PREDICTION OF HEALTH USING WEARABLE DEVICES WITH MACHINE LEARNING TECHNIQUES

A PROJECT REPORT (PHASE-2)

by

AASHUTHOSH S (VJC20CS001)
ANANDHU S (VJC20CS020)
CRISTIN SILJO (VJC20CS040)
GEORGE GEO (VJC20CS054)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY, VAZHAKULAM

MAY 2024

PREDICTION OF HEALTH USING WEARABLE DEVICES WITH MACHINE LEARNING TECHNIQUES

A PROJECT REPORT (PHASE-2)

by

AASHUTHOSH S (VJC20CS001)
ANANDHU S (VJC20CS020)
CRISTIN SILJO (VJC20CS040)
GEORGE GEO (VJC20CS054)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

under the guidance

of

Ms.Ierin Babu

Assistant Professor, CSE Dept.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY, VAZHAKULAM

MAY 2024

VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY, VAZHAKULAM

Department of Computer Science and Engineering

Vision

Moulding socially responsible and professionally competent Computer Engineers to adapt to the dynamic technological landscape

Mission

1. Foster the principles and practices of computer science to empower life-long learning and build careers in software and hardware development.
2. Impart value education to elevate students to be successful, ethical and effective problem-solvers to serve the needs of the industry, government, society and the scientific community.
3. Promote industry interaction to pursue new technologies in Computer Science and provide excellent infrastructure to engage faculty and students in scholarly research activities.

Program Educational Objectives

Our Graduates

1. Shall have creative aid critical reasoning skills to solve technical problems ethically and responsibly to serve the society.
2. Shall have competency to collaborate as a team member and team leader to address social, technical and engineering challenges.
3. Shall have ability to contribute to the development of the next generation of information technology either through innovative research or through practice in a corporate setting
4. Shall have potential to build start-up companies with the foundations, knowledge and experience they acquired from undergraduate education

Program Outcomes

1. **Engineering knowledge**:Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**:Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design / development of solutions**:Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems**:Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**:Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:**Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:**Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:**Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice
9. **Individual and team work:**Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
10. **Communication:**Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and unread in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**:Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

1. Ability to integrate theory and practice to construct software systems of varying complexity
2. Able to Apply Computer Science skills, tools and mathematical techniques to analyse, design and model complex systems
3. Ability to design and manage small-scale projects to develop a career in a related industry.

VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY, VAZHAKULAM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the project report (phase-2) entitled **"PREDICTION OF HEALTH USING WEARABLE DEVICES WITH MACHINE LEARNING TECHNIQUES"** is a bonafide record of the work done by **AASHUTHOSH S(VJC20CS001), ANANDHU S(VJC20CS020), CRISTIN SILJO(VJC20CS040), GEORGE GEO(VJC20CS054)** in partial fulfillment of the requirements for the award of the **Degree of Bachelor of Technology** in **Computer Science and Engineering** of APJ Abdul Kalam Technological University.

Internal Supervisor **External Supervisor**

Project Coordinator **Head of Department**

ACKNOWLEDGEMENT

First and foremost, I thank **God Almighty** for His divine grace and blessings in making all these possible. May He continue to lead us in the years. It is my privilege to render our heartfelt thanks to our beloved Manager, **Rev. Msgr. Dr. Pius Malekandathil** , our Director **Rev. Dr. Paul Parathazham** and our Principal **Dr. K K Rajan** for providing us the opportunity to do this project report (phase-2) during the Fourth year (2024) of my B.Tech degree course. I am deeply thankful to my Head of the Department, **Dr. Amel Austine** for his support and encouragement. I would like to express my sincere gratitude and heartfelt thanks to our Project Guide **Ms.Ierin Babu**,Assistant Professor, Department of Computer Science and Engineering for her motivation, assistance and help for the project. I also express sincere thanks our Project Coordinator **Ms.Dona Jose**,Assistant Professor, Department of Computer Science and Engineering for her guidance and support. I also thank all the staff members of the Computer Science and Engineering Department for providing their assistance and support. Last, but not the least, I thank all my friends and family for their valuable feedback from time to time as well as their help and encouragement.

DECLARATION

I undersigned hereby declare that the project report "Prediction of Health Using Wearable Devices with Machine Learning Techniques", submitted for partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology of the APJ Abdul Kalam Technological University is a bonafide work done by myself under the supervision of Ms.Ierin Babu. This submission represents ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Vazhakulam

Date:

AASHUTHOSH S

ANANDHU S

CRISTIN SILJO

GEORGE GEO

ABSTRACT

Nowadays health related issues have started to increase and people have started to focus more on overall health. Busy-going people most of whom may not have time and proper resources to monitor their health and physical mode of checkups may not be always useful as hospitals may be busy with scheduled appointments. This is where our Health Monitoring app comes in use. The app includes features such as heart rate monitoring, temperature monitoring, and indication of blood oxygen level.It also includes recommendations for exercise type and food dietary plans and comments on the health status of the individual with prediction models using both machine learning and deep learning. It predicts potential health risks after scanning heart rate, SpO2, body temperature, and underlying symptoms if any improve overall health. In an era where time constraints and busy schedules often hinder traditional health checkups, The app emerges as a valuable tool for maintaining and enhancing one's health. With its user-friendly interface and data-driven insights, it encourages individuals to make informed choices and prioritize their health in the midst of their hectic lives. The app employs deep learning techniques MultiLayer Perceptron(MLP). As we continue to advance in the realm of health technology, the app serves as a promising and accessible solution for a healthier future.

Contents

List of Figures

List of Abbreviations

MLP	Multi-Layer Perceptron
RNN	Recurrent Neural Networks
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
IoT	Internet of Things
IoMT	Internet of Medical Things
ELM	Extreme Learning Machine
DL	Deep Learning
GC	Graph Clustering
NLTK	Natural Language Toolkit
NCD	Non-Communicable Disease

BP	Blood Pressure
CNN	Convolutional Neural Networks
NLP	Natural Language Processing
HR	Heart Rate
ML	Machine Learning
HCPS	Healthcare Cyber Physical Systems
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristic
ICD	International Classification Of Diseases
WHO	World Health Organization
MIMIC-III	Medical Information Mart for Intensive Care III)

EHR	Electronic Health Records
FM	Factorization Machine
HR	Hit Ratio
NDCG	Normalized Discounted Cumulative Gain
HPO	Human Phenotype Ontology
ICF	Item-based Collaborative Filtering
NAIS	Neural Attentive Item Similarity
FISM	Factored Item Similarity Model
HTML	HyperText Markup Language
CSS	Cascading StyleSheets
PHP	Perl Hypertext Processor
IDE	Integrated Development Environment

Chapter 1

INTRODUCTION

The convergence of wearable technology and Machine Learning(ML) holds immense potential for revolutionizing healthcare. Wearables, with their ability to continuously track various physiological and behavioral metrics, provide a rich source of data that can be harnessed to predict and manage health conditions. ML, on the other hand, offers powerful algorithms capable of identifying patterns and relationships within this data, leading to personalized insights and predictions about a user's health.

The given study delves into the exciting realm of predicting health using wearable devices and machine learning techniques. We will explore the various types of wearable sensors and the data they collect, delve into the capabilities of machine learning algorithms in analyzing this data, and examine the potential applications of this emerging technology in various healthcare domains. We will also discuss the challenges and limitations associated with this approach, and outline future directions for research and development.

As we embark on this journey, it is important to remember that the integration of wearables and machine learning for health prediction has the potential to democratize healthcare, empowering individuals to take proactive control of their health and well-being. By harnessing the power of these technologies, we can pave the way for a future of personalized, preventative, and predictive healthcare, ultimately leading to better health outcomes for all.

1.1 Problem Statement

Traditional healthcare systems primarily rely on episodic, in-person visits to healthcare providers. This approach fails to provide continuous real-time monitoring of an individual's health status, limiting early detection and prevention of health issues.

Access to healthcare services is unequal, with many individuals, especially in remote or underserved areas, facing challenges in obtaining regular check-ups and medical attention. This inequality results in health disparities and delayed diagnoses

Healthcare costs are soaring, and there is a need for cost-effective solutions that reduce the burden on healthcare systems.

1.2 Objective

1. To design an app which has a simple and efficient user interface.
2. To develop a system that provides continuous, real-time monitoring of an individual's health parameters through wearable devices and sensors.
3. To recommend proper and adequate dietary information to the users based on their health status.

1.3 Scope

1. Identifying and warning about potential health risks based on the health status of the user.
2. Optimized diet recommendation

Chapter 2

LITERATURE SURVEY

2.1 Realizing an Efficient IoMT-Assisted Patient Diet Recommendation System Through Machine Learning Model [1]

Patient/dietician recommendation systems are designed to assist users in choosing the best diets and foods for their specific health needs and preferences. These systems utilize a variety of methods to achieve this goal, including incorporating nutritional information from cloud systems and employing artificial intelligence algorithms. While current systems have shown promise in improving patient health outcomes, there are still areas for improvement, such as addressing the timing fragmentation of recommendations and modeling the nutritional composition of specific food items. This research aims to address these limitations by implementing machine learning models like Recurrent Neural Networks, Naive Bayes, and Long Term Short Memory to provide users with accurate and personalized food recommendations.

2.1.1 Implementation Details

The methodology employed in this study involves a comprehensive approach to data processing, feature importance identification, and classification utilizing both deep learning and Machine Learning(ML) techniques. The initial step encompasses data processing, where normalization and encoding techniques are applied to ensure uniformity and compatibility with ML algorithms. Feature importance is determined using the Random Forest Classifier, shedding light on key features within the dataset. The classification phase employs a combination of deep learning classifiers, including Multi-Layer Perceptron (MLP),Recurrent Neural Network(RNN),Long Short-Term Memory(LSTM) , and Gated Recurrent Unit(GRU). Additionally, ML classifiers such as Naive Bayes and Logistic Regression are applied, taking into account feature independence and probability calculations.The architecture of the study as shown in Figure 2.1 involves a meticulous process of data collection, where information from 30 patients is gathered using Internet of Things (IoT) and Cloud methods. The subsequent data processing step includes normalization for scaling data, encoding to convert nominal values into numeric format, and Optimal Feature Visualization to identify crucial features in the dataset. For data classification, Deep Learning methods such as MLP, RNN, LSTM, and GRU are employed to categorize the data. A comparative analysis of ML and Deep Learning models is conducted, and the study recommends the most effective model based on the evaluation metrics applied to all suggested models. This two-fold approach ensures a thorough exploration of both Deep Learning and traditional ML techniques for health prediction using wearable device data.

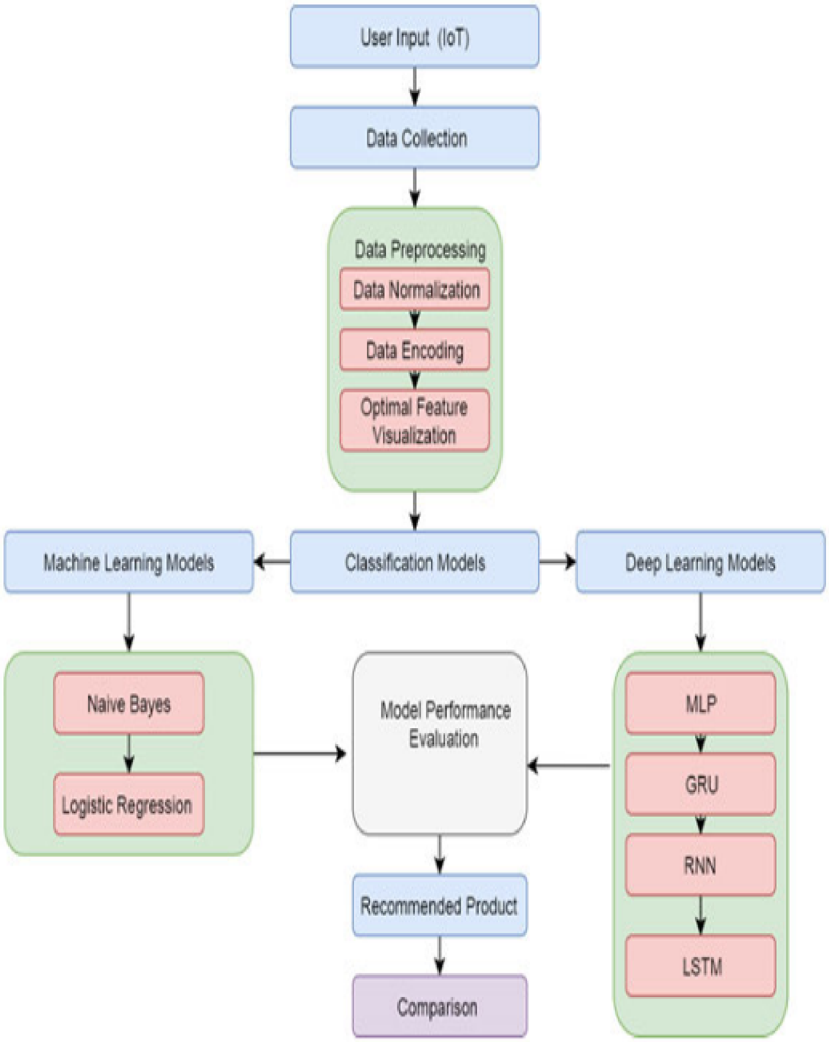


Figure 2.1: Architecture Diagram

2.1.2 Experimental Analysis

The dataset utilized in this research comprises approximately 1000 products and data collected from 30 patients through the Internet of Things (IoT) and cloud methods. A comprehensive set of experiments was conducted on a Core-i3 system with 8GB RAM, utilizing Google Colab with

13GB from Google Colab Laboratory. The dataset, consisting of 21 features and 16933 records, was divided into three sections: training set, cross-validation set, and testing set. The experimentation involved the use of K-Fold Cross Validation, with 70% of the dataset designated as the training set and the remaining 30% for testing purposes. Notably, various Machine Learning (ML) and deep learning classifiers, including MLP, RNN, LSTM, GRU, Naive Bayes, and Logistic Regression, were employed to assess training accuracies. The results indicated that LSTM achieved the highest training accuracy at 96.5%, outperforming other classifiers.

Moving on to testing accuracies, the deep learning classifiers demonstrated superior performance compared to machine learning models. LSTM attained an impressive testing accuracy of 97.74%, followed closely by GRU with 96.10%. In contrast, machine learning models such as Naive Bayes and Logistic Regression achieved testing accuracies of 93.96% and 93.80%, respectively. Further analysis of testing and validation scores for each classifier revealed the robust performance of LSTM and GRU in accurately predicting health outcomes. The precision, recall, and F1 measure scores emphasized LSTM's exceptional performance, particularly in distinguishing between allowed and not allowed classes. Overall, the study showcases the efficacy of employing deep learning techniques, especially LSTM, for health prediction based on wearable device data.

2.1.3 Advantages

1. Effective in handling linear relationships
2. Useful in Traditional Classification Problems
3. Enhanced Healthcare Management
4. Can automatically learn intricate data representations.
5. Early Detection of Health Issues

2.1.4 Disadvantages

1. Prone to Overfitting
2. Effective training requires large data
3. Demand significant computational resources
4. Complex Implementation

2.2 A Novel Time-Aware Food Recommender-System Based on Deep Learning and Graph Clustering [2]

The realm of food recommender systems has witnessed the development of TDLGC, a novel approach that addresses the limitations of existing systems. Time-aware food recommender-system based on Deep Learning and Graph Clustering(TDLGC) incorporates user similarity, food similarity, time factor, and community aspects to deliver personalized food recommendations. The system operates in two phases: user-based rating prediction and food-based rating prediction.

TDLGC, an advanced food recommender system, overcomes limitations by integrating user similarity, food similarity, time factors, and community aspects for personalized recommendations. In two phases, it predicts user-based ratings through community and similarity matrices and employs a Deep Learning Clustering Algorithm for food-based predictions. TDLGC uniquely combines collaborative and content-based models, aligning recommendations with user preferences and historical ratings. It introduces an innovative time-aware similarity measure for evolving user preferences. To address cold start challenges, TDLGC introduces a trust-aware food recommender, establishing a trust network based on follower-following statements. Its graphical representation, with edge weights from user ratings-based similarity and the trust network, efficiently manages sparse datasets.

2.2.1 Implementation Details

A. Methodology

This advanced food recommendation system combines graph clustering and deep learning for personalized suggestions. By analyzing ingredients and nutritional data, the system identifies groups of similar food items through graph clustering. At the same time, a Deep-Learning model discerns detailed connections between users and food items, learning from personal preferences and past interactions. The outcome is a dynamic recommendation system that goes beyond basic similarities, delivering users tailored suggestions based on nuanced connections within the data.

B. Architecture

Food recommendation system's effectiveness relies on a well-orchestrated process involving seamless data collection and preprocessing, recommendation generation, and evaluation. To kick-start the process, data is gathered meticulously from a variety of sources, such as extensive food databases, user profiles, and interactions. Subsequently, a rigorous preprocessing phase is implemented to guarantee the data's quality and consistency. This meticulous preparation lays the groundwork for subsequent stages, ensuring a solid foundation for the smooth operation of the recommendation system.

Transitioning to the recommendation generation phase, the system as depicted in Figure 2.2 adopts an advanced two-phase approach. This involves a comprehensive analysis of both the content of food items and the nuanced preferences of individual users. Through this process, the system crafts personalized recommendations, ensuring that users receive suggestions precisely tailored to their unique tastes and dietary inclinations. This personalized touch significantly enhances the overall user experience, offering recommendations that resonate seamlessly with each user's preferences.

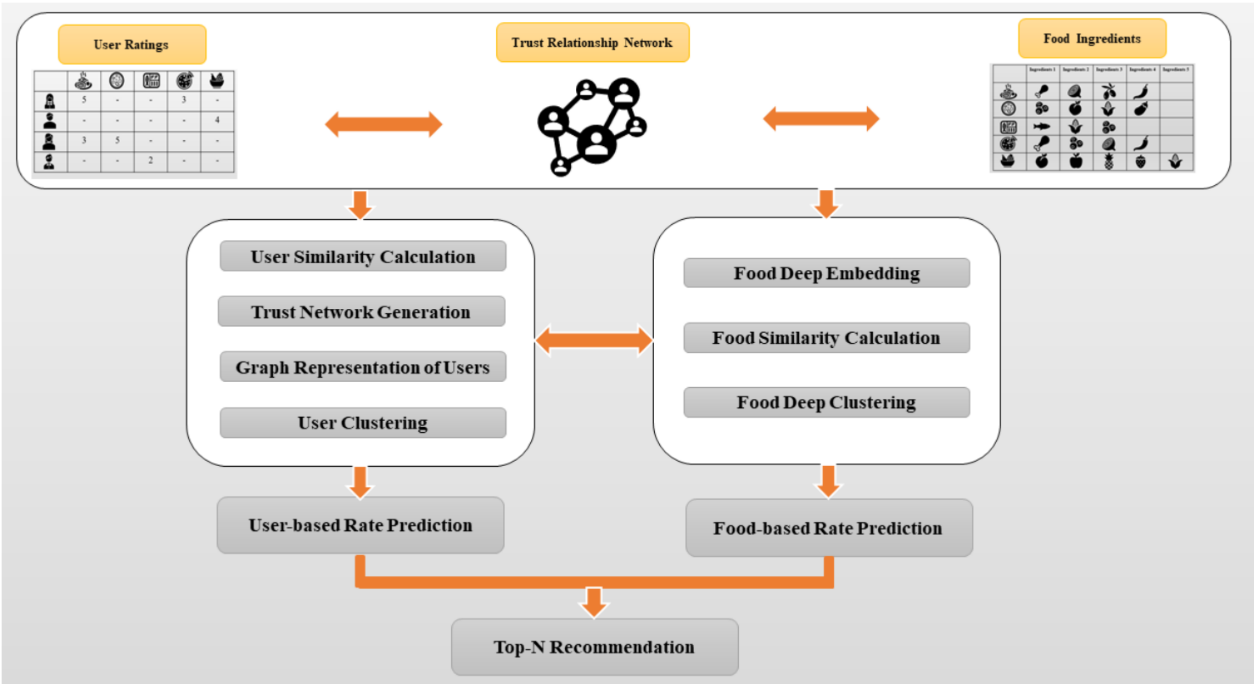


Figure 2.2: Conceptual framework of the developed model.

The recommendation evaluation component takes center stage. Various metrics, such as accuracy, precision, and recall, are meticulously employed to gauge the system's performance. This quantitative assessment not only validates the reliability of recommendations but also guides iterative improvements, ensuring that the food recommendation system continually evolves to meet the dynamic preferences and expectations of users.

C. Proposed Method

The authors propose a novel time-aware food recommender system called TDLGC, which integrates Deep Learning (DL) and Graph Clustering (GC) to consider user trust networks and historical user ratings. The system consists of two phases: user-based rating prediction and food-based rating prediction.

In the user-based rating prediction phase, user similarities and trust networks are generated using user ratings and follower-following networks. A novel time-aware graph clustering algorithm is proposed to cluster users based on user similarities, trust networks, and historical ratings. Finally, new user-based ratings are predicted using user clusters, user similarities, and historical ratings.

In the food-based rating prediction phase, food ingredients are embedded using a deep learning technique, and food similarities are assessed using embedding vectors. Finally, the rating of unseen foods is predicted using food similarities. After both phases, Top-N food recommendations are generated for the target user.

The proposed TDLGC system addresses the limitations of existing food recommender systems by considering user communities, user trust networks, and historical user ratings. The system's two-phase approach effectively captures user preferences and food similarities, leading to more accurate and personalized food recommendations.

2.2.2 Experimental Analysis

A. Dataset

The proposed food recommender system required a detailed dataset encompassing user-food interactions and ingredient information. Existing public food datasets lacked user-food rating data, necessitating a custom dataset creation. The authors collected user-food ratings and other relevant information from the popular Allrecipes.com website. This dataset covered food items categorized into 27 groups and spanned from 2000 to 2018. To convert user ratings into a suitable format, binary implicit feedback was generated, indicating user-food interactions. The final dataset included over 68,000 users, 45,000 foods, and 33,000 ingredients, along with over a million ratings.

To identify food ingredients from crawled text, Natural Language Processing (NLP) techniques were implemented. The authors employed a simple string matching technique from Natural Language Toolkit(NLTK) to identify ingredients against a predefined list. Before feeding the data into the recommender system, ingredient formalization and preprocessing were performed. This included tokenization, stemming, and stop-word removal, resulting in more refined ingredient information.The multi-step data collection and preprocessing approach yielded a comprehensive dataset, providing a solid foundation for developing an effective food recommender system.

2.2.3 Advantages

1. Time-aware: Considers the time factor
2. Content-aware:Considers the content of food items
3. User-aware:Considers the user's preferences and past interactions

2.2.4 Disadvantages

1. Data dependency:Requires a large amount of data
2. Computational complexity
3. Cold start problem

2.3 Early-Stage Risk Prediction of Non-Communicable Disease Using Machine Learning in Health CPS [3]

Cyber-Physical Systems (CPS) combine the physical world with the cyber world, allowing for communication and feedback between them. This technology has the potential to revolutionize healthcare, by enabling early-stage risk prediction of Non-Communicable Diseases (NCDs). This paper proposes a closed-loop Machine Learning(ML)-powered Healthcare Cyber-Physical System (HCPS) for early-stage risk prediction of NCDs, considering diabetes as an example. The proposed system uses a verified training dataset and a dynamic test dataset, which allows it to be applied to real-time data from wearable sensors. The paper concludes by discussing the results of several experiments that demonstrate the effectiveness of the proposed framework.

2.3.1 Implementation Details

The rising prevalence of NCDs, particularly diabetes, necessitates the development of an effective early-stage risk prediction system. To address this challenge, a closed-loop ML-powered HCPS is proposed. This HCPS aims to accurately predict NCD risk in individuals by processing real-time data from wearable sensors, utilizing verified training and dynamic test datasets, and achieving high prediction accuracy. However, the HCPS must be robust and generalizable to a diverse patient population, detect NCD risk early enough for intervention, and be easy to use and implement in various healthcare settings. The success of the HCPS will be evaluated based on its ability to accurately predict NCD risk, enable early intervention, and facilitate widespread adoption through user-friendly and scalable implementation.

Health Monitoring Sensors , a transformative technology in the realm of healthcare, encompasses a diverse array of sensors designed for a wide range of purposes.These meticulously crafted sensors, strategically placed on or within the body, continuously gather data on various health parameters, providing an unprecedented level of insight into an individual's well-being.This real-time data stream, encompassing heart rate, BP, sleep patterns, and a multitude of other vital metrics, empowers healthcare providers with invaluable information for diagnosis, treatment planning, and monitoring patient progress.Beyond its diagnostic capabilities, the wearable network extends its reach into preventive healthcare, enabling the early detection of potential health issues before they escalate into more severe conditions.By identifying subtle changes in an individual's health patterns, wearable sensors can trigger timely interventions, preventing the onset of chronic diseases and improving overall health outcomes.The wearable network's ability to provide real-time feedback to users empowers individuals to make informed decisions about their health and lifestyle choices.By monitoring their heart rate during exercise, for instance, individuals can optimize their workout routines, maximizing their fitness goals while maintaining a safe and effective exercise regimen.

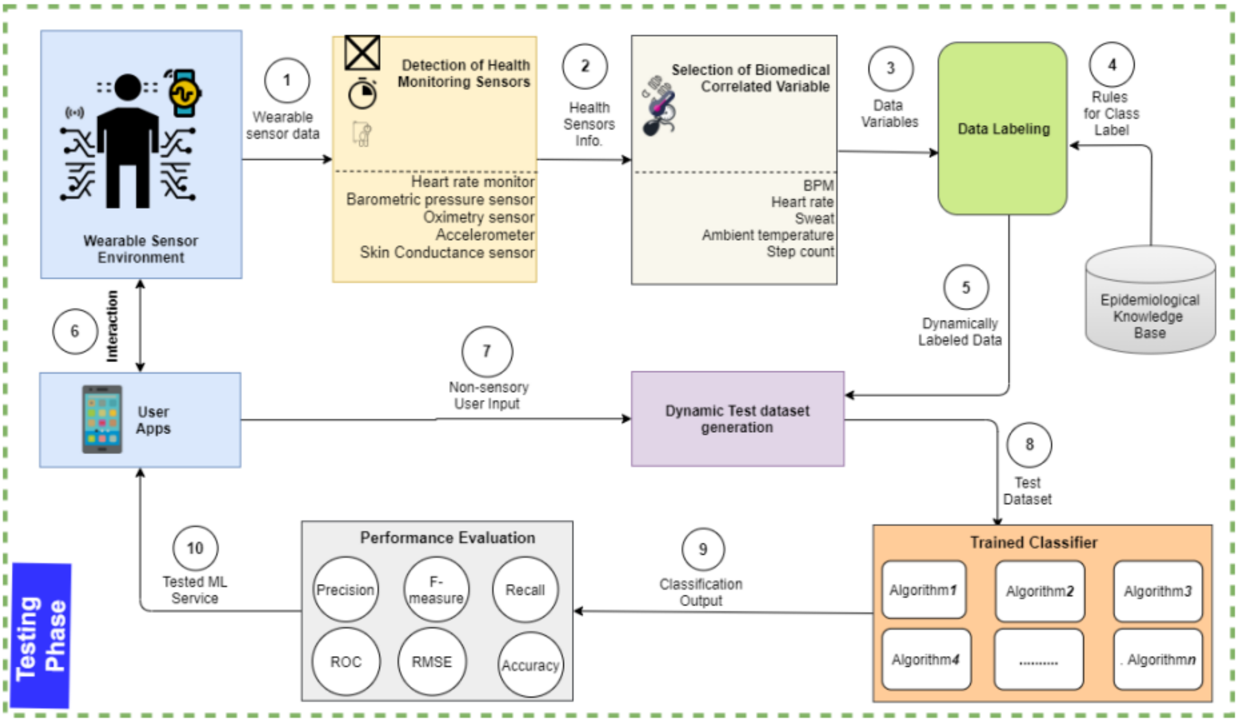


Figure 2.3: Architecture Diagram

Similarly, tracking sleep patterns can reveal potential sleep disturbances, allowing individuals to adopt lifestyle modifications or seek medical intervention to improve sleep quality and overall well-being.The wearable network's versatility extends to home rehabilitation, providing real-time feedback and guidance to individuals undergoing physical therapy or recovering from injuries.By monitoring progress and adherence to rehabilitation protocols, wearable sensors can accelerate recovery times and enhance the effectiveness of rehabilitation programs.In the realm of safety monitoring, wearable sensors can detect falls and other emergencies, alerting caregivers or emergency services promptly, ensuring timely assistance and potentially saving lives.The wearable network's potential extends to treatment assessment, providing healthcare providers with objective data on patient response to medications or therapies.By analyzing changes in vital signs and other health parameters, clinicians can optimize treatment plans, ensuring personalized and effective care for each patient.Despite the immense promise of the wearable network, challenges remain to be addressed, primarily in sensor accuracy, battery life, and user-friendly applications.Continuous advancements in sensor technology are addressing accuracy concerns, leading to more reliable and precise data collection.Battery life limitations are being tackled through innovative power management techniques and energy-efficient sensor designs. User-friendliness is being enhanced through intuitive interfaces, personalized data visualization, and seamless integration with smartphones and other devices. As these challenges are overcome, the wearable network is poised to revolutionize healthcare management, transforming the way monitoring, diagnosis, treatment, and prevention of diseases occur, leading to a healthier and more empowered populace.

Knowledge Base

The Knowledge Base illustrated in architecture in Figure 2.3 encompasses verified datasets, ontology structures, and predefined rules tailored for data labeling , including risk, symptom, and disease ontology alongside medical rules for attribute determination. Serving as reusable information for queries and analysis, this epidemiological knowledge base significantly enhances the classification system's performance. Its pivotal role spans both training and testing phases within the proposed system. During training, the verified dataset from this knowledge base becomes instrumental, training the classifier across various ML classification algorithms. In contrast, during testing, the knowledge base aids in implementing rules, assigning data labels, and extracting vital features essential for predicting Non-Communicable Diseases (NCDs) from sensor-generated data. This comprehensive utilization of the knowledge base streamlines functionality across phases, ensuring reliability and accuracy in disease prediction tasks.

2.3.2 Experimental Analysis

The proposed research introduces an innovative ML-based HCPS targeting diabetes prediction. It revolutionizes conventional ML approaches by integrating an epidemiological Knowledge Base, significantly minimizing pre-processing steps. This novel approach relies on a meticulously vetted training dataset endorsed by medical professionals, ensuring robustness and reliability in the end-user application. Moreover, it implements rules to efficiently extract health data from raw sensor data, streamlining the pre-processing stage remarkably. The testing phase adopts a multi-stage process, combining sensory and non-sensory data to create a dynamic test dataset that aligns with the training data structure. Despite the involvement of medical practitioners in establishing the knowledge base, it does not impede the training process, instead fortifying the system's reliability and robustness. Leveraging low-computational ML algorithms, this method efficiently processes raw sensory data in an IoT-embedded HCPS environment. Its computational efficiency primarily stems from using a verified training dataset to train the classifier and employing a dynamic test dataset for evaluation. The comprehensive system processes outlined in subsequent sections detail the methodologies, algorithms, and workflow for training and testing the ML model within the HCPS environment. In conclusion, this innovative ML-powered HCPS system effectively predicts diabetes, leveraging an epidemiological knowledge base to streamline pre-processing complexities and ensure reliability and accuracy in healthcare predictions.

EVALUATION

The proposed model follows a rigorous evaluation process, adhering to widely accepted ML evaluation measures applied to the dynamic test dataset to assess classifier performance. This evaluated ML service is subsequently provided to end-user applications. Various parameters gauge the early-stage disease risk prediction, ensuring a comprehensive evaluation. Parameters include True Positive (TP) for correctly identified NCD risks, False Positive (FP) for incorrectly identifying risk-free individuals as at risk, True Negative (TN) for accurately identifying risk-free individuals, and False Negative (FN) for incorrectly identifying risk individuals as risk-free. Additionally, evaluation metrics encompass Correctly Classified instances, Incorrectly Classified instances, and the Kappa statistic, indicating agreement level, with higher values suggesting better performance. Root Mean Square Error (RMSE) measures prediction error concentration around the best-fit line, while TP Rate, FP Rate, Precision, Recall, F-measure, and Receiver Operating Characterstic (ROC) area delve into classifier accuracy, false predictions, precision-recall balance, and Receiver Operating Characteristic curve closeness to accuracy. Overall accuracy, considering TP, TN, FP, and FN, summarizes classifier performance comprehensively. This detailed evaluation across multiple metrics ensures a thorough assessment, aiding in understanding the model's strengths and areas for improvement in early-stage disease risk prediction.

2.3.3 Advantages

1. Comprehensive Assessment
2. Specificity
3. Threshold Independence
4. Visualization Capability

2.3.4 Disadvantages

1. Sensitivity to Imbalance
2. Complexity

3. Sensitivity to Thresholds
4. Insensitivity to Cost Factors

2.4 Long Short-Term Memory Recurrent Neural Networks for Multiple Diseases Risk Prediction by Leveraging Longitudinal Medical Records [4]

2.4.1 Implementation Details

In the implemented model, input features undergo normalization using mean and variance, specifically applied to continuous variables such as age. Categorical variables like gender and ethnicity are encoded using One-Hot Encoding to represent them effectively. To address long-term dependencies in the data, hidden units, including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), are employed. The output layer utilizes a sigmoid activation function for binary classification, ensuring a suitable representation of predicted probabilities. The choice of a binary cross-entropy loss function serves to penalize deviations between true labels and predicted probabilities, optimizing the model for effective classification tasks. This comprehensive approach in feature processing, architecture design, and loss function selection collectively contributes to the model's robustness and accuracy in capturing patterns within the given dataset.

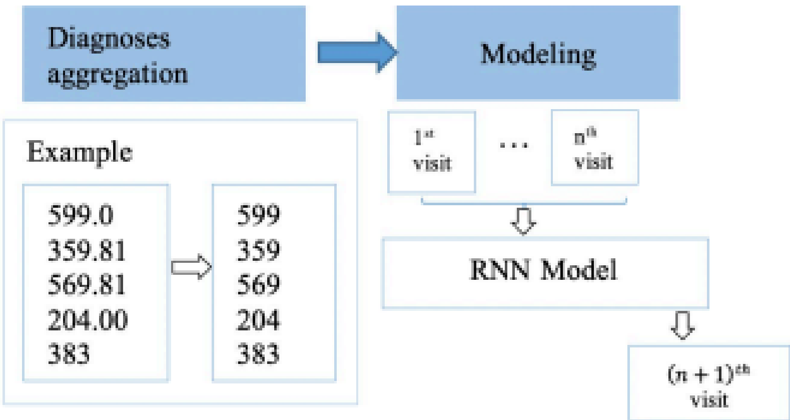


Figure 2.4: Architecture Diagram

In the context of patient-specific disease prediction, this model employs demographic data, historical diagnoses, and hospital stay information to predict the likely diseases a patient may experience in their upcoming hospital visit. Leveraging Recurrent Neural Networks (RNNs), the model as depicted in Figure 2.4 captures temporal patterns inherent in the patient's historical health records.

Adopting a ”many-to-one” architecture, where input comprises information from multiple historical visits, the model outputs the disease risk associated with the next hospital visit. Emphasizing the utilization of longitudinal data, the model aims to accurately predict disease risks by considering the comprehensive patient history, contributing to a more robust and personalized approach to healthcare forecasting.

2.4.2 Experimental Analysis

In this study, multi-disease prediction is conducted based on two independent datasets: the publicly available MIMIC-III (Medical Information Mart for Intensive Care III) database and a private dataset named GenCare, comprising inpatients with general care from a hospital in Shenzhen, China. MIMIC-III is a large, de-identified health-related database covering over forty thousand patients who stayed in intensive care units at the Beth Israel Deaconess Medical Center between 2001 and 2012. Diagnoses in patients' electronic health records are encoded using International Classification of Diseases (ICD) codes, a standard classification issued by the World Health Organization (WHO). The ICD system categorizes diseases hierarchically, considering causes, anatomical sites, and clinical symptoms, with ICD-10 widely used internationally and ICD-9 still in use in the United States. The study utilizes the current version of the MIMIC-III database (v1.4) and focuses on disease prediction based on these comprehensive datasets.

The results demonstrated that LSTM networks can predict future disease risks for patients with the exact-match score of 98.90% in MIMIC dataset and 95.12% in GenCare dataset based on 3-digit ICD codes aggregation, while 96.60% and 96.83% using 4-digit ICD code aggregation for these two datasets, respectively. In addition, the outcomes of this study could be developed as a function support module in a hospital information system, which facilitates healthcare professional's decision-making at the point of need.

2.4.3 Advantages

1. Handling Long Sequences
2. Avoiding the Vanishing Gradient Problem
3. Handling Variable Length Sequences
4. Memory Cell
5. Gradient Flow Control

2.4.4 Disadvantages

1. Higher Data Sparsity

2. Computational Complexity
3. Overfitting
4. Long Training Times

2.5 A Medical-History-Based Potential Disease Prediction Algorithm [5]

2.5.1 Implementation Details

This hybrid approach seamlessly integrates three distinct models, namely NAIS, DeepICF, and DeepFM, to enhance disease prediction. The data processing phase involves representing diseases using dense vectors through an embedding matrix. The extraction of relations is achieved using MLP with batch normalization, contributing to the model's ability to discern complex patterns. The incorporation of low and high-order relations is facilitated through a Factorization Machine, enriching the model's understanding of disease dynamics. Disease likelihood is determined through a sigmoid function in the output layer, and the prediction process is guided by a loss function that incorporates an L2 regularization term. This comprehensive approach aims to optimize disease prediction accuracy by leveraging the strengths of multiple models and sophisticated techniques throughout the various stages of the process.

The hybrid approach presented in Figure 2.5 offers a comprehensive recommendation system by seamlessly integrating various recommendation methods. Leveraging DeepICF, the model incorporates item relations and user-item representations, enhancing the precision of its suggestions. Additionally, the utilization of a Factorization Machine (FM) proves pivotal in efficiently handling high-dimensional and sparse features, contributing to improved modeling accuracy. The model delves into feature insights, considering correlations such as gender and purchase preferences to further enhance its recommendation capabilities. Through a meticulous factorization process, auxiliary vectors are employed to address feature sparsity and capture crucial feature relations, consolidating a robust and versatile system for delivering refined suggestions across diverse contexts.

2.5.2 Experimental Analysis

This experiment utilized a public dataset containing 103,886 pairs of samples representing 1355 genes and 6563 phenotypes, establishing connections between genes and Human Phenotype Ontology (HPO) terms. The evaluation protocol involved a leave-one-out approach, where the last

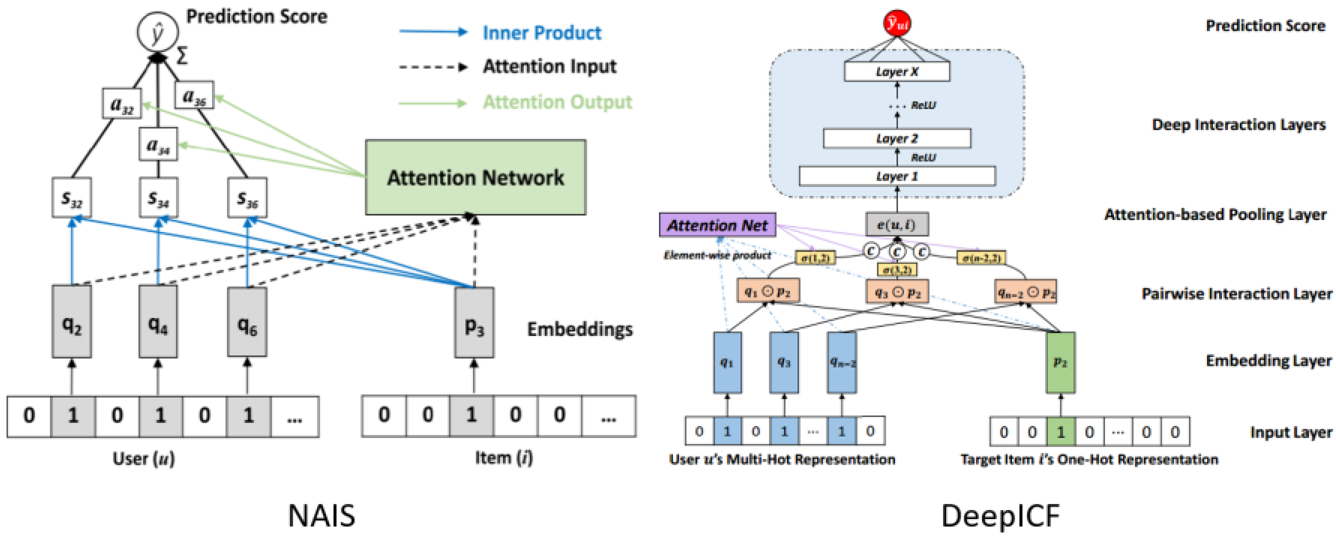


Figure 2.5: Architecture Diagram of NAIS & DeepICF

phenotype for each gene served as the testing data, and the remaining phenotypes were used for training. Negative testing examples were sampled by selecting 99 random phenotypes not linked to the gene. Supervised learning models were trained with a sampling ratio of 4 for negative examples, resulting in a dataset nearly five times larger. Two evaluation metrics, Hit Ratio at rank k (HR@k) and Normalized Discounted Cumulative Gain at rank k (NDCG@k), were employed to assess model performance, with k set to 10. Competing models included FISM, NAIS, and DeepICF, each designed for embedding-based recommendations. The proposed model, integrating diverse recommendation methods, outperformed all competitors in terms of HR@10 and NDCG@10. The attention network in NAIS contributed to its superior performance over FISM, while DeepICF demonstrated significant improvements, showcasing the importance of high-order relations in latent features. The proposed model achieved the best overall performance by considering both high-order and low-order relations in phenotype combinations. The growth patterns of HR@10 and NDCG@10 during training steps further illustrated the effectiveness of the proposed model in ranking positive phenotypes. The study provides a comprehensive evaluation of the proposed model's superiority over existing methods, affirming its potential for accurate disease predictions in a medical context.

2.5.3 Advantages

1. DeepICF enhances recommendation accuracy
2. Factorization Machines improves robustness of the system
3. Feature Relations leads to more insightful and personalized recommendations
4. Factorization Process handles feature sparsity provides accurate recommendations

2.5.4 Disadvantages

1. Complexity due to multiple models being integrated.
2. Data Dependency as it relies on diverse data sources
3. Training Overhead
4. Interpretability

Chapter 3

PROPOSED SYSTEM

The proposed medical app system seamlessly integrates sensors, a mobile application, and a centralized server to enable continuous monitoring and analysis of user's health data. Leveraging ML and DL techniques, the system processes real-time information to predict potential health risks. A sophisticated diet recommendation engine utilizes this analysis to suggest personalized dietary plans, considering individual nutritional needs, preferences, and allergies. Comprehensive health reports are generated, offering users actionable insights and facilitating informed decision-making. The user-friendly interface allows individuals to access their reports, receive diet recommendations, and provide valuable feedback for system refinement. Robust security measures ensure the confidentiality of health data, while notifications and alerts keep users informed about potential health issues. This holistic approach aims to empower users with actionable insights and personalized interventions for proactive health management.

3.1 Process Overview

1. **Configuration Setup** This allows advanced users to fine-tune the system to their preferences. They can choose which data points to prioritize (e.g., sleep, heart rate variability), select specific ML algorithms for different purposes (e.g., activity recognition, nutrition recommendation), and customize the recommendation format (e.g., recipes, meal plans, simple tips).

2. **Tracking Activity**. Beyond basic metrics, the device might capture more nuanced data like workout type and even environmental factors like temperature and humidity. This allows for more context-aware analysis and recommendations (e.g., suggesting outdoor activities on sunny days or recommending adjustments based on workout intensity).

3. **Data Synchronisation** Data synchronization is the invisible engine that keeps the entire IoMT-assisted patient diet recommendation system humming. It's the bridge between the

user, their wearable device, and the powerful algorithms that generate personalized insights. Wearable devices constantly stream data in real-time, providing instant feedback and enabling immediate analysis. This is ideal for monitoring workouts or capturing fleeting moments of activity. Data can be synchronized via Bluetooth , WiFi or cellular modes.

4. **Data Preprocessing** This stage is crucial for ensuring accurate analysis and recommendations. Techniques like outlier detection remove erroneous data points while missing value imputation fills in gaps without skewing results. Data normalization ensures all metrics are on the same scale for proper comparison and analysis.

5. **Data And Recommendation Generation** ML algorithms play a key role here. Supervised learning models can predict future activity levels or health outcomes based on past data. Unsupervised learning can identify hidden patterns in the data, suggesting unexpected correlations or potential health risks. The system might also integrate knowledge bases containing nutritional information, food pairings, and exercise routines to generate more specific recommendations.

6. **Data And Recommendation Visualization** Presenting recommendations in an easily digestible format is crucial for user engagement. Interactive dashboards with charts and graphs can help users visualize progress and trends. Personalized reports summarizing key insights and actionable steps can motivate behavior change.

7. **Data Analysis and Recommendation Forwarding** Analyzed data like activity trends and goal progress are also forwarded. This allows users to monitor their overall health picture and identify areas needing improvement. They can compare their performance against benchmarks or previous data to gauge progress and celebrate achievements.

8. **Decision Making** The user remains at the center of the process. They can accept or reject recommendations, adjust goals based on their feedback, and provide input on the system's effectiveness. This empowers them to make informed decisions about their health, ultimately owning their wellness journey.

9. **Goal Synchronisation**. This involves not just setting goals, but also specifying the timeframe (e.g., daily, weekly, monthly) and the desired level of granularity (e.g., specific steps, overall activity level, dietary changes). The system might also suggest personalized goals based on the user's profile (e.g., age, fitness level, health conditions).

3.1.1 Process Flow Diagram

Users can customize their preferences, track advanced metrics, and monitor progress through data visualization and analysis. ML algorithms generate recommendations, while users make informed decisions and set goals for their health journey. This is how the process flow of the system works and it is shown in Figure 3.1.

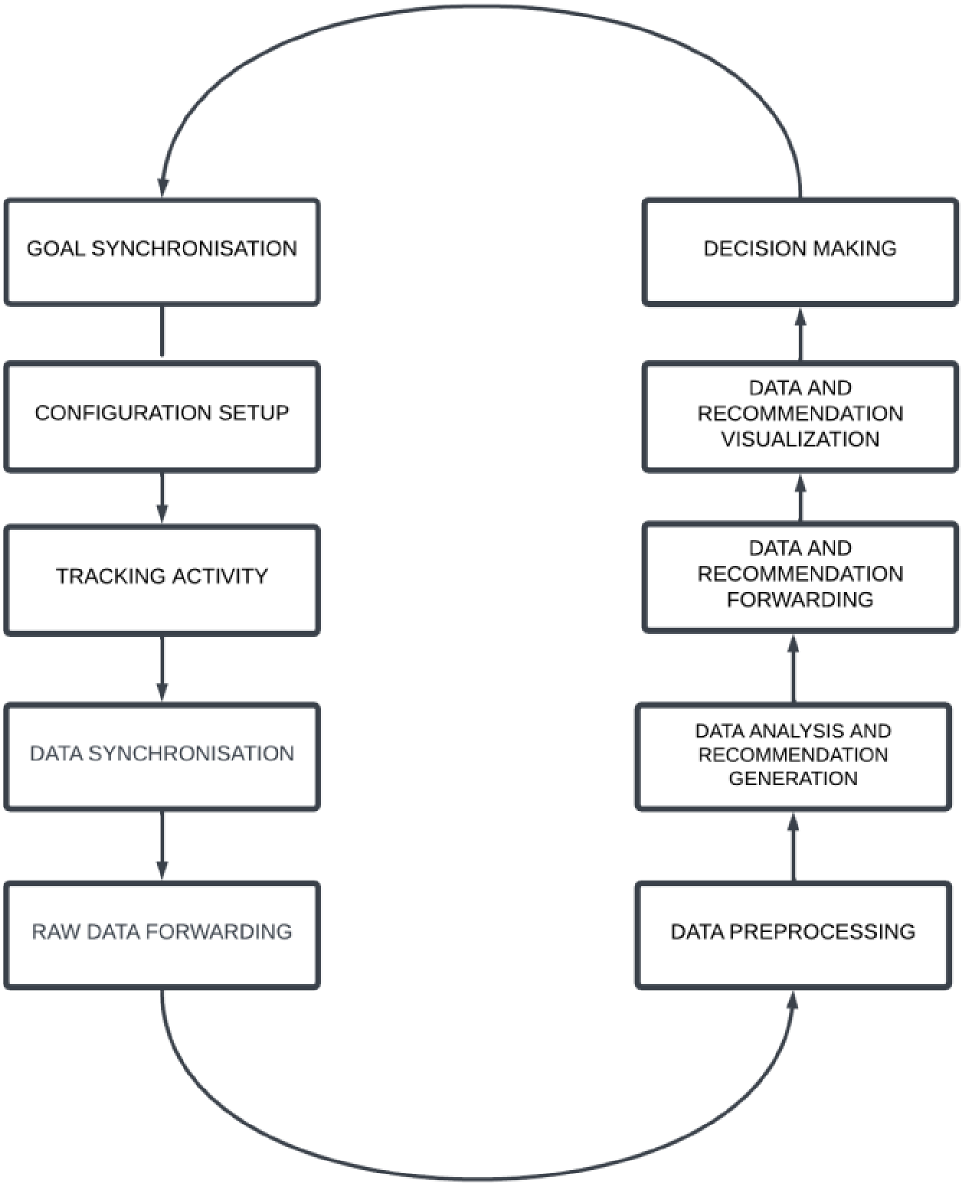


Figure 3.1: Process Flow Diagram

3.1.2 Architecture Diagram

The wearable device-based health risk predictor system as shown in Figure 3.2 is a personalized system that uses data collected by a wearable device to predict the user's risk of developing various health conditions. The system then provides the user with timely and relevant recommendations for improving their health.

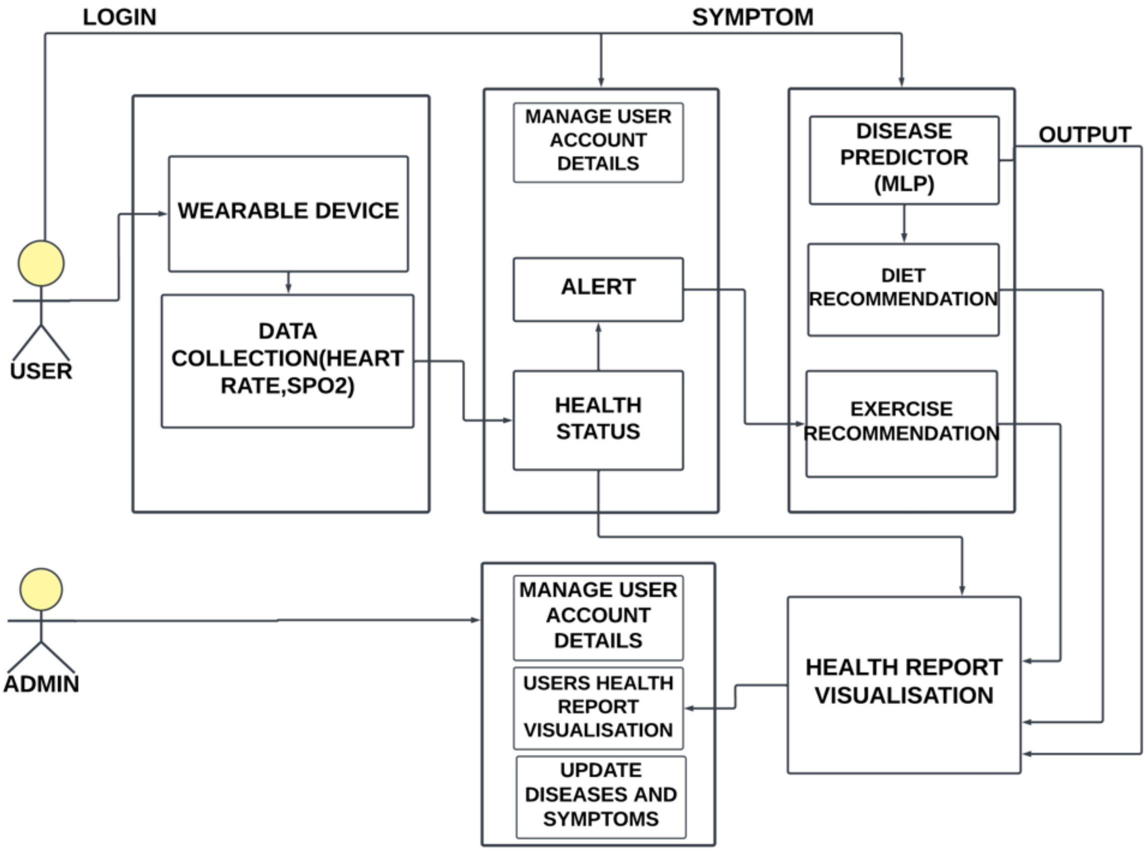


Figure 3.2: Component Architecture Diagram

3.1.3 Use Case Diagram

The use case diagram depicted in Figure 3.3 shows the prediction system uses a model to predict the future health risk of the user and the user interface interacts with the model. The model is used to predict future health risks, while the application interface is used to interact with the model.

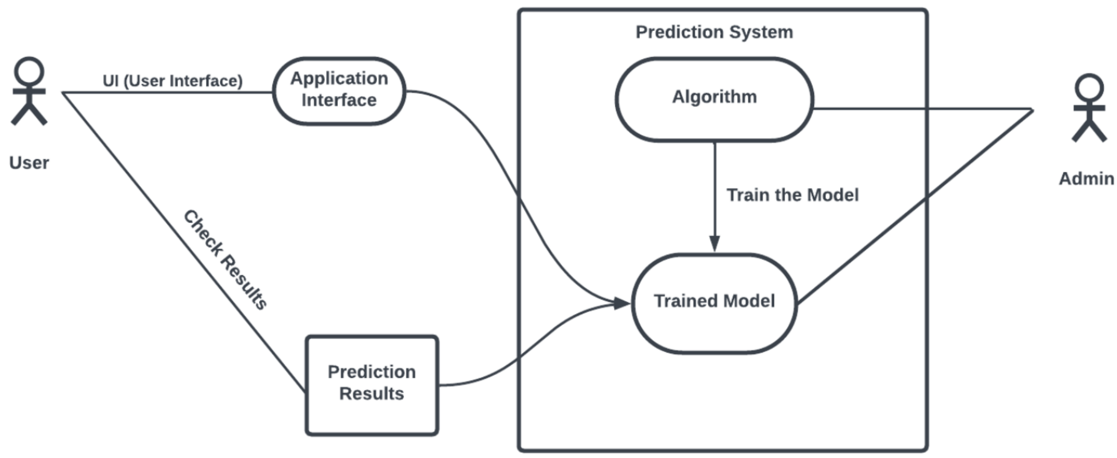


Figure 3.3: Use Case Diagram

3.1.4 Data Flow Diagram

The diagrams shown in Figures 3.4, 3.5, 3.6 and 3.7 depicts the admin and user module's functionality and data exchange with other system modules. Data is prepped, analyzed for disease patterns, and used to detect diseases and identify health risks. Then, the system recommends diets and generates reports summarizing patient health.

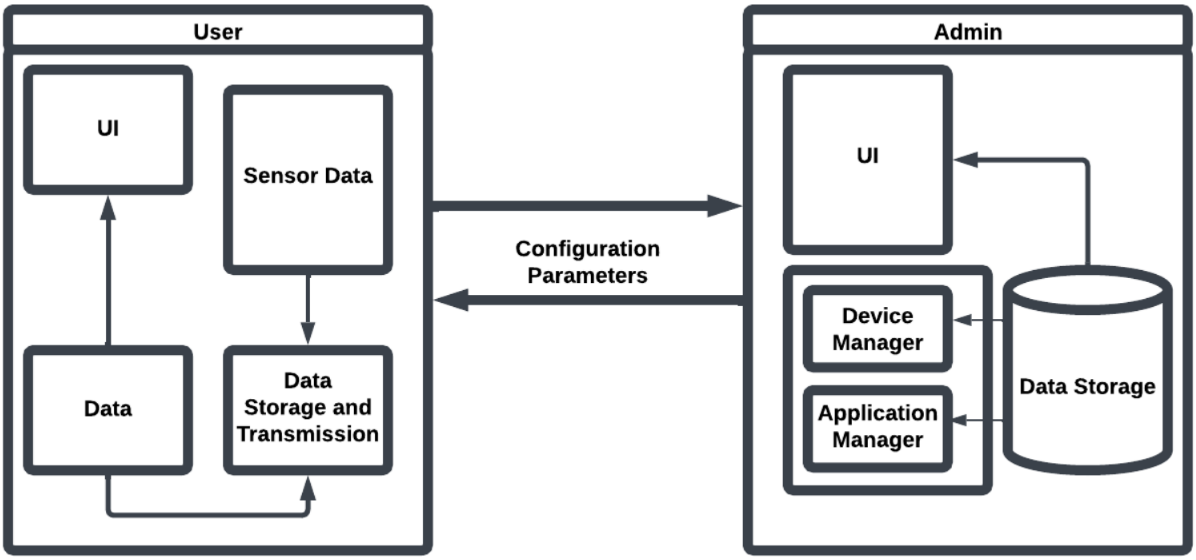


Figure 3.4: Data Flow Diagram establishing relation between User and Admin

3.1.5 Data Flow Diagram of Admin

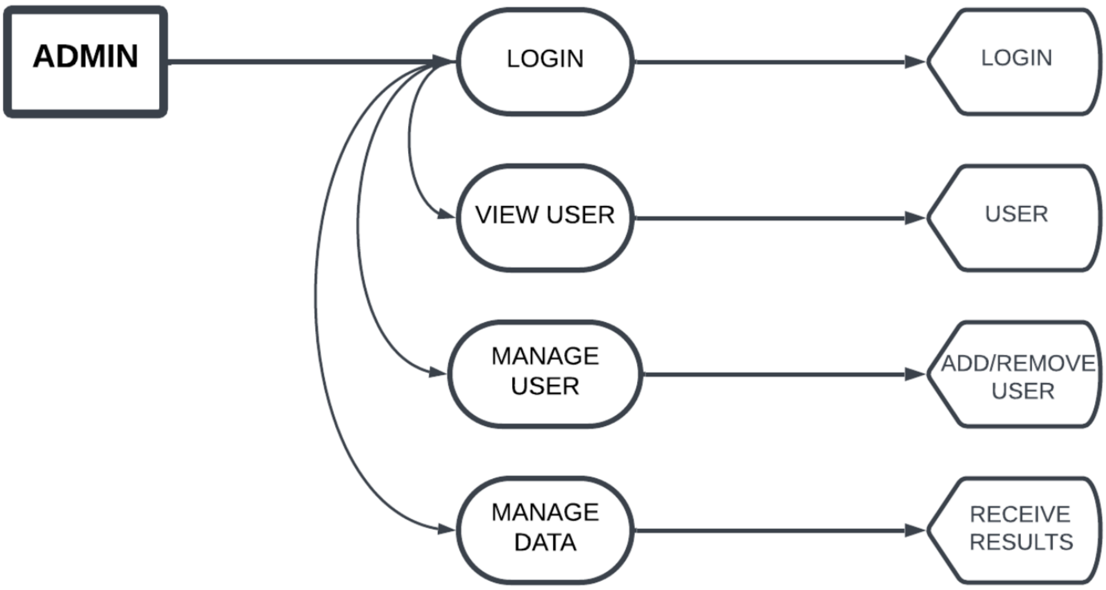


Figure 3.5: Data Flow Diagram Level 1-Admin

3.1.6 Data Flow Diagram of User

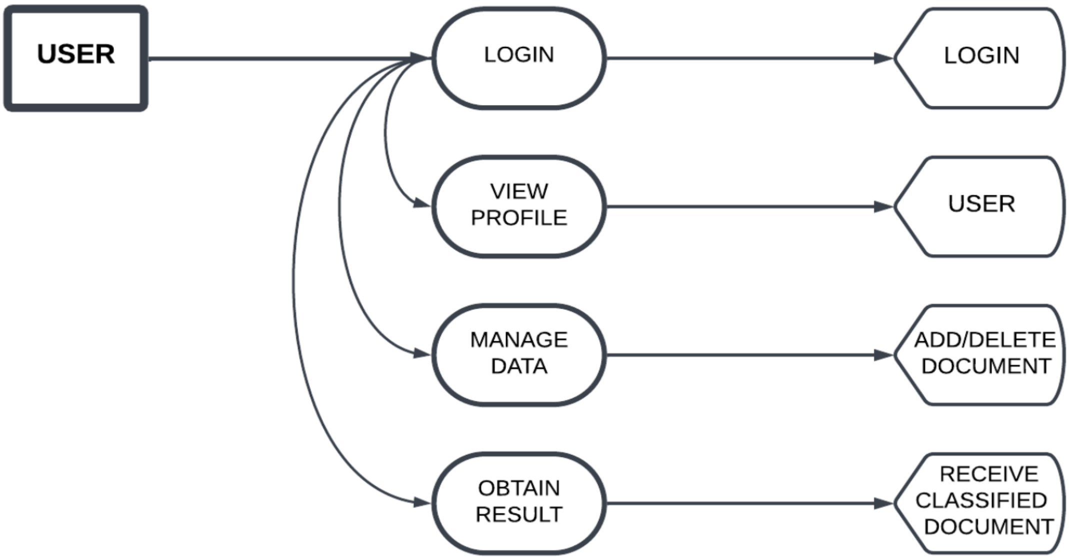


Figure 3.6: Data Flow Diagram Level 0-User

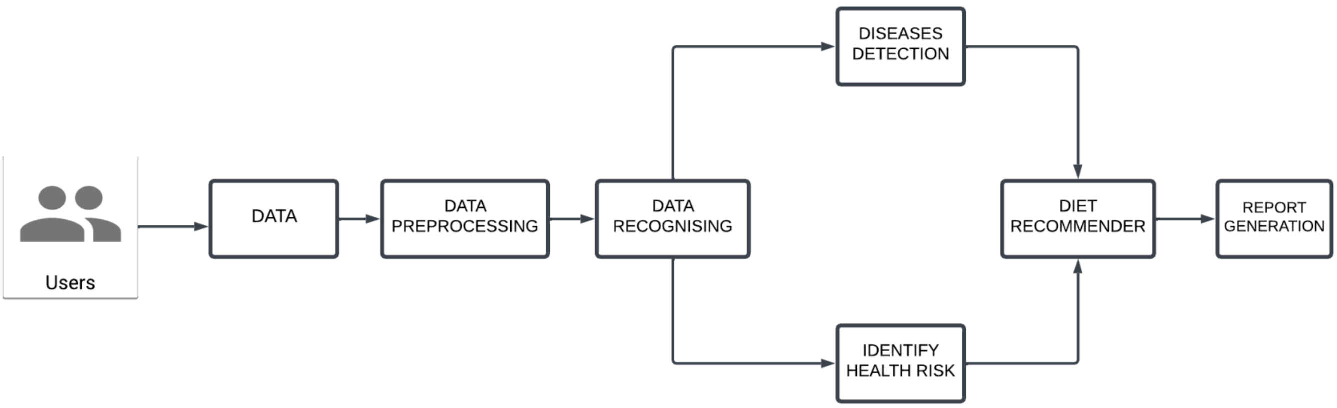


Figure 3.7: Data Flow Diagram Level 1-user

3.1.7 Class Diagram

The class diagram for the patient health monitoring system as depicted in Figure 3.8 has two classes, user and admin.

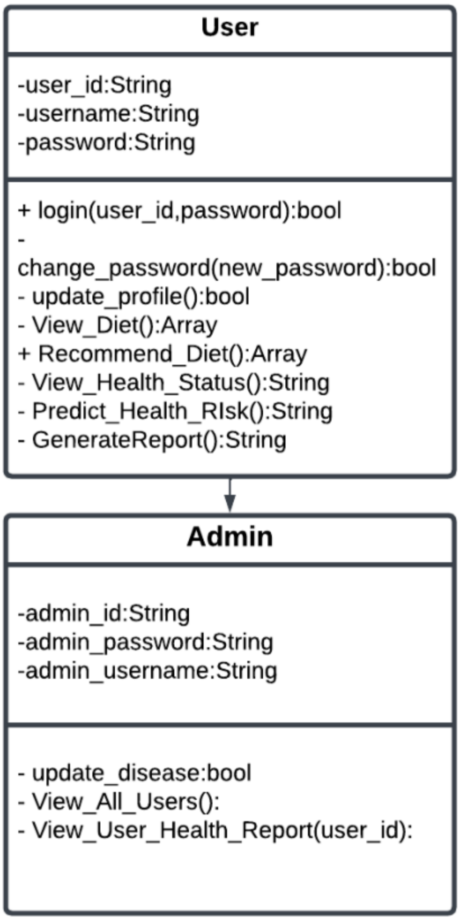


Figure 3.8: Class Diagram

3.2 System Requirements

The basic requirements for the proposed systems can be divided into two. Hardware requirements and Software requirements.

3.2.1 Hardware requirements

The hardware requirements consist of a laptop with a quadcore processor with 8GB RAM or more. Along with this, a Hard Disk of 500GB or above is also required. For recording data such as heart rate, sp02 , and body temperature we require hardware modules like ESP32 and MAX30102.

3.2.2 Software requirements

The softwares are generally for coding and execution of algorithms,software platforms used in this system are HTML,CSS, Javascript ,PHP,Python and MySQL. IDE(Integrated Development Environment) such as Android Studio and Arduino IDE are in use.

3.2.3 HTML

HTML, or Hypertext Markup Language, is the standard language for creating web pages and applications on the World Wide Web. It uses tags to structure content and define how elements are displayed and interacted with by users.

3.2.4 CSS

CSS, or Cascading Style Sheets, is a language used to style and format HTML documents. It allows web developers to control the layout, appearance, and presentation of web pages, including aspects like fonts, colors, spacing, and responsiveness.

3.2.5 Javascript

JavaScript is a programming language commonly used for creating interactive and dynamic elements on web pages. It enables developers to add functionality such as animations, form validations, and event handling to enhance user experience.

3.2.6 PHP

PHP is a server-side scripting language used for web development. It is a system software requirement for web servers like Apache or Nginx. PHP interpreters execute PHP code, generating dynamic web pages. It is commonly used in conjunction with databases like MySQL for dynamic content.

3.2.7 Python

Python is a high-level programming language known for its readability and versatility. It's a system software requirement for various applications, including web development, data analysis, machine learning, and automation.Python interpreters execute Python code, making it platform-independent.Python 3.6 has been utilized for the working of project.

3.2.8 Apache

The Apache HTTP Server is a free and open-source cross-platform web server software, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

3.2.9 MySQL

MySQL is a database management system that helps software applications store, retrieve, and manage data efficiently. It's essential for applications that need organized data storage, fast data retrieval, security features, scalability for growth, performance optimization, seamless integration with programming languages, backup and recovery capabilities, making it a critical component of many software systems.

3.2.10 XAMPP

XAMPP is like a pre-packaged toolbox for developers. It installs all the essential software components needed to run a web server on your local machine. This means you can develop and test websites without needing to install them on a remote server.

3.2.11 Android Studio

This is the official IDE for developing Android applications. It offers a comprehensive set of tools for building features, designing user interfaces, and testing your Android app.

3.2.12 Arduino IDE

The Arduino IDE is software used to program Arduino microcontroller boards. It's essential for writing, compiling, and uploading code to make these boards function. The IDE provides tools like a code editor, compiler, libraries management, and a serial monitor for communication. It's compatible with major operating systems like Windows, macOS, and Linux, making it widely accessible for developers.

3.3 Implementation Details

Our disease prediction system leverages wearable sensors and machine learning to transform health monitoring. It utilizes a dataset that consists of numerous symptoms related to illness which has binary values(0 or 1) the combination of these values gives a disease as output. The hardware module MAX30102 captures vital signs like heart rate, body temperature, and blood oxygen and passes data via Wi-Fi module present in ESP32. The data provided in the dataset undergoes preprocessing to ensure its quality for analysis. Here, noise is filtered out, measurements are standardized for consistency across users, and key characteristics are extracted from the raw readings. To make the data more manageable for analysis, dimensionality reduction techniques condense the information while preserving what's important. Additionally, the data is aggregated over time intervals to provide insights into health trends. Machine learning takes the preprocessed data and prepares the diagnostic information. This involves identifying relevant factors like symptoms, exploring the range of health conditions the model will encounter, and assigning unique numerical codes to each diagnosis. The categorical diagnostic data is then converted to a numerical format using this code system. Before deploying the model, the balance of different health outcomes within the data is assessed. If certain conditions are underrepresented, techniques like random oversampling can be used to create a more balanced dataset. This is crucial for ensuring the model's accuracy. Finally, a rule-based system steps in to recommend diet and medication based on the predicted disease. This simpler approach is preferred over machine learning for exercise recommendations due to its ease of implementation and the lack of extensive training data required. Overall, this system integrates wearable sensors, data processing, machine learning, and a rule-based system to create a comprehensive disease prediction and management solution.

3.4 Methodology

3.4.1 Data Acquisition

Data Acquisition is an important step before training a diseases prediction model. Here are the general steps you might follow:

1. **Data Capture with MAX30102 Sensor**: The process begins with the MAX30102 sensor capturing vital health metrics such as SpO2 and heart rate. This sensor utilizes photoplethysmography (PPG) to measure the volumetric variations of blood circulation.

2. **Transmission via ESP32 Microcontroller**: The captured data is then transmitted through the ESP32 microcontroller, which serves as the intermediary between the sensor and the application. ESP32 facilitates seamless Wi-Fi connectivity, ensuring efficient data transmission to designated platforms.

3. **Integration with Application and Modules**: Through Wi-Fi connectivity, the transmitted data seamlessly integrates with our application and other modules, such as the Disease Predictor and Diet Recommender. This integration allows for comprehensive health tracking and analysis in real-time.

4. **Facilitation of Health Benchmarks Evaluation**: The transmitted data plays a crucial role in evaluating health benchmarks. By analyzing SpO2 levels, heart rate patterns, and body temperature, the system establishes personalized health benchmarks for users.

5. **Proactive Health Monitoring**: With established health benchmarks, the system enables

proactive health monitoring. Deviations from personalized benchmarks trigger alerts, prompting timely intervention and preventive measures.

6. **Strategizing Intervention**: Utilizing the analyzed data, the system formulates intervention strategies tailored to individual health profiles. These strategies encompass lifestyle adjustments, medication adherence reminders, and personalized health recommendations to optimize well-being.

3.4.2 Data Preprocessing

1. **Signal Filtering and Artifact Removal**: Begins by applying digital signal processing techniques tailored to the MAX30102 sensor data to filter out noise and remove artifacts. This ensures that only reliable measurements of SpO2 and heart rate are considered for further analysis.

2. **Normalization of Health Metrics**: Normalize the SpO2 , heart rate and body temperature data to ensure consistency and comparability across different individuals. This step is crucial for accurately assessing deviations from individual health baselines. This enhances the efficiency of subsequent analysis and modeling tasks.

3. **Null Value Checking And Removal**: Utilize the isnull() method from the Pandas library to identify any null values present within the dataset. If null values are found, use the dropna() method to remove rows containing null values.

4. **Duplicate Value Checking and Removal**:Additionally, consider checking for duplicate rows in the dataset using the duplicated() method.If duplicates are found and deemed unnecessary, use the drop$_{d}uplicates()methodtoremoveduplicaterows.$

3.4.3 Training

1. **Identification of Relevant Categorical Variables**: Start by identifying categorical variables specific to the health domain dataset, focusing on attributes like symptoms, medical history, and diagnostic findings. In our project, particular attention is given to the ”diagnosis” variable, representing different medical conditions or prognoses.

2. **Exploration of Diagnostic Classes**: Conduct an in-depth exploration of the unique diagnostic classes present in the ”diagnosis” variable. This step involves understanding the diverse range of medical conditions or prognoses encountered in the dataset, which serves as the basis for subsequent mapping creation.

3. **Mapping Scheme Design**: Create a mapping scheme tailored to the identified diagnostic classes, where each unique diagnosis is assigned a corresponding numerical code. This mapping scheme establishes a standardized representation of medical conditions, essential for computational analysis and machine learning applications.

4. **Conversion of Categorical Data**: Apply the established mapping scheme to convert the categorical ”diagnosis” data into numerical format, replacing diagnosis labels with their corresponding numerical codes. This conversion enables seamless integration of diagnostic information into machine learning models for predictive analysis.

5. **Validation of Mapping Accuracy**: Validate the accuracy and consistency of the mapping scheme by cross-referencing original diagnosis labels with their numerical representations. Ensuring accurate mapping is crucial for preserving the interpretability and integrity of diagnostic data throughout the conversion process.

6. **Incorporation into Machine Learning Workflow**: Integrate the transformed numerical diagnosis data into the machine learning workflow for model training and evaluation. By converting categorical diagnoses into numerical representations, our project streamlines computational processing, enabling efficient predictive modeling and analysis tasks tailored to health prognosis prediction.

3.4.4 Testing

1. **Dataset Balance Assessment**: Start by evaluating the balance of classes within the health prognosis dataset, focusing on the distribution of different prognostic outcomes. This step involves examining the frequency of each prognostic category to identify potential class imbalances.

2. **Identification of Imbalanced Prognostic Categories**: Identify specific prognostic categories that exhibit imbalance, where certain health outcomes are underrepresented compared to others. Understanding which prognostic categories are imbalanced is crucial for determining the appropriate corrective measures

3. **Implementation of Random Oversampling**: Apply Random Oversampling as a method to address imbalanced prognostic categories by artificially increasing the number of instances in minority categories. This technique involves randomly replicating samples from the minority prognostic categories to achieve a more balanced distribution

4. **Validation of Oversampling Results**: Validate the effectiveness of Random Oversampling on the dataset by assessing the post-sampling class distribution. It’s essential to ensure that

oversampling successfully mitigates class imbalance without introducing bias or distorting the original data distribution.

5. **Impact Evaluation on Predictive Modeling**: Evaluate the impact of balanced dataset on predictive modeling tasks specific to health prognosis prediction. Compare the performance metrics of predictive models before and after oversampling to assess how addressing imbalance influences accuracy, sensitivity, and specificity.

3.4.5 Rule-Based System for matching diet, exercise, and medication

Rule-Based System entails that an output variable such as diet, exercise, and medication are mapped onto disease predicted from an existing dataset that has mapped disease with output variables. Now it is important to mention that not all diseases can be treated via exercise hence, diet and medication are output variables that give suggestive methods to treat diseases and the significance of these parameters helps in treating diseases. A Rule-Based method is chosen for the reason it is simpler to map the output to the input, ML techniques require training overhead, and applying prediction based on diet and medication using a single variable disease seems unnecessary, hence the given method is adopted.

3.4.6 Evaluating the Model

Once the model is trained, it is evaluated using the testing set. The performance of the model is evaluated using various metrics, such as accuracy, precision, recall, and F1 score. If the performance of the model is not satisfactory, the model can be tuned by changing hyperparameters of both the models.

1. **Accuracy**: It measures the proportion of correctly classified instances over the total number of instances.

2. **Precision**:It measures the proportion of true positives over the total number of predicted values.

3. **Recall**:It measures the proportion of true positives over the total number of actual positives.

4. **F1 Score**:It is a harmonic mean of precision and recall, and it combines both metrics into a single score. F1 Score is a good metric for evaluating models when the classes are imbalanced.

Chapter 4

PERFORMANCE ANALYSIS

To perform a comprehensive analysis of disease prediction in MLP model, it's essential to evaluate key performance metrics such as accuracy, precision, recall, and F1 score. Additionally, visualizing the learning curves provides valuable insights into the models' performance across different training set sizes, aiding in the identification of underfitting or overfitting. These assessments help assess the models' effectiveness and identify areas for improvement.

1. **Accuracy**: The accuracy of the model is determined by the proportion of properly identified instances in the test dataset, as well as by precision, recall, F1 score. The calculation is as follows: Accuracy is equal to $\frac{TP+TN}{TP+TN+FP+FN}$, where TP, TN, FP, FN, and FP represent the number of true positives, true negatives, false positives, and false negatives, respectively.

2. **Precision**: The precision of the model is defined as the proportion of correctly classified positive instances out of all the instances classified as positive. It can be computed as follows: Precision = $\frac{TP}{TP+FP}$.

3. **Recall**: The recall of the model is defined as the proportion of correctly classified positive instances out of all the positive instances in the test dataset. It can be computed as follows: Recall = $\frac{TP}{TP+FN}$.

4. **F1 score**: The harmonic mean of precision and recall is the F1 score, which may be calculated as follows: F1 score is equal to $2 \times \frac{precision \times recall}{precision+recall}$.

5. **Learning Curves**: Learning curves show the model's performance as it learns from the training data. Below are the learning curves for accuracy and loss.

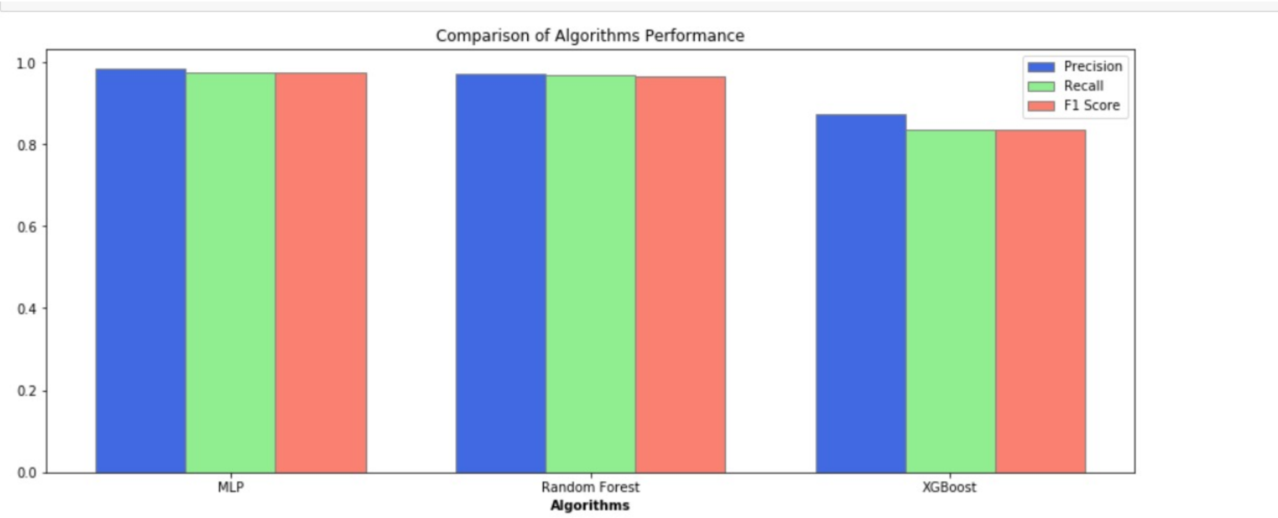


Figure 4.1: Comparison graph

The graph as depicted in Figure 4.1 compares the performance of MLP, Random Forest, and XGBoost algorithms. Here, MLP achieves the highest precision, recall, and F1 score, indicating its strength in correctly classifying relevant data and balancing precision with recall.

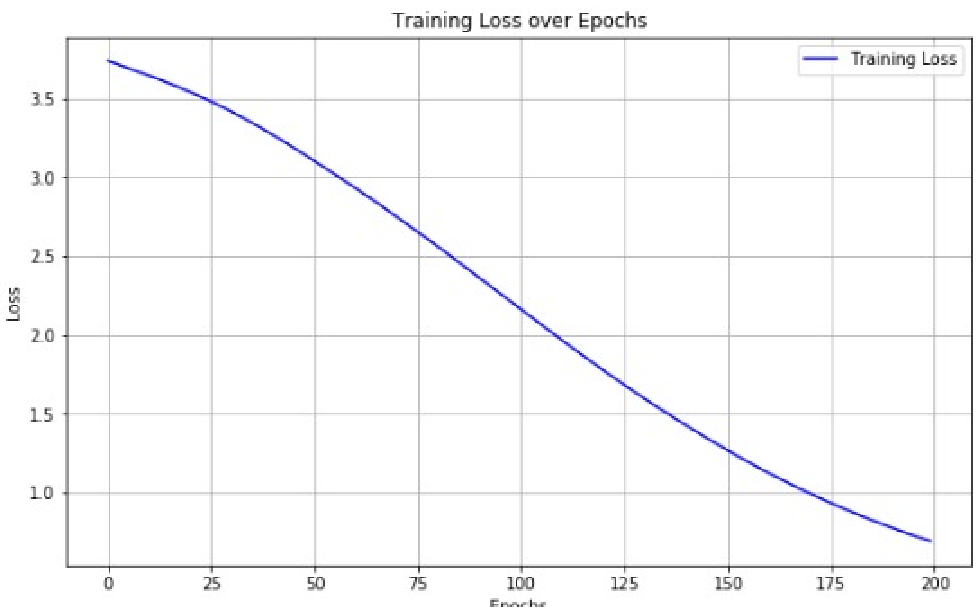


Figure 4.2: Loss Curve

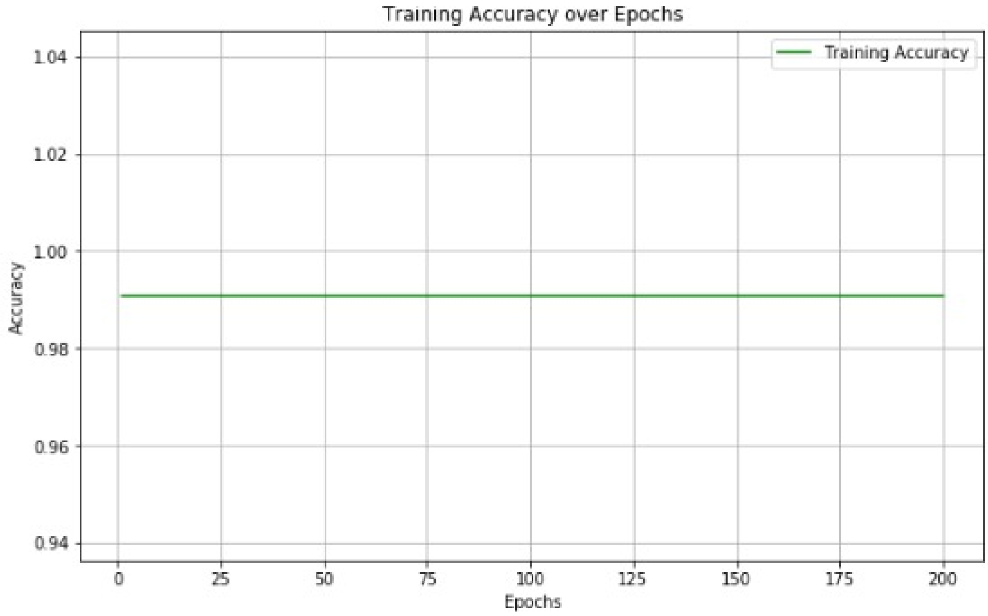


Figure 4.3: Accuracy Curve

In the context of training an MLP (Multi-Layer Perceptron), the training loss as depicted in Figure 4.2 over epochs graph becomes a roadmap to success. Each epoch represents one full pass through your training data. As the MLP sees more data (reflected by the increasing number of epochs on the x-axis), the ideal scenario is for the training loss on the y-axis to steadily decrease. This signifies the MLP is learning the patterns in your data and improving its predictions.A smooth, downward sloping curve indicates the MLP is efficiently minimizing the error between its predictions and the actual targets.

The graph as shown in Figure 4.3 depicts the model's learning process over training iterations (epochs). As the epochs progress (x-axis), the accuracy of classifying training examples rises (y-axis), indicating the model is grasping the training data. However, the flattening curve towards the end suggests the model is nearing its peak accuracy on this specific data.

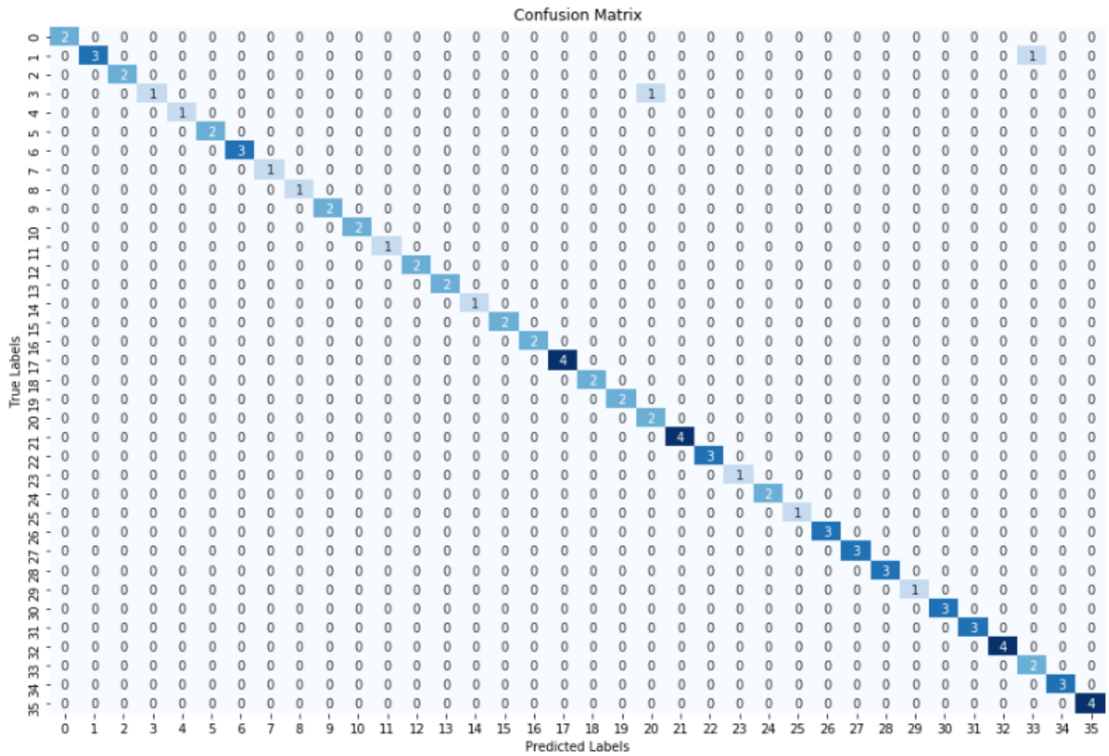


Figure 4.4: Confusion Matrix

A confusion matrix as illustrated in Fig 4.4 is a table that visualizes the performance of an MLP model on a classification task. In the context of disease prediction based on symptoms (binary values of 0 or 1), the confusion matrix would typically have two rows and two columns, representing the two possible outcomes (disease or no disease).

The analysis of the confusion matrix reveals promising aspects of the disease prediction model. The high number of True Negatives indicates the model excels at identifying healthy individuals. This is valuable because it allows for quick dismissal of the disease, potentially reducing unnecessary procedures and anxiety for patients. Additionally, the low False Positive rate suggests the model rarely misclassifies healthy people as having the disease, minimizing the risk of unnecessary worry and treatment. These findings highlight the model's effectiveness in accurately ruling out illness, a crucial step in early diagnosis.

Chapter 5

CONCLUSION

The combination of wearable devices and ML techniques has the potential to revolutionize healthcare by enabling personalized and predictive health monitoring. Wearable devices can continuously collect a wide range of physiological data, which can then be analyzed by ML algorithms to identify patterns and predict potential health risks.This information can be used to provide timely interventions and improve patient outcomes.Currently, wearable devices and ML conditions, including cardiovascular disease, diabetes, and sleep apnea. These systems are still under development.As wearable devices and ML techniques continue to improve, they are expected to play an increasingly important role in healthcare. These technologies have the potential to revolutionize the way we monitor and manage our health, leading to better health outcomes for everyone.

References

[1] Iwendi, Celestine, et al. ”Realizing an efficient IoMT-assisted patient diet recommendation system through machine learning model.” IEEE access 8 (2020): 28462-28474.

[2] FARRAHI, VAHID. ”A Novel Time-Aware Food Recommender-System Based on Deep Learning and Graph Clustering.”

[3] Ferdousi, Rahatara, M. Anwar Hossain, and Abdulmotaleb El Saddik. ”Early-stage risk prediction of non-communicable disease using machine learning in health CPS.” IEEE Access 9 (2021): 96823-96837.

[4] Wang, Tingyan, Yuanxin Tian, and Robin G. Qiu. ”Long short-term memory recurrent neural networks for multiple diseases risk prediction by leveraging longitudinal medical records.” IEEE journal of biomedical and health informatics 24.8 (2019): 2337-2346.

[5] Hong, Wenxing, et al. ”A medical-history-based potential disease prediction algorithm.” Ieee Access 7 (2019): 131094-131101.

Appendix A

DATA ACQUISITION FROM HARDWARE

```
#include <Wire.h>
#include "MAX30105.h"
#include "heartRate.h"
#include "spo2_algorithm.h"
#include <WiFi.h>
#include <MySQL_Generic.h>
MAX30105 particleSensor;
boolean startflag = 0, continueflag = 0;

const int ledPin = 5;

const char* ssid = "hms";
const char* psk = "hms@1234";

IPAddress server( 192,168,127,185);
uint16_t server_port = 3306;

char user[] = "health_user";
char password[] = "123456789";

char default_database[] = "health_monitoring";
char default_table[] = "health_data";



char query[128];
char INSERT_DATA[] = "INSERT INTO %s.%s (spo2, heartrate,temperature) VALUES
    ('%s', '%s', '%s')";

MySQL_Connection conn((Client*)&client);
```

```
void dbConnect(String O2, String bpm, String temperature) {
  MySQL_Query query_mem = MySQL_Query(&conn);

  Serial.println(String(O2) + ", " + String(bpm)+ ", " + String(temperature)
     );
  if (conn.connected()) {
    sprintf(query, INSERT_DATA, default_database, default_table, String(O2).
       c_str(), String(bpm).c_str(), String(temperature).c_str());
    Serial.println(query);

    if (!query_mem.execute(query)) {
      Serial.println("Update/Insert error");
    } else {
      Serial.println("Data Updated/Inserted");
    }
  } else {
    Serial.println("Disconnected from Server. Can’t Update/Insert.");
  }

}
uint32_t irBuffer[100];
uint32_t redBuffer[100];

int32_t bufferLength;
int32_t spo2;
int8_t validSPO2;
int32_t heartRate;
int8_t validHeartRate;
float temperatureF = 0;

void setup() {
  Serial.begin(9600);

  pinMode(ledPin,OUTPUT);
```

```
  if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) {
    Serial.println ("Sensor connection failure");
    while (1) {
      delay (10);
    }
  }
    WiFi.begin(ssid, psk);
  while (WiFi.status() != WL_CONNECTED) {
    delay(20);
    MYSQL_DISPLAY0(".");
  }
  Serial.println("WiFi connected");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

  MYSQL_DISPLAY1("Connected to network. My IP address is:", WiFi.localIP());
  MYSQL_DISPLAY3("Connecting to SQL Server @", server, ", Port =",
      server_port);
  MYSQL_DISPLAY5("User =", user, ", PW =", password, ", DB =",
      default_database);
  delay(700); /* Start the DHT11 Sensor */
}

void loop() {
  boolean processFlag = true;
  const byte RATE_SIZE = 10;
  byte rates[RATE_SIZE];
  byte avg_rates[RATE_SIZE];
  long del[25], timeDelay;
  byte rateSpot = 0, avgCnt = 0, loopCnt = 0;
  long lastBeat = 0;
  float beatsPerMinute;
  int beatAvg, Avg_beat, HRbpm;

  particleSensor.setup();
```

```
particleSensor.setPulseAmplitudeRed(0x0A);
particleSensor.setPulseAmplitudeGreen(0);

Serial.println("Make proper Contact with sensor");
delay (1500);
if (processFlag == true) {
  long irValue = particleSensor.getIR();
  if (irValue > 50000) {
    startflag = true;
    continueflag = true;

    Serial.println("Contact detected");
    delay (1500);
  }

  Serial.print("Calibrating HR, ");
  while (startflag == true) {
    long irValue = particleSensor.getIR();
    if (irValue > 50000) {
      if (checkForBeat(irValue) == true) {
        long delta = millis() - lastBeat;
        lastBeat = millis();

        beatsPerMinute = 60 / (delta / 1000.0);
        if (beatsPerMinute < 255 && beatsPerMinute > 20) {
          rates[rateSpot++] = (byte)beatsPerMinute;
          rateSpot %= RATE_SIZE;
          beatAvg = 0;
          for (byte x = 0 ; x < RATE_SIZE ; x++)
            beatAvg += rates[x];
          beatAvg /= RATE_SIZE;

          avgCnt++;
          del[avgCnt] = delta;
```

```
          String progress = String (map (avgCnt, 0, 25, 0, 100));
          // Serial.println ("*" + progress + "#");
          Serial.println(progress + "%");

          if (avgCnt >= 11 && avgCnt <= 20) {
            avg_rates[loopCnt++] = (byte)beatAvg;
            loopCnt %= RATE_SIZE;
          }
          else {
            for (byte y = 0 ; y < RATE_SIZE ; y++)
              Avg_beat += avg_rates[y];
            Avg_beat /= RATE_SIZE;
          }
          if (avgCnt >= 25) {
            HRbpm = Avg_beat;
            startflag = false;
            processFlag = false;
            avgCnt = 0;
            delay (100);
          }
        }
      }
    }
  }
}

if (continueflag == true) {
  processFlag = true;
  byte ledBrightness = 60;
  byte sampleAverage = 4;
  byte ledMode = 2;
  byte sampleRate = 100;
  int pulseWidth = 411;
  int adcRange = 4096;
```

```
particleSensor.setup(ledBrightness, sampleAverage, ledMode, sampleRate,
   pulseWidth, adcRange);
Serial.print("Calibrating SPO2");
while (processFlag == true) {
  bufferLength = 100;

  for (byte i = 0 ; i < bufferLength ; i++) {
    while (particleSensor.available() == false)
      particleSensor.check();

    redBuffer[i] = particleSensor.getRed();
    irBuffer[i] = particleSensor.getIR();
    particleSensor.nextSample();
  }
  maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength,
     redBuffer, &spo2, &validSPO2, &heartRate, &validHeartRate);
  startflag = true;

  avgCnt = 0;
  while (startflag == true) {
    for (byte i = 25; i < 100; i++) {
      redBuffer[i - 25] = redBuffer[i];
      irBuffer[i - 25] = irBuffer[i];
    }

    for (byte i = 75; i < 100; i++) {
      while (particleSensor.available() == false)
        particleSensor.check();

      redBuffer[i] = particleSensor.getRed();
      irBuffer[i] = particleSensor.getIR();
      particleSensor.nextSample();

      avgCnt ++;
      if (validSPO2 == 1) {
```

```
        String progress = String (map (avgCnt, 0, 50, 0, 100));
        Serial.println (progress + "%");
        if (avgCnt == 50) {
          startflag = 0;
          processFlag = 0;
          delay (500);
        }
      }
      else {
        avgCnt = 0;
      }
    }
    maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength,
        redBuffer, &spo2, &validSPO2, &heartRate, &validHeartRate);
  }
}


Serial.print("Calibrating for Temperature");
processFlag = true;
particleSensor.setup(0);
particleSensor.enableDIETEMPRDY();
delay (1500);

temperatureF = 0;
while (processFlag == true) {
  for (int dl = 0; dl <= 30; dl++) {
    String progress = String (map (dl, 0, 30, 0, 100));
    Serial.println (progress + "%");
    temperatureF += particleSensor.readTemperatureF();
    delay (100);
  }
  temperatureF = temperatureF / 30;
  processFlag = false;
  delay (1500);
```

```
    }

    Serial.println ("Heart Rate: " + String (HRbpm) + " bmp");
    Serial.println ("Oxygen Saturation: " + String (spo2) + " %");
    Serial.println ("Temperature: " + String (temperatureF) + " F");
    continueflag = false;

    digitalWrite(ledPin,HIGH);
    delay(1000);
    digitalWrite(ledPin,LOW);
    delay(1000);


    MYSQL_DISPLAY("Connecting...");
    if (conn.connectNonBlocking(server, server_port, user, password) !=
       RESULT_FAIL) {
      delay(1000);
      //updateItemInScannedTable(object);
      dbConnect(String (spo2) , String (HRbpm),String (temperatureF));
      // sprintf(query, "INSERT INTO %s.%s (humidity, temp, pressure,
         distance,rainfall) VALUES ('%.2f', '%.2f', '%.2f', '%.2f', '%.2f')
         ",
      // default_database, default_table, h, t,pressure,bmp.readPressure(),
         distanceCm,rainAnalogVal);
      conn.close();
      // digitalWrite (LED_Pin, LOW);
    } else {
      MYSQL_DISPLAY("\nConnect failed. Trying again on the next iteration.");
    }
  }


  delay(1000); /* Wait for 1000mS */
}
```

Appendix B

DATA PREPROCESSING WITH MODEL TRAINING AND MODEL TESTING

```
import pandas as pd
df=pd.read_csv('Training.csv')
df.head()

df.info()

df.isna().sum()

df.duplicated().sum()

df.nunique()

df.drop_duplicates(inplace=True)

with open('map.txt', 'w', encoding='utf-8') as f:
    f.write(str(maps))
y=df.iloc[:,-1:]
y.head()

x=df.iloc[:,:-1]
x.tail()

y=df['prognosis']
from collections import Counter#checking the dataset balance
Counter(y)

from imblearn.over_sampling import RandomOverSampler
ros = RandomOverSampler()
xres,yres = ros.fit_resample(x,y)
```

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(xres,yres,train_size=0.8,
   random_state=40)

from sklearn.metrics import accuracy_score, classification_report,
   confusion_matrix
from sklearn.neural_network import MLPClassifier
import joblib

# Define and train the MLP classifier
clf_mlp = MLPClassifier(hidden_layer_sizes=(10,), random_state=42)
clf_mlp.fit(xtrain, ytrain)

# Make predictions on the test set
ypred_mlp = clf_mlp.predict(xtest)



# Save the trained MLP model
joblib.dump(clf_mlp, "mlp.pkl")

accuracy = accuracy_score(ytest, clf_mlp.predict(xtest))
print("Accuracy :", accuracy)
```

Model Testing

```
import pandas as pd
import numpy as np
import joblib
dd=pd.read_csv('testing.csv')


df=dd.iloc[40:41,:-1]
```

```
df.isna().sum()

import numpy as np

with open('map.txt', 'r', encoding='utf-8') as f:
    content = f.read()

content = content.replace("nan", "np.nan")
maps = eval(content)

model=joblib.load("mlp.pkl")

import pandas as pd



df_encoded = pd.get_dummies(df)

pred = model.predict(df_encoded)

predicted_disease = prognosis_mappings.get(pred[0], "Unknown")
print("Predicted Disease:", predicted_disease)
```

Appendix C

SCREENSHOTS

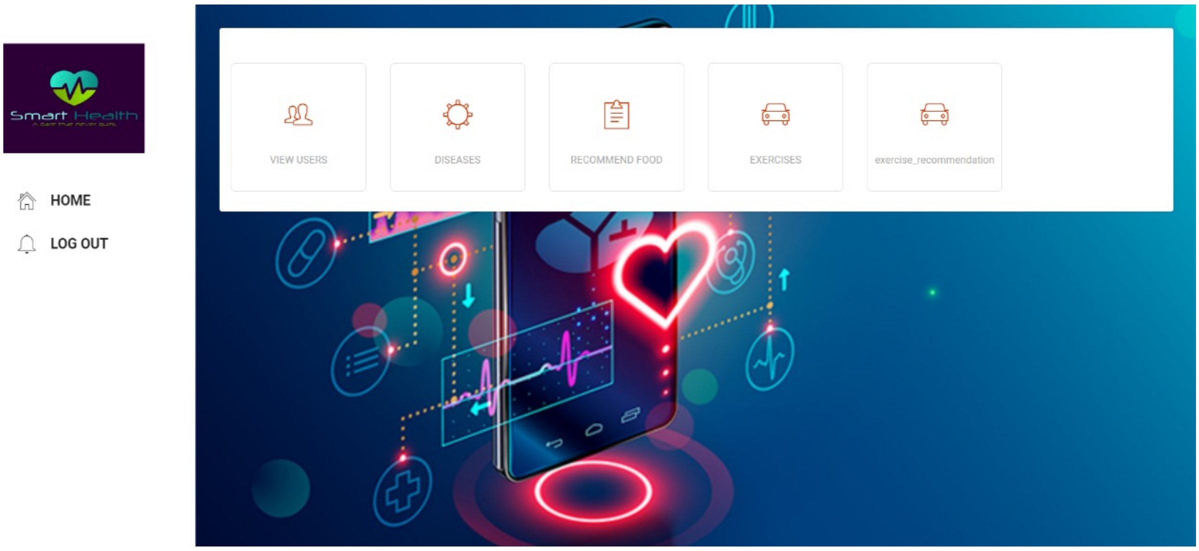
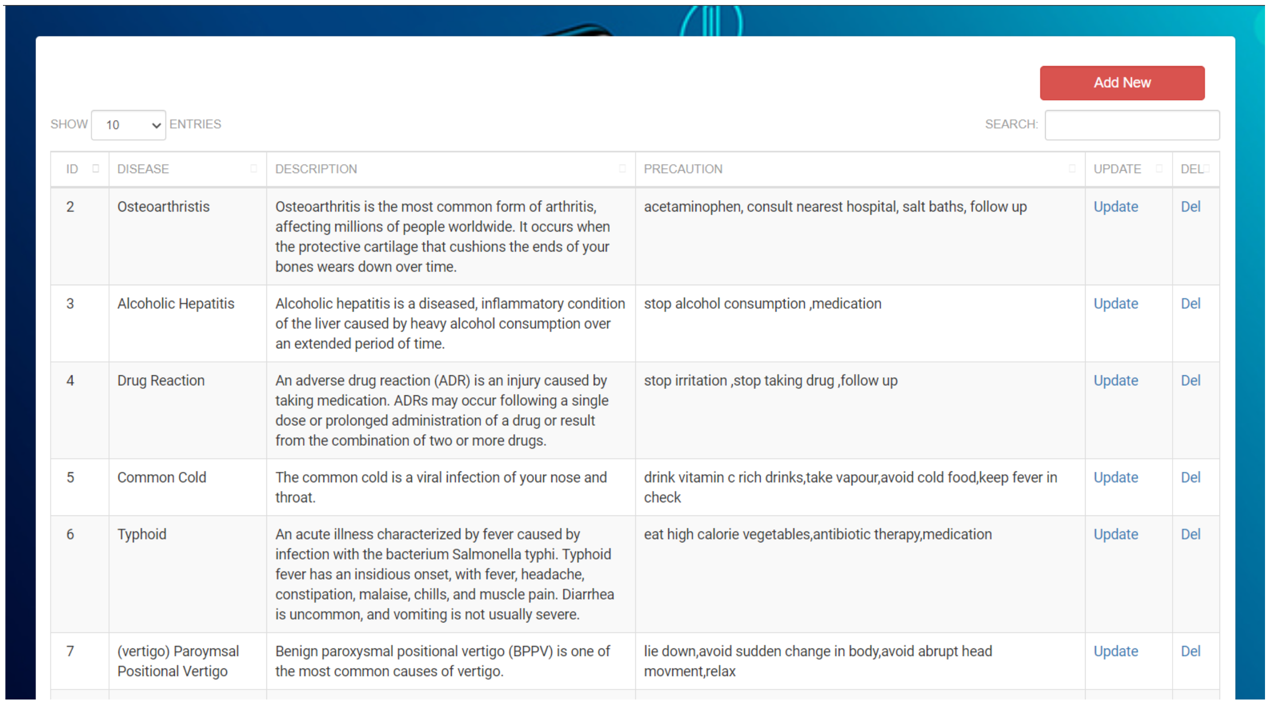


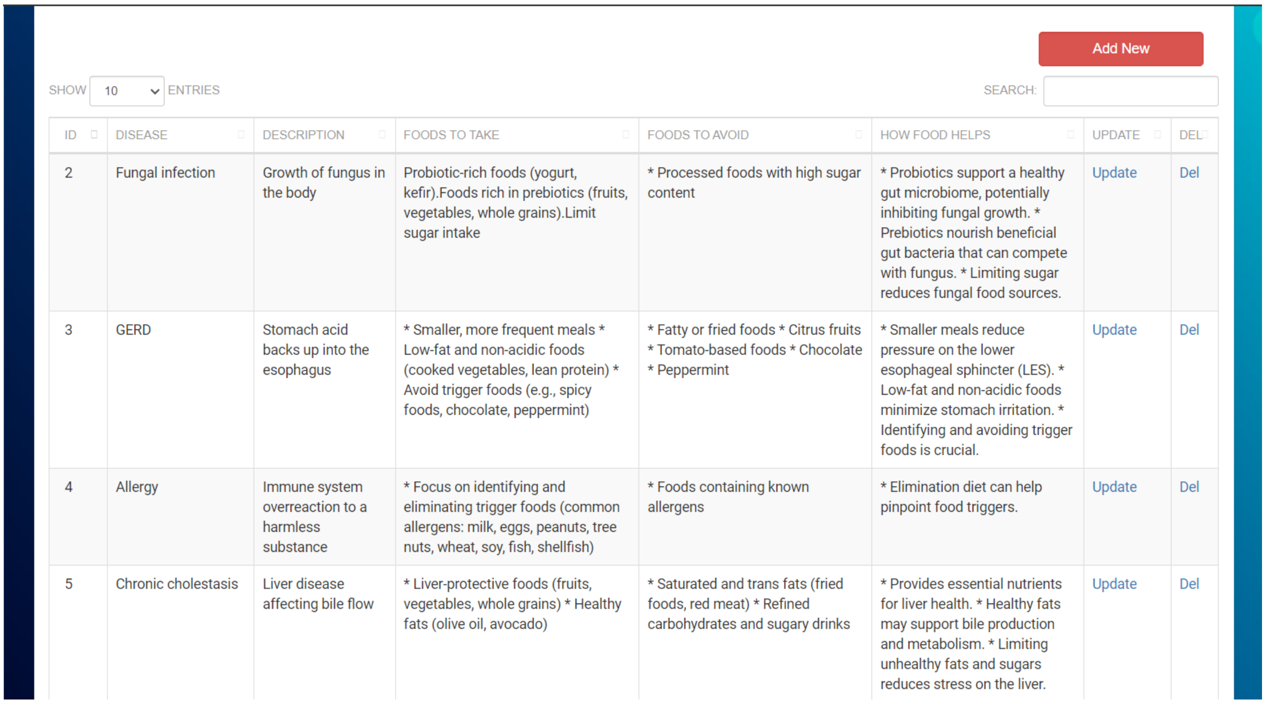
Figure 1: Admin Dash Board


Add New

SHOW 10 ENTRIES SEARCH:

ID	DISEASE	DESCRIPTION	PRECAUTION	UPDATE	DEL
2	Osteoarthristis	Osteoarthritis is the most common form of arthritis, affecting millions of people worldwide. It occurs when the protective cartilage that cushions the ends of your bones wears down over time.	acetaminophen, consult nearest hospital, salt baths, follow up	Update	Del
3	Alcoholic Hepatitis	Alcoholic hepatitis is a diseased, inflammatory condition of the liver caused by heavy alcohol consumption over an extended period of time.	stop alcohol consumption ,medication	Update	Del
4	Drug Reaction	An adverse drug reaction (ADR) is an injury caused by taking medication. ADRs may occur following a single dose or prolonged administration of a drug or result from the combination of two or more drugs.	stop irritation ,stop taking drug ,follow up	Update	Del
5	Common Cold	The common cold is a viral infection of your nose and throat.	drink vitamin c rich drinks,take vapour,avoid cold food,keep fever in check	Update	Del
6	Typhoid	An acute illness characterized by fever caused by infection with the bacterium Salmonella typhi. Typhoid fever has an insidious onset, with fever, headache, constipation, malaise, chills, and muscle pain. Diarrhea is uncommon, and vomiting is not usually severe.	eat high calorie vegetables,antibiotic therapy,medication	Update	Del
7	(vertigo) Paroymsal Positional Vertigo	Benign paroxysmal positional vertigo (BPPV) is one of the most common causes of vertigo.	lie down,avoid sudden change in body,avoid abrupt head movment,relax	Update	Del

Figure 2: User Records in Admin Page



Add New

SHOW 10 ENTRIES SEARCH:

ID	DISEASE	DESCRIPTION	FOODS TO TAKE	FOODS TO AVOID	HOW FOOD HELPS	UPDATE	DEL
2	Fungal infection	Growth of fungus in the body	Probiotic-rich foods (yogurt, kefir).Foods rich in prebiotics (fruits, vegetables, whole grains).Limit sugar intake	* Processed foods with high sugar content	* Probiotics support a healthy gut microbiome, potentially inhibiting fungal growth. * Prebiotics nourish beneficial gut bacteria that can compete with fungus. * Limiting sugar reduces fungal food sources.	Update	Del
3	GERD	Stomach acid backs up into the esophagus	* Smaller, more frequent meals * Low-fat and non-acidic foods (cooked vegetables, lean protein) * Avoid trigger foods (e.g., spicy foods, chocolate, peppermint)	* Fatty or fried foods * Citrus fruits * Tomato-based foods * Chocolate * Peppermint	* Smaller meals reduce pressure on the lower esophageal sphincter (LES). * Low-fat and non-acidic foods minimize stomach irritation. * Identifying and avoiding trigger foods is crucial.	Update	Del
4	Allergy	Immune system overreaction to a harmless substance	* Focus on identifying and eliminating trigger foods (common allergens: milk, eggs, peanuts, tree nuts, wheat, soy, fish, shellfish)	* Foods containing known allergens	* Elimination diet can help pinpoint food triggers.	Update	Del
5	Chronic cholestasis	Liver disease affecting bile flow	* Liver-protective foods (fruits, vegetables, whole grains) * Healthy fats (olive oil, avocado)	* Saturated and trans fats (fried foods, red meat) * Refined carbohydrates and sugary drinks	* Provides essential nutrients for liver health. * Healthy fats may support bile production and metabolism. * Limiting unhealthy fats and sugars reduces stress on the liver.	Update	Del

Figure 3: Disease Dataset

Add New

SHOW 10 ENTRIES SEARCH:

ID	DISEASE	DESCRIPTION	FOODS TO TAKE	FOODS TO AVOID	HOW FOOD HELPS	UPDATE	DEL
2	Fungal infection	Growth of fungus in the body	Probiotic-rich foods (yogurt, kefir).Foods rich in prebiotics (fruits, vegetables, whole grains).Limit sugar intake	* Processed foods with high sugar content	* Probiotics support a healthy gut microbiome, potentially inhibiting fungal growth. * Prebiotics nourish beneficial gut bacteria that can compete with fungus. * Limiting sugar reduces fungal food sources.	Update	Del
3	GERD	Stomach acid backs up into the esophagus	* Smaller, more frequent meals * Low-fat and non-acidic foods (cooked vegetables, lean protein) * Avoid trigger foods (e.g., spicy foods, chocolate, peppermint)	* Fatty or fried foods * Citrus fruits * Tomato-based foods * Chocolate * Peppermint	* Smaller meals reduce pressure on the lower esophageal sphincter (LES). * Low-fat and non-acidic foods minimize stomach irritation. * Identifying and avoiding trigger foods is crucial.	Update	Del
4	Allergy	Immune system overreaction to a harmless substance	* Focus on identifying and eliminating trigger foods (common allergens: milk, eggs, peanuts, tree nuts, wheat, soy, fish, shellfish)	* Foods containing known allergens	* Elimination diet can help pinpoint food triggers.	Update	Del
5	Chronic cholestasis	Liver disease affecting bile flow	* Liver-protective foods (fruits, vegetables, whole grains) * Healthy fats (olive oil, avocado)	* Saturated and trans fats (fried foods, red meat) * Refined carbohydrates and sugary drinks	* Provides essential nutrients for liver health. * Healthy fats may support bile production and metabolism. * Limiting unhealthy fats and sugars reduces stress on the liver.	Update	Del

Figure 4: Table showing Disease,Description and its precautions.

Add New

SHOW 10 ENTRIES SEARCH:

ID	BODY PART	EQUIPMENT	NAME	TARGET	INSTRUCTIONS	UPDATE	DEL
1	waist	body weight	3/4 sit-up	abs	Lie flat on your back with your knees bent and feet flat on the ground. Place your hands behind your head with your elbows pointing outwards. Engaging your abs slowly lift your upper body off the ground; curling forward until your torso is at a 45-degree angle. Pause for a moment at the top; then slowly lower your upper body back down to the starting position. Repeat for the desired number of repetitions.	Update	Del
2	waist	body weight	45Â° side bend	abs	Stand with your feet shoulder-width apart and your arms extended straight down by your sides.Keeping your back straight and your core engaged; slowly bend your torso to one side lowering your hand towards your knee.Pause for a moment at the bottom then slowly return to the starting position.Repeat on the other side.;Continue alternating sides for the desired number of repetitions.	Update	Del
3	upper legs	body weight	all fours squad stretch	quads	Start on all fours with your hands directly under your shoulders and your knees directly under your hips.Extend one leg straight back keeping your knee bent and your foot flexed.Slowly lower your hips towards the ground feeling a stretch in your quads.Hold this position for 20-30 seconds.Switch legs and repeat the stretch on the other side.	Update	Del
4	back	cable	alternate lateral pulldown	abs	Sit on the cable machine with your back straight and feet flat on the ground.Grasp the handles with an overhand grip slightly wider than shoulder-width apart.Lean back slightly and pull the handles towards your chest squeezing your shoulder blades together.Pause for a moment at the peak of the movement then slowly release the handles back to the starting position.Repeat for the desired number of repetitions.	Update	Del

Figure 5: Exercise Data

SHOW 10 ENTRIES SEARCH:

ID	DISEASE	PREVENTION TIPS	EXERCISES	MEDICATION	UPDATE	DEL
1	Fungal infection	Keep skin dry and clean. Wear breathable clothing and shoes. Avoid tight-fitting clothing. Practice good hygiene. Don't share personal items like towels or shoes.	No specific excercises applicable	Topical antifungal medications (creams, ointments). Oral antifungal medications in severe cases (prescribed by doctor).	Update	Del
2	GERD	Maintain a healthy weight. Eat smaller, more frequent meals. Avoid trigger foods (spicy, fatty, acidic). Don't eat close to bedtime. Elevate the head of your bed.	Light cardio exercises like walking or swimming can help manage weight and improve digestion (consult doctor before starting).	Medications to reduce stomach acid (proton pump inhibitors, H2 blockers).	Update	Del

Figure 6: Food Recommendation

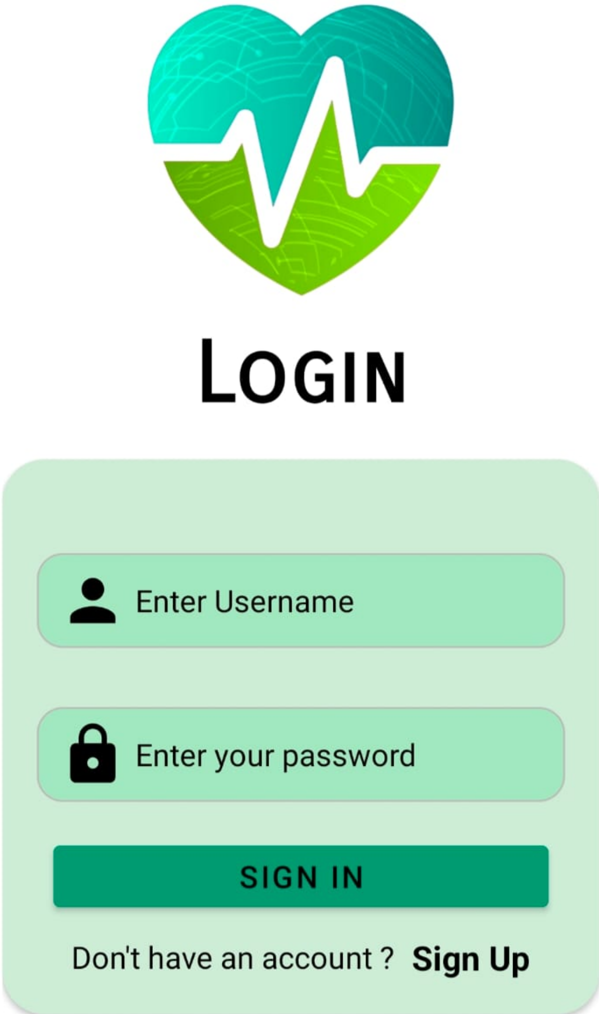


Figure 7: User Login

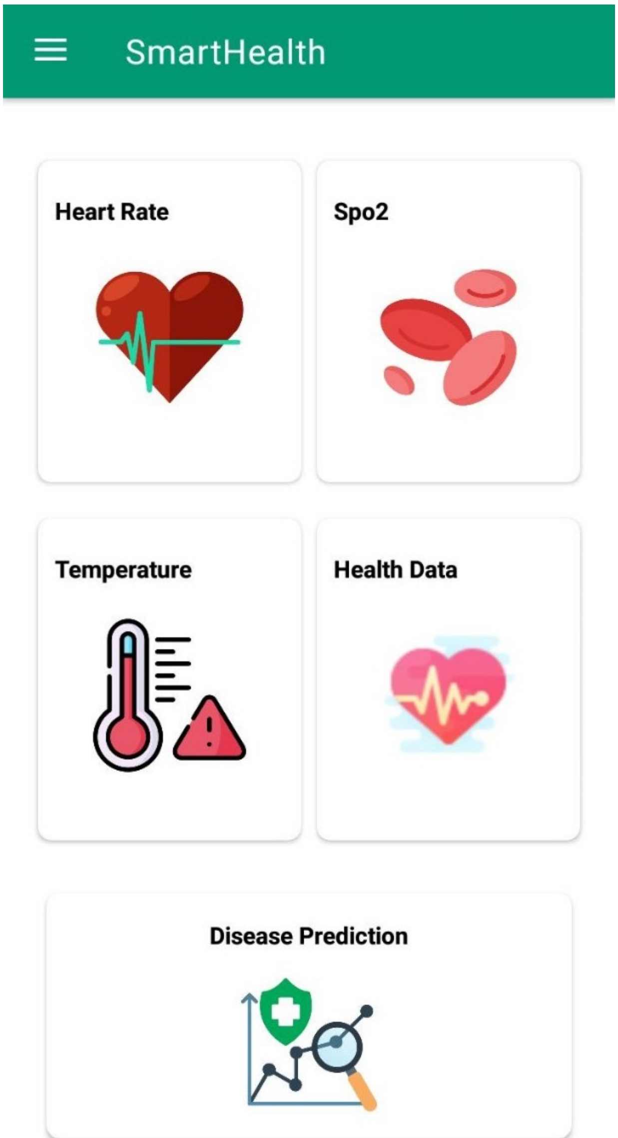


Figure 8: User Dashboard

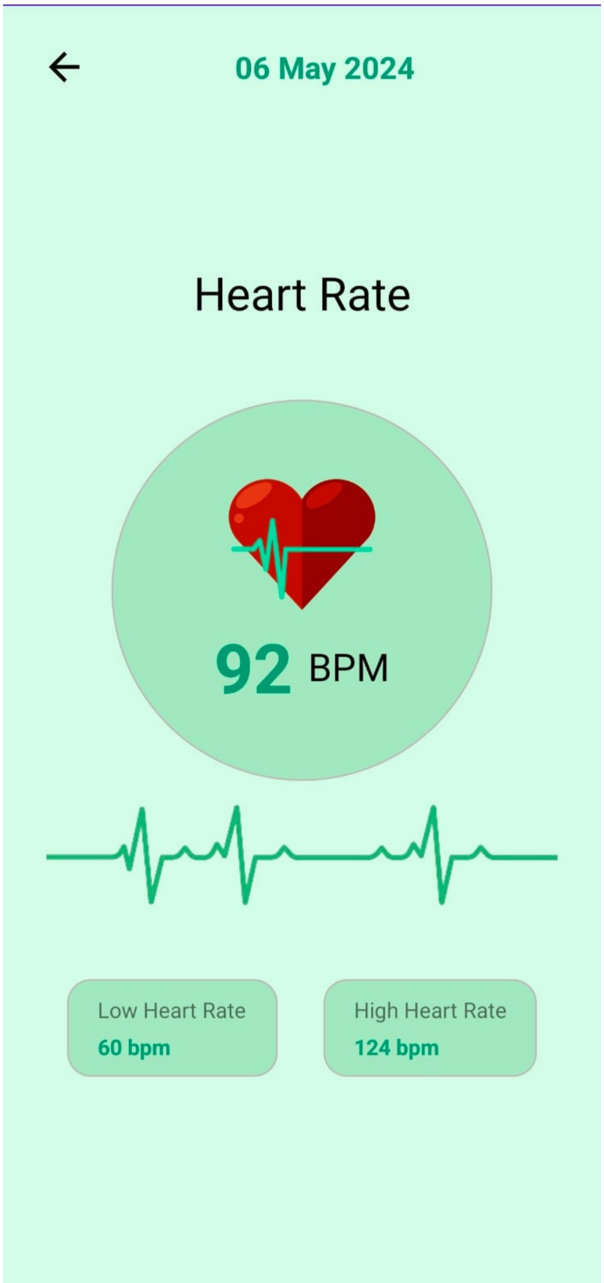


Figure 9: Heart Rate

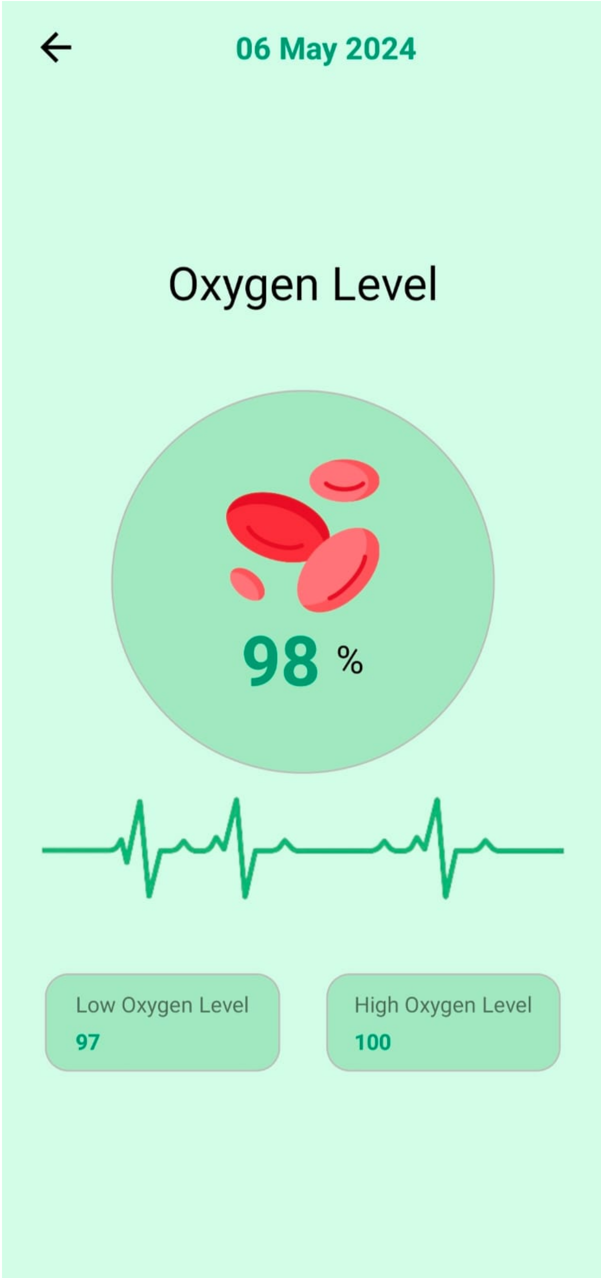


Figure 10: Blood Oxygen Level

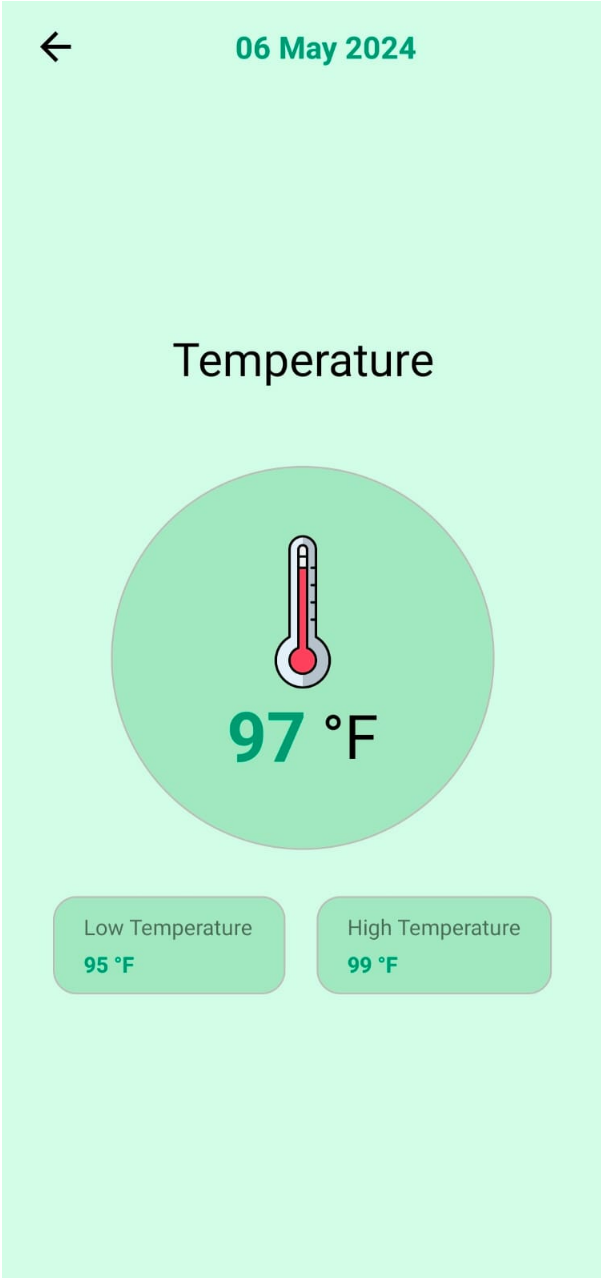


Figure 11: Temperature Status

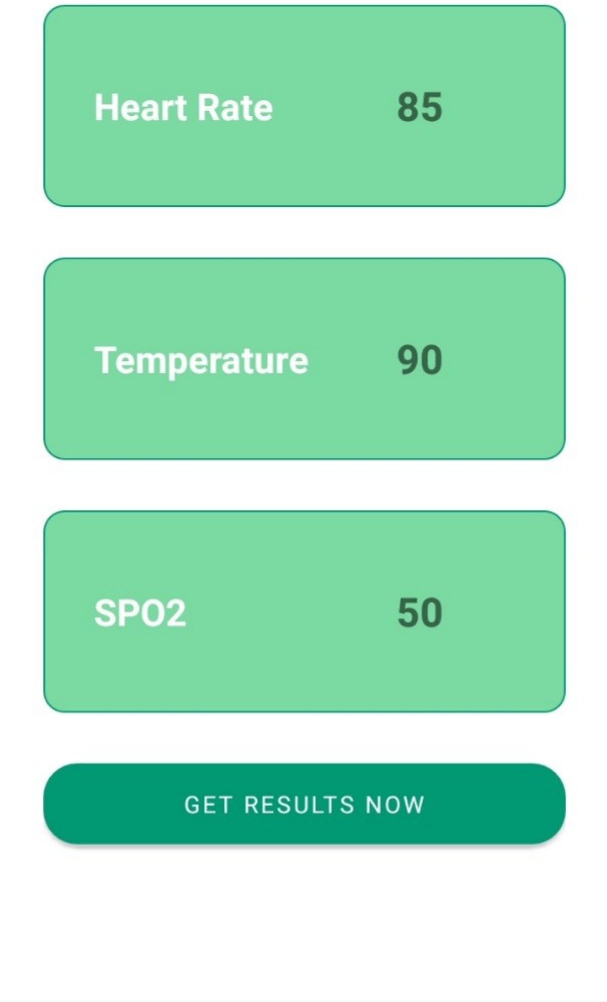


Figure 12: Health data

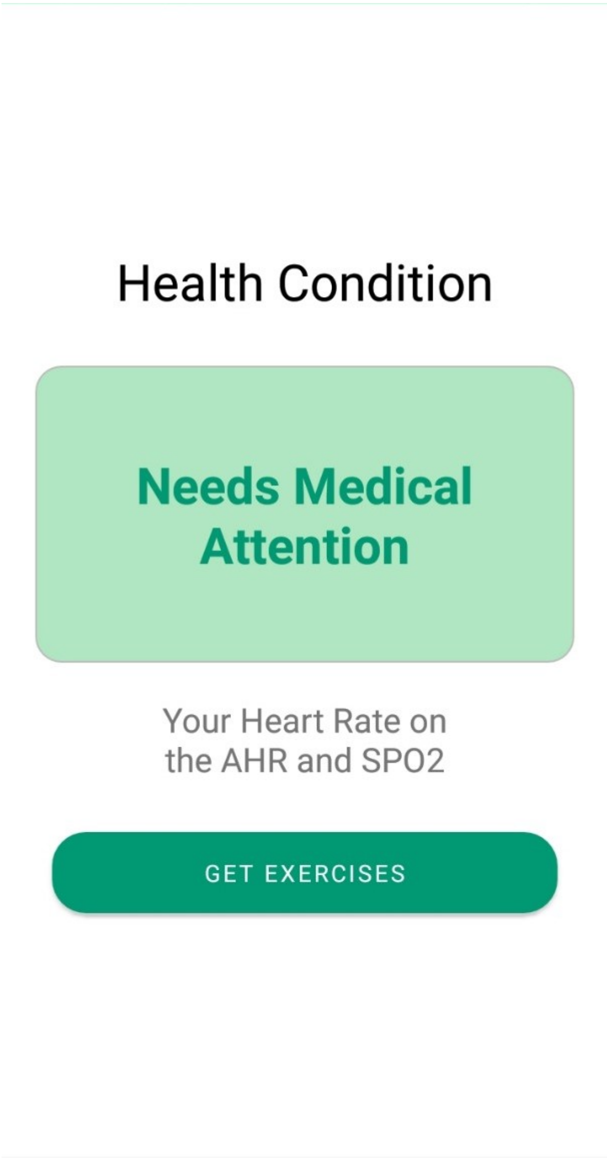


Figure 13: Health Status

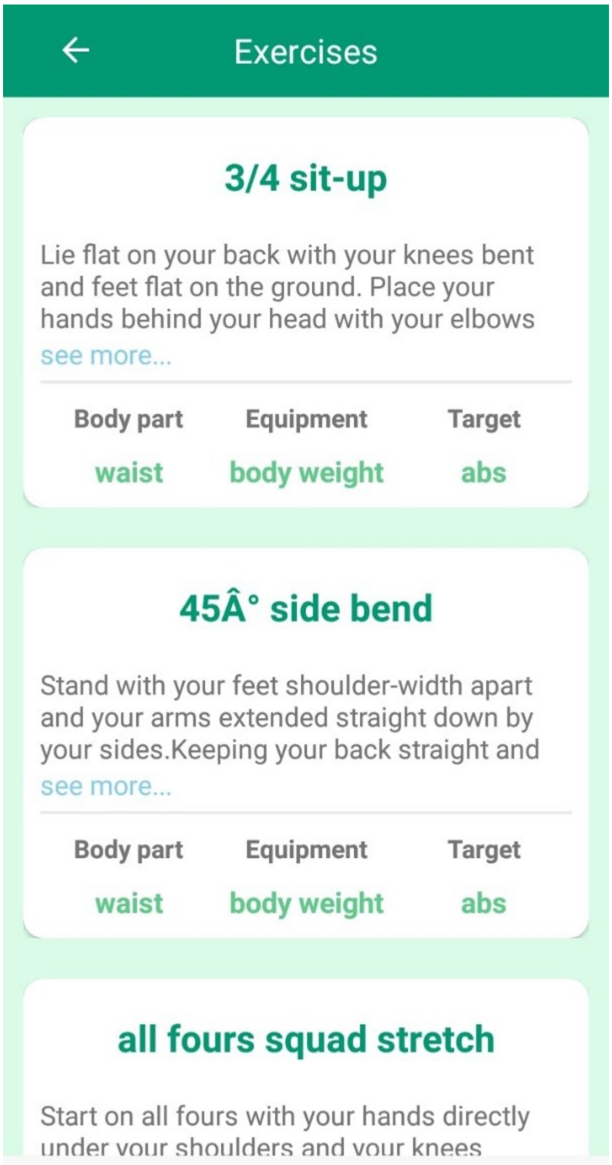


Figure 14: List of Exercises

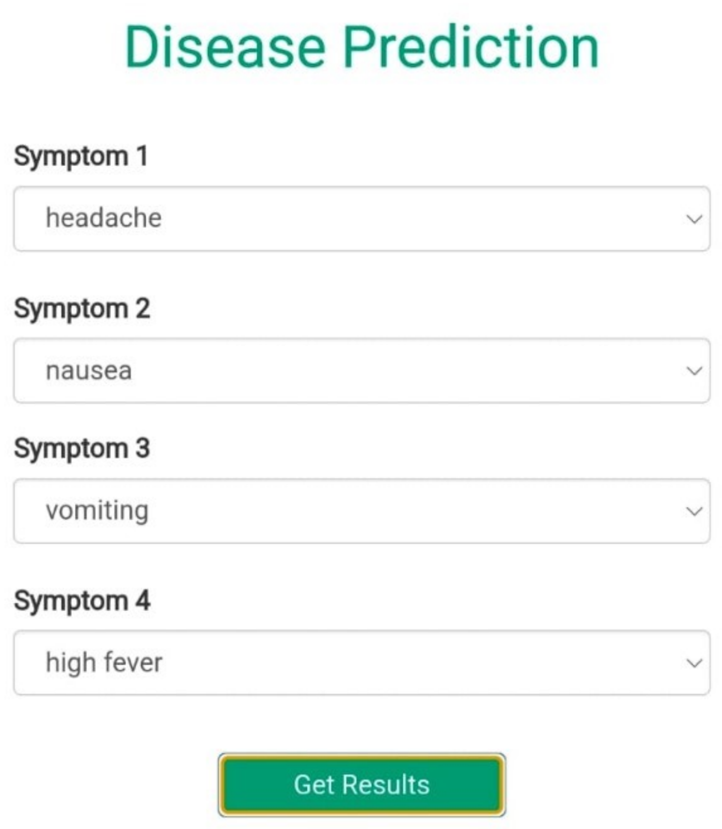


Figure 15: Symptom Dropdown

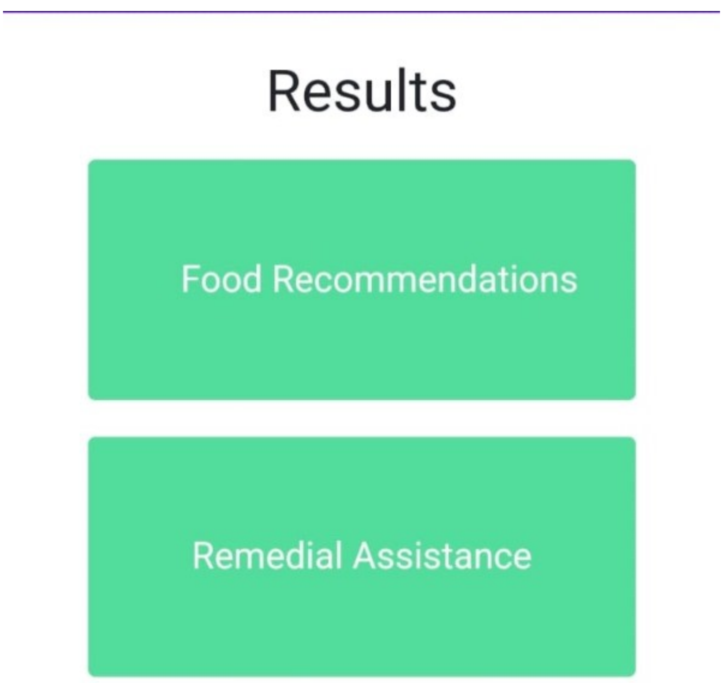


Figure 16: Result of Predicted Diseases

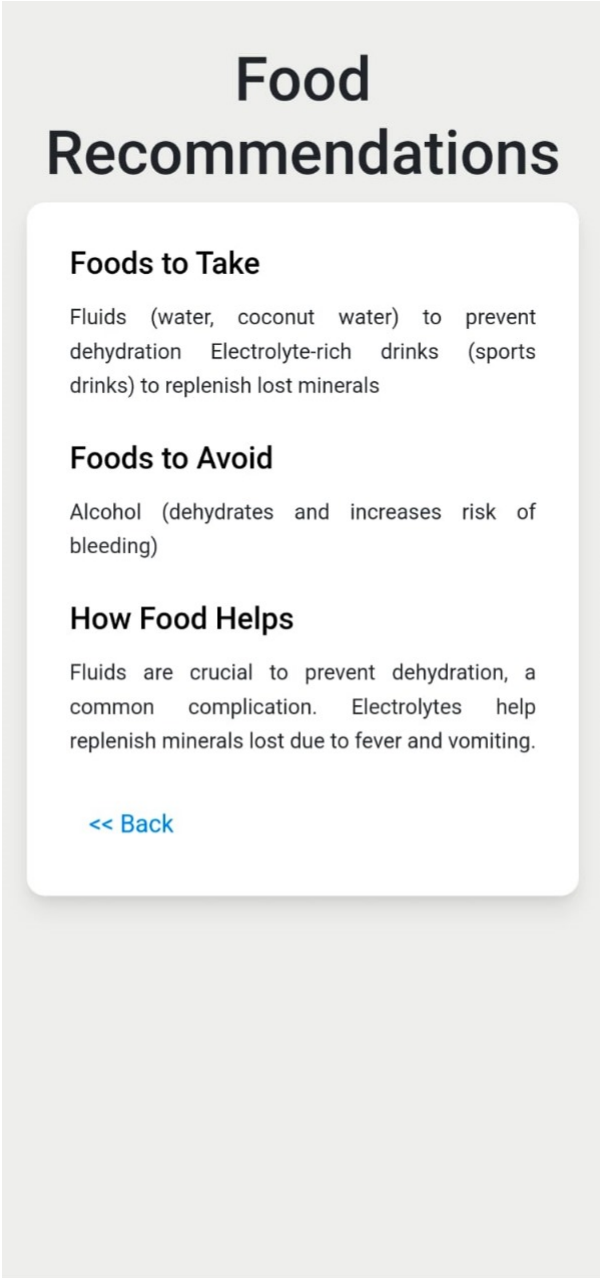


Figure 17: Food Recommendation

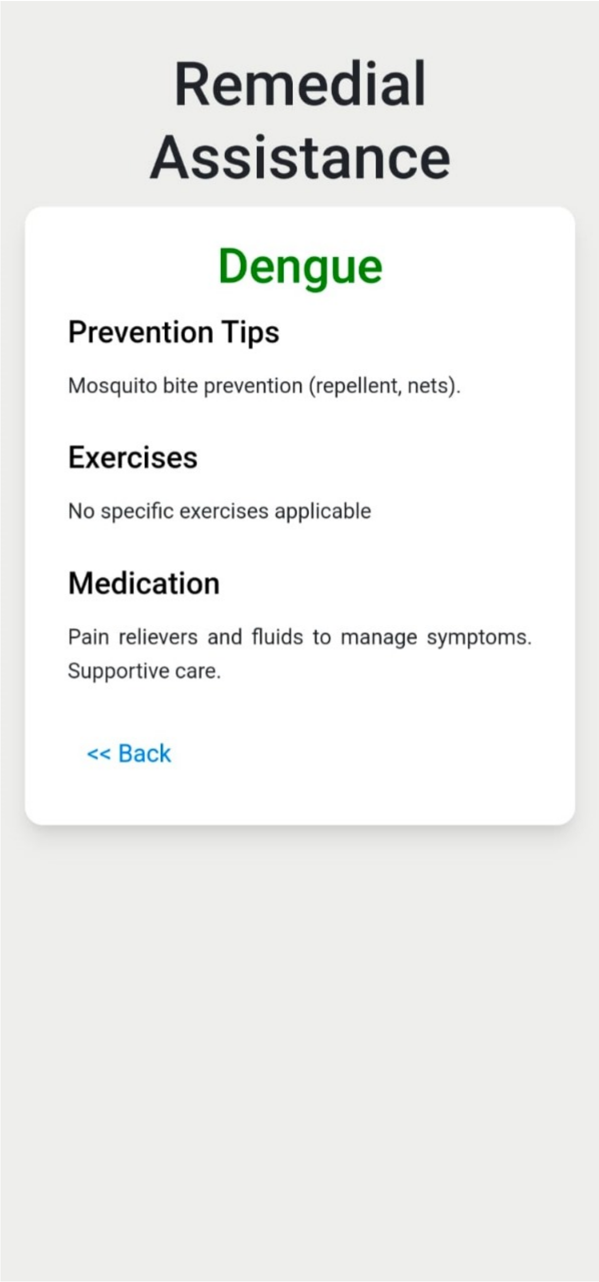


Figure 18: Remedial Assistance