, using a single IMU, cannot compute joint kinematics, restricting detailed analysis of specific joint movements during gait.
- **Synchronization Challenges:** Synchronized videos, taken by non-professionals, may lack precision, affecting the accuracy of correlating observational data with measured parameters.
- **Feasibility Constraints:** Instrumented validation work on patients in the hospital is currently infeasible, limiting direct validation of the proposed system against gold standard methods.

[7] Wenyao Xu, et al.,proposes that, Smart Insole is a portable system for gait analysis, integrating low-cost sensors. It enables precise analysis in daily life, reducing

the need for specialized labs. The system's affordability and mobility extend applications to fall prevention, life behavior analysis, and networked wireless health systems, fostering pervasive gait analysis.

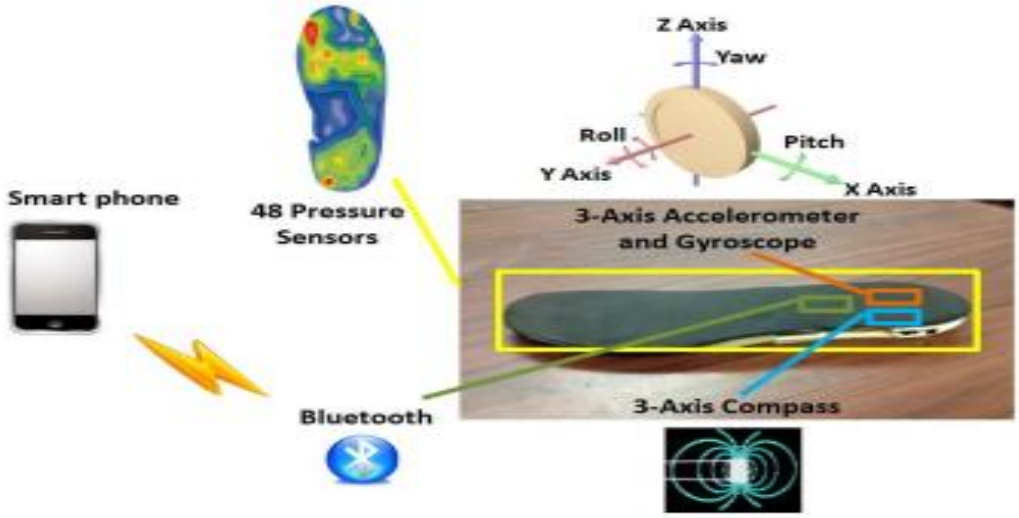


Fig 2.7 : Insole Architecture

The paper presents Smart Insole, integrating pressure sensors, accelerometers, and gyroscopes (Fig. 2.7). This portable system enables accurate gait analysis in daily life, offering a cost-effective alternative to traditional lab setups.

Drawbacks:

- **The Sensor Placement Variability:** Accuracy may be affected by variations in the placement of sensors within the insole.
- **User Compliance:** The system relies on consistent daily wear, and non-compliance may impact data quality.
- **Limited Depth of Analysis:** The system's depth of gait analysis may not match the comprehensive insights provided by traditional gait labs.

[8] Lin Shu, et al., introduces an innovative in-shoe plantar pressure measurement system using a textile pressure sensing array. The system incorporates a wireless data acquisition system with three configuration modes. Experimental results demonstrate its effectiveness in capturing plantar pressure during various activities, showcasing its potential for clinical and athletic applications.

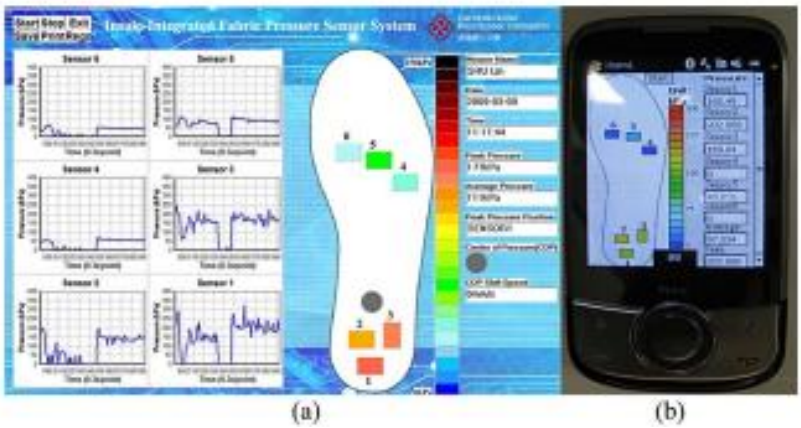


Fig 2.8: Plantar pressure data interface for (a) desktop and laptop and (b) smart phone.

The paper's essence is captured in Fig 2.8, revealing user interfaces for desktop and smartphone configurations. These interfaces visualize real-time plantar pressure data, offering insights into the innovative in-shoe measurement system's capabilities and applications in diverse settings, from clinics to daily outdoor activities.

Drawbacks:

- **Limited sensor coverage:** The system uses a small number of sensors, potentially missing detailed pressure information from specific areas of the foot.
- **Temperature sensitivity:** Lack of comprehensive temperature compensation may affect accuracy in environments with significant temperature variations.
- **Crosstalk error:** Discrete pressure measurements may introduce crosstalk errors, impacting the precision of the obtained plantar pressure data.

[9] Giuseppe Lamola, et al investigates gait modifications in diabetic patients, revealing a wider base of support and reduced foot excursion with disease progression. These biomechanical changes suggest impaired balance control and highlight the importance of early intervention to prevent foot ulcers, emphasizing the need for larger-scale longitudinal studies.

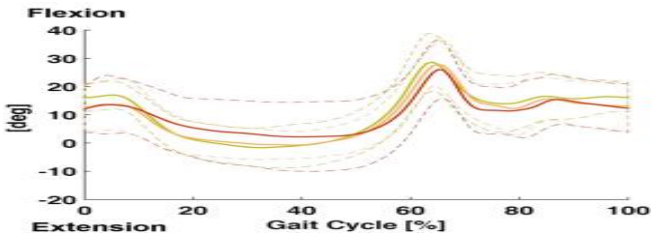


Fig 2.9 Comparison of the metatarso-phalangeal flexion-extension between the three groups, showing a trend of progressive reduction of ROM of flexion-extension with the advancing of the disease

Fig 2.9 illustrates the comparison of metatarso-phalangeal flexion-extension among diabetic patients at different disease stages. It demonstrates a progressive reduction in range of motion with disease advancement, indicating diminished foot mobility. These findings underscore the relevance of foot segmental kinematics in assessing diabetic foot complications.

Drawbacks

- **Small Sample Size:** The study's limited participant pool may reduce the generalizability of findings and limit the statistical power of the analysis.
- **Lack of Longitudinal Data:** While the study identifies early biomechanical indicators, longitudinal data could provide clearer insights into disease progression and the efficacy of interventions.
- **Absence of Neuropathy Severity Assessment:** Not investigating the extent of neuropathy limits understanding of its contribution to gait alterations and foot complications.

[10] Karim Gariani et al proposes that Plantar fasciitis (PF) management in diabetic patients follows a three-tiered approach: conservative measures, steroid injections, and, if needed, surgical intervention or shockwave therapy. Surgical options include endoscopic or open fasciotomy. Infectious PF is rare but requires immediate intervention. ESWT's efficacy is inconclusive, while surgery yields varied success rates.

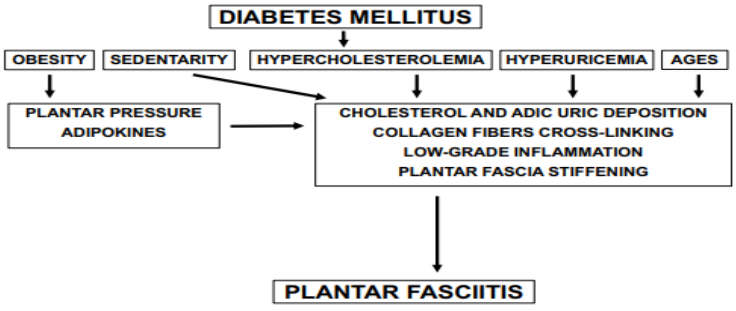


Fig 2.10 Proposed pathogenic mechanisms for the development of plantar fasciitis in diabetic patients.

Fig 2.10 depicts the proposed pathogenesis of plantar fasciitis (PF) in diabetic patients. It highlights the role of metabolic factors like hyperglycemia, advanced glycation end-products (AGEs), and altered collagen structures in PF development, emphasizing the complex interplay between metabolic dysfunction and mechanical stress on tendons.

Drawbacks

- **Limited Evidence:** The paper relies heavily on retrospective and heterogeneous studies, lacking robust evidence, especially regarding diabetic patients.
- **Lack of Comparative Analysis:** It lacks direct comparisons between different treatment modalities and their efficacy, particularly in diabetic populations.
- **Sparse Discussion on Complications:** While it briefly mentions complications of surgical interventions, it lacks in-depth discussion on potential adverse outcomes, especially in diabetic patients who may be more prone to complications.

2.3 Drawbacks of Existing System

- **Inadequate Limited Accessibility:** Existing systems often require specialized and costly equipment, making them inaccessible to smaller clinics and research facilities.
- **Labor-Intensive Process:** Manual data collection and analysis in some systems are time-consuming and dependent on operator expertise, leading to inconsistencies and errors.
- **Confined to Laboratories:** Some systems are confined to laboratory settings, restricting the analysis of gait patterns in real-world environments where natural movement may differ.
- **Wired Sensors Restriction:** Reliance on wired sensors can restrict the freedom of movement for patients during gait analysis, potentially affecting the accuracy of results.
- **Complex Setup Procedures:** Setting up and calibrating existing systems for gait analysis can be a complex process, requiring trained personnel and significant time.

2.4 Problem Statement

“To develop a wireless gait analysis system that is portable, lightweight for real-time monitoring and medical diagnosis using Arduino Uno and FSR sensor.”

Input: The input Data collected from our Podiatric device built using IoT technology

Output: The result is the Prediction of disease due to foot pressures

2.5 Proposed Method

- The Gait analysis system for podiatric analysis where the FSR are used to measure pressure
- Wireless data transmission eliminates the hindrance caused due to long wires
- To develop our model we use embedded system development life cycle
- AI application is employed to predict the type of disease of the foot due to diabetes in patients using ML algorithms
- ML algorithms like SVM, KNN, Logistic regression and Random Forest are used.

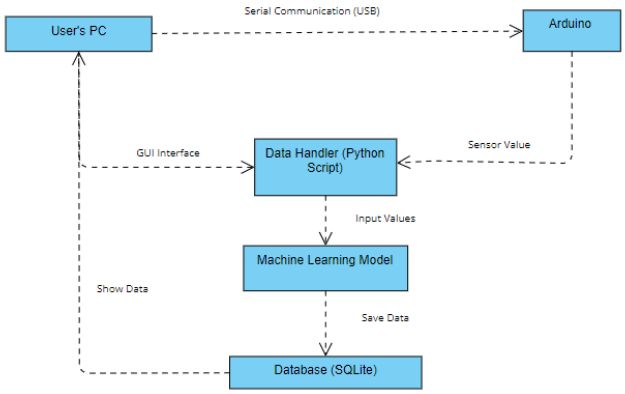


Fig 2.5.1: Architectural Design

In the architecture diagram, a person places their foot on a sensor, initiating data collection. The sensor signals are transmitted to a microcontroller connected to a Bluetooth module for wireless communication. The microcontroller preprocesses the data using PLX DAQ software, enhancing its quality. Subsequently, machine learning models like KNN, Random Forest, and SVM analyze the preprocessed data to predict potential foot-related diseases. This comprehensive process integrates real-time data collection, wireless transmission, preprocessing, and advanced machine learning, culminating in a robust system capable of accurate disease prediction based on dynamic foot pressure mapping.

CHAPTER 3

REQUIREMENT ENGINEERING

Chapter 3

REQUIREMENT ENGINEERING

The requirements for a system are the descriptions of what the system should do, the services that it provides and the constraints on the operation. These requirements reflect the needs of customers for a system that serves a certain purpose such as controlling a device, placing an order or searching an information. The process of finding out, analysing, documenting and checking these services and constraints is called requirements engineering (RE).

3.1 Software and Hardware tools used

a) Software requirements:

- **Coding Language :** Python.
- **Operating system :** Windows 10 with MS EXCEL.
- **Library :** Matplotlib, numpy, Pandas, Sckit-learn.
- **tool :** Anaconda Navigator IDE 3.7.4 , jupyter Notebook, Arduino IDE

b) Hardware requirements:

- ATMEGA 328 – 1
- FSR Sensor – 4
- Bluetooth module HC-05 – 1
- Mobile phone Buzzer - 1
- LED - 4
- Resistors – 4
- Jumper wires

3.2 Conceptual/ Analysis Modelling

In broad terms, conceptual modelling is the process of developing a graphical representation (or model) from the real world. In the context of collaborative problem solving, it provides an easily understood representation of the system for the different stakeholders involved. The process of conceptual modelling requires decisions to be taken regarding the scope and level of detail of the model. These decisions should generally be a joint agreement between the modeller and the problem owners i.e., the stakeholders who require the model to aid decision making. It also requires assumptions to be made about the situation concerned.

A conceptual model is a representation of a system, made of the composition of concepts which are used to help people know, understand, or simulate a subject the model represents. It is also a set of concepts. In contrast, physical models are physical objects; for example, a toy model which may be assembled, and may be made to work like the object it represents. Unified Modelling Language (UML) is a standard language for specifying, visualizing, constructing and documenting the artifacts of software systems. There are 5 types of UML diagrams namely: use case diagrams, sequence diagrams, activity diagrams, state chart diagrams and class diagrams.

3.2.1 Use case diagram

The purpose of a use case diagram is to capture the dynamic aspect of a system. In its simplest form, a use case identifies the actors involved in an interaction and names the type of interaction. Use cases are documented using a high-level use case diagram. The set of use cases represents all of the possible interactions that will be described in the system requirements. Actors in the process, who may be humans or other systems, are represented as stick figures. Each class of interaction is represented as a named ellipse. Lines link the actors with the interaction. Optionally, arrowheads may be added to lines to show how the interaction is initiated. Include relationships are used to depict common behaviour that are shared by multiple use cases. Each use case should be documented with a textual description. Following is the use case diagram :

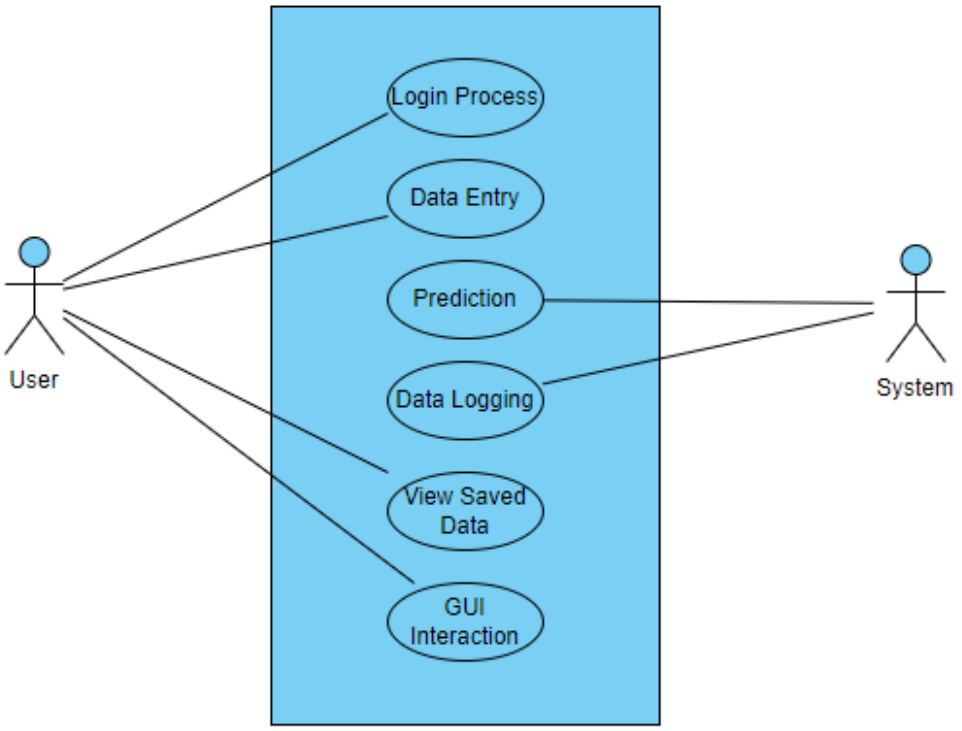


Fig 3.1: Use Case Diagram

3.2.2 Activity Diagram

The basic purpose of activity diagram is to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities. It is used to show message flow from one activity to another. Activity is a specific operation of the system. It can be regarded as a form of a structured flowchart combined with traditional data flow diagram. They are constructed from a number of shapes, connected with arrows. The important shapes are: ellipses indicate actions, a black circle represents the start of the workflow, an encircled circle indicates the end of the workflow, diamonds represent decisions and bars represent split (start) or join (end) of concurrent activities. Arrows begin from the start and run towards the end which indicate the order in which activities happen**.**

The activity diagram for the podiatric gait analysis project depicts a systematic flow of actions. It begins with the user initiating the system, triggering data collection.

The system, using Arduino Uno, performs pressure mapping and processes the information.

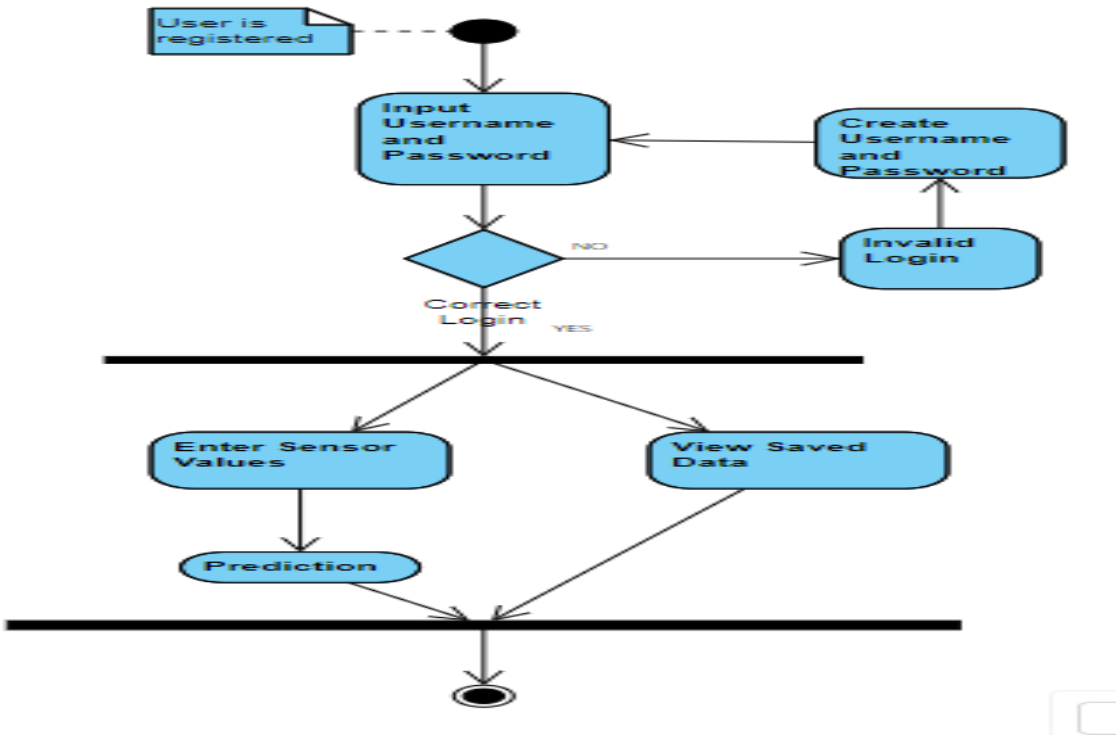


Fig 3.2: Activity diagram

Simultaneously, machine learning models analyze the data for disease prediction. The results are then sent to the mobile app for graphical representation. Admins can access and monitor the system, ensuring its functionality. This sequential representation highlights the dynamic interactions between users, the embedded system, and administrators, providing a comprehensive overview of the project's workflow from data collection to disease prediction and system monitoring

3.2.3 State chart diagram

State chart diagrams define different states of an object during its lifetime and these states are changed in response to events. They are useful to model reactive systems. Reactive systems can be defined as a system that responds to external or internal events. It is used to describe the flow of control from one state to another state. States are defined as a condition in which an object exists and they change when some event is triggered. States are represented as rectangles; the arrows represent stimuli that force a transition from one state to another. The important purpose of state chart diagrams is to model lifetime of an object from creation till termination.

The state chart diagram for the podiatric gait analysis project illustrates the system's dynamic states and transitions.

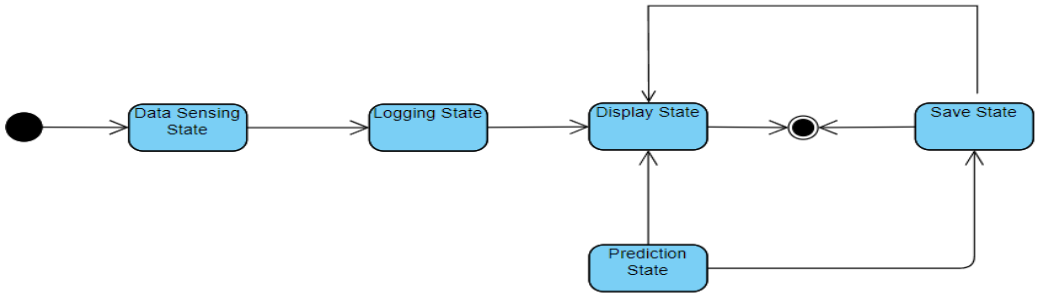


Fig3.3: State chart diagram

The initial state represents the system at rest. Upon user initiation, it transitions to the active state, initiating data collection. The system then progresses through states like "Pressure Mapping," "Data Processing," and "Machine Learning Analysis." Depending on the results, it transitions to states such as "Disease Prediction" or "Normal Analysis." Admin interactions trigger transitions to states like "System Monitoring" or "Configuration." The diagram captures the dynamic nature of the embedded system, depicting various states and transitions throughout its operational lifecycle.

3.2.4 Class diagram

A class diagram is a graphical representation of the structure and relationships within a system's classes, showcasing the static design elements. In this diagram, classes are depicted as rectangles, each containing attributes and methods. Arrows indicate associations between classes, showcasing how they are connected. Additionally, multiplicity notations signify the cardinality of associations. The diagram may include inheritance relationships, portrayed by a solid-line arrow with a triangular arrowhead pointing to the superclass. Interfaces are represented by lollipop symbols, and abstract classes are italicized.

Stereotypes, such as <<entity>> or <<controller>>, can provide additional information about the classes. Overall, a class diagram provides a comprehensive overview of the system's architecture, highlighting the key classes, their attributes, methods, and relationships, contributing to a clearer understanding of the static structure of the software system.

The class diagram for the podiatric gait analysis project portrays the system's structural organization. Key classes include "User," "System," Associations depict interactions, such as the "User" initiating the system and the "System" interacting with the "Mobile App" for data display.

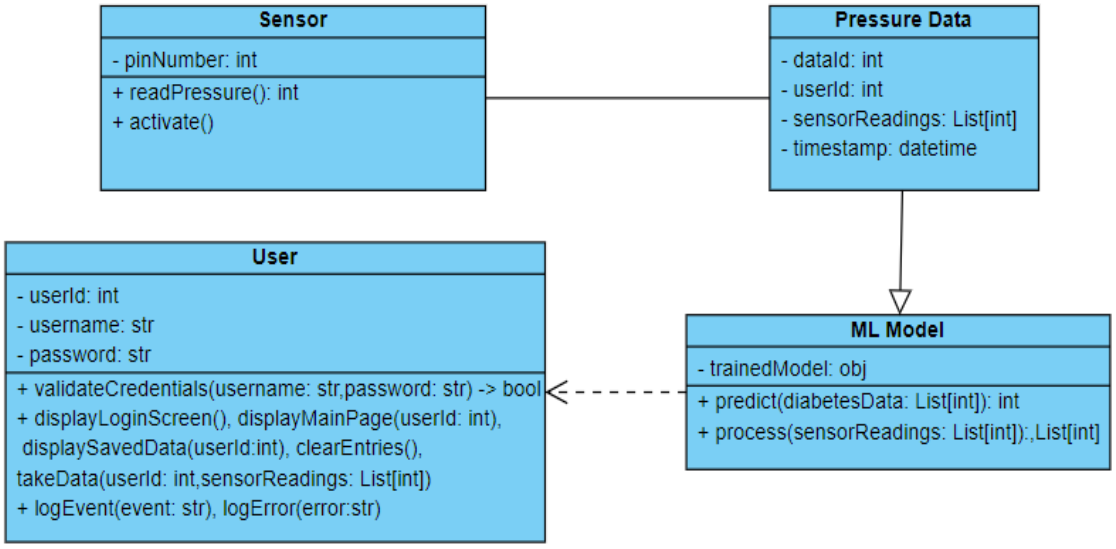


Fig3.4: Class Diagram

Additional classes like "Machine Learning Models" and "Data Processing" encapsulate specific functionalities. Inheritance relationships, such as the specialization of "Machine Learning Models," add granularity. The class diagram succinctly represents the project's architectural blueprint, showcasing the relationships and interdependencies among classes essential for the seamless functioning of the embedded gait analysis system.

3.3 Software Requirements Specification

The software requirements document (sometimes called the software requirements specification or SRS) is an official statement of what the system developers should implement. It should include both the user requirements for a system and a detailed specification of the system requirements. Sometimes, the user and system requirements are integrated into a single description. In other cases, the user requirements are defined in an introduction to the system requirements specification. If there are a large number of requirements, the detailed system requirements may be presented in a separate document.

An SRS document typically includes these elements: the purpose of the software being developed, an overall description of the software, the functionality of the software or what is supposed to do, performance of the software in a productive environment etc. Using SRS ensures that all the requirements are fulfilled.

i. User requirements:

- Accuracy: The system must provide precise and reliable foot pressure mapping to ensure accurate analysis for effective diagnosis and treatment planning.

- Portability: Users require a portable and lightweight system for flexibility in conducting gait analysis in various healthcare settings, including clinics, research labs, and sports facilities.
- User-Friendly Interface: The system should feature an intuitive and easy-to-use interface, enabling podiatrists, researchers, and clinicians to operate it efficiently without extensive training.

ii. System requirements

System requirements are more detailed description of the software system's functions, services and operational constraints. They are clearly articulated statements of what a system must be able to do in order to satisfy stakeholder needs and requirements and are derived from the user requirements. They are further defined in three categories namely: functional requirements, non-functional requirements and domain requirements.

Functional Requirements

Functional requirements are the what requirements — What is this system designed to do? As the name suggests, it describes the functionality of the software. These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. In some cases, they also explicitly state what the system shouldn't do. The functional requirements are basically a list of functionalities that must be included in the system. The functional requirements of our project are as follows:

- Applying the Logistic regression, KNN, SVM and Random forest classifier algorithms for retrieval of best algorithm for prediction.
- The system should be able to communicate with the user with appropriate responses.
- The system should be able to analyze the parameters and predict whether a patient has a podiatric disease or not

Non-Functional Requirements:

Non-functional requirements are the how requirements — How will the system do what it's designed to do? As the name suggests, these are requirements that are not directly concerned with the specific services delivered by the system to its users. These requirements describe how each feature should behave under what conditions, what

limitations there should be, and so on. They are often applicable to the system as a whole, rather than individual system features or services. Non-functional requirements are often more critical than individual functional requirements. The non-functional requirements of our project are as follows:

- **Performance**: Timely processing and analysis of data.
- **Accuracy**: High precision in sentiment analysis and fake review detection.
- **Security**: Robust measures to safeguard user data.
- **Reliability**: Consistent system performance without failures.
- **Scalability**: Ability to handle increased data or user loads.

CHAPTER 4

PROJECT PLANNING

Chapter 4

PROJECT PLANNING

Before starting a project, it is essential to determine the tasks to be performed and properly manage allocation of tasks among individuals involved in the software development. Hence, planning is important as it results in effective software development. Project planning is an organized and integrated management process, which focuses on activities required for successful completion of the project. It prevents obstacles that arise in the project such as changes in project's or organization's objectives, non-availability of resources and so on. It helps in better utilization of resources and optimal usage of the allotted time for a project.

Project scheduling is the process of deciding how the work in a project will be organized as separate tasks, and when and how these tasks will be executed. A Gantt chart is a project management tool assisting in the planning and scheduling of projects of all sizes. Project management timelines and tasks are converted into a horizontal bar chart, showing start and end dates, scheduling and deadlines, including how much of the task is completed per stage. The project plan of our project is illustrated as shown in Figure 4.1:

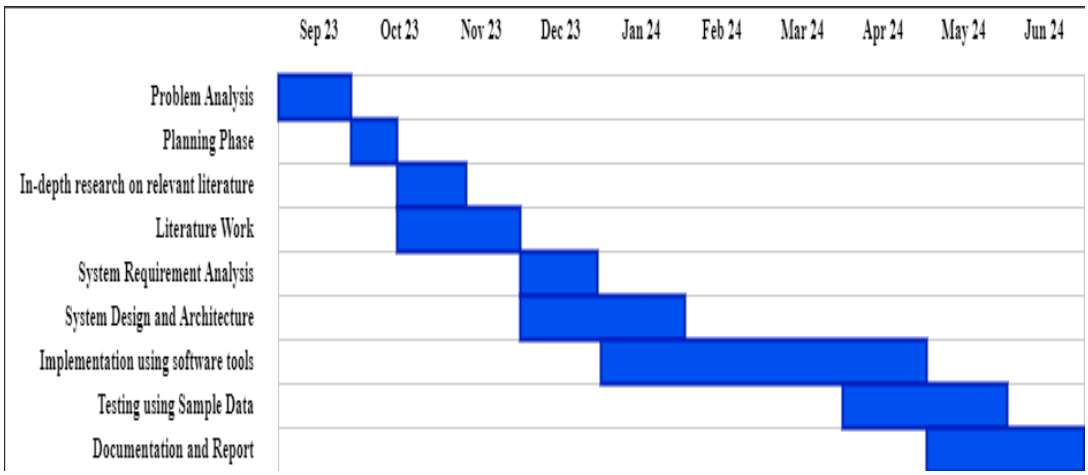


Fig4.1:Gantt Chart

CHAPTER 5

SYSTEM DESIGN

Chapter 5

SYSTEM DESIGN

5.1 System Architecture

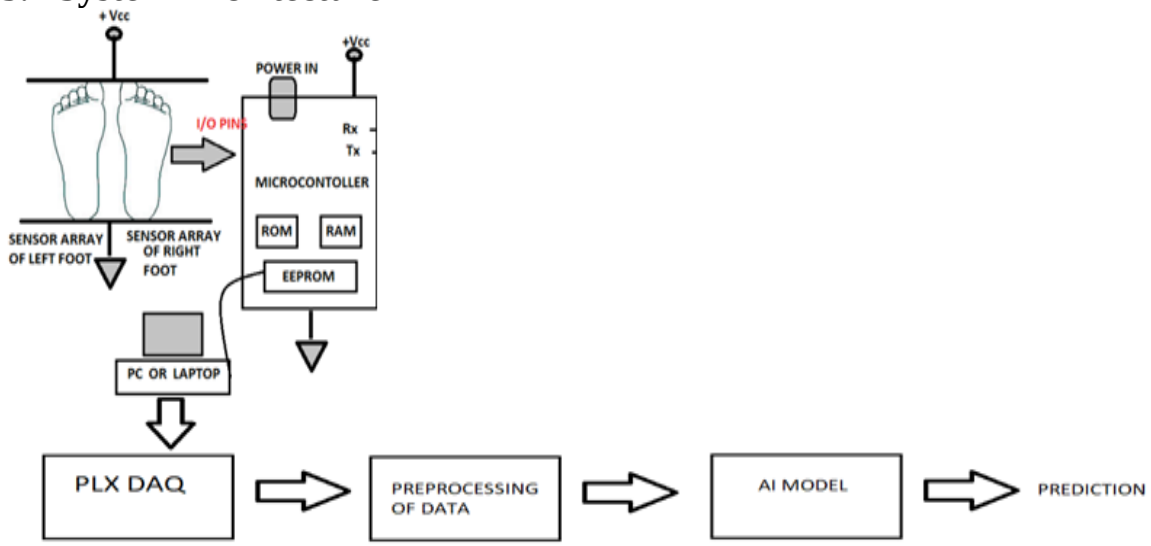


Figure 5.1: Architecture of the proposed system

The complete project encompasses a system architecture that integrates Arduino-based sensor data collection, machine learning prediction, and a graphical user interface (GUI) for user interaction. The architecture can be outlined as follows:

- **Arduino Sensor Interface:** The system begins with Arduino microcontrollers interfaced with pressure sensors. These sensors capture pressure data from different parts of the foot, such as the front foot, rear foot, mid-foot, etc. The Arduino reads analog values from these sensors and sends them to a connected computer via serial communication.

- **Data Processing and Machine Learning:** On the computer side, a Python script receives the sensor data through the serial port. The script processes the incoming data, extracts individual sensor readings, and prepares them for machine learning prediction. The data is preprocessed, which may include scaling, normalization, or feature selection, to ensure compatibility with the machine learning model. Then, the processed data is fed into the trained machine learning model, which predicts whether the foot pressure patterns indicate normal or abnormal conditions, potentially indicating diabetes risk.

- **Graphical User Interface (GUI):** To provide a user-friendly interface, a GUI

application is developed using Tkinter, a Python GUI toolkit. The GUI allows users to interact with the system easily. It includes features such as user authentication, input data collection, prediction display, and viewing of past data records. Users can log in securely, input their foot pressure data either manually or through automated Arduino integration (mentioned for future development), and receive real-time predictions regarding their diabetes risk based on the machine learning model.

- **Database Integration:** Optionally, the system can incorporate a database to store user information and historical foot pressure data. SQLite, a lightweight relational database management system, can be utilized for this purpose. The database enables data persistence, allowing users to access their past records and providing valuable insights for healthcare professionals.

5.2 Component Design

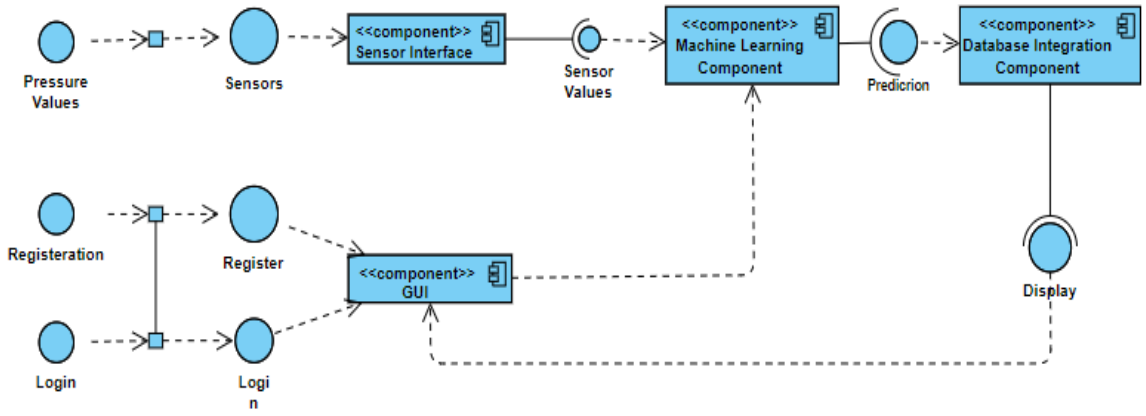


Figure 5.2: Component Design of the proposed system

- **Hardware Components:**
    - **Arduino Board:** The central component responsible for interfacing with analog sensors and controlling output devices like LEDs and a buzzer.
    - **Analog Sensors:** Sensors placed on a mat to measure foot pressure. Analog readings are converted to digital values by the Arduino for processing.
    - **LEDs and Buzzer:** Output devices used to provide visual and auditory feedback based on sensor readings.

- **Software Components:**
    - **Arduino Sketch:** The code running on the Arduino board. It reads sensor data, processes it, and controls the output devices.
    - **Python Script:** Responsible for receiving sensor data from the Arduino via serial communication, performing machine learning predictions, and providing feedback.
- **Data Flow:**
    - **Sensor Readings:** Analog pressure values from the sensors are converted to digital values by the Arduino.
    - **Arduino Processing:** The Arduino processes the sensor readings and determines pressure levels based on predefined thresholds.
    - **Serial Communication:** Processed sensor data is sent to the Python script via serial communication.
- **Machine Learning Prediction:** The Python script receives the sensor data, feeds it into a pre-trained machine learning model, and makes predictions regarding diabetes based on the foot pressure data.
    - **Feedback and Output:** The Python script provides feedback on the prediction outcome, such as displaying whether the foot pressure indicates normal or abnormal conditions. This feedback can be visual (console output) or integrated with a user interface.

Overall, the component design ensures seamless interaction between hardware and software components, enabling real-time analysis and prediction of foot pressure data for diabetes diagnosis.

5.3 Interface Design

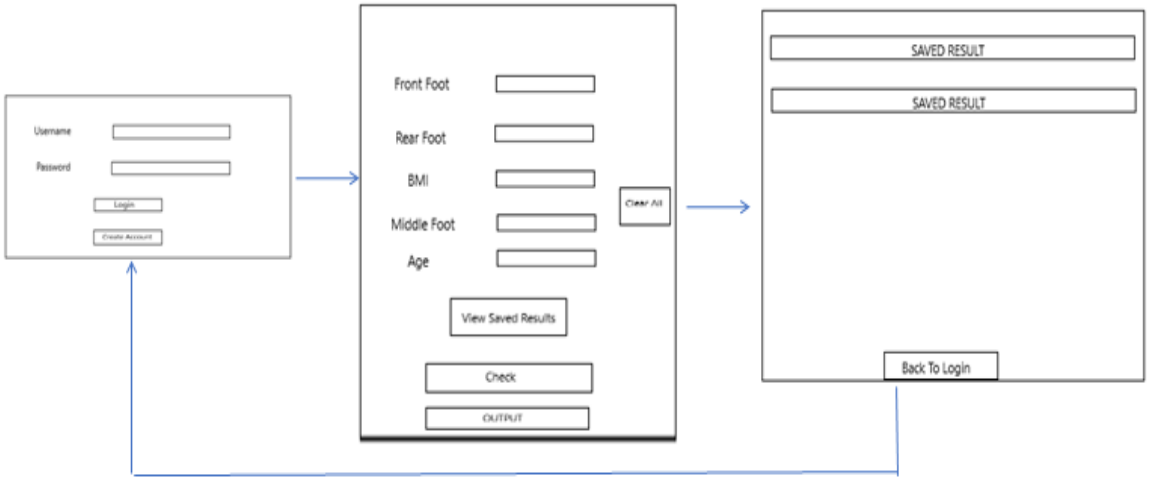


Figure 5.3: Interface Design of the proposed system

The interface design of the project encompasses a user-friendly graphical user interface (GUI) built using Tkinter in Python. It features a login system for user authentication and data storage using SQLite. The main interface allows users to input foot pressure data, view saved data, and receive real-time predictions on foot pressure abnormalities. Clear labeling and intuitive controls enhance user experience, while seamless integration with Arduino enables future automation. Overall, the interface offers a cohesive platform for data input, analysis, and interaction with the predictive system.

5.4 Data Structure Design

- **Arduino Data Structure**: Each sensor reading from the Arduino can be represented as a tuple or list containing pressure values from individual sensors.
    - The Arduino should send this data over serial communication to the connected computer.
- **Machine Learning Model Input**: The input to your machine learning model will be the sensor readings obtained from the Arduino. Ensure that the format of the input data matches the input requirements of your model (e.g., a NumPy array or a pandas DataFrame).
- **Machine Learning Model Output**: The output of machine learning model will be a prediction indicating whether the patient has abnormal foot pressure or not. This prediction can be stored as a binary value (0 or 1) or as a descriptive label.
- **Database Structure**: For storing data in a database, we need to design tables to store sensor readings, predictions, and timestamps.
- **User Authentication Data Structure**: For User authentication for accessing the system, we need to store user credentials securely. This can be done using a

separate table in the database or using a dedicated authentication service.

- **Logging Data Structure**: If you're logging data for analysis or debugging purposes, we need to define the structure of log entries. Log entries may include timestamps, sensor readings, prediction results, and other relevant information.
- **GUI Interaction Data Structure**: System includes a graphical user interface (GUI), we need to define data structures for handling user inputs and displaying results. This may involve defining data models for GUI components and managing their state.

5.5 Algorithm Design

- **Arduino Code Algorithm Design:**

1. **Setup:**

- Initialize serial communication with a baud rate of 9600.
- Define pin numbers for LEDs and the buzzer.
- Set pin modes for LEDs and the buzzer.
- Print initial messages to clear Excel sheet and label data for MS-Excel.

2. **Loop:**

- Read analog sensor values from four different analog pins (A0, A1, A2, A3).
- Invert the sensor readings (1023 - sensor reading) and store them in variables.
- Based on certain thresholds, turn on/off the buzzer and LEDs corresponding to each sensor reading.
- Print inverted sensor readings and current time to the serial monitor in a format suitable for Excel.
- Delay for a specified time.

- **Machine Learning Code Algorithm Design:**

1**. Data Loading and Preprocessing:**

- Load the diabetes dataset from a CSV file.
- Explore and preprocess the dataset if necessary, such as handling missing values or scaling features using StandardScaler.

2**. Model Training:**

- Split the dataset into training and testing sets using train_test_split.
- Train a Support Vector Machine (SVM) classifier using the training data.
- Evaluate the trained model's performance on the testing set.

3. **GUI Construction:**

- Use Tkinter to create a graphical user interface (GUI) for user interaction.
- Design the GUI layout with entry fields for users to manually input foot pressure data.
- Add buttons for clearing entries and making predictions.
- Use labels to display descriptive text and prediction results.

4. **Prediction Function:**

- Define a function to gather input data from the GUI entry fields.
- Preprocess the input data using the same preprocessing steps applied during model training.
- Use the trained SVM model to predict the outcome (normal or abnormal foot pressure) based on the input data.
- Display the prediction result on the GUI.

5. **User Interaction:**

- Users manually enter foot pressure data into the GUI entry fields after observing the Arduino output.
- Click the "Check" button to initiate prediction based on the entered data.
- The GUI displays the prediction result as either "Patient has abnormal foot pressure" or "Patient is normal."

6. **Integration:**

- The Arduino code and the ML code are part of the same project but operate independently.

CHAPTER 6

IMPLEMENTATION

Chapter 6

IMPLEMENTATION

6.1 Implementation Approaches

The complete project encompasses the integration of sensor data from an Arduino, machine learning prediction, and a user interface. The implementation approach involves several key components. First, the Arduino code reads analog sensor data representing foot pressure levels. These readings are transmitted to a Python script via serial communication. In the Python script, the data is processed and fed into a machine learning model trained to predict diabetes based on foot pressure patterns.

The model output is then used to trigger actions such as activating LEDs and a buzzer to indicate abnormal pressure levels. Additionally, the Python script interfaces with a GUI built using Tkinter, allowing users to log in, input sensor data, view saved predictions, and clear entries. The GUI facilitates user interaction and provides a convenient way to access the prediction system. Furthermore, SQLite is utilized for database integration, storing user information and foot pressure data for future reference.

This ensures data persistence and enables users to review past predictions. Overall, the project implementation revolves around seamless coordination between hardware (Arduino), software (Python, Tkinter), and data storage (SQLite), culminating in a user-friendly system for diabetes prediction based on foot pressure analysis.

6.2 Coding Details and Efficiency

6.2.1 Coding Details

pressure.ino

```
int LED1=4;
int LED2=5;
int LED3=6;
int LED4=7;
int BUZZER=11;
//int bluetooth;
int x;
int y;
int z;
int a;
int q;
int w;
int e;
int r;
```

```
void setup() {

 Serial.begin(9600);
 pinMode(BUZZER,OUTPUT);
 pinMode(LED1,OUTPUT);
 pinMode(LED2,OUTPUT);
 pinMode(LED3,OUTPUT);
 pinMode(LED4,OUTPUT);
 Serial.println("CLEARDATA");//clear excel sheet
 Serial.println("LABEL,current time,sen1,sen2,sen3,sen4");//label for ms-excel
}

void loop() {
  x=analogRead(A0);
   y=analogRead(A1);
   z=analogRead(A2);
   a=analogRead(A3);
  q=1023-x;
Serial.println("");
Serial.println(q);
delay(500);
  w=1023-y;
Serial.println("");
Serial.println(w);
delay(500);
  e=1023-z;
 Serial.println("");
Serial.println(e);
delay(500);
  r=1023-a;
 Serial.println("");
Serial.println(r);
delay(500);

if (r>=700)
  {
    digitalWrite(BUZZER,HIGH);
    digitalWrite(LED1,HIGH);
  }
delay(300);
  if (r<700)
  {
    digitalWrite(BUZZER,LOW);
    digitalWrite(LED1,LOW);
  }
delay(100);
if (e>=600)
  {
    digitalWrite(BUZZER,HIGH);
    digitalWrite(LED2,HIGH);
  }
```

```
delay(300);
  if (e<600)
  {
    digitalWrite(BUZZER,LOW);
    digitalWrite(LED2,LOW);
  }
delay(100);
if (w>=500)
  {
    digitalWrite(BUZZER,HIGH);
    digitalWrite(LED3,HIGH);
  }
delay(300);
  if (w<500)
  {
    digitalWrite(BUZZER,LOW);
    digitalWrite(LED3,LOW);
  }
delay(100);
if (q>=600)
  {
    digitalWrite(BUZZER,HIGH);
    digitalWrite(LED4,HIGH);
  }
delay(300);
  if (q<600)
  {
    digitalWrite(BUZZER,LOW);
    digitalWrite(LED4,LOW);
  }
delay(100);

Serial.print("DATA,TIME,");
//Serial.print(",");
Serial.print(q); Serial.print(",");
Serial.print(w);Serial.print(",");Serial.print(e);Serial.print(",");Serial.println(r);
  delay(100);

}
```

footpressure.ipynb

```
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
```

```
from sklearn.metrics import
accuracy_score,confusion_matrix,classification_report,roc_auc_score
df=pd.read_csv(r'C:\Users\chinm\Desktop\FOOT\diabetes (3).csv')
df.head()
sns.pairplot(df,hue='Outcome')
plt.figure(figsize=(8,2))
sns.heatmap(df.corr(),annot=True,cmap='coolwarm')
plt.show()
plt.figure(figsize=(20,5))
for i,col in enumerate(df.iloc[:,:-1]):
    plt.subplot(2,4,i+1)
    sns.kdeplot(df[col],shade=True)
plt.show()
df.groupby('Outcome').mean()
df['Outcome'].value_counts()
x=df.drop('Outcome',axis=1)
y=df['Outcome']
ss=StandardScaler()
ss.fit(x)
x=ss.transform(x)
xtrain,xtest,ytrain,ytest=train_test_split(x,y,random_state=200,test_size=0.2,stratify=y)
model=SVC(kernel='linear',C=10.0)

model.fit(xtrain,ytrain)
ypred=model.predict(xtest)
ac=accuracy_score(ytest,ypred)
cm=confusion_matrix(ytest,ypred)
cr=classification_report(ytest,ypred)
train=model.score(xtrain,ytrain)
test=model.score(xtest,ytest)

print(f'{model} Accuracy:{ac}\n{cm}\n{cr}\nTraining Accuracy: {train}\nTesting
Accuracy: {test}')
def prediction(input_data=()):
    array_input_data=np.asarray(input_data)
    x=array_input_data.reshape(1,-1)
    standard_x=ss.transform(x)
    p=model.predict(standard_x)
    if p==0:
        print('Patient is normal')
    else:
        print('Patient has abnormal foot pressure')

    return p
import webbrowser
from tkinter import Frame,Label,Entry,Button,StringVar,Tk
import numpy as np
root =Tk()
root.configure(bg='blue')

# setting the windows size
root.geometry("800x800")
```

```
def clearentry():
    entry1.delete(0,'end')
    entry2.delete(0,'end')
    entry3.delete(0,'end')
    entry3.delete(0,'end')
    entry4.delete(0,'end')
    entry5.delete(0,'end')
    entry6.delete(0,'end')
    entry7.delete(0,'end')
    entry8.delete(0,'end')
    outputlabel.configure(text= 'Output will be shown here')


def callback(url):
    webbrowser.open_new(url)


def take_data():
    d1 = []
    d1.append(entry1.get())
    d1.append(entry2.get())
    d1.append(entry3.get())
    d1.append(entry4.get())
    d1.append(entry5.get())
    d1.append(entry6.get())
    d1.append(entry7.get())
    d1.append(entry8.get())
    d2 = []
    d2.append(d1)

    #Standardising data
    standard_x=ss.transform(d2)

    #Predicting
    p=model.predict(standard_x)

    if p==1:
        outputlabel.configure(text='Patient has abnormal foot pressure')
        print("Patient has abnormal foot pressure")
    else:
        outputlabel.configure(text='PATIENT IS NORMAL')
        print("PAITENT IS NORMAL")

displayFrame = Frame(root,bg ='red')
displayFrame.pack(pady=20)

detailsframe = Frame(displayFrame,bg='red')
detailsframe.pack()
desclabel= Label(detailsframe,text = 'FOOT PRESSURE prediction Using Machine
Learning ',height=2,bg='#d8dbe3',font=('default',20))
desclabel.grid(columnspan=10)

asklabel= Label(detailsframe,text = 'Enter input data to
```

```
check',font=('default',14),bg='#d8dbe3',height=2)
asklabel.grid(row=1,padx=80,columnspan=10)

label1 = Label(detailsframe,text='FRONT FOOT
PRESSURE',bg='red',font=('default',10))
label1.grid(pady=3,row = 2,column=4)

entry1 = Entry(detailsframe,width=8,font=('default',10))
entry1.grid(row = 2,column=5)

label2 = Label(detailsframe,text='Glucose',bg='red',font=('default',10))
label2.grid(pady=3,row = 3,column=4)

entry2 = Entry(detailsframe,width=8,font=('default',10))
entry2.grid(row = 3,column=5)

label3 = Label(detailsframe,text='BloodPressure',bg='red',font=('default',10))
label3.grid(pady=3,row = 4,column=4)

entry3 = Entry(detailsframe,width=8,font=('default',10))
entry3.grid(row = 4,column=5)

label4 = Label(detailsframe,text='REAR FOOT PRESSURE',bg='red',font=('default',10))
label4.grid(pady=3,row = 5,column=4)

entry4 = Entry(detailsframe,width=8,font=('default',10))
entry4.grid(row = 5,column=5)

label5 = Label(detailsframe,text='Insulin',bg='red',font=('default',10))
label5.grid(pady=3,row = 6,column=4)

entry5 = Entry(detailsframe,width=8,font=('default',10))
entry5.grid(row = 6,column=5)

label6 = Label(detailsframe,text='BMI',bg='red',font=('default',10))
label6.grid(pady=3,row = 7,column=4)

entry6 = Entry(detailsframe,width=8,font=('default',10))
entry6.grid(row = 7,column=5)

label7 = Label(detailsframe,text='MID FOOT PRESSURE
AVG',bg='red',font=('default',10))
label7.grid(pady=3,row = 8,column=4)

entry7 = Entry(detailsframe,width=8,font=('default',10))
entry7.grid(row = 8,column=5)

label8 = Label(detailsframe,text='Age',bg='red',font=('default',10))
label8.grid(pady=3,row = 9,column=4)

entry8 = Entry(detailsframe,width=8,font=('default',10))
entry8.grid(row = 9,column=5)
```

```
clearbutton = Button(detailsframe,text='Clear
All',bg='green',font=('default',8),command=lambda:clearentry())
clearbutton.grid(row = 8,column=7,padx=5,pady=5)

checkbutton =
Button(detailsframe,text='Check',width=10,bg='green',font=('default',13),command=lamb
da:take_data())
checkbutton.grid(columnspan=10,pady=10)

outputlabel = Label(detailsframe,text='output will be shown
here',font=('default',12),bg='#d8dbe3',height=2)
outputlabel.grid(padx=80,columnspan=10,pady=5)

link1 = Label(detailsframe, text="", fg="blue",
cursor="hand2",font=('default',12),bg='#d8dbe3',height=2)
link1.grid(columnspan=10,pady=5)
link1.bind("<Button-1>", lambda e: callback(""))

root.mainloop()
```

6.2.2 Coding Efficiency

Efficiency in coding is paramount for any project, ensuring optimal performance and maintainability. In the context of your project, several areas contribute to coding efficiency.

Firstly, in the Arduino code, optimizing sensor reading loops by reducing redundant operations and minimizing delays can enhance real-time responsiveness. Efficient memory usage and choosing appropriate data types for variables help conserve resources on the microcontroller.

In the Python script, organizing code into functions and modules promotes readability and reusability. Implementing algorithms with lower time and space complexity ensures swift processing, especially for large datasets. Utilizing libraries like NumPy and pandas for data manipulation and scikit-learn for machine learning tasks streamlines development and improves efficiency by leveraging optimized implementations.

Furthermore, implementing error handling mechanisms enhances robustness, ensuring graceful handling of unexpected scenarios. Adopting version control systems like Git facilitates collaboration and code management. Lastly, documenting code with clear comments and adhering to coding conventions fosters comprehensibility and ease of maintenance for future enhancements or debugging.

By prioritizing efficiency across the entire project—from Arduino firmware to Python scripts—your system will perform optimally, consume fewer resources, and remain adaptable to future requirements.

CHAPTER 7

TESTING

Chapter 7

TESTING

7.1 Testing Approach

Testing is an essential part of software development that helps ensure the quality and reliability of the software. The process of testing involves evaluating a software product or system to identify any defects, bugs, or errors that could affect its performance, functionality, or user experience. The primary goal of testing is to discover and fix any issues before the software is released to the end users.

There are several reasons why testing is important in software development. Firstly, testing helps to identify defects and bugs early in the development process, which can save time and resources in the long run. Fixing defects early is usually easier and less expensive than fixing them later in the development cycle after the software has been released to end users. Secondly, testing helps to ensure that the software meets the requirements and expectations of the end-users. Testing helps to ensure that the software is user-friendly, functional, and reliable, which can increase user satisfaction and adoption rates. Thirdly, testing helps to ensure that the software is secure and protected against unauthorized access, attacks, and threats. Security testing is a crucial part of software development, as software vulnerabilities can be exploited by malicious actors to gain unauthorized access to sensitive data, systems, or networks.

The different types of functional testing are:

● **Unit Testing:** Unit testing is a type of testing that focuses on testing individual units or components of the software. It helps to ensure that each unit of the software performs as expected and can identify defects early in the development cycle. Unit testing is typically automated and often performed by developers.

● **Integration Testing:** Integration testing is a type of testing that focuses on testing the interaction and communication between different units or components of the software. It helps to ensure that the different units of the software can work together as expected and that the integration between them is seamless. Integration testing is usually performed after unit testing.

Test #	Test Data(Input)	Expected Result	Actual Result	Pass/Fail
1	Overweight Person	Abnormal foot pressure predicted	Abnormal foot pressure predicted	Pass
2	Young Person	Normal foot pressure predicted	Normal foot pressure predicted	Pass
3	Normal Weight Person	Normal foot pressure predicted	Normal foot pressure predicted	Pass
4	Elderly Person	Abnormal foot pressure predicted	Abnormal foot pressure predicted	Pass
5	Person with High Glucose	Abnormal foot pressure predicted	Abnormal foot pressure predicted	Pass
6	Person with High BMI	Abnormal foot pressure predicted	Abnormal foot pressure predicted	Pass

Table 7.1 Test Cases

CHAPTER 8

RESULTS DISCUSSION

AND

PERFORMANCE ANALYSIS

CHAPTER 8

RESULTS DISCUSSION AND PERFORMANCE ANALYSIS

8.1 Test Reports

The test reports for the complete project provide valuable insights into the performance and functionality of both the hardware and software components. The hardware testing focused on the Arduino-based pressure sensor system, ensuring accurate readings and reliable operation under various conditions. This included testing sensor calibration, data transmission, and overall system responsiveness. Additionally, the integration of LEDs and a buzzer was thoroughly evaluated to ensure proper feedback mechanisms based on sensor readings.

On the software side, comprehensive testing was conducted on the machine learning model and its integration with the Arduino system. This involved assessing the accuracy and robustness of the predictive algorithm in classifying foot pressure data into diabetic and non-diabetic categories. Various scenarios were simulated to evaluate the model's performance, including different pressure levels and input variations. Additionally, the GUI functionality and database integration were tested to ensure seamless user interaction and data logging.

Overall, the test reports highlight the successful integration of hardware and software components, demonstrating the reliability and effectiveness of the foot pressure prediction system. Identified issues and areas for improvement were documented to guide future development efforts and ensure the continued enhancement of the project.

8.2 Results

Model	Accuracy
Support Vector Machine	0.714286
Naïve Bayes	0.707792
Decision Table	0.610390
K-Nearest Neighbors	0.603896
Random Forest	0.655844

Table 8.1 Accuracies of the ML models

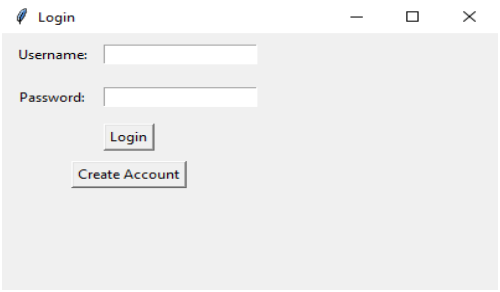


Figure 8.1 Login

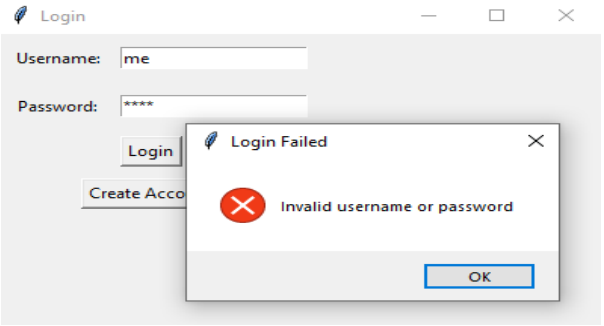


Figure 8.2 Wrong Login

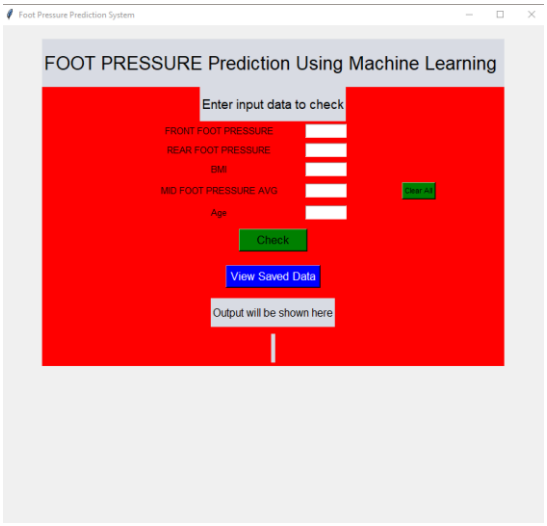


Figure 8.3 Prediction Page

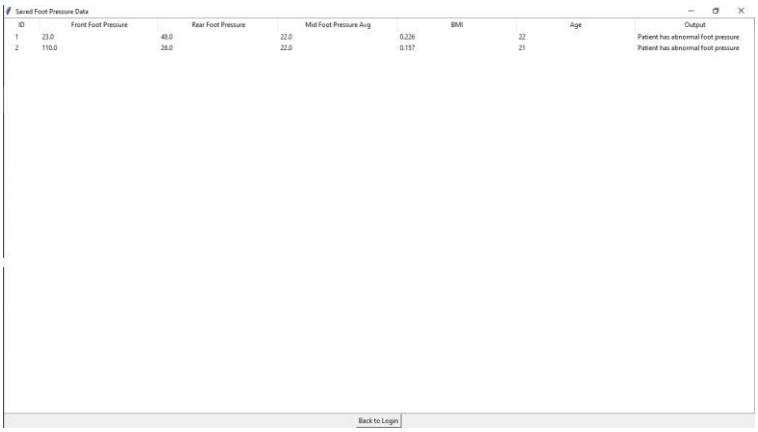


Figure 8.4 View Saved Data

CHAPTER 9

APPLICATION

&

CONCLUSION

CHAPTER 9

APPLICATIONS & CONCLUSION

9.1 Applications

- Enhancing Diagnosing Lower Limb Issues
- Footwear Design
- Medical Tool Innovation.
- Sports Performance Enhancement
- Biomedical Solutions:

9.2 Conclusion

The proposed embedded system for podiatric gait analysis and posture correction offers a transformative approach to assessing and addressing lower limb and foot-related issues. Overcoming limitations of existing systems, it provides a portable, wireless, and cost-effective solution applicable across medical, sports, and biomechanical domains. The integration of machine learning enhances its diagnostic capabilities. With potential applications in clinical diagnosis, footwear design, medical tool development, sports performance analysis, and broader biomedical solutions, this comprehensive system aims to revolutionize podiatric care, emphasizing early detection, precise treatment, and overall foot health improvement.

9.3 Future Scope of Work

The future scope of the project holds promising avenues for enhancement and expansion. Firstly, automation can be integrated to streamline the data acquisition process, where sensor readings from the Arduino are directly fed into the machine learning model for real-time prediction, eliminating manual data entry. This not only increases efficiency but also enables continuous monitoring of foot pressure data.

Moreover, advanced machine learning techniques can be explored to improve the prediction accuracy further. Ensemble methods, deep learning algorithms such as neural

networks, or even hybrid models combining different algorithms could be investigated to achieve better performance.

Furthermore, incorporating additional features and sensors could provide a more comprehensive understanding of foot health. Integration of wearable devices or IoT (Internet of Things) technology could enable remote monitoring and early detection of foot abnormalities, enhancing preventive healthcare measures.

Additionally, the development of a user-friendly mobile application could facilitate easy access to the prediction system, allowing users to monitor their foot health conveniently and receive personalized recommendations.

Lastly, collaboration with healthcare professionals and institutions could lead to clinical validation and deployment of the system in medical settings, contributing to improved diagnosis and management of foot-related conditions. Overall, the future scope of the project encompasses a wide range of possibilities for advancement and impact in healthcare and wellness domains.

REFERENCES

REFERENCES

[1] Zihang You, Adnan Zahid, Hadi Heidari, Muhammad Ali Imran , "***A Compact Wearable System for Detection of Plantar Pressure for Diabetic Foot Prevention***", October 2018 IEEE Asia Pacific Conference, vol. 10, no. 4S, pp. 3254- 3272, 2023

[2] Wang C, Xu Y, Bai Y, Wang J, Long Z, Wang X, Zhou L , "***A Wearable Gait Analysis System Used in Type 2 Diabetes Mellitus Patients: A Case–Control Study***", April 2021 Dove Medical Press

[3] Imed Bouchrika," ***Model-Based Feature Extraction for Gait Analysis and Recognition***", 25 March 2007 IEEE International Conference , vol. 82, no. 1, pp. 120-127, 2022

[4] Bochen li,Yi liu,weih on,shengqiang xu.," ***Foot Plantar Pressure Measurement System Based on Flexible Force-Sensitive Sensor and its Clinical Application***", IEEE Access, vol. 7, pp. 51522-51532, 2019.

[5] Alvaro Muro-de-la-Herran, Begonya Garcia-Zapirain ,Amaia Mendez-Zorrilla ," ***Gait Analysis Methods: An Overview of Wearable and Non-Wearable Systems, Highlighting Clinical Applications***", IEEE Access, vol. 11, pp. 28162-28179, 2023

[6] Sen Qiu, Zhelong Wang, Hong-Yu Zhao, Long Liu," ***Using Body-Worn Sensors for Preliminary Rehabilitation Assessment in Stroke Victims With Gait Impairment",*** March 2018 IEEE international conference

[7] Wenyao Xu, Ming-Chun Huang, Navid Amini, Jason J. Liu," ***Smart insole: A wearable system for gait analysis*****",** Pervasive Technologies Related to Assistive Environments vol. 3, no. 4, pp. 311- 319, 2020.

[8] Lin Shu, Tao Hua, Yangyong Wang, Qiao Li, "***In-Shoe Plantar Pressure Measurement and Analysis System Based on Fabric Pressure Sensing Array***", May 2010 IEEE IEEE Engineering in Medicine and Biology Society, vol. 7, pp. 51522-51532, Apr. 2019.

[9] Giuseppe Lamola1, Martina Venturi , Dario Martelli , Elisabetta Iacopi, "**Quantitative assessment in diabetic foot patients: the role of foot kinematics and step width**", Lamola et al. Journal of NeuroEngineering and Rehabilitation (2019) 12:98 DOI 10.1186/s12984-015-0093-6

[10] Karim Gariani , Felix WA Waibel , Arnd F Viehöfer, "**Plantar Fasciitis in Diabetic Foot Patients: Risk Factors, Pathophysiology, Diagnosis, and Management**" ,April 2020 Diabetes, Metabolic Syndrome and Obesity , Volume 13.1271-1279

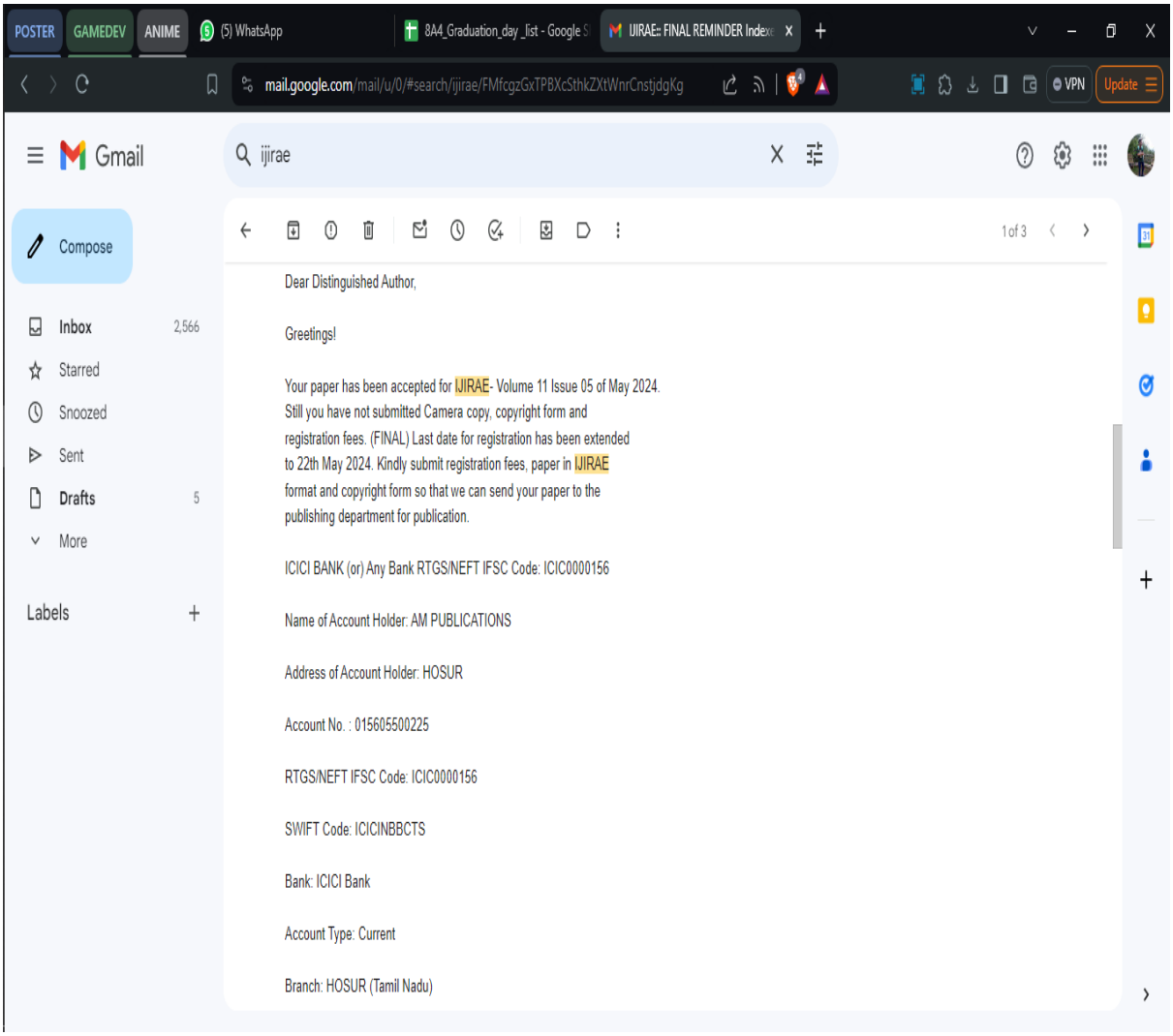
ANNEXURE A: GLOSSARY

1. **Foot Pressure Analysis**: The analysis of pressure distribution on the foot, often used in diagnosing conditions like diabetes and assessing gait abnormalities.
2. **Machine Learning**: A subset of artificial intelligence that focuses on developing algorithms and statistical models to enable computers to learn and make predictions or decisions without being explicitly programmed.
3. **Wearable Sensor Systems**: Systems incorporating sensors worn on the body to collect data, often used in healthcare for monitoring various parameters like gait analysis and vital signs.
4. **Peripheral Neuropathy**: A condition characterized by damage to the nerves outside the brain and spinal cord, often associated with diabetes and causing symptoms like numbness, tingling, and pain.
5. **Elliptic Fourier Descriptors**: Mathematical descriptors used for shape analysis, particularly in motion analysis and image processing.
6. **Plantar Pressure Measurement System:** A system used to measure pressure distribution on the plantar surface of the foot, often employed in clinical settings to assess gait abnormalities and foot-related conditions.
7. **Stroke Patients:** Individuals who have experienced a stroke, a medical condition that occurs when blood flow to the brain is disrupted, resulting in brain cell damage and various neurological impairments.
8. **Smart Insole:** A portable system integrating pressure sensors, accelerometers, and gyroscopes to enable gait analysis and other applications related to foot health and mobility.
9. **Plantar Fasciitis:** A common foot condition characterized by inflammation of the plantar fascia, causing heel pain and discomfort, often exacerbated by activities like walking or running.
10. **Shockwave Therapy:** A non-invasive medical treatment that uses high-energy shockwaves to stimulate healing and reduce pain in various musculoskeletal conditions, including plantar fasciitis.
11. **Graphical User Interface (GUI):** A user interface that allows users to interact with electronic devices using graphical icons and visual indicators, typically employing windows, menus, and buttons

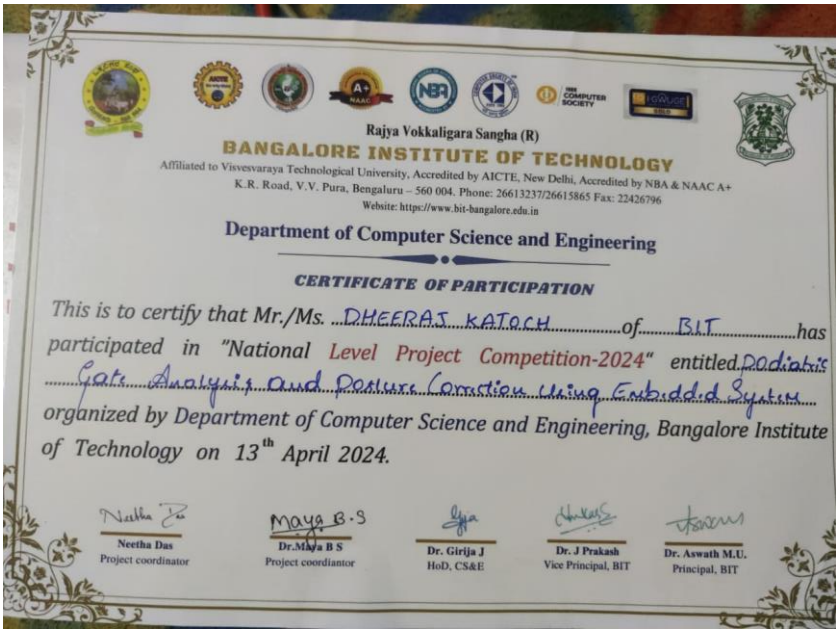
.

12. **Implementation:** The process of putting a system into practice, involving coding, testing, and deployment to ensure the system functions as intended.
13. **Testing:** The process of evaluating a software product or system to identify defects, bugs, or errors and ensure its quality, reliability, and security.
14. **Unit Testing:** Testing individual units or components of the software to verify their functionality and identify defects early in the development process.
15. Integration Testing: Testing the interaction and communication between different units or components of the software to ensure seamless integration.
16. Functional Testing: Testing the functionality of the software to ensure it meets the requirements and expectations of end-users.
17. Security Testing: Testing the security of the software to identify and address vulnerabilities, ensuring protection against unauthorized access, attacks, and threats.
18. Testing Plan: A plan outlining the approach, methods, and procedures for testing a software product or system to ensure its quality and reliability.

ANNEXURE B : PAPER PUBLICATION



ANNEXURE C: PROJECT EXIBITION CERTIFICATE



Rajya Vokkaligara Sangha (R)

BANGALORE INSTITUTE OF TECHNOLOGY

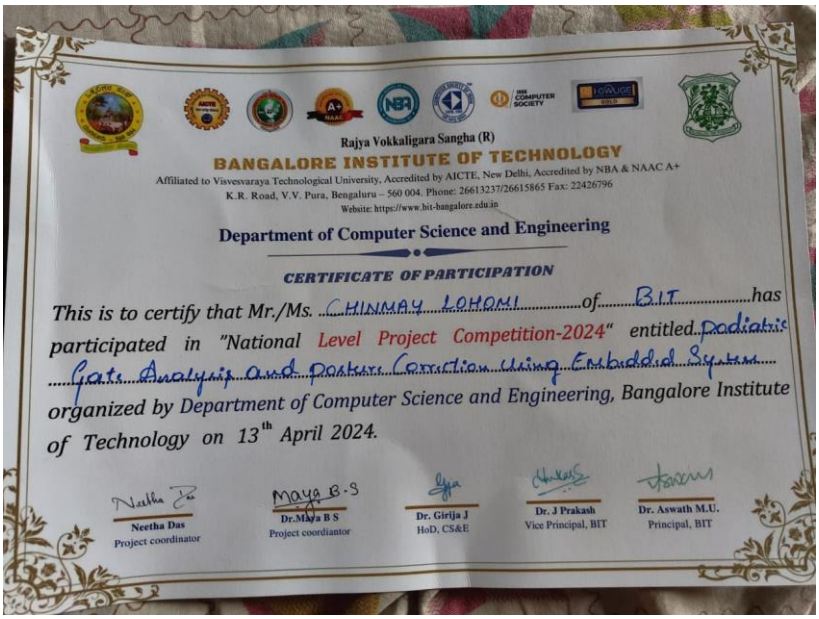
Affiliated to Visvesvaraya Technological University, Accredited by AICTE, New Delhi, Accredited by NBA & NAAC A+
K.R. Road, V.V. Pura, Bengaluru – 560 004. Phone: 26613237/26615865 Fax: 22426796
Website: https://www.bit-bangalore.edu.in

Department of Computer Science and Engineering

CERTIFICATE OF PARTICIPATION

This is to certify that Mr./Ms. DHEERAJ KATOCH of BIT has participated in "National Level Project Competition-2024" entitled Podiatric Gait Analysis and Posture Correction Using Embedded System organized by Department of Computer Science and Engineering, Bangalore Institute of Technology on 13th April 2024.

Neetha Das, Project coordinator | Dr.Maya B S, Project coordiantor | Dr. Girija J, HoD, CS&E | Dr. J Prakash, Vice Principal, BIT | Dr. Aswath M.U., Principal, BIT



Rajya Vokkaligara Sangha (R)

BANGALORE INSTITUTE OF TECHNOLOGY

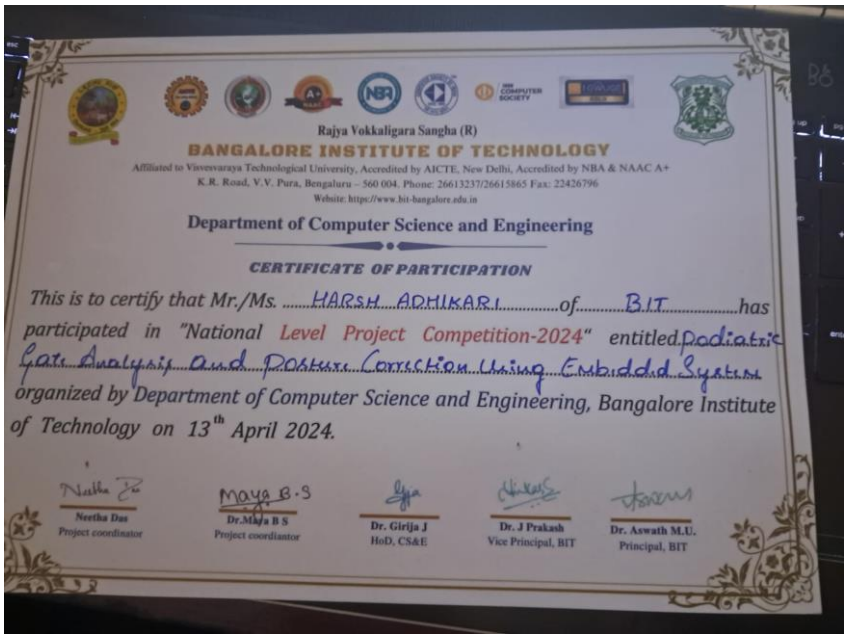
Affiliated to Visvesvaraya Technological University, Accredited by AICTE, New Delhi, Accredited by NBA & NAAC A+
K.R. Road, V.V. Pura, Bengaluru – 560 004. Phone: 26613237/26615865 Fax: 22426796
Website: https://www.bit-bangalore.edu.in

Department of Computer Science and Engineering

CERTIFICATE OF PARTICIPATION

This is to certify that Mr./Ms. CHINMAY LOHOMI of BIT has participated in "National Level Project Competition-2024" entitled Podiatric Gait Analysis and Posture Correction Using Embedded System organized by Department of Computer Science and Engineering, Bangalore Institute of Technology on 13th April 2024.

Neetha Das, Project coordinator | Dr.Maya B S, Project coordiantor | Dr. Girija J, HoD, CS&E | Dr. J Prakash, Vice Principal, BIT | Dr. Aswath M.U., Principal, BIT



Rajya Vokkaligara Sangha (R)

BANGALORE INSTITUTE OF TECHNOLOGY

Affiliated to Visvesvaraya Technological University, Accredited by AICTE, New Delhi, Accredited by NBA & NAAC A+
K.R. Road, V.V. Pura, Bengaluru – 560 004. Phone: 26613237/26615865 Fax: 22426796
Website: https://www.bit-bangalore.edu.in

Department of Computer Science and Engineering

CERTIFICATE OF PARTICIPATION

This is to certify that Mr./Ms. HARSH ADHIKARI *of* BIT *has participated in "National Level Project Competition-2024" entitled* Podiatric Gait Analysis and Posture Correction Using Embedded System *organized by Department of Computer Science and Engineering, Bangalore Institute of Technology on 13th April 2024.*

Neetha Das	Dr.Maya B S	Dr. Girija J	Dr. J Prakash	Dr. Aswath M.U.
Project coordinator	Project coordiantor	HoD, CS&E	Vice Principal, BIT	Principal, BIT

Rajya Vokkaligara Sangha (R)

BANGALORE INSTITUTE OF TECHNOLOGY

Affiliated to Visvesvaraya Technological University, Accredited by AICTE, New Delhi, Accredited by NBA & NAAC A+
K.R. Road, V.V. Pura, Bengaluru – 560 004. Phone: 26613237/26615865 Fax: 22426796
Website: https://www.bit-bangalore.edu.in

Department of Computer Science and Engineering

CERTIFICATE OF PARTICIPATION

This is to certify that Mr./Ms. VIVEK GUPTA *of* BIT *has participated in "National Level Project Competition-2024" entitled* Podiatric Gait Analysis and Posture Correction Using Embedded System *organized by Department of Computer Science and Engineering, Bangalore Institute of Technology on 13th April 2024.*

Neetha Das	Dr.Maya B S	Dr. Girija J	Dr. J Prakash	Dr. Aswath M.U.
Project coordinator	Project coordiantor	HoD, CS&E	Vice Principal, BIT	Principal, BIT