UNIVERSITY of RWANDA
College of Science and Technology

Website: www.aceiot.ur.ac.rw
Mail: aceiot@ur.ac.rw

AFRICAN CENTER OF EXCELLENCE IN INTERNET OF THINGS

Research Thesis Title:

Developing an IoT-based Conversational AI Recommender Assistant for Vital Sign Predicted Anomalies

A dissertation** submitted **in partial fulfilment of the requirements for the award of masters of science degree in internet of things: wireless intelligent sensor network or Embedded computing system

Submitted By :
Name: Akram Ali Omar (Ref. No: 221000559)

December, 2022

UNIVERSITY of RWANDA
College of Science and Technology

ACEIoT
Website: www.aceiot.ur.ac.rw
Mail: aceiot@ur.ac.rw

AFRICAN CENTRE OF EXCELLENCE IN INTERNET OF THINGS

Research Thesis Title: Developing an IoT-based Conversational AI Recommender Assistant for Vital Sign Predicted Anomalies

A dissertation submitted in partial fulfilment of the requirements for the award of masters of science degree in internet of things: wireless intelligent sensor network or Embedded computing system

Submitted By**:**
Akram Ali Omar (REF.NO: 221000559)

Supervised by:
- Dr. Jimmy Nsenga
- Dr. Frederic Nzanywanyingoma

December, 2022

Declaration

I, Akram Ali Omar, Master ‘student at African Center of Excellence in internet of things(ACEIoT), College of Science and Technology (CST), University of Rwanda. I declare that this research thesis is my own original work and it has never been presented before anywhere in the world.

Akram Ali Omar

Ref: 221000559

Signed: ………………………….

Date: ……/……/……

Bonafide certificate

This is to certify that this submitted Research Thesis work report is a record of the original work done by **Akram Ali Omar** (**Ref. Nu: 221000559**), M.Sc. of Internet of Things with Embedded Computing Systems (ECS) Student at College of Science and Technology (CST), University of Rwanda, the Academic year 2020/2022.

This work has been submitted under the supervision:

Main Supervisor: Dr. Jimmy Nsenga

Date: 20/12/2022

Signature :

Co-Supervisor: Dr. Frederic Nzanywanyingoma

Date: 20/12/2022

Signature:

The Head of Masters and Trainings

Dr. James Rwigema

Signature:

Date:

Acknowledgement

Thanks to the African Centre of Excellence, Internet of Things (ACEIoT), who supported me from the first day of my studies and IA as a whole. I extend my thanks to the Director of the ACEIoT, Professor. Damien HANYURWIMFURA, Head of Masters' studies Dr. RWIGEMA James and Postgraduate & Research Administrator, Mr. BENJAMIN Hakizimana. I also extend my supreme gratitude to my supervisors Dr. Frederic Nzanywanyingoma and Dr. Jimmy Nsenga for their invaluable support and advice during the whole duration of my research project. A special thanks go to Mr. Samuel Rutunda from Digital Umuganda and Mr Marvin Muyango a Master graduate student from ACEIoT, THEOFRIDA, MAGINGA a Phd student at ACEIoT for collaborate and share their knowledge and experience with me during the whole process of my research project. Finally, I would like to express my special thanks to my wives for the sincere support they have given me during the two year of my masters studies.

Abstract

In most real-time scenarios such as emergency first response or a patient self-monitoring using a wearable device, it is likely that accessing a healthcare physician for assessing potential vital sign anomalies and provide a recommendation will be impossible; thus potentially putting the patient at risk. Leveraging the latest advances in Natural Language Processing (NLP), this study presents a research-driven design and development of a cloud-based conversational AI platform trained to predict vital signs anomalies and provides recommendations from a dataset created by physicians. To reinforce the learning of the virtual assistant, the Conversation Driven Development (CDD) methodology has been adopted to involve end users in the testing process in the early phase. The proposed platform will help to manage the consequences of low physician-patient ratios especially in developing countries. A part from this thesis. I have already submitted my first paper about my research project. The paper was submitted to conference 8th International Conference on Machine Learning Technologies (ICMLT 2023) which has already been accepted for publication.

Table of Content

Contents

List of Figures

List of Tables

List of Acronyms

ACEIoT: African Center of Excellence in Internet of Things

AI: Artificial Intelligence

GSM: Global System for Mobile

HW: Hardware

IDE: Integrated Development Environment

IoT: Internet of Things

ANN: Artificial Neural Network

K- NN: K- Nearest Neighbors

DNN: Deep Neural Network

LCD: Liquid Crystal Display

MCU: Microcontroller Unit

ML: Machine Learning

SW: Software

NLP: Natural Language Processing

NLU: Natural Language Understanding

CDD: Conversation Driven Development

BP: Blood Pressures

WSN: Wireless Sensor Networks

nCOVID-19: Corona Virus disease of 2019

IP: Internet Protocol

TCP: Transmission Control Protocol

GPRS: General Packet Radio Services

UART: Universal Asynchronous Receiver-Transmitter

RGB: Red Green Blue

OLED: Organic Light-Emitting Diode

I2C: Inter-Integrated Circuit

LED: Light-Emitting Diode

CSV: Comma-Separated Values

JSON: JavaScript Object Notation

HTTP: Hypertext Transfer Protocol

OEM: Original Equipment Manufacturer

PDL: Program Description Language

SDK: Software Development Kit

HSRF: Health Service Recommendation Framework

KNN-LS-SVM: K-nearest neighbors (K-NN) Based Least Squares Support Vector Machine (LS-SVM)

DDTRS: Disease Diagnosis and Treatment Recommendation System

DPCA: Density-Peaked Clustering Analysis

API: Application Programming Interface

CHAPTER 1

INTRODUCTION

1.1 Introduction

Vital signs are measurements of the body's essential physiological functions that provide critical information about the health state of patients [3]. During a clinical examination of the patient, the measurement of vital signs such as blood pressure, respiratory rate, temperature, and so on plays a major role in determining if the patient is at risk of deterioration during an emergency [1]. Monitoring vital signs is crucial to determine whether or not a patient is in good health, potentially unhealthy, or at risk of health deterioration [1]. The vital signs data are also relied on to define which medical intervention is required to stabilize the patient's condition [3] and may also influence the decision-making of triage teams, especially for critically ill casualties [4].

The integration of health-based IoT devices with chatbots greatly contributes to providing a quality health service [6]. IoT devices can only collect a large amount of data from the physical environmental domain, it cannot address all challenges especially when it comes to monitoring the health status of the patient [6]. According to [6] the integration of chatbots to IoT can overcome many challenges such as interoperability, device management, and application, data management and context, user interaction with IoT systems, searching and discoverability, monitoring, and reporting, a cognitive burden in education the end users, configuration, lack of automated error reporting and support. AI chatbots as intelligent systems can assist the patient as well as collect important information from the patient for assessment. It improves the remote diagnosis and interaction between doctors and patients in a simple and natural way and involves patients in labeling abnormal patterns in recorded data using natural language.

1.2 Background and Motivation

Unexpected pattern changes or anomalies in a patient's vital signs can infer a potential health problem which if not identified and/or treated on time can result in adverse consequences and even death [2]. According to a retrospective record review study that was performed from June 2020 to January 2021 at the ED of King Abdul-Aziz University Hospital (KAUH) [5], the patients with urgent and less argent respiration rate has higher mortality rate with percentage of 42.9% and 11.1% respectively. Those with urgent and less urgent oxygen saturation has significantly higher mortality rate with the percentage of 14.3% and 36.6% respectively for as shown in Figure 1. The patients with low systolic blood pressure and those with high systolic blood pressure have a motality rate of 9.5% and 7.1% respectively. The patients with low and high diastolic blood pressures (BP) have a significantly higher mortality rate with a percentage of 10.2% and 5.4% respectively as shown in Figure 2. From the review, we can conclude that urgent changes in vital signs can increase the risk of death.

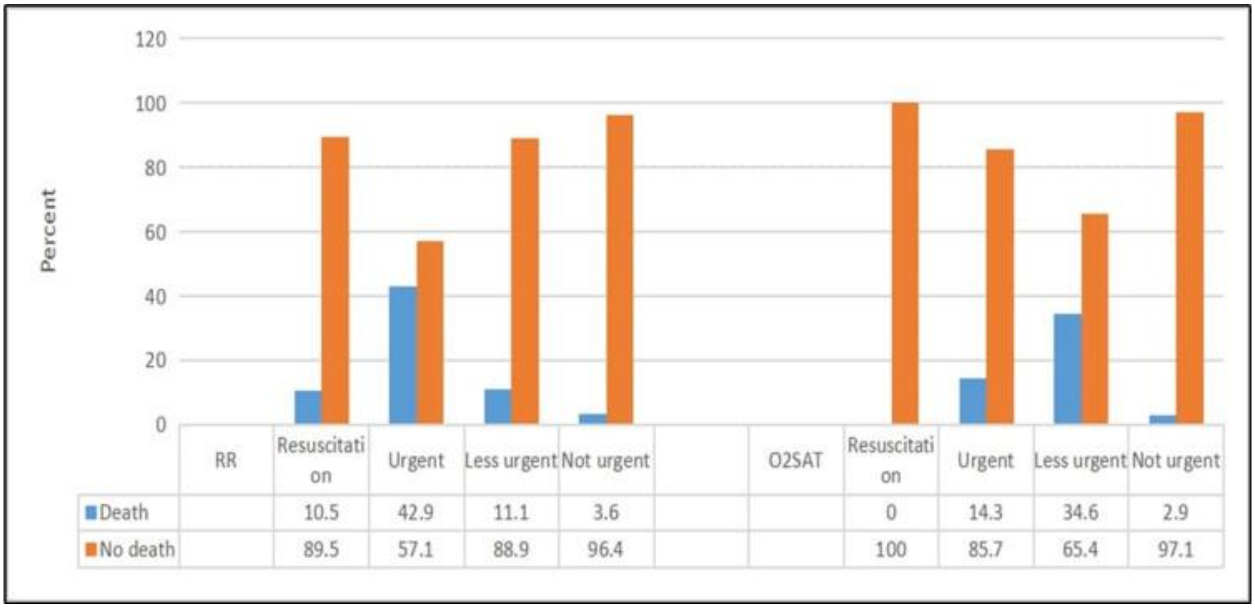


Figure 1:Relationship between death and respiratory rate and oxygen saturation

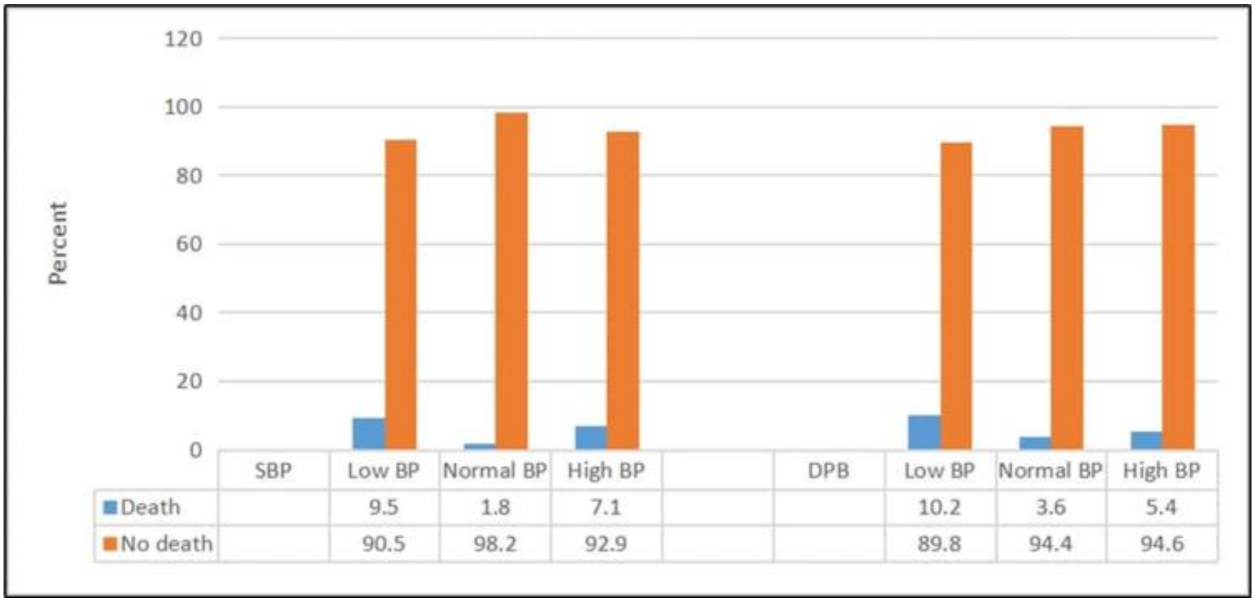


Figure 2: Relationship between death and systolic and diastolic blood pressures

Vital signs data can help health experts to know which medical intervention is required to stabilize the patient's condition [3]. Vital signs may also influence patient treatment as well as the decision-making of triage teams, especially for critically ill patients [4]. According to [7], five common vital signs play an important role in patient status which are: body temperature, blood pressure, blood oxygen saturation(Spo2), heart rate, and respiration rate. His study focuses on 3 vital signs which are temperature, pulse rate, and blood oxygen saturation.

The vital signs benefits may be easily and efficiently achievable if the vital signs are monitored and disseminated in real-time with the use of embedded technologies [7]. With embedded technology advancement, embedded systems are cheaper, have small sizes, and have high performance [7]. This provides a portable and cheaper medical device with the opportunity to automate medical procedures for the patients to self-service care by themselves to minimize hospitalizations and doctor visits. Therefore, as the vital signs device can be possessed by end users of different backgrounds and disciplines, this creates a need to have an intelligent system that can assist such categories of users in result interpretation like a physician. This system is called a virtual assistant.

During the vital signs anomaly prediction, the virtual assistant will be aware of the user context so that it can provide relevant details to the user via chat display. In the case of medical experts, health professionals, and nurses the virtual assistant can disseminate vital readings, anomalies, and recommendations in a more technical way. But for non-expert users at home, the system can diagnose the vital signs values, and anomalies as well as provide home remedies recommendations using natural language

1.3 Problem Statement

In many cases, vital signs data are analyzed by human experts such as nurses and physicians. As stated by [1], this dependency creates a bottleneck for real-time prediction of vital signs anomalies for a large number of patients, anytime and anywhere. The causes of this bottleneck are: (1) a few numbers of health professionals capable of determining vital signs changes and their implications, (2) unavailability of specialized experts at the desired moment to provide medical support while busy with other patients or other critical responsibilities, and (3) poor or lack of communication network to enable efficient remote communication between the place where the vital signs are being collected and healthcare professionals at the hospital. The real-time and rapid prediction requirement becomes very important in some emergency scenarios when a patient shows signs of clinical deterioration that may lead to death. It is, therefore, necessary to explore technology solutions that may enable the prediction of vital signs data with less or without dependencies to busy health experts and propose recommendations in case of anomaly. Even though many solutions have been proposed currently for self-monitoring and self-interpretation of the vital signs, the challenges are still at large. To begin with self-monitoring of vital signs, most of the existing solutions are expensive and need some knowledge to interact with it. Secondly, most of existing solutions are relying on real physicians to provide medical recommendations to remote patients which is not an effective way in a situation where by the physician busy and cannot provide an immediate intervention upon emergency clinical deterioration. I addition, the solutions are collecting vital signs data without considering the contextual information from the patient. Last but not the least, some existing solutions are not made for home use provided that the solution still relies on various types of health data that may not be available from home.

1.4 Study Objectives

1.4.1 General Objective

The main objective of this project is to present research-driven design and development integrating the Internet of Things (IoT), AI, and Natural Language Processing (NLP) to build an autonomous conversational AI assistant that monitors patients' vital signs, predicts anomalies, and provides recommendations of the best practices for handling vital signs abnormalities.

1.4.2 Specific Objectives

To achieve the above goal, the following steps have to be accomplished:

i) Review state-of-the-art research works related to the studied challenge.
ii) Designing and deploying the Conversational AI Recommender Assistant integrated with embedded IoT device to leverage the vital signs ML diagnosis inference in the natural conversation with users
iii) Designing and deploying cloud based and age AI based model for vital signs anomaly detection
iv) Design and develop a rule-based recommendations web system for detected anomaly
v) Evaluating and publishing the results from the deployed solution prototype.

1.5 Hypotheses

This study assumed the following hypotheses:

1. The proposed solution can decrease the consequences of low physician-patient ratios.
2. The proposed solution can minimize unnecessary patient admission to the hospital.
3. The real-time vital signs AI anomaly detection can minimize the risk of the deterioration of the patient condition.

1.6 Study Scope

According to [8], five common vital signs plays an important role in patient status which are: body temperature, blood pressure, blood oxygen saturation(Spo2), heart rate, and respiration rate. Due to limited resources including physiological sensors and availability of datasets for an AI model training. This study was focusing on 3 vital signs which are temperature, pulse rate, and blood oxygen saturation and the anomaly detection was based on open datasets. In the case of anomaly recommendations, this study was based on rule-based recommendation system.

1.7 Significance of the Study

This study has been motived by the fact that, in most of developing countries, there are limited number of medical experts capable of determining and interpreting vital signs which may cause a bottle-neck in the situations (like accidents, emergencies, disasters, etc.) where there is a large number of patients needing urgent medical attention. In addition to that, there are also situations whereby the patient is at home and needs remote vital-signs monitoring as well as urgent remote medical. As the doctor-patient ratio continues to decrease especially in developing countries, relying on doctors for real-time dissemination of medical data such as vital signs becomes difficult, thus potentially putting at risk patients that may need rapid intervention. Therefore, conversational AI assistants offer a complementary solution to relieve the workload of doctors in dealing with recurring requests such as interpreting patients' vital signs and providing recommendations in case of anomaly.

This proposed study provides a solution with a lot of benefits such as the system will have the ability to monitor the patient's vital signs in real time. Furthermore, the study provides a conversational AI platform trained to predict vital signs anomalies and provides recommendations from a living recommendation dataset created by physicians. With these benefits, the solution will manage and overcome the consequences of low physician-patient ratios, especially in developing countries. Furthermore, the solution will reduce the patient's unnecessary trips to the hospital.

1.8 Organization of the Study

This work has been organized into six chapters that are highlighted as follows:

Chapter 1 introduces the whole idea of the proposed study which includes concepts used in this study. Chapter 2 presents a review of the state-of-the-art research works related to the studied challenge; Chapter 3 describes the proposed methodology used in the study, methodologies used for the ML process, recommendation system, virtual assistant, and integration of technologies used. Chapter 4 presents system analysis and system design which includes the high-level architecture of the proposed solution, system-level design, machine learning models, block diagram of vitals device, the architecture of Virtual assistant, and recommendation system designs. Chapter 5 discusses the results and analysis of project implementation and outputs. Chapter 6 is the last chapter which presents the conclusion and recommendations for further enhancements of the proposed solution.

1.9 Conclusion

Real-time self-monitoring of vital signs for individuals is not enough if it lacks remote medical attention or recommendations from a medical expert. This is the challenge that this study is trying to address. This study is proposing an IoT-based conversational AI recommender assistant for vital signs-predicted anomalies. The main idea of this study is to provide a solution that will allow the users to self-monitoring and interpretation of vital signs using a vital signs device without relying on existing medical experts. With the final goal of providing recommendations on the detected anomalies from the vital signs. The rest of this document highlights step by step process of archiving this goal.

2 CHAPTER 2

LITERATURE REVIEW

This chapter describes a review of relevant-related existing studies, the state-of-the-art research works related to the studied challenge, and an overview of existing technologies used to drive solutions for integration of IoT and conversation virtual assistants, and highlights the gap from the existing literature toward the studied challenges.

2.1 Vital signs monitoring and anomaly detection

The issue of real-time patient vital signs anomaly prediction has been paid attention to by many researchers, and several solutions with different technologies have been proposed. [5, 6] proposed a framework for vital signs anomaly detection in medical wireless sensor networks (WSN) without the intervention of health professionals. The framework conducts sequential data analysis on a base station (smartphone, computer) and alerts the healthcare profession upon any clinical deterioration. Medical experts are required for immediate intervention upon emergency clinical deterioration which sometimes may be busy and not able to respond on time. A real-time patient monitoring system was proposed in [3] which integrated with vital signs wearable sensors equipped with location tracking to monitor the patient's vital signs remotely. The real-time vital signs data are transmitted to a tablet device that runs a vital signs anomaly algorithm to detect anomalies. When anomalies are detected the tablet device generates an alert in the user interface. However, in a situation where a patient needs urgent medical attention, the system is not able to provide any medical instructions or recommendations on how to stabilize the conditions without relying on a medical professional to physically meet the patient. [9], [10], and [11] proposed a vital sign data transfer system that uses different psychological sensors to monitor vital signs during emergencies and sends the collected data to the hospital. However, these solutions failed to consider the automatic detection of anomalies on the transferred data, instead, the health professionals have to do it manually. Youssef et al. [12] proposed a wearable technology to investigate the monitoring of the vital signs of hospitalized patients. The solution also explores a hybrid machine learning algorithm of kNN-LS-SVM for predicting future vital signs values about one hour ahead.

2.2 Conversation AI virtual assistant in healthcare

Battineni et al. [13] proposed the design of an AI chatbot for the diagnostic assessment of the patient and suggests instantaneous actions up to the exposition to nCOV-19. The chatbot can also measure the level of infection through responses from a predefined questionnaire and notify the designated doctors when nCOV-19 signs become worse. However, the proposed system is not able to automatically detect the health status of the patient without asking the patient explicitly. This may not be an effective way to identify the severity of the patient's condition especially when the patient is not able to do so. The chatbot will not be able to know the severity of the patient if the user fails to provide feedback. [17] and [18] develop a chatbot for self-diagnosis and self-healthcare treatment. The bot uses an AI for diagnosing various diseases based on the information and symptoms provided by the user through a chatbot using human’s natural language. The AI model is trained based on online sources and databases. AI model for diagnosis is only relying on the information provided by the user, which is not effective, since the user may provide false information and get the wrong treatment. The user’s health condition may be critical and not able to provide relevant information about his/he health issues. Ko et al. in [20], proposed a vital signs chatbot to collect vital signs data from COVID patients admitted to COVID Virtual Ward. The patient is provided with vital signs monitoring device to monitor their own vital signs. The patient uses a conversation AI assistant to send their vital signs reading to the hospital. The chatbot can also remind the patient to send the vital signs reading within a desired period of time. However, the vital signs device is not integrated with the chatbot, the vital signs readings are manually sent by patient which may sometimes the patient send incorrect readings.

2.3 Medical recommendation systems

In [13], a health service recommendation framework (HSRF) was proposed. The framework gathers the consumer’s health status and takes into consideration the various contexts of the user to find the appropriate service for them. However, this solution is not designed to be used at home by nontechnical users provided that the health status and the context of the users must be manually provided by a medical professional. In [15], a Disease Diagnosis and Treatment Recommendation System (DDTRS) was proposed that uses a Density-Peaked Clustering Analysis (DPCA) algorithm to analyze historical inspection records to recommend diagnosis and treatment plans to doctors and patients. However, this solution is not made for home use provided that the solution still relies on various types of health data that may not be available from home.

2.4 Integration of IoT and conversion AI virtual assistant in healthcare for the case of vital signs

Zaki et al. [16] developed an Artificial Intelligence (AI) Chatbot for diagnostic assessment of the patient's vital signs without direct interactions with real physicians. The system comprises contact vision-based real-time monitoring of vital signs. The solution provides a diagnostic evaluation of patient conditions using AI and recommends immediate measures. The focus of the chatbot is to receive additional information from the patient and provide an appropriate response. However, the system cannot provide a recommendation in case of an anomaly. Sivaraj et al. in [19] developed a "MediBOT" end-to-end voice-based chatbot with four main features which are disease prediction using ML techniques, reading handwritten prescriptions using deep learning techniques, predicting skin diseases using image processing with deep learning, and reading vital signs from the user using smart IoT device. based on the data collected from these components, the system can recommend detected disease preventive measures to the patient. This system is not applicable to the user who just needs to know if he/she has any vital signs anomalies and get instant recommendations on how to get rid of those anomalies.

2.5 The summary and identified gap

Despite the wide range of existing solutions. The current state-of-the-art solutions for patient self-monitoring of their own vital signs with automatic recommendations still have some significant gaps. Existing solutions still depend on medical experts for anomaly detection and recommendations to remote patients with vital signs anomalies. Existing solutions do not provide a portable and cheap device affordable to low-income individuals that could help them to self-monitoring their health status to get earlier diagnosis and guidance for pre-treatment especially when medical professionals are not available Existing solutions are not flexible enough to get the data from any vital signs device, most of them are device-specific solutions. Existing solutions are not effective for a large number of patients, since they still rely on physical physicians to interpret the vital signs data and consult the patient.

To overcome the limitations of the existing solutions, this study presents a research-driven design and development integrating the Internet of Things (IoT), AI, and Natural Language Processing (NLP) to build an autonomous conversational AI assistant that monitors patient's vital signs, predicts anomalies, and provides recommendations of the best practices for handling vital signs abnormalities.

3 CHAPTER 3

RESEARCH METHODOLOGY

This chapter describes different methodologies, tools, techniques, processes, and procedures that have been used to achieve the stated objectives of this study. Scientific methods and experiments for conducting research have also been highlighted in this section.

3.1 Research Process

The research process started with analysis of existing research in the field of interest for the purpose of getting a critical understanding of the field of a studied challenge as well as identifying a research problem. This has been conducted along with formulating the research topic based on the gaps identified from the literature review. Along with the motivating research topic, the research proposal was written and presented for approval. After the proposal has been approved, the research process proceeded with clearly defined steps highlighted as follows:

(1) Review state-of-the-art analysis of research works
(2) Identifying prototyping resources (chatbot framework, cloud virtual server, dataset, edge AI platform, virtual embedded simulator)
(3) Developing and deploying conversational AI recommender assistant
(4) Developing and deploying a vital signs app (API service) for IoT and chatbot integration
(5) Training and deploying AI anomaly detection model on vital signs app
(6) Developing and deploying a rule-based anomaly recommendation web system.
(7) Training and simulating vital signs edge AI anomaly model on Proteus
(8) Performing system-level designs for vital signs embedded device prototype
(9) Perform an experiment on a real embedded board
(10) Analysis of results and challenges
(11) Thesis preparation and Publication of results.

These process can be summarized as shown in Figure 3.

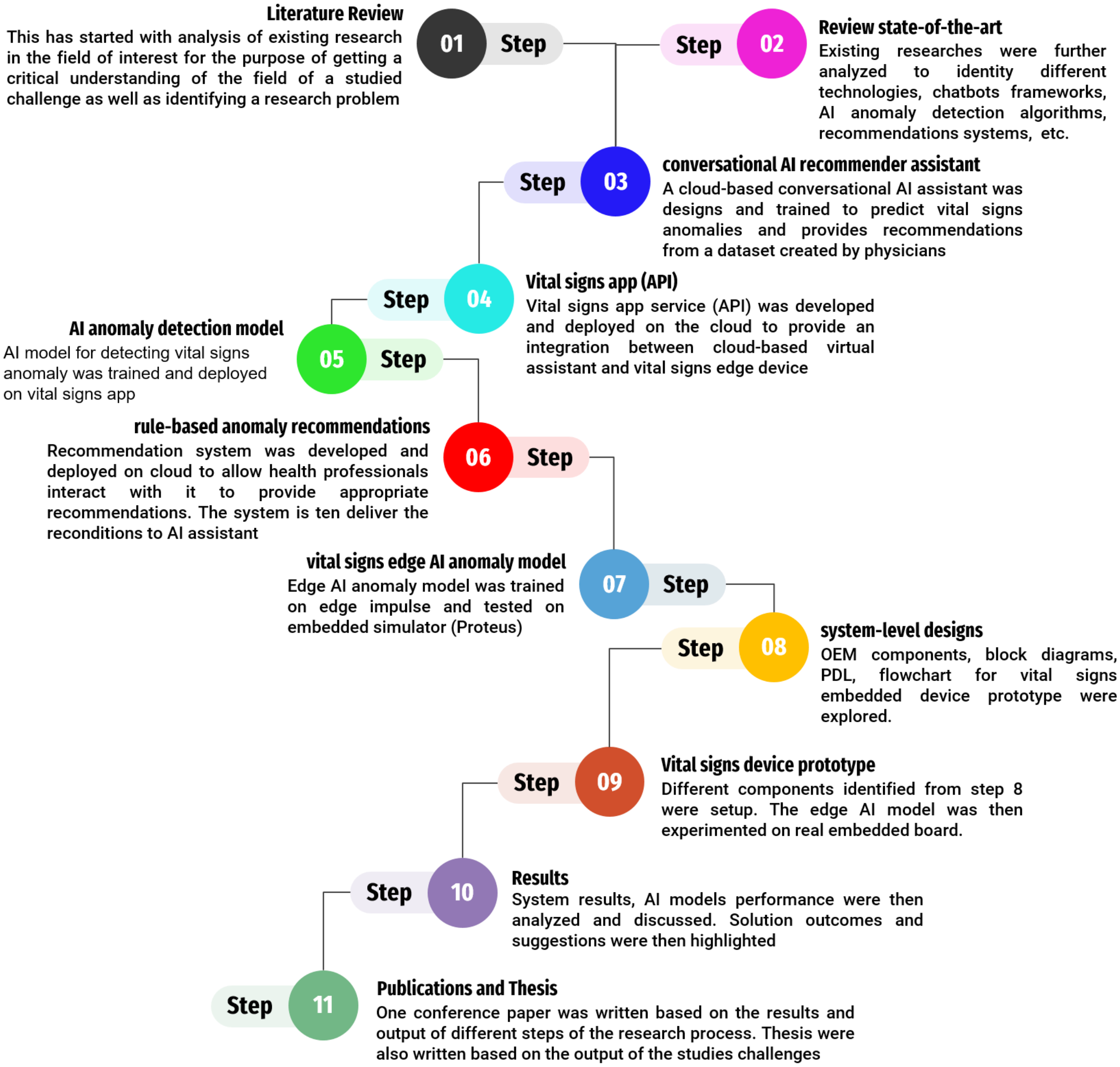


Figure 3: Research Process

3.2 Conversational AI recommender assistant

The development of a virtual assistant was started by selecting the appropriate tools for building chatbots. The following are the methods used for identifying and selecting the appropriate tools for building a virtual assistant (chatbot).

3.2.1 Literature review

The goal of this method is to find out what criteria the previous studies used to select the tools for building chatbots with justified reasons including their strengths and weakness.

3.2.2 Hands-on with existing tools

With this method, we looked at different popular tools on the market and evaluate them according to our needs. Various tools were identified and evaluated including Dialogow, Rasa, Chatlayer, and Chatfuel. However, most of these existing tools have limitations for research projects such as vendor locking and closed source [13].

3.2.3 Consulting experts

Additional inputs and suggestions were obtained from experienced experts from companies dealing with chatbot development. Digital Umuganda is one of the companies we consulted.

3.2.4 Conversational AI recommender assistant tools stack

From the above methods, various tools were compared and analyzed. This study selected to use the RASA framework since it is open source and provides an intuitive and user-friendly interface for building chatbots by providing examples that the system can learn from how people say things and how conversations go. The following tools for building rasa chatbot were selected:

Rasa open source: open source conversation platform that allows you to understand the conversation and connect to different messaging channels and other external systems through a set of APIs.

Rasa x: A user interface for Conversation-Driven Development (CDD) for listening to your user's conversations and using those insights to improve the AI assistant

Rasa SDK: A python-based SDK for the development of custom actions with rasa.

Pycharm IDE: Python IDE for creating, running, and debugging the rasa chatbot.

4GB RAM Linux server: Virtual Machine used for running rasa chatbot.

3.3 Cloud-based Anomaly Detection Machine learning process

The artificial neural network(ANN) model was trained using the tensorflow library in python. In the machine learning process, the dataset was required for training. An Existing human vital signs dataset was used where each data sample was associated with a pre-defined label. The dataset was first passed through preprocessing stage for cleaning. The dataset was then imported using panda's library. Features and labels were then extracted from the dataset and saved into separate variables which were then divided into training and testing sets. An ANN model was built and trained using the Keras library which acts as an interface to artificial neural networks. The resulting model was then saved on a separate file and deployed on the vital signs app for inferencing. Figure 4 shows the machine learning model training process.

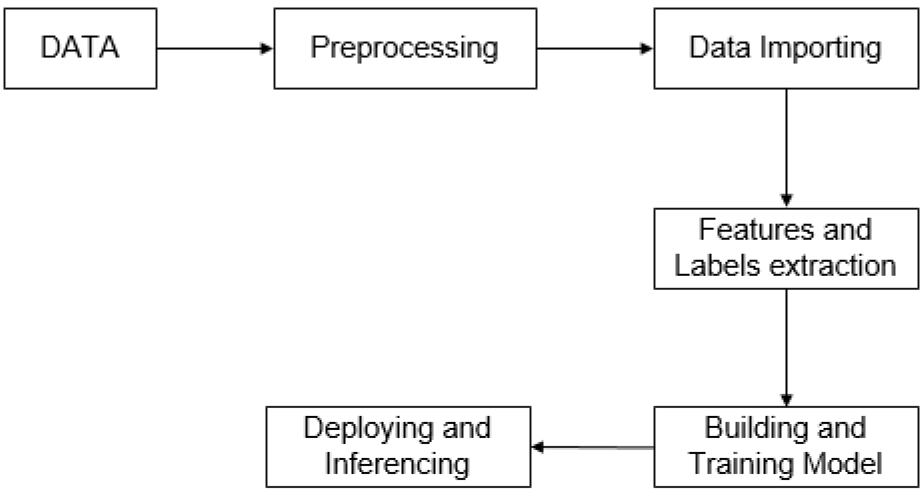


Figure 4: A cloud-based ANN model training process

3.4 Vital signs app

To allow an integration between embedded IoT device, recommendation system and our conversational AI virtual assistant, a REST web service known as vital signs app has been created and deployed on the cloud. Vital signs app has been developed using the python flask library to relay information between IoT devices, recommendation system and conversational virtual assistant. This communication passes through various steps as shown in Figure 5. The process start with a virtual assistant request for vital signs data, vital signs anomalies or recommendations from vital signs app. Upon the receiving of request, the vital signs app check from vital signs if there is any recent vital signs data read from the specified embedded device. The embedded vital signs device will only send the data to the vital signs app if it is connected to the internet which then the data is saved to database with the current time and date. When the data is loaded from database, the vital signs data, anomaly detection is performed as well the recommendations are loaded from recommendation system which then sent to virtual assistant depending on specified request.

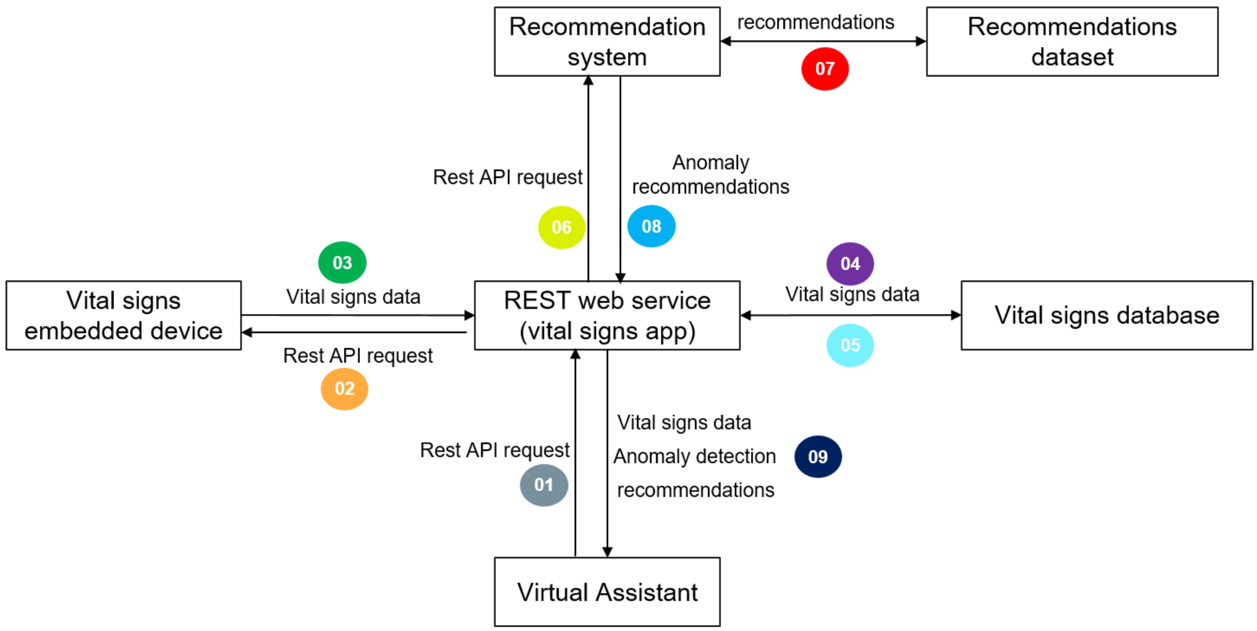


Figure 5: Integration Process of vital signs app.

3.5 Rule-based recommendation app

This is a web-based application designed using python and MySQL databases. The system allows health professionals to provide appropriate recommendations based on detected anomalies on the vital signs data. Recommendations datasets will be stored on a database and loaded when it is requested by the vital signs app. As shown in the Figure 6, The process of developing this recommendation system was started by reviewing existing studies in order to identify different technologies and approaches used to develop recommendation system in healthcare. Since this recommendation system is rule-based, different health professionals have been consulted in order to build a rule-based recommendation dataset. Then the designs of the system were created which was then implemented using python flask library.

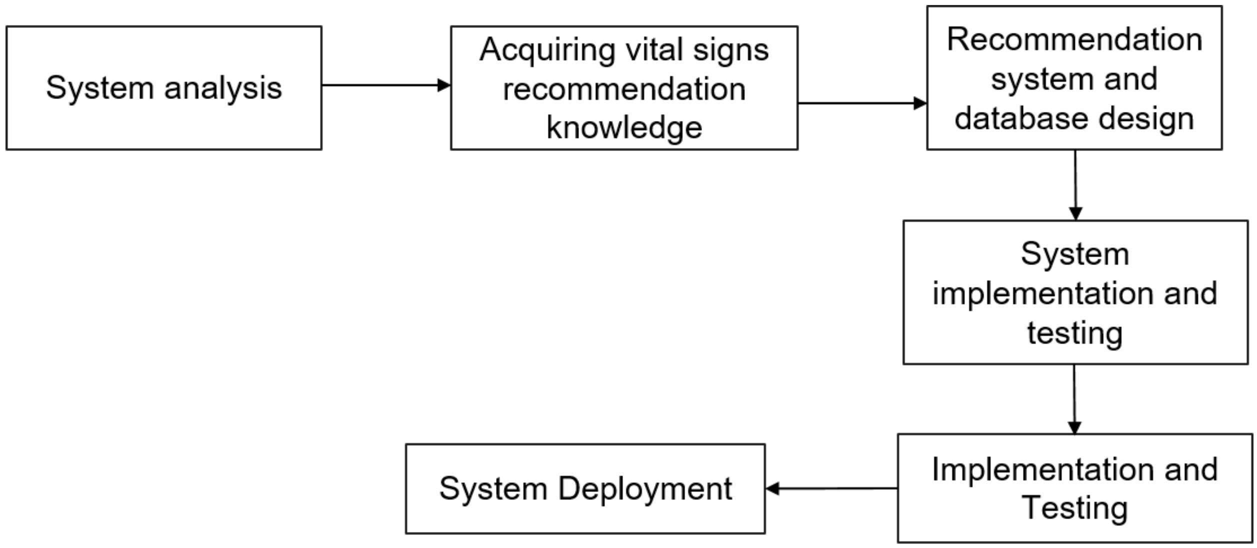


Figure 6: Rule-based recommendation system development process

3.6 Edge AI process

To get a lightweight and optimized AI model for anomaly detection that can be deployed on a low-power embedded device, a machine learning technique known as k-nearest neighbors (K-NN) was used. K-NN is based on distance-learning between data. Classification is performed based on majority votes of nearest neighbors. New data is classified based on its distance from multiple neighboring sample data. The model training and testing process is presented in Figure 7 which starts with acquiring an open dataset of human vital signs used for training and testing the model. Assumptions were made that the dataset was collected using the same psychological sensors for measuring vital signs. The data are transformed into features. The next step is to build and train the model using python. The model is tested with different K to identify the optimum value which provide the best results. The resulting model was then packed in a form of an Arduino library which was then imported and compiled on a targeted processor architecture using Arduino IDE and deployed on either an embedded simulator or a real-embedded device.

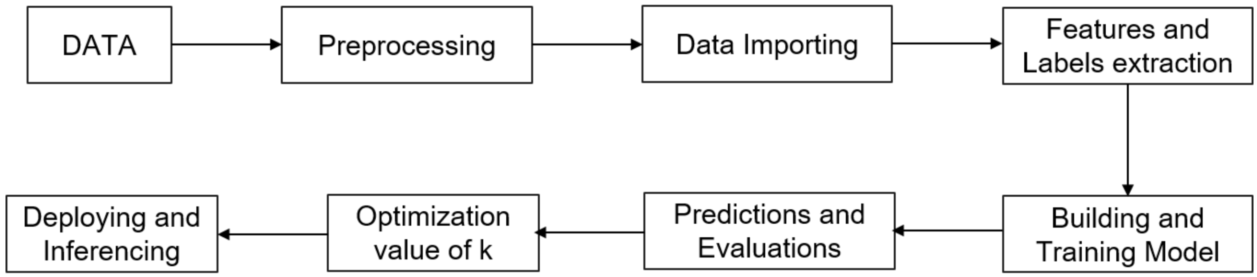


Figure 7: KNN model training process

3.7 System development method

AGILE methodology was used as a system-developing method. According to [21] the agile method breaks tasks into small increments (iterations) with minimal planning. Iterations are in short time frames that typically for 1 weak. Each iteration involves planning, requirements analysis, design, coding, unit testing, and acceptance testing. At the end of the iteration, progress was demonstrated to the supervisor through a weakly progress meeting. Figure 8 shows the agile development process. Scrum is a subset of Agile. It is a lightweight process framework for agile development, and the most widely-used one [21].

Figure 8: Agile methodology

4 CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

In this chapter, the analysis and designs of different components of the proposed solution were described. The overall architecture of the proposed solution was described which demonstrates how different components of the system are integrated together to achieve the designed output. For a conversational AI virtual assistant, a detailed architecture of the different components used to build the assistant was described. For the rule-based recommendation system the functions requirements, system models, and database models were described. For the case of the vital signs app a list of API services provided by the app were described. For the embedded IoT device prototype, system block diagram, Original Equipment Manufacturer (OEM) components, Program Description Language (PDLs), and flowchart were described.

4.1 System architecture

4.1.1 Integration design of IoT, AI, and NLP chatbot to build a virtual vital sign assistant

Figure 9 shows the overall architecture of the proposed solution integrating 3 technologies including embedded IoT, ML for anomaly detection, and NLP for conversational AI. From left to right, embedded IoT allows to collection of vital signs data from a patient and transmit them to the cloud. In this research, 3 psychological parameters from the patient have monitored the body temperature, heart rate, and blood oxygen saturation. Next in the cloud, ML is used to detect anomalies from vital signs data and eventually provide recommendations in case of anomaly. The training for anomaly detection uses a deep neural network (DNN) while the recommendation system is based on a rule-based AI. When a user consults vital sign data, he/she communicates with a chatbot based on NLP. Indeed, NLP enables to establish natural conversation with end users about vital signs raw data, anomalies, or recommendations. The natural conversation feature is important to enable the adoption of the system by non-experts.

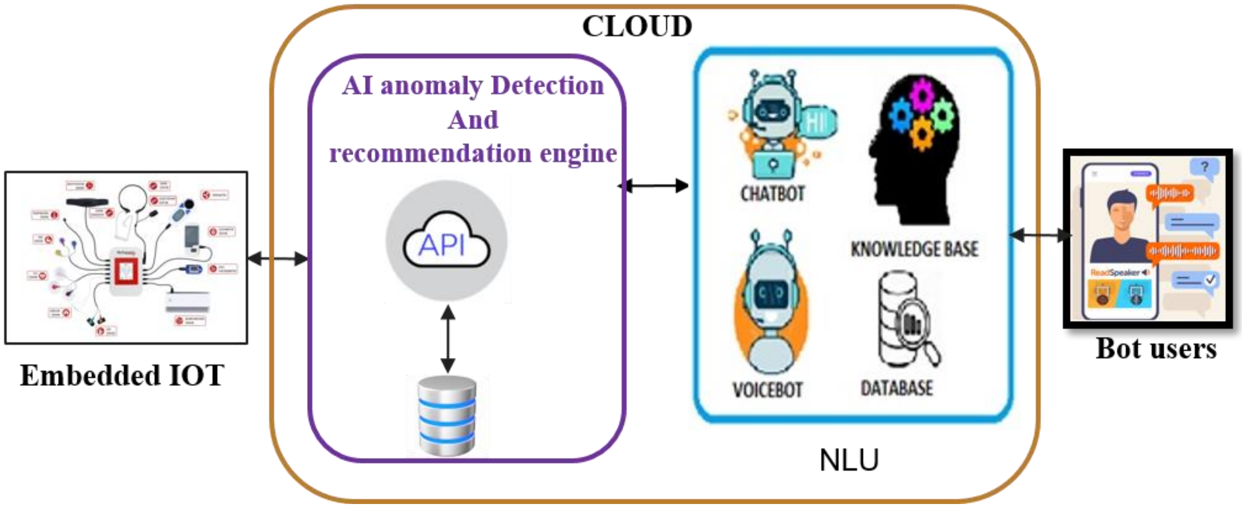


Figure 9: Architecture Integration of enabling technologies

4.1.2 High-level system architecture for our conversational AI virtual assistant

This section presents the overall system architecture of the proposed solution as shown in Figure 10. The patient is tagged with an embedded IoT device for collecting physiological parameters. The data collected in an embedded IoT device are sent wirelessly to the vital sign app running on the cloud. The vital sign app will use the ANN model to identify anomalies in the vital signs. The vital signs app is integrated with a virtual assistant located on the cloud to deliver vital signs data and detect anomalies to users. Users will use a conversation UI as an interface to the virtual assistant to request vital signs values, and vital signs anomalies as well as ask for clinical recommendations from the vital assistant. The virtual assistant will communicate with a knowledge-based recommendation system to identify what medical procedures to be recommended to the patient based on the detected anomaly.

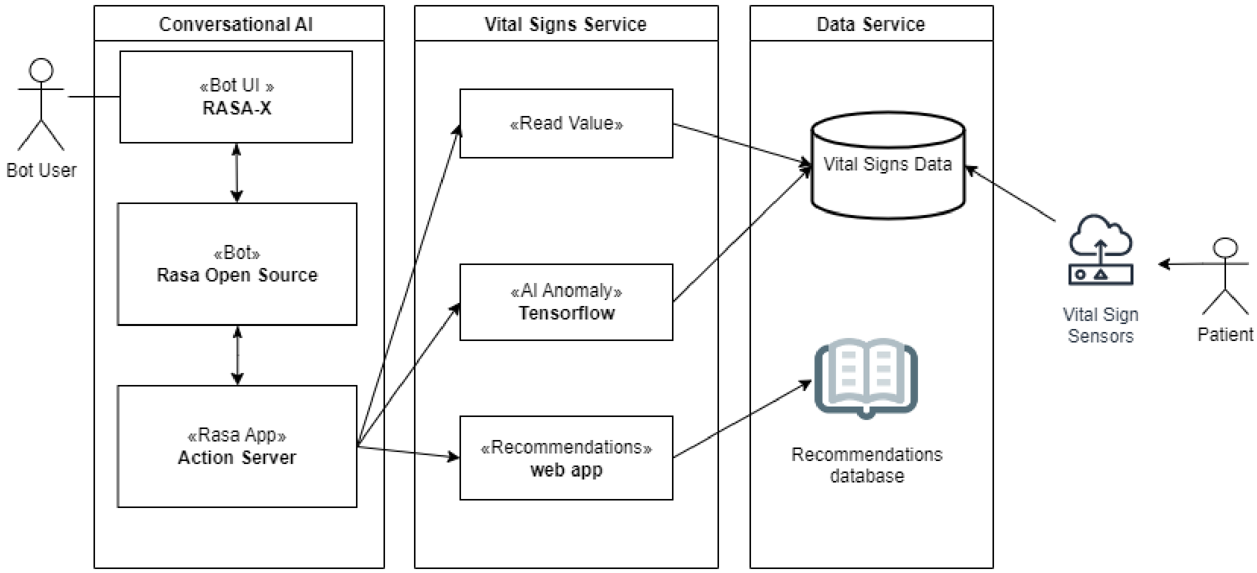


Figure 10: The high-level architecture of our conversational AI virtual assistant

4.2 Conversational AI chatbot design

Rasa provides various modules to train the built-in model based on the input data from the user and provide a customer's response to the user.

4.2.1 RASA NLU

This module receives the user input text and identifies the intents and entities from it as well as performs response retrieval. Table 1 shows the list of intents used in this project and its descriptions.

Table 1: List of Intents and its description

Intent	Description
greet	user greets the bot with a message
goodbye	user says bye to the bot
affirm	user agrees to the bot’s response
user_diagnostic_enquiries	user asks the bot for the patient’s health status
user_what_is_wrong	use asks the bot for vital signs readings
user_what_is_suggested	user asks the bot for the best practice to get rid of vital signs anomalies
bot_challenge	user asks the bot if he/she is talking to a human or a bot

Intent	Description
vital_signs	user request for the reading of the specific vital sign
age	User select the age of the patient
context	User provides context of the patient
Set_code	User enter the device key to access vital signs
Exit	User reset the conversation with the bot
device_number	User provides device number of the device to be communicated
Health	User specify list of historical health issues he/she have

4.2.2 Rasa Rules

The module is used by the dialogue manager to handle the short piece of conversations that always follows the same way. That is whenever the intent belonging to the user input text matches with the intent in the rules, then an action will be performed. Figure 11 shows a list of rules used in this project. For example, if the user says “goodbye” the bot will detect an intent as goodbye and will render utter_goodbye action as a response to the user.

```
version: "2.0"
rules:
- rule: Say goodbye anytime the user says goodbye
  steps:
  - intent: goodbye
  - action: utter_goodbye
- rule: Say 'I am a bot' anytime the user challenges
  steps:
  - intent: bot_challenge
  - action: utter_iamabot
- rule: out-of-scope
  steps:
  - intent: nlu_fallback
  - action: utter_out_of_scope
```

Figure 11: List of Rules to call actions for the detected user intent.

4.2.3 Rasa stories

Rasa stories is a rasa module that defines the set of intents and responses sequence that the user may follow during the conversation. Figure 12 shows some stories for this project. from the story with the story name “happy path greeting” whenever the user says “Hi”. The intent “greeting” will be detected, then the bot's four responses, utter_medbot_greet, utter_medbot_intro, utter_service_list, and utter_ask_for_help are generated which form one set of input and response sequences. This conversation will proceed with the next set of inputs and responses when the user asks for vital signs anomalies that belong to the intent “user_diagnostic_enquiries”. with detected intent, the response “diagnostic_response_action” will be generated by a bot that displays the vital signs anomaly to the user.

```yaml
- story: hapy path greeting
  steps:
  - intent: greet
  - action: utter_medbot_greet
  - action: utter_medbot_intro
  - action: utter_service_list
  - action: utter_info_needed
  - action: utter_age
- story: hapy path diagnosing enquires
  steps:
  - action: utter_ask_for_help
  - intent: user_diagnostic_enquiries
  - action: diagnostic_response_action
- story: hapy path what is wrong
  steps:
  - intent: user_what_is_wrong
  - action: check_abnormal_vital_signs_
  - intent: user_what_is_suggested
  - action: suggested_action
- story: hapy path vital sign readings
  steps:
  - intent: vital_signs
    entities:
      - vital_signs: "temperature"
  - slot_was_set:
      - vital_signs: "temperature"
  - action: action_request_vital_signs
```

```yaml
- story: Story from Conversation
  steps:
  - intent: greet
  - action: utter_medbot_greet
  - action: utter_medbot_intro
  - action: utter_service_list
  - action: utter_info_needed
  - action: utter_age
  - intent: age
    entities:
      - age: "Adult"
  - slot_was_set:
      - age: "Adult"
  - action: utter_device_number
  - intent: device_number
    entities:
      - device_number: "DVS0001"
  - slot_was_set:
      - device_number: "DVS0001"
  - intent: health
    entities:
      - health_status: "Diabetics"
  - slot_was_set:
      - health_status: "Diabetics"
- story: hapy path start diagnosis
  steps:
  - action: utter_ask_for_help
  - intent: user_diagnostic_enquiries
  - action: diagnostic_response_action
  - intent: user_what_is_wrong
  - action: check_abnormal_vital_signs_action
```

Figure 12: Stories with a sequence of intents and actions.

4.2.4 Domain.yml

A domain is a rasa module that contains everything the rasa assistant knows. It defines the context in which the assistant operates. The actions shown in Figure 11 and Figure 12 are the responses that are defined in this module. For example, for the user input text “Hi” which belongs to the intent “greet” in Table 1 the bot will pick the response “utter_medbot_greet” with the response text “I am designed or trained to analyze vital signs” defined in domain.yml and send it the user.

4.2.5 Custom Actions

Custom actions are an optional rasa module used to create custom responses to the user intent. custom actions are defined in actions.py which is run by an action server known as RASA SDK.

In this project, custom actions were created to run code for API calls for vital signs data and database queries. The response of the user intent can be from the domain.yml or can be the custom action defined in actions.py.

According to the stories shown in Figure 12, for the user diagnosing inquiry that belongs to the intent “user_diagnostic_enquiries” the custom action “diagnostic_response_action” will be generated. Figure 13 shows the definition of diagnostic_response_action within actions.py. The custom action calls an API to retrieve vital signs anomalies from the recently stored vital signs reading.

```
class ActionDiagnosticResponseAction(Action):
    def name(self) -> Text:
        return "diagnostic_response_action"
    def run(
            self,
            dispatcher,
            tracker: Tracker,
            domain: "DomainDict",
    ) -> List[Dict[Text, Any]]:
        output = check_anomalies()
        if len(output) > 0:
            if output == "Abnormal":
                dispatcher.utter_message(template="utter_abnormal_response",
                abnormal_response="The Patient condition is not normal.
                "The patient needs an agent medical attention")
            elif output == "Normal":
                dispatcher.utter_message(template="utter_normal_response",
                      normal_response=str("The Patient condition is Normal."))
        else:
            dispatcher.utter_message(template="utter_no_data",
                                     no_data="No data available")
```

Figure 13: Source codes of diagnostic_response_action action

4.2.6 Chatbot System

Figure 14 shows the chatbot system developed using the modules discussed above. Users will interact with a chatbot using the RASA UI tool known as rasa x by posting messages and asking queries to the assistant. The message posted to the rasa x is passed to the RASA NLU module. NLU MODULE will generate intents and extract entities from the user message with a certain confidence level and send the one with the highest confidence level to RASA CORE for generating a response corresponding to the detected intent. the response may be generated from either the domain.yml or from custom actions. custom-actions-based responses are generated from the custom code which runs certain API calls or database queries.

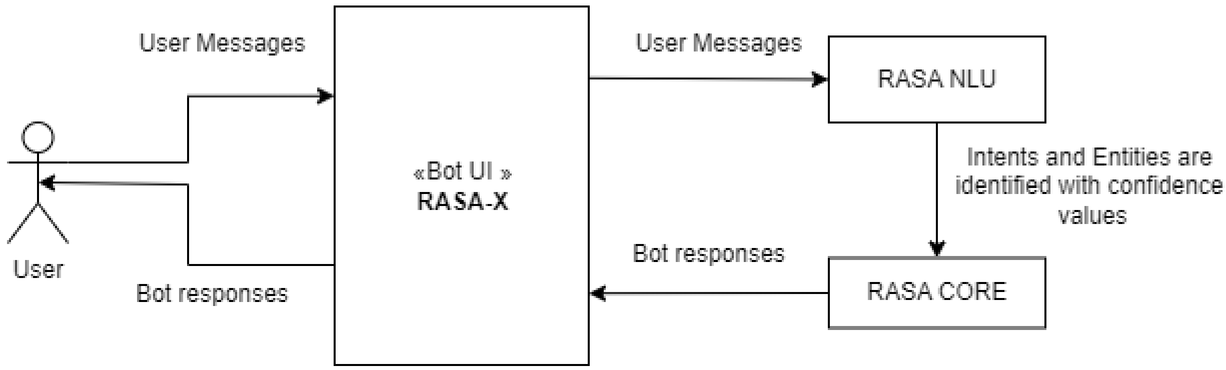


Figure 14: Chatbot system model

4.2.7 Intent classification and entity recognition

There are various techniques offered by rasa on how intents can be classified and how entities can be extracted from user utterances. Using a mix of different methods can make you rasa have a good performance in intent classifications and entity extractions. In this project, the following techniques have been used for entity extraction and intent classifications as shown in Figure 15.

4.2.7.1 DIETClassifer

It is a machine-learning-based technique for classifying intents and extracting entities from user utterances. It is very powerful for extracting custom entities. It requires a lot of training data to achieve better results.

Rasa comes with default configurations of different parameters of DIETClassifier including the number of epochs, embedding dimensions, Number of transformer layers, use_masked_language_model, etc.

To get a good performance of intent classification the DIETClassifier was optimized by tuning these parameters. To start with the number of epochs, since our training data are small, the number of epochs was increased from the default value of 300 epochs to 400 epochs to achieve better performance. To predict missing tokens from user utterances, the language model needs to be trained. In our DIETClassier, we set use_masked_language_model to true to gain additional domain knowledge from our training data and solve the challenge of predicting missing tokens. For the case of Embedding dimensions, it was set to the default value of 20 at the beginning when the training data was small. But we kept increasing it whenever training data were increased. And finally, we set the embedded dimension to 30. The number of transformer layers was set to 4 to influence the word in a sentence and avoid confusion in some intents.

4.2.7.2 Ducking Entity Extraction

It is a prebuilt model for extracting dates, numbers, email address, url, etc. it doesn't require any training data for assistant to extract this information. In this project Duckling entity extraction has been used to extract device numeric key from the user utterance.

4.2.7.3 RegexEntityExtractor

It is an entity extraction technique that allows defining the specific patterns of how the entity that would like to extract shows follow. It is a suitable technique for entities which follows specific patterns like phone numbers, user IDs, account number, etc. in this study, regex has been used to extract device number which has to follow specific patterns for example DVS0001.

```
pipeline:
  - name: WhitespaceTokenizer
  - name: LexicalSyntacticFeaturizer
  - name: RegexFeaturizer
  - name: RegexEntityExtractor
  - name: CountVectorsFeaturizer
    analyzer: char_wb
    min_ngram: 1
    max_ngram: 4
  - name: DucklingEntityExtractor
    url: "http://duckling:8000"
    dimensions: ["time", "number"]
  - name: DIETClassifier
    epochs: 400
    use_masked_language_model: True
    embedding_dimension: 30
    transformers_layers: 4
```

Figure 15: RASA configuration file

4.2.8 Vital signs app

Vital signs app is a RESTful web service developed to enable an integration between different components of the system. The service was implemented using the python flask library. As shown in Figure 16, the web service has four API endpoints which are accessed via HTTP requests. API endpoint 1 fetches the recent vital signs data from the database and returns the data to the client in

JSON format. API endpoint 2 predict the vital signs anomaly using a machine learning model and return the output to the client in JSON format. API endpoint 3 fetches appropriate recommendations from the database based on detected anomalies and returns the list of recommendations to the client in JSON format. Endpoint 4 adds the vital signs sensor data into the database.

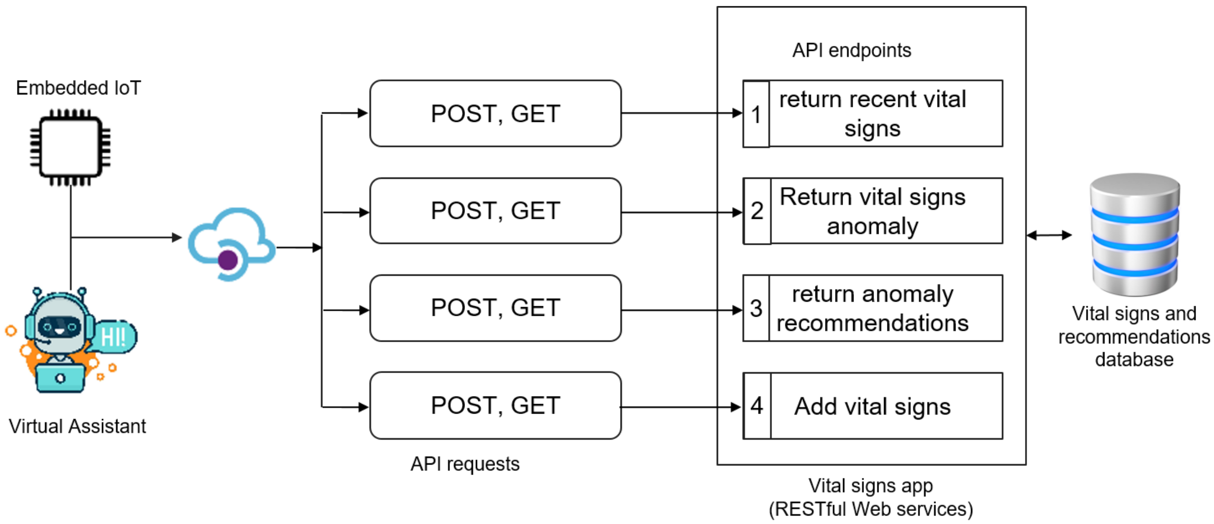


Figure 16: Architecture of vital signs app (RESTful web service)

4.3 Cloud-based AI vital signs anomaly detection model

4.3.1 The training dataset

The open dataset of human vital signs anomalies used in this project is found in [22]. The dataset is called “Human vital signs” which contains about 25000 data samples and six columns recorded in a CSV format as shown in Table 2. Some of the samples were taken from healthy people and were labeled as normal. Some samples were taken from people with abnormal vital signs and were labeled as Abnormal.

Table 2: Sample of collected vital signs data. recorded in a CSV format

Timestamp	HR (BPM)	RESP (BPM)	SpO2 (%)	TEMP (*C)	OUTPUT
134	44	15	100	41	Abnormal

Timestamp	HR (BPM)	RESP (BPM)	SpO2 (%)	TEMP (*C)	OUTPUT
79	45	15	100	40	Abnormal
133	47	15	100	41	Abnormal
78	48	15	100	41	Abnormal
425	48	16	100	40	Abnormal
250	48	16	100	41	Abnormal
44	49	15	100	41	Abnormal

4.3.2 Training ML model

The vital sign dataset was divided into two sets. 80 % of the data was used for training ANN models and 20 % was used to test the model. The model was trained using four features (Heart Rate, Respiration Rate, Blood Oxygen Saturation, and Body temperature) and one label (Output). ANN model was trained using TensorFlow in python which consists of input layer of 20 input neurons, two dense fully connected hidden layers, each having 10 neurons, and output layer of 2 output neurons. To achieve a good performance of the model, the model has been trained with a different number of epochs, and finally, a good performance of the model was obtained with 30 epochs and a learning rate of 0.001.

4.4 Rule-based recommendation system

A rule-based recommendations web system has been designed using MySQL database and python whereby health professionals interact with it to provide appropriate recommendations based on different anomalies detected and other information from the patient. The appropriate recommendations are generated based on patient information (e.g. patient age, medical history) and vital signs values and are accessed by an assistant using a vital signs app.

4.4.1 Functional requirements of recommendation system

i) The system must allow health professionals to provide recommendations based on detected anomalies

ii) The system must all health professionals to access the list of detected vital sings anomalies

iii) The system must allow health professionals to manage vital signs datasets with respect to the patient age range

iv) The system must allow the admin to manage health professional information

4.4.2 Use case diagram

Figure 17 shows the use case diagram for the recommendation system. The use case diagram shows the interactions for different user roles with the system. The main identified for this system are health professionals and administrators.

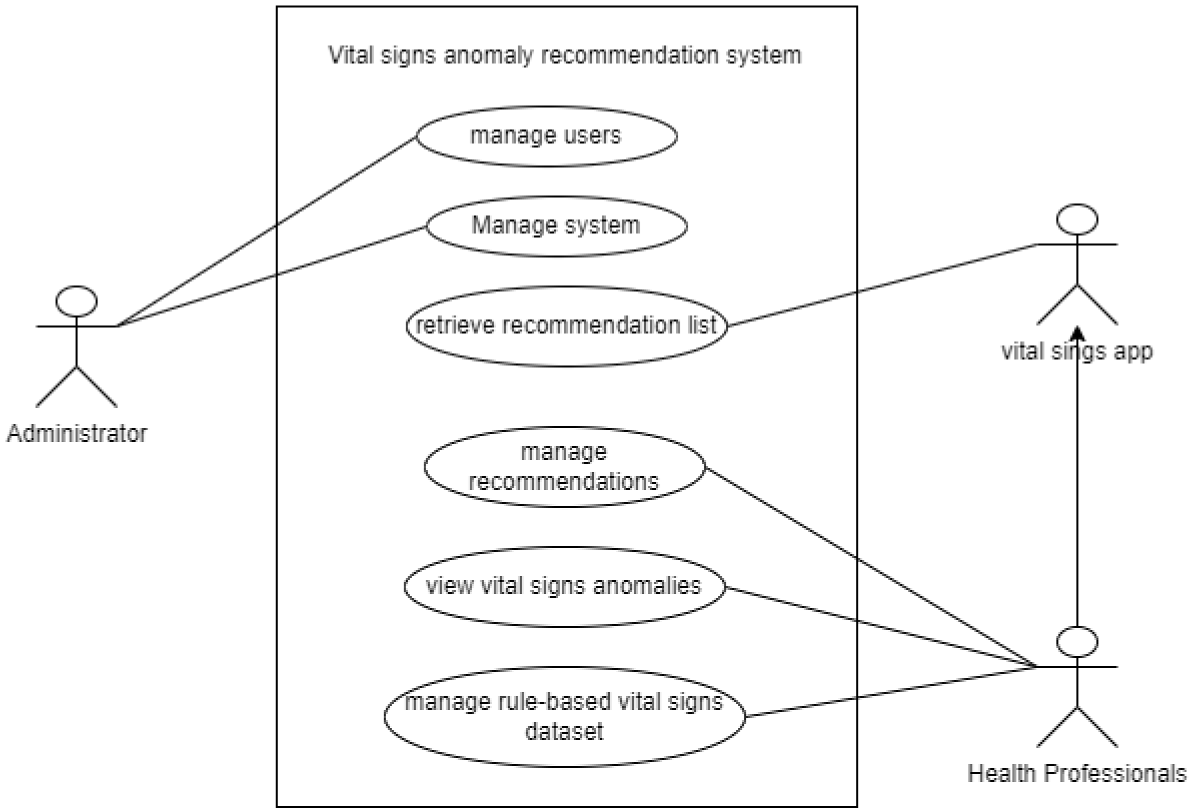
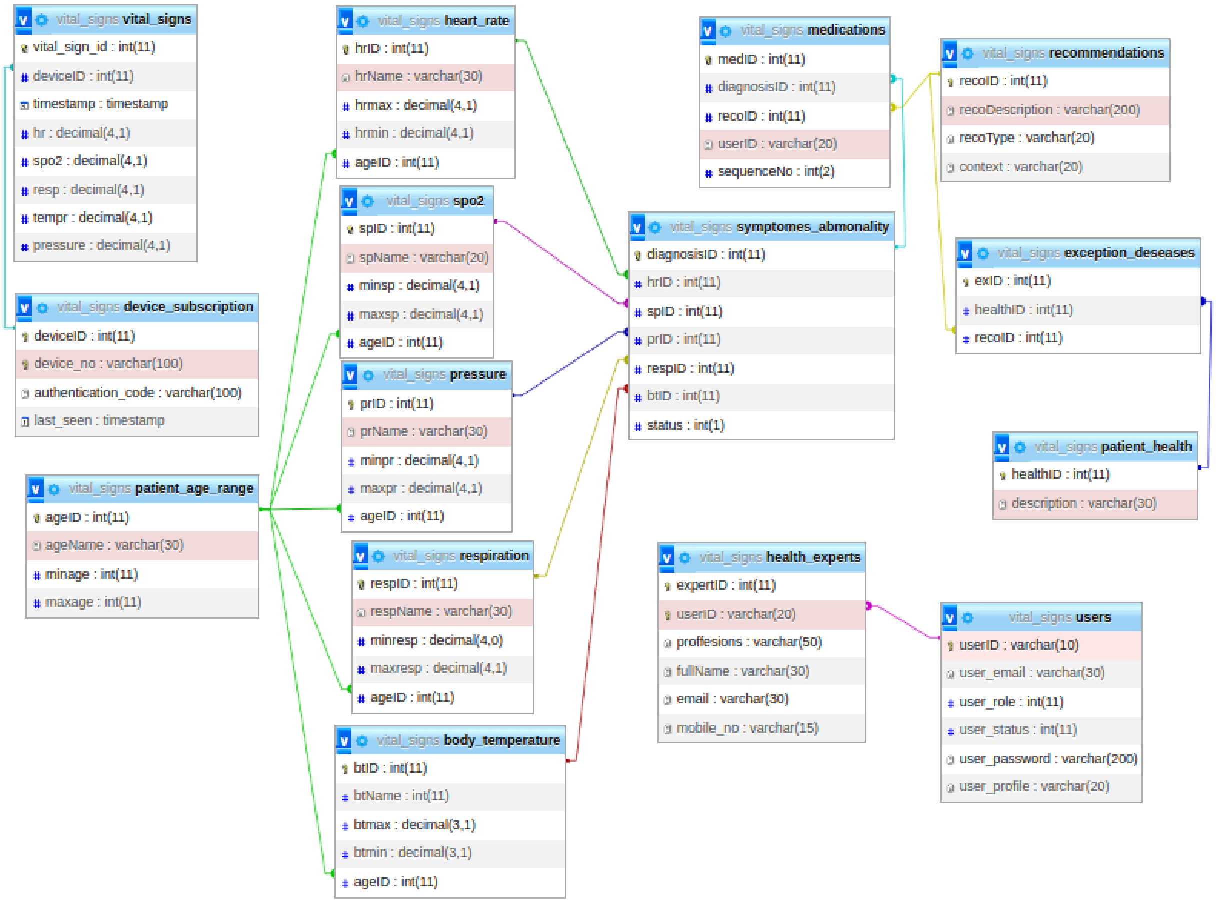


Figure 17: Usecase diagram for recommendation system

4.4.3 Database Design

Figure 18 presents a database model or the recommendation system. The database consists of 15 tables that store different data needed for building a recommendations dataset. The tables (heart rate, spo2, pressure, respiration, body temperature) store threshold values for different vital signs with respect to age range. Vital signs table stores vital signs sensor data collected from different embedded devices. symtomes_abnomality table store different abnormalities detected from vital signs data which are associated with different recommendations. Recommendations are also based on the health background of the patient.



4.4.4 Building rule-based recommendation datasets.

The recommendations dataset has been prepared by first defining threshold values for each vital sign with respect to the age range. Table 3 , Table 4, Table 5, Table 6, and Table 7 present the threshold values for temperature, heart rate, Spo2, respiration rate, and pressure respectively. Age has been categorized into four groups which are Adult (age≥8 years), child (1 year ≤ age <8 years), Infant (1 month ≤ age <1 year), Neonate (1 day≤ age < 1 month).

Table 3: Temperature threshold values

AGE	LOW	NORMAL	HIGH
Neonate	36.05- 36.05°C	36.4 - 37.3°C	≥37.5 °C
Infant	36.0 - 36.4°C	36.5 - 37.3°C	≥37.4°C

Child	≤35.5°C	35.9 -36.8°C	37.5 - 38.4°C
Adult	≤35.1°C	35- 36.1°C	37.1 - 38°C

Table 4: Heart rate threshold values

AGE	LOW	NORMAL	HIGH
Neonate	≥107	120 - 160	≤181
Infants	93- 107	80-140	161 -181
Children	52- 88	75-120	115 -156
adult	>43	60 - 99	<104

Table 5: Spo2 threshold values

AGE	LOW	NORMAL	AGE
Neonate	≥92%	93 -100%	Neonate
Infant	≥92%	93 -100%	Infant
Children	≥94%	95 -100%	Children
Adult	≥94%	95 -100%	Adult

Table 6: Respiration rate threshold values

AGE	LOW	NORMAL	HIGH
Neonate	≥25	30 - 40	≤66
Infant	22 - 25	20 - 40	58 - 66
Child	14 - 21	15 - 25	25 - 53
Adult	>11	15 - 20	<22

Table 7: Blood Pressure threshold values

AGE	LOW	NORMAL	HIGH
Neonate	≤60/45	67/35 -84/53	>90/60
Infant	≤70/60	87/53 - 90/60	≥110/75
Child	≤80/55	95/53 - 105/66	≥110/79
Adult	≤109/76	121/80 - 129/89	≥133/84

4.4.5 Recommendation system dashboard

Figure 18 shows the dashboard of the recommendation system where by health professionals and administrator use to interact with the system. Through the dashboard, users can also access vital signs trends of the connected devices as well as provides recommendations to the detected anomalies.

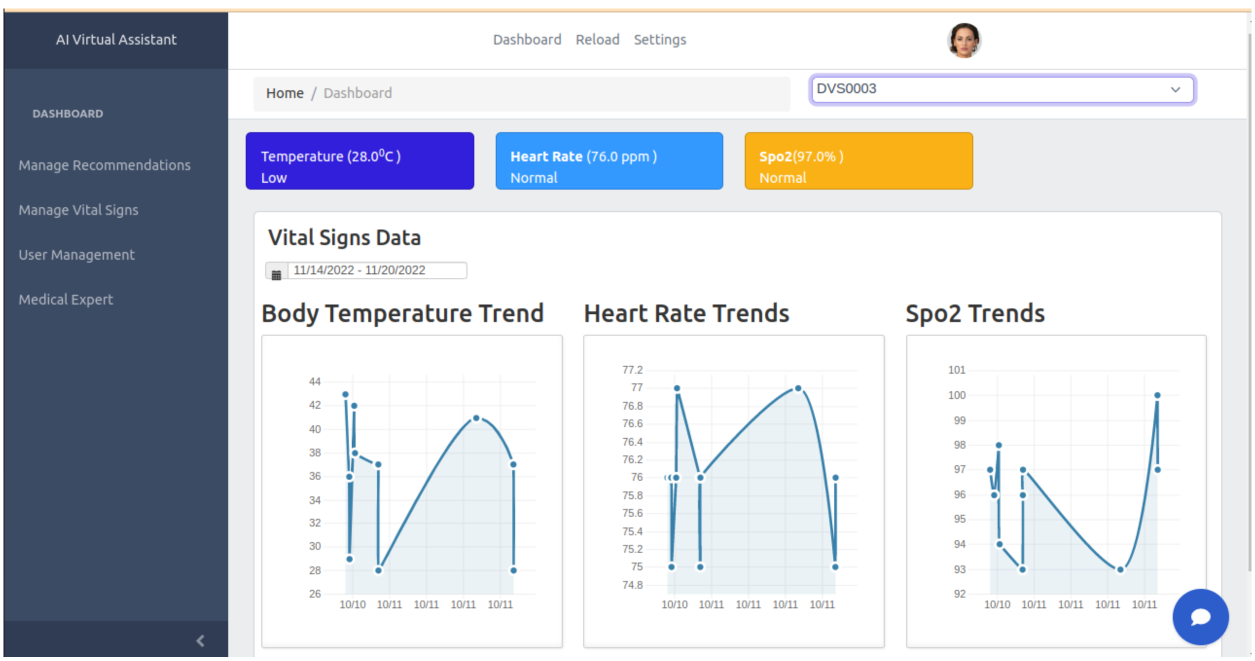


Figure 18: Recommendation system dashboard

4.5 Embedded system-level design

4.5.1 Generic (IoT) Embedded System Block Diagram

Figure 19 presents the generic HW embedded system hardware modeling. The hardware modeling shows that the device is made up of four main units which are the sensing unit, actuation unit, processing unit, and communication unit. The sensing unit comprising of psychological sensors for measuring vital signs data from the patient. The detected vital signs data are then sent to the processing unit. The sensing unit comprises the microcontroller which processes and analyzes the data using an embedded machine-learning model. The processing output is then sent to the actuation unit which presents the output in different forms. The actuation unit comprises of (1) a display for displaying the vital signs and a predicted anomaly, (2) a buzzer for producing a sound alert for inform user when an anomaly is detected, and (3) a flashlight that indicate the severity of vital signs anomaly. The communication unit is responsible for send the vital signs data to the cloud for storage and further processing.

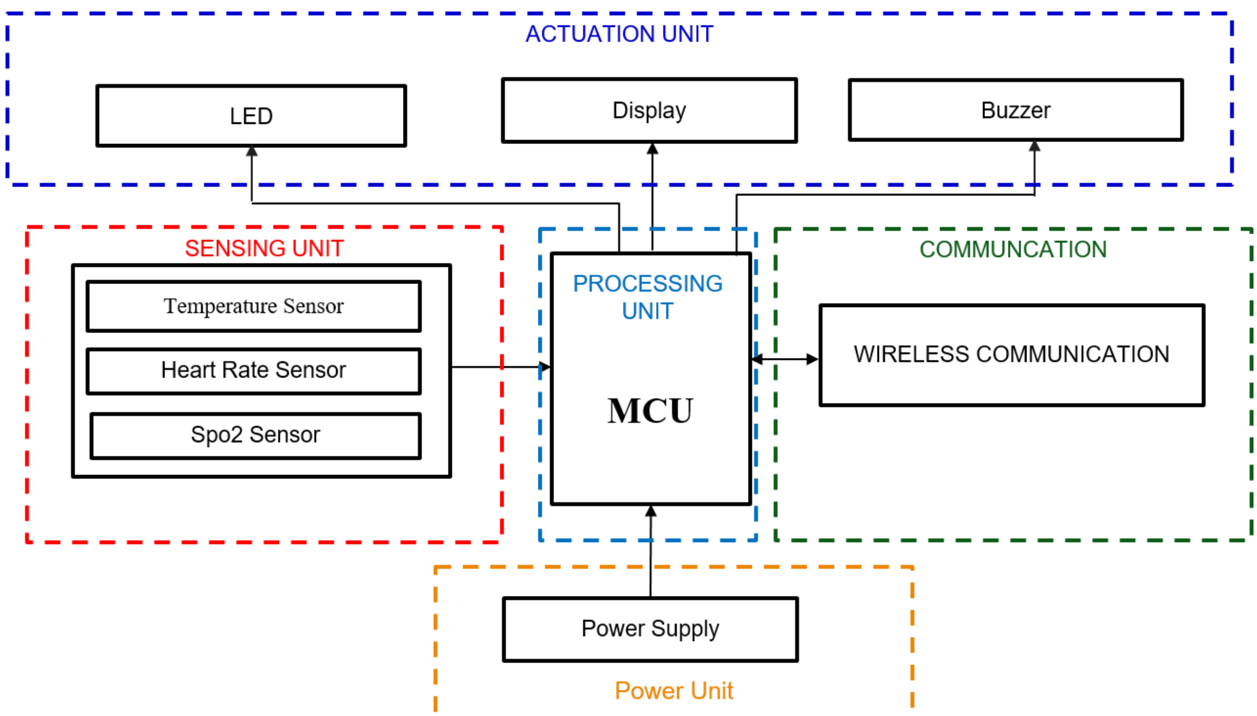


Figure 19: Generic (IoT) Embedded System Block Diagram

4.5.2 OEM component specifications

Table 8 shows the list of selected OEM components available on the market for building the proposed embedded IoT device prototype.

Table 8: OEM component specifications

Component Name	Model	Price (RWF)	Size	Power Specification		
				Active Current (mA)	Sleep Current (µA)	Power supply range (V)
Breadboard	Solderless Breadboard PRO25 ,R13	3,500	165.1 x 54.29 x 9.68mm	-	-	-
Power Supply	Breadboard power supply module	2,500	51 x 32 x 19mm	700	-	3.3 - 5
Microcontroller	Arduino Nano 33 BLE sense	80,000	45x18mm	15mA per I/O pin	1.0 µA	3.3
Contactless body temperature sensor	MLX90614	25,000	11.5 x 16.5mm	2	5 µA	4.5 - 5
Pulse oximeter	MAX30100	7,500	18.8mm (L) x 14.4mm (W) x 3.0mm (H)	1200	10 µA,	3.3
Display	0.96 OLED	5,500	1.09 x 1.07 x 0.17 inches	20	1	5
Buzzer	Buzzer – 5V COM22 ,R33	500	12 x 9.5mm	15	-	5
LED	RGB LED Diode Lights	500	5mm	20	-	3.3 - 4

Component Name	Model	Price (RWF)	Size	Power Specification		
				Active Current (mA)	Sleep Current (µA)	Power supply range (V)
GSM Module	Smallest SIM 800L GPRS GSM BRD56 , R11	16,000	15.8x17.8x2.4 mm	1000	1	3.4 – 4.4
Battery Holder	9V Battery Holder with ON/OFF Switch COM41 ,R14	1,500	68mm x 33mm x 21mm	-	-	-
Battery	14500 1300mah 3.7 V lithium ion rechargeable battery COM32	5,300	50mm * 14mm	-	-	3.7

4.5.3 From generic functional hardware block to specific hardware components block

Figure 20 present a block diagram with selected OEM components available on the market.

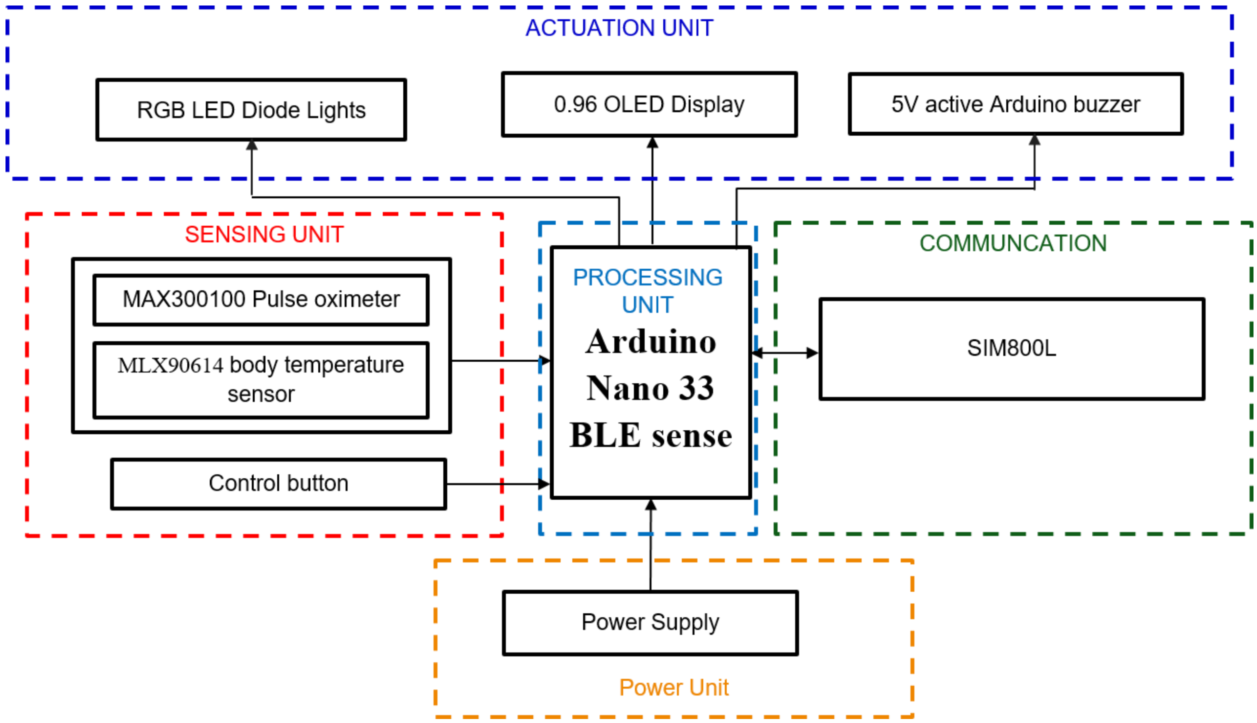


Figure 20: Hardware Specific (IoT) Embedded System Block Diagram

4.5.4 Components description

4.5.4.1 Arduino Nano 33 BLE

The Arduino Nano 33 BLE Sense [23] is a tiny development board with a Cortex-M4 microcontroller embedded with three sensors that are motion sensors, a microphone, and BLE. It has the capability of running embedded machine learning. In this study, this board was used to build an embedded IoT device prototype of monitoring of vital signs. Figure 21 shows the pinout of the board.

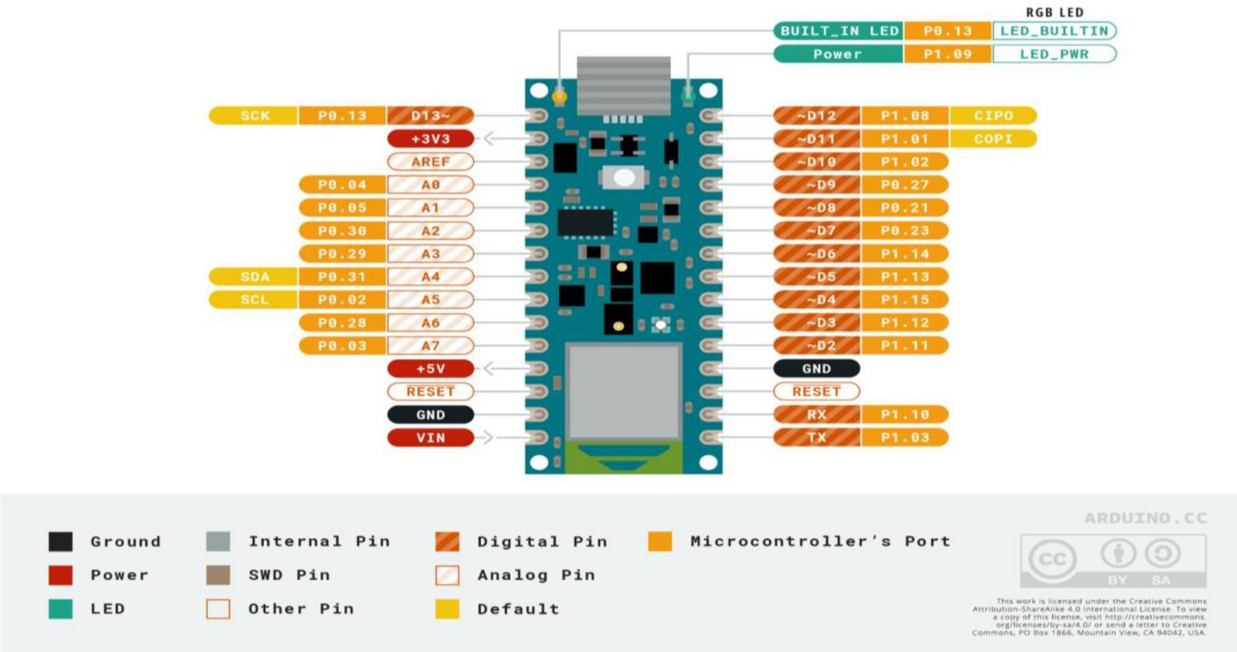


Figure 21: ARDUINO NANO 33 BLE SENSE

4.5.4.2 Contactless body temperature sensor (MLX90614)

The MLX90614 sensor [24] is a Contactless Body Temperature Sensor that uses infrared radiation emitted by the body to measure the temperature. The sensor can measure the temperature of the body ranging from -70° C to 382.2°C. It can be connected to microcontroller using I2C protocol. Figure 22 shows the pinout of the MLX90614 sensor.

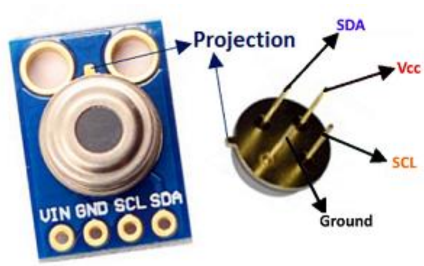


Figure 22: MLX90614 Pinout

4.5.4.3 Pulse oximeter(MAX30100)

The MAX30100 pulse oximeter [25] is an integrated pulse oximetry and heartrate monitoring solution. It comprises of two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart-rate signals. This sensor was used in this study to measure the blood oxygen saturation and heart rate from human being. It can be interfaced with a microcontroller board using I2C protocols. Figure 23 shows the pinout of the MAX30100 pulse oximeter.

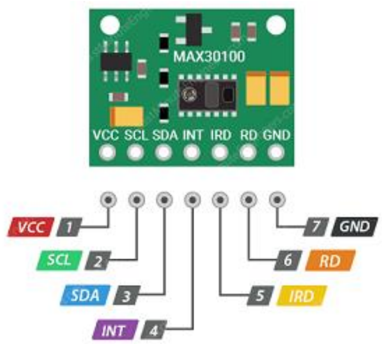


Figure 23: MAX30100 pulse oximeter pinout

4.5.4.4 Display (0.96 OLED)

OLED display [26] is one of the most attractive displays available for a microcontroller. it has a good pixel density and view angle which make it good for small pixel graphics. It can be interfaced with microcontroller using either I2C or SPI. OLED display was used in this study to display the vital signs values and anomaly prediction output. Figure 24 shows the pinout of OLED display.

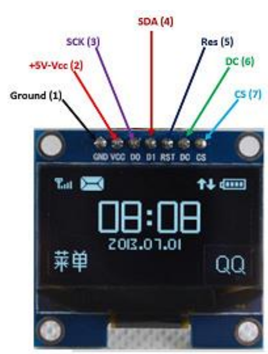


Figure 24: SSD1306 OLED Display Pinout

4.5.4.5 Buzzer (5V active Arduino buzzer)

A buzzer component to add sound features to the system. active buzzer [27] is a type of can be used to generate tones and sounds from a 5V signal. It has two pins for interfacing it to microcontroller or any other system. This component was used in this study to generate sound alert when anomaly is detected in vital signs. Figure 25 shows the pinout of active buzzer.

Figure 25: Active buzzer pinout

4.5.4.6 LED (RGB LED Diode Lights)

RGB LED [28] is a multi-color flash led which is mainly used for indication purpose. The are two types of RGB leds (1) the common cathode one and (2) the common anode one. This component was used in this study for indicating whether the vital signs are normal or abnormal by flashing different color. Figure 26 shows the pinout of RGB led.

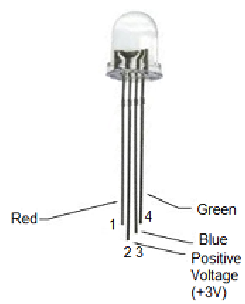


Figure 26:RGB led flasher pinout

4.5.4.7 GSM SIM800L

The SIM800L [29] is a GSM module that gives GSM capability to microcontroller. it can be connected to mibile network to perform massaging and calling functionalities. It can also be connected to the internet using IP, TCP, or GPRS. It can be interfaced with microcontroller using serial interface(UART). This module was used in this study to send the vital signs data to the cloud. Figure 27 shows the pinout of the SIM800L.

Figure 27: SIM800L pinout

4.5.4.8 14500 1300mah 3.7 V lithium ion rechargeable battery COM32

The ICR 1200mAh 14500 [33] is a lithium-battery for powering small powered electronics such as led flash lights. It is a rechargeable battery with up to 1000 cycles which is more suitable for devices located in remote. Figure 28 shows the layout of the ICR 1200mAh 14500 battery.

Figure 28: Lithium ion rechargeable battery

4.5.5 Circuit Diagram of the proposed embedded IoT vital signs device

Figure Figure 29 represent a schematic diagram of the embedded device prototype. All important components of the prototype a shown in the diagram including sensors, actuators, microcontroller, communication and module.

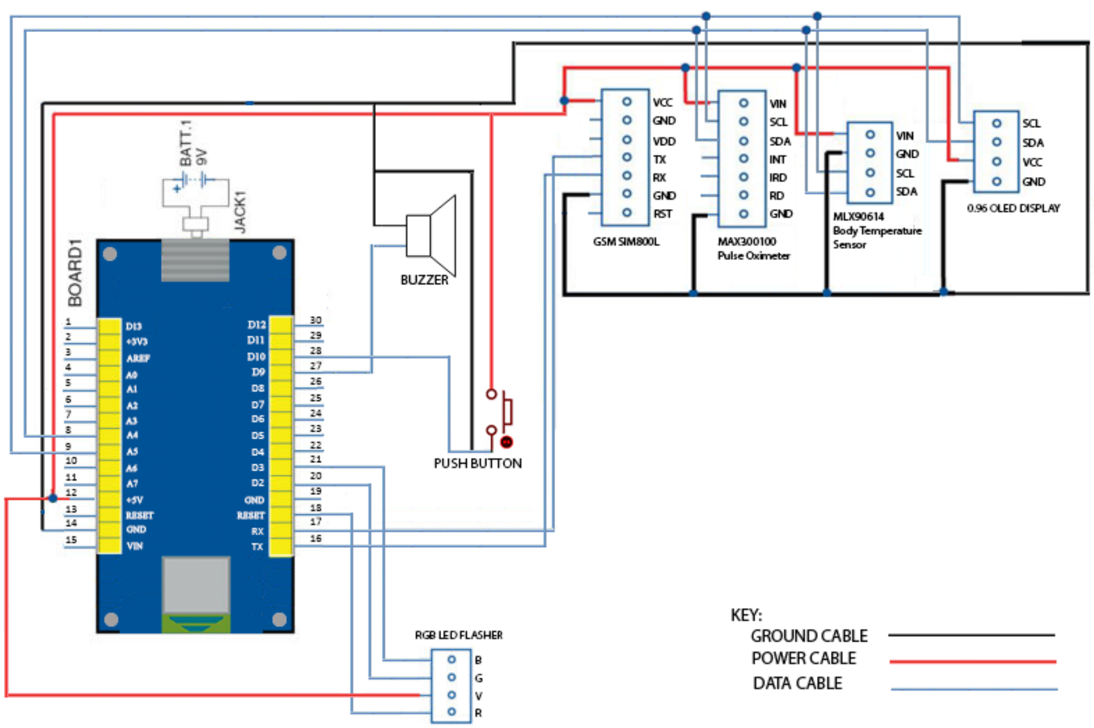


Figure 29: Schematic Diagram of the proposed embedded device

4.5.6 Embedded system Prototype

Figure 30 shows the embedded system prototype composed of different prototyping components including micro controller, sensors, actuators and GSM module.

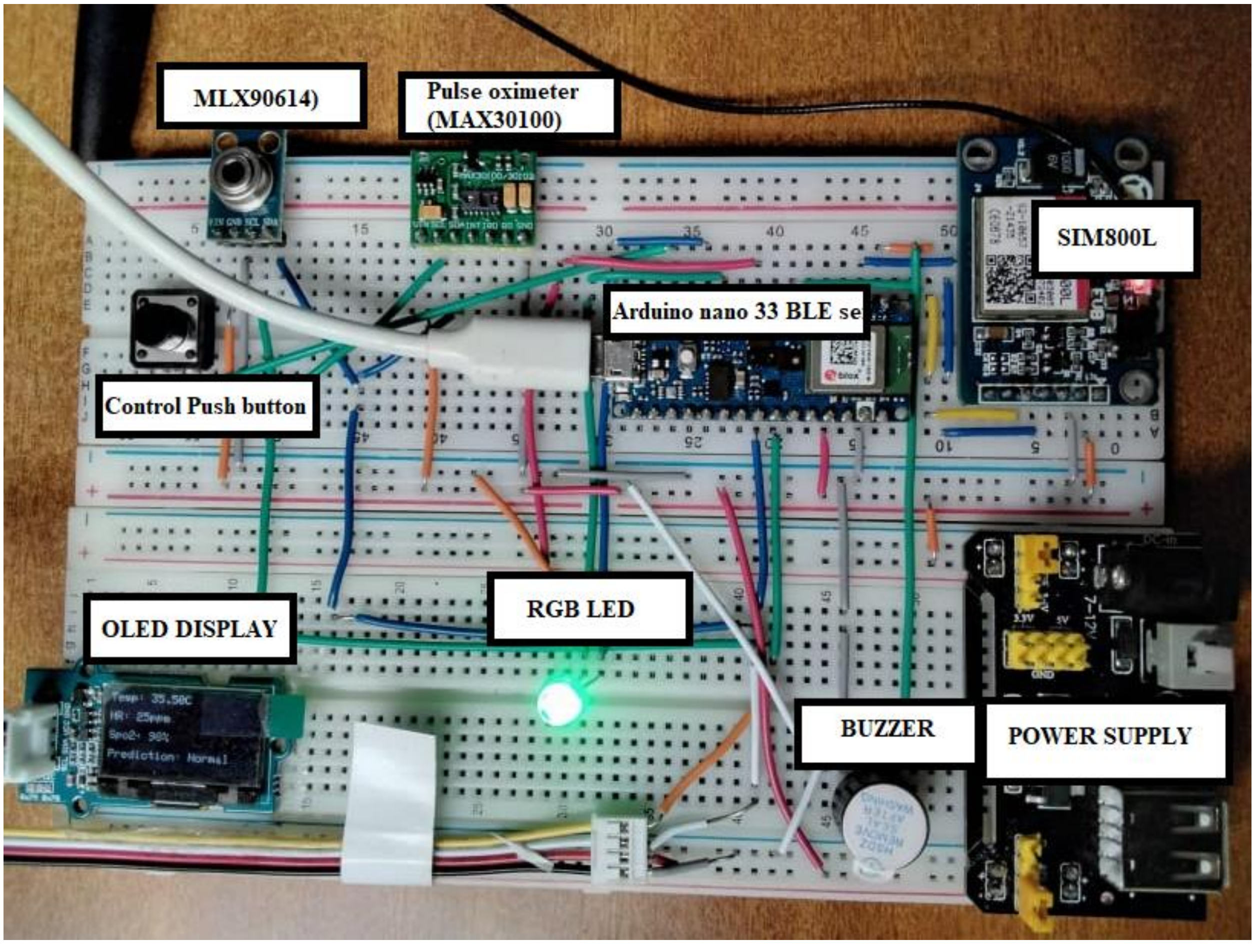
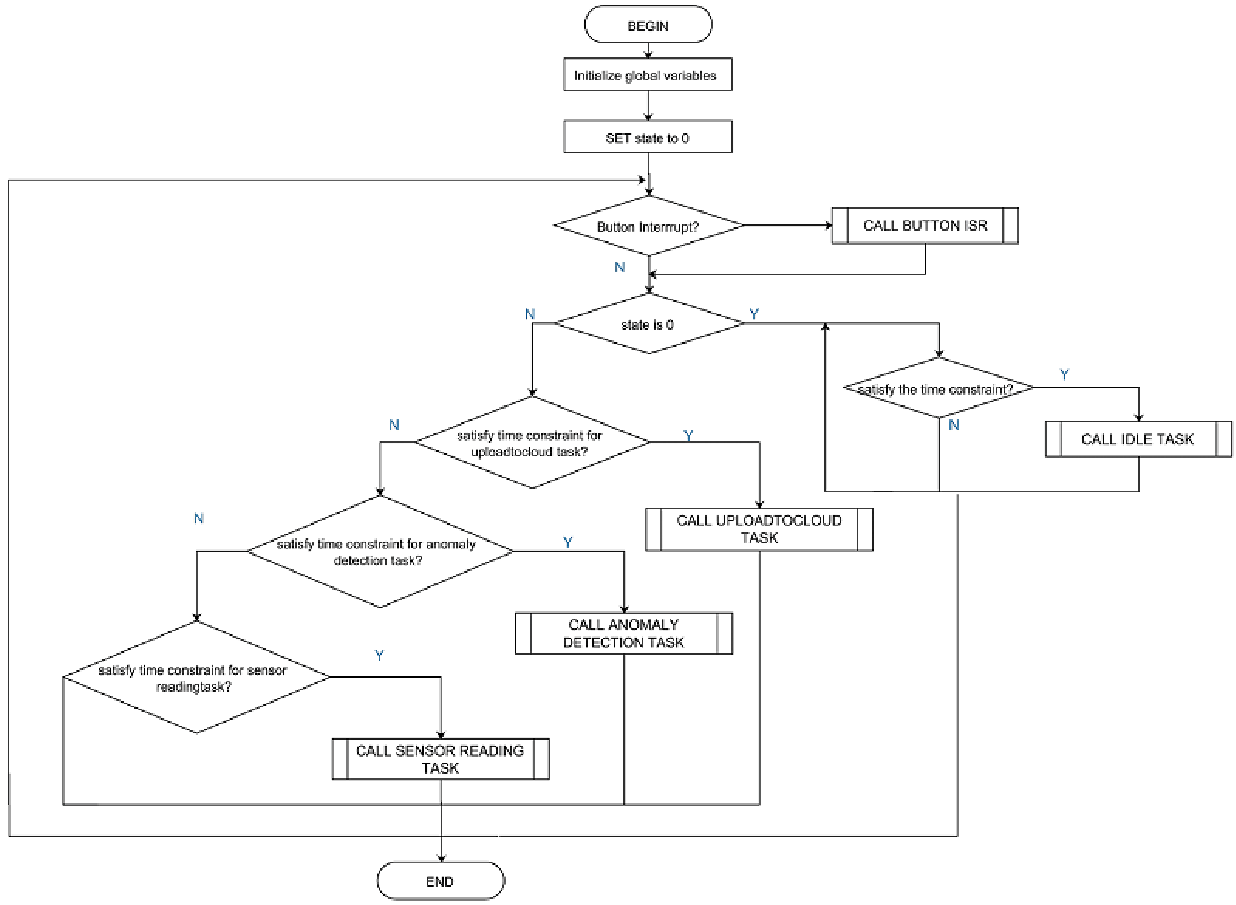


Figure 30: Embedded System Prototype

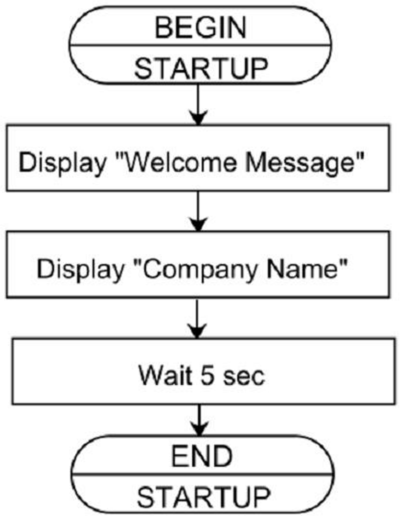
4.5.7 Flowchart diagram for embedded system Prototype

Figure 31 (a), (b), (c), (d), (f), (g), (h) represent the algorithms in terms of flowchat for various sub-programs of the embedded system prototype.

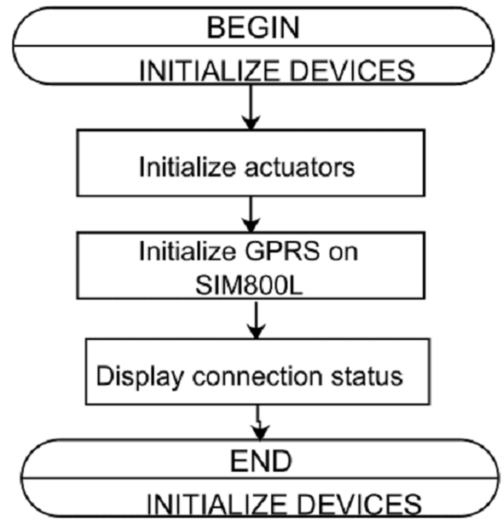


(a)

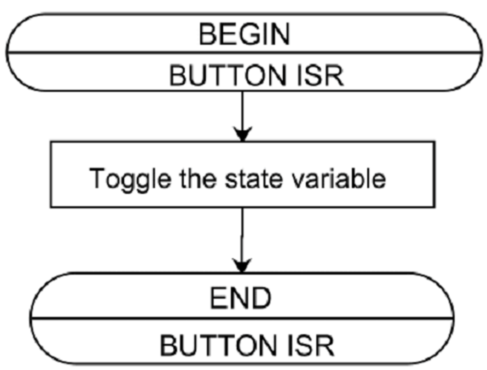
Figure 31: Flowchart Diagram


BEGIN
STARTUP
Display "Welcome Message"
Display "Company Name"
Wait 5 sec
END
STARTUP

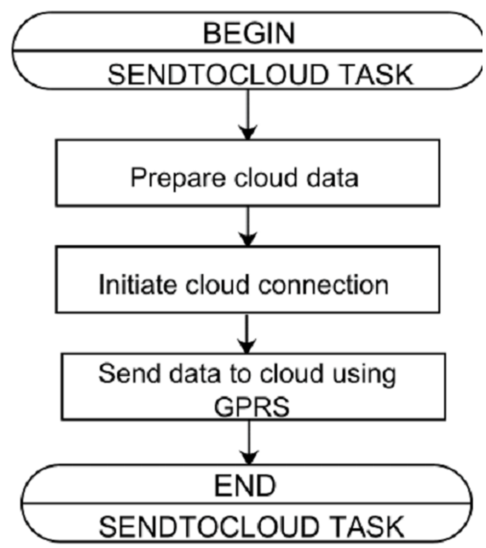
(b)


BEGIN
INITIALIZE DEVICES
Initialize actuators
Initialize GPRS on SIM800L
Display connection status
END
INITIALIZE DEVICES

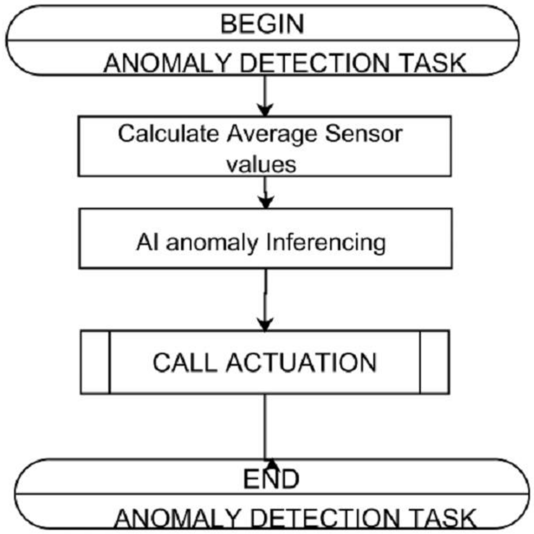
(c)


BEGIN
BUTTON ISR
Toggle the state variable
END
BUTTON ISR

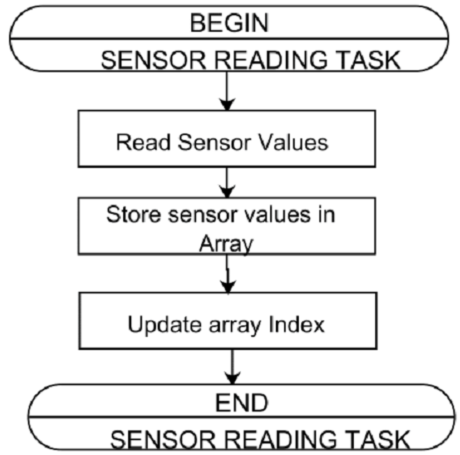
(d)


BEGIN
SENDTOCLOUD TASK
Prepare cloud data
Initiate cloud connection
Send data to cloud using GPRS
END
SENDTOCLOUD TASK

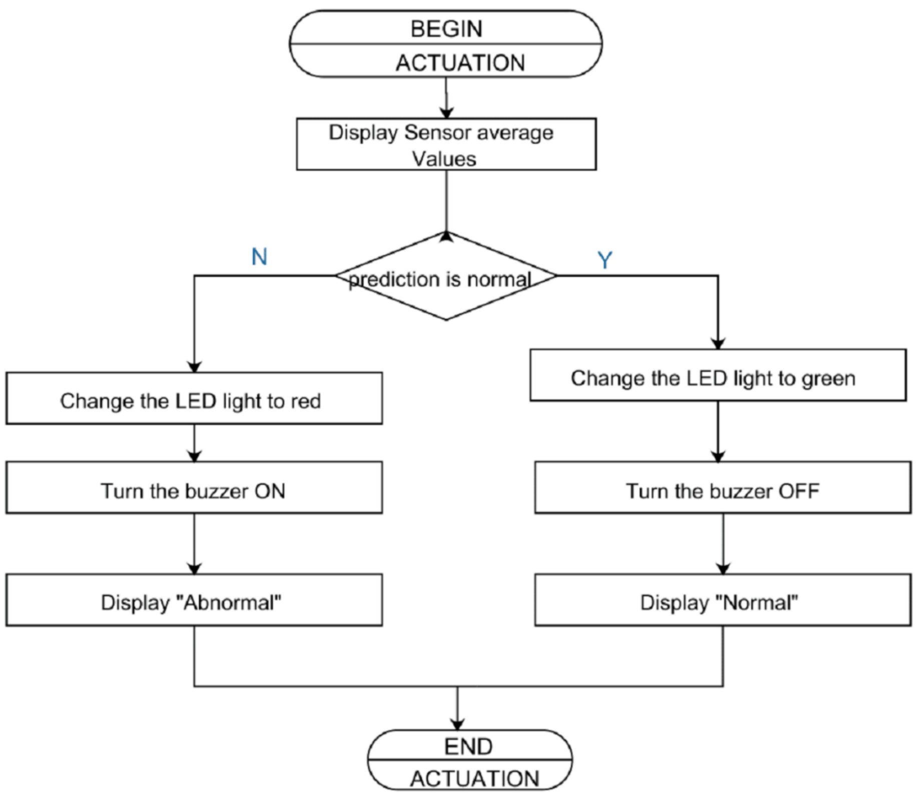
(e)


BEGIN
ANOMALY DETECTION TASK
Calculate Average Sensor values
AI anomaly Inferencing
CALL ACTUATION
END
ANOMALY DETECTION TASK

(f)


BEGIN
SENSOR READING TASK
Read Sensor Values
Store sensor values in Array
Update array Index
END
SENSOR READING TASK

(g)


BEGIN
ACTUATION
Display Sensor average Values
prediction is normal
N
Y
Change the LED light to red
Turn the buzzer ON
Display "Abnormal"
Change the LED light to green
Turn the buzzer OFF
Display "Normal"
END
ACTUATION

(h)

4.5.8 System PDL

Figure 32 represent algorithm for embedded system porotype in terms of PDL. The embedded system code is divided into different subprograms.

ACTUATION SUB-PROGRAM

```
BEGIN/ACTUATION
   Display average sensor values
   IF anomaly is detected THEN
      Turn the LED light to red
      Turn the buzzer ON
      Display message "Abnormal"
   ELSE
      Turn the LED light to green
      Turn the buzzer OFF
      Display message "Normal"
   ENDIF
END/ACTUATION
```

SENSOR_READING_TASK SUB-PROGRAM

```
BEGIN/SENSOR_READING_TASK
  DO WHILE time constraint satisfy
     Read sensor values
     Store sensor values in array
     Update array index
  ENDDO
END/SENSOR_READING_TASK
```

STARTUP SUB-PROGRAM

```
BEGIN/STARTUP
   Display "WELCOME MESSAGE"
   Display "CAMPANY NAME"
   Wait 5 Sec.
END/STARTUP
```

BUTTON_ISR SUB-PROGRAM

```
BEGIN/BUTTON_ISR
   Toggle the state
   variable
END/BUTTON_ISR
```

SEND_TO_CLOUD_TASK SUB-PROGRAM

```
BEGIN/SEND_TO_CLOUD_TASK
  Prepare cloud data
  Initiate cloud connection
  Send data to cloud using GPRS
END/SEND_TO_CLOUD_TASK
```

INITIALIZE_DEVICES SUB-PROGRAM

```
BEGIN/INITIALIZE_DEVICES
      Initialize Actuators
      Initialize GPRS on SIM800L
      Initialize OLED Display
      Display Connection Status
END/INITIALIZE_DEVICES
```

ANOMALY_DETECTION_TASK SUB-PROGRAM

```
BEGIN/ANOMALY_DETECTION_TASK
  Calculate Average Sensor Values
  Perform AI anomaly Inference
  CALL ACTUATION
END/ ANOMALY_DETECTION_TASK
```

MAIN-PROGRAM

```
BEGIN
      state = 0
      vitalSgns = array of vital signs
      readings
      avgVitalSigns = array of average vital
      signs reading
      CALL STARTUP
      CALL INITIALIZE_DEVICES
      DO FOREVER
            IF button interrupt is detected
            THEN
                  CALL BUTTON_ISR
            REPEAT
                  CALL IDLE_TASK
                  Wait 2 sec.
            UNTIL state is 1

            REPEAT
                  CALL SENSOR_READING_TASK
                  CALL
                  ANOMALY_DETECTION_TASK
                  CALL SEND_TO_CLOUD_TASK
            UNTIL state is 0
      ENDDO
END
```

Figure 32: System PDL

4.6 Edge AI model training

4.6.1 Data acquisition: data acquisition

The model training was based on the open dataset which was described in section 3.4.1.

4.6.2 K-NN algorithm design with jupyter notebook

The datasets were uploaded as CSV files to online jupyter notebooks known as google colab.

the data were spited into training and test dataset. Where by 30% of the data were used for testing the model and 70% of the data were used for training. The model was trained with the default value of K = 1. The model was then tested with different value of K and calculate error rate. The Figure 33 shows the relationship of K with error rate. The Figure 33 shows that the error rate is smallest when K =4. Therefore, the optimum value of K for predicting the best output is 4.

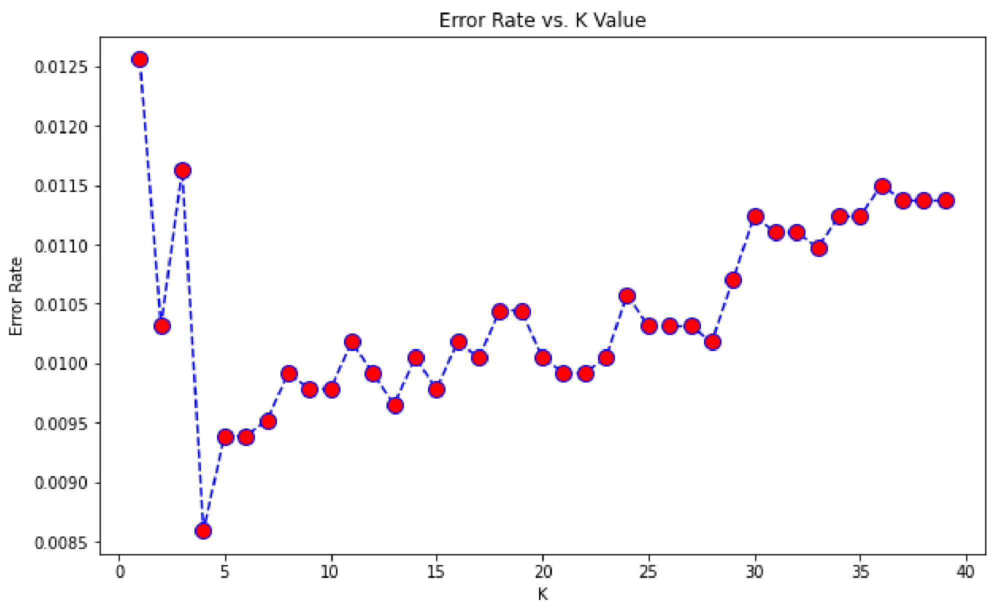


Figure 33: Relationship between error rate and K value

K-NN algorithm

Figure 34 shows the K-NN algorithm to classify the new data. The algorithm starts by setting parameter K. The distance between data to all training data is calculated and sorted in ascending order. Then shorted distance is determined with the order of K neighbors and associated with the corresponding class. From K nearest neighbors the number of each class is determined and set the class as class data to be evaluated.

Euclidian formula for calculating distance between data:

$$d_i = \sqrt{\sum_{i=1}^{p} \left(x_{2i} - x_{1i}\right)^2}$$

Where:

d = Distance formed

x_1 = Sample Data

p = Dimension Data

x_2 = Data Test / Testing

i = Variable Data

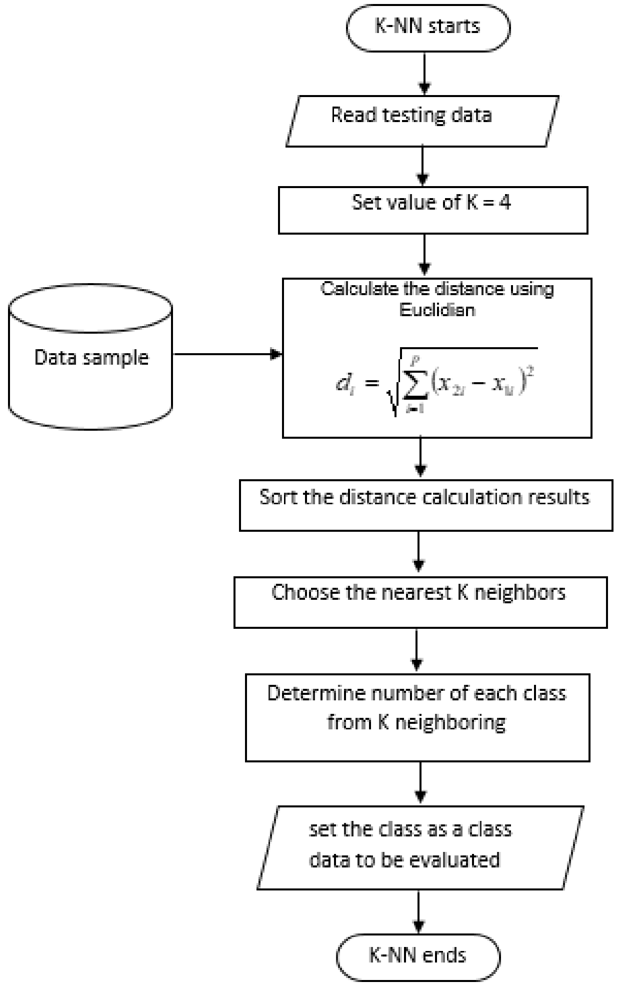


Figure 34: Flowchart of KNN Algorithm

5 CHAPTER 5

SYSTEM RESULTS AND ANALYSIS

5.1 Recommender virtual assistant

5.1.1 Conversation-Driven Development (CDD)

Rasa framework employs CDD. [30] CDD is used to improve the chatbot assistant through the analysis of user interactions. This method has been motivated by the fact that users will unpredictably say something with their own words that the chatbot didn't anticipate before.

5.1.2 CDD process

CDD has been performed cyclically through the following steps:

- The first step is that the chatbot was shared with different test users who are not familiar with how the assistant works from the inside. Test users include non-expert users, health professionals as well as emergency units.
- With the use of rasa-x, the test user conversations were then reviewed periodically as shown in Figure 35 which shows the list of user messages with predicted intents.
- From the list of user conversations in Figure 35, the predicted messages were then annotated to the appropriate intent and added as NLU training data.
- The assistant is then tested to check if it responds as expected.
- The performance of the assistant was measured and tracked regularly to identify any failure.

Sentence	Predicted Intent and Confidence
Which vital sign is most important?	user_what_is_wrong (1.0)
Is body weight considered during the measurement of standard vital signs?	user_what_is_wrong (0.94)
Is the vital signs range differ from children adult and elderly?	user_what_is_wrong (0.98)
What are normal range of vital signs for adult?	user_what_is_wrong (0.96)
what are the normal range of vital signs?	user_what_is_wrong (0.98)
Why is checking vital signs important?	user_what_is_wrong (0.79)
How many essential primary vital signs for physiological functions of living organism?	user_what_is_wrong (0.51)

Figure 35: NLU inbox showing a list of users messages

5.1.3 Recommender Virtual Assistant conversation demonstration

Figure 36 and Figure 37 shows the conversations demo for vital signs recommendation assistant.

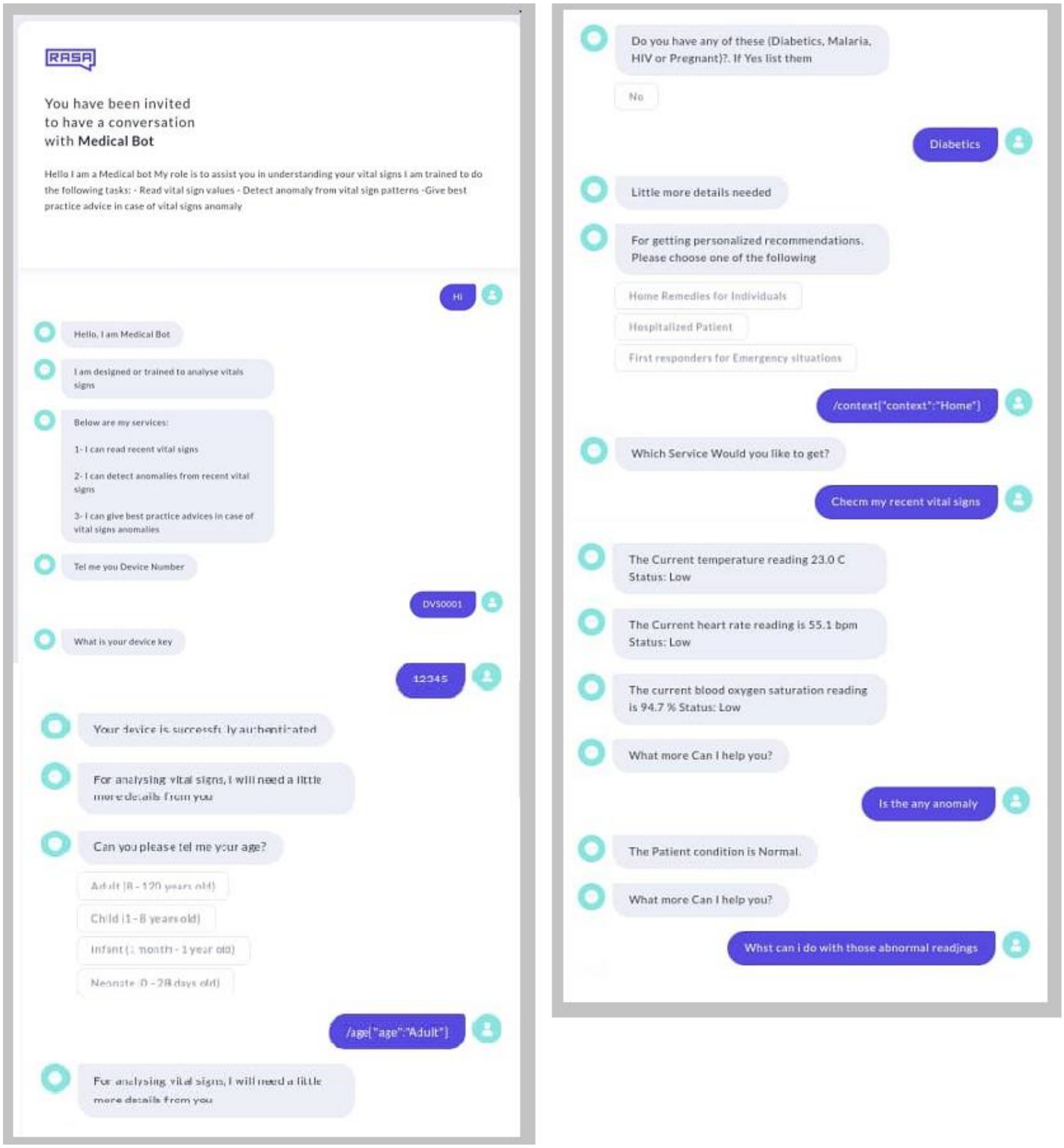


Figure 36: Chatbot conversation demo part 1

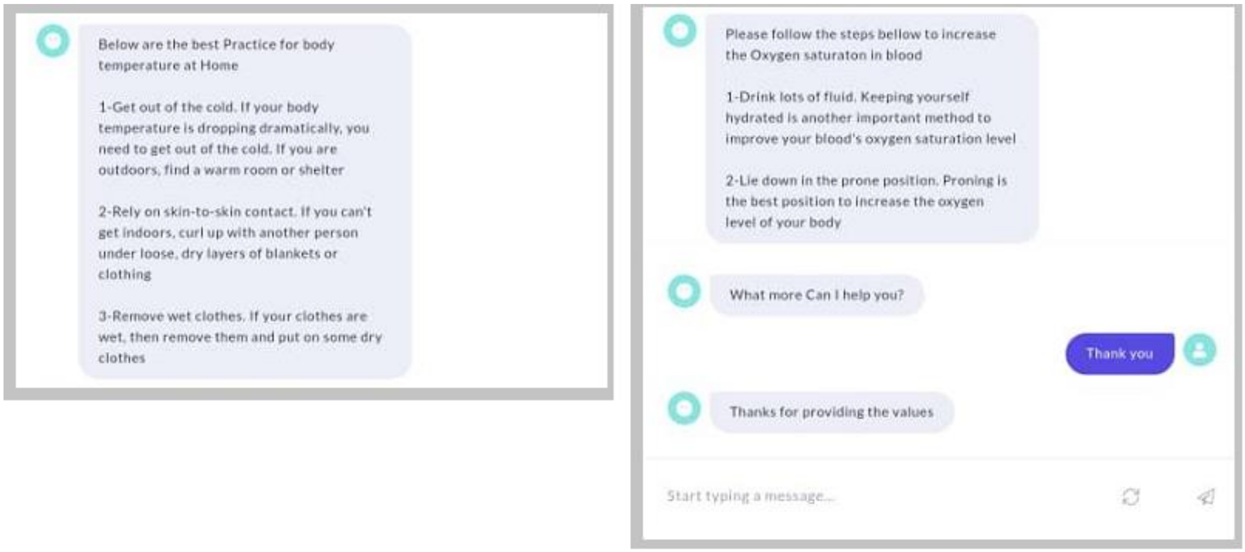


Figure 37: Chatbot conversation demo part 2

5.1.4 Testing and evaluation of RASA core and RASA NLU

RASA Chabot is composed of deep learning models for performing accurate predictions and producing accurate results. The accuracy of the chatbot is based on how best the best can recognize the intents and entities of user utterances and how best it can predict the response back to the user. To calculate the accuracy of the stories, test stories similar to the stories were created whereby intents with corresponding actions are written to define the conversation flow of the bot. RASA offers multiples commands for checking the accuracy of both RASA core and RASA NLU.

5.1.4.1 Testing and evaluation of an NLU model

RASA NLU test is performed by running "rasa test nlu" command. It will check whether intents and entities were correctly classified for a given user utterance. The test results are presented in terms of a confusion matrix and histogram. Correct predictions are presented by diagonal entries while incorrect predictions are presented on either side of the diagonal.

DIETClassifer was used to classify intents and extract entities. Table 9 and Figure 38 show a summary of DIET as an intent classifier which presents the classified intent using the confusion matrix and the prediction confidence distribution respectively.

Table 9 shows the results of predicting intents using DIETclassifer with a Confusion Matrix. It shows that there are only 4 questions that are at either side of the diagonal, whereas the

DIETclassifer predicts them incorrectly. Other questions which are located on the diagonal were predicted correctly and have correct responses.

Figure 38 presents the predicting probability distribution chart for intent classification which shows that no question was classified wrongly with a confidence. All questions were correctly classified with confidence more than 0.8. Figure 39 presents the predicting probability distribution chart for entity extraction which shows that no entity was predicted wrongly with. All questions were correctly classified with confidence more than 0.99.

For entity extraction, Table 10 represents an entity confusion matrix using DIETClassifier. The table shows that among 486 entities, only 48 entities were extracted using DIET and 438 entities were missing.

Table 9: Intent confusion matrix

		Predicted Label															
True Label		affirm	age	bot_challange	context	deny	device_number	exit	goodbye	greet	health	set_code	User_diagnostic_e	user_what_is_sugg	user_what_is_wro	Vital_signs	Total
	affirm	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
	age	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	8
	bot_challange	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	5
	context	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	7
	deny	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	7
	device_number	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	3
	exit	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	6
	goodbye	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	9
	greet	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	17
	health	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	9
	set_code	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	3
	User_diagnostic_enquiries	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	21
	user_what_is_suggested	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	25
	user_what_is_wrong	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	24
	Vital_signs	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	21
	Total	4	8	5	0	7	3	0	13	16	1	0	21	27	25	20	150

Table 10: Entity confusion matrix of entity extraction using DIETClassifier

True	Predicted Label							
		age	context	device_number	health_status	no_entity	vital_signs	Total

	age	8	0	0	0	0	0	8
	context	0	8	0	0	0	0	8
	device_number	0	0	3	0	0	0	3
	health_status	0	0	0	8	0	0	8
	no_entity	0	0	0	0	428	0	828
	vital_signs	0	0	0	0	0	21	21
	Total	8	8	3	8	428	21	486

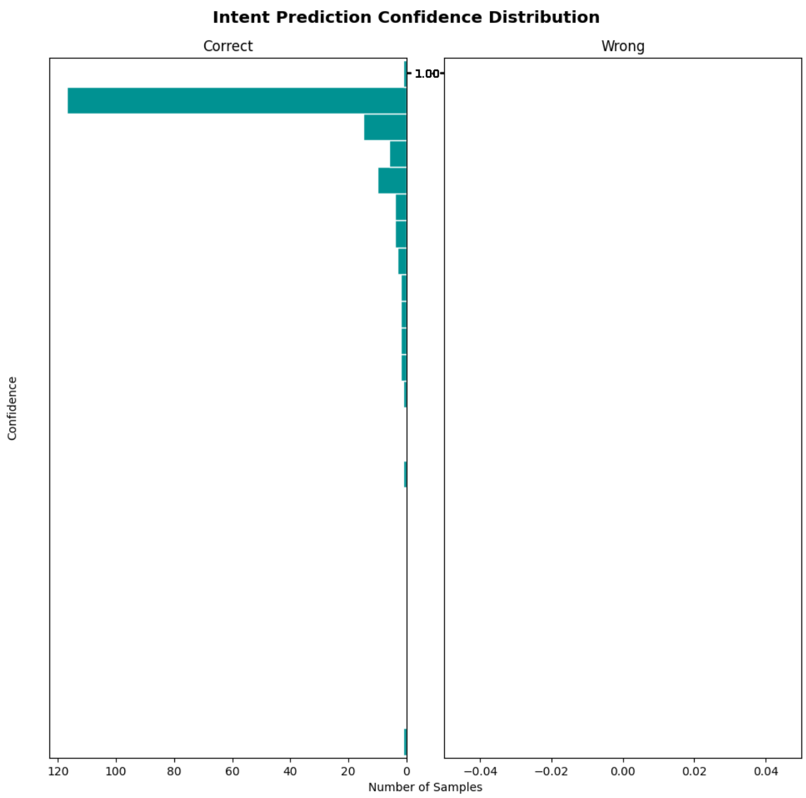


Figure 38: The predicting probability distributions for Intent

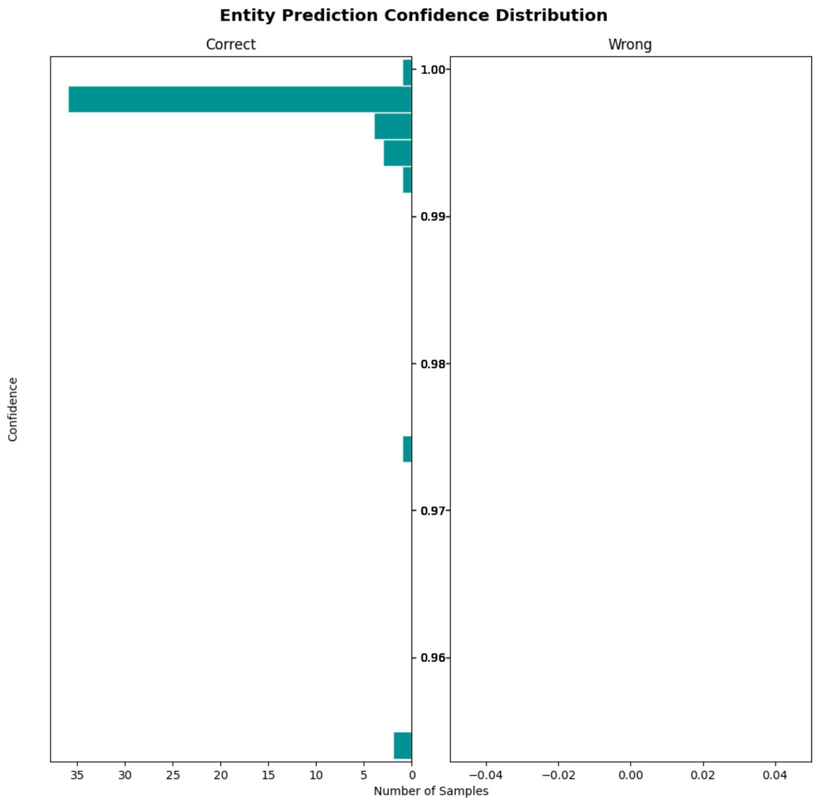


Figure 39: Entity Prediction confidence distribution using DIETClassifier

Table 11 represent an entity confusion matrix using RegexEntityExtractor. The table shows that among 476 entities, only 9 entities were extracted using regex and 467 entities were missing.

Table 11: Entity confusion matrix of entity extraction using RegexEntityExtractor

		Predicted Label						
		age	context	device_number	health_status	no_entity	vital_signs	Total
True Label	age	0	0	0	0	8	0	8
	context	0	0	0	0	8	0	8
	device_number	0	0	1	0	2	0	3
	health_status	0	0	0	8	0	0	8
	no_entity	0	0	0	2	426	0	428
	vital_signs	0	0	0	0	21	0	21
	Total	0	0	1	10	465	0	476

Table 12 represents the summary of RASA model of cross-validation test of RASA NLU models for various techniques of intent classifications and entity extract.

Table 12: Rasa models evaluation summary

Evaluation Process	Accuracy	F1-score	Precision
Intent evaluation results			
Model Training	1.000	1.000	1.000
Model Testing	0.793	0.769	0.775
Entity evaluation results			
Entity extractor: RegexEntityExtractor			
Model Training	0.912	0.000	0.000
Model Testing	0.911	0.000	0.000
Entity extractor: DIETClassifier			
Model Training	1.000	1.000	1.000
Model Testing	0.958	0.600	0.711

5.1.4.2 Testing and evaluation of rasa core model

The RASA Core test is performed by running the “rasa test core” command. The command will check whether the response and actions to the users are correct or not.

Figure 40 represents the evaluation results of the rasa core model on the action level. The figure shows that all 25 actions were correctly executed with an accuracy of 100 %.

Figure 41 represents the evaluation results of the rasa core model at the end to end level. This means that rasa core can predict the next action to be performed by just considering the user utterance. The figure shows that all 9 stores were predicted correctly with an accuracy of 100%. This means that, there are no failed stories.

```
rasa.core.test  - Evaluation Results on ACTION level:
rasa.core.test  -  Correct:          25 / 25
rasa.core.test  -  F1-Score:         1.000
rasa.core.test  -  Precision:        1.000
rasa.core.test  -  Accuracy:         1.000
rasa.core.test  -  In-data fraction: 1
```

Figure 40: Evaluation Results on ACTION level

```
rasa.core.test  - Evaluation Results on END-TO-END level:
rasa.core.test  -  Correct:          9 / 9
rasa.core.test  -  Accuracy:         1.000
```

Figure 41: Evaluation Results on END-TO-END level

Table 13 represents action confusion matrix. From the matrix, it can be seen that all elements are at the diagonal which means that all actions were predicted correctly.

Table 13: Action confusion matrix

		Predicted Label							
True Label		action_listen	action_request_vital_signs	activity_tracking_action	check_abnormal_vital_sign_ction	diagnostic_response_action	exit_action	utter_medbot_great	utter_medbot_great
	action_listen	9	0	0	0	0	0	0	0
	action_request_vital_signs	0	1	0	0	0	0	0	0
	activity_tracking_action	0	0	9	0	0	0	0	0
	check_abnormal_vital_sign_ction	0	0	0	1	0	0	0	0
	diagnostic_response_action	0	0	0	0	1	0	0	0
	exit_action	0	0	0	0	0	1	0	0
	utter_medbot_great	0	0	0	0	0	0	1	0
	utter_medbot_intro	0	0	0	0	0	0	0	1

5.2 Cloud-based AI vital signs anomaly detection model training and inferencing

5.2.1 Cloud-based AI training output

To achieve a good performance of the model, the model has been trained with a different number of epochs, and finally, a good performance of the model was obtained with 30 epochs and a learning rate of 0.001. From Figure 42 and Figure 43, it has been shown that the model has achieved a validation accuracy of 97.92 % with a validation loss of 6.05 % and a training accuracy of 97.53 %with a training loss of 6.33 %. The model has been tested with the testing dataset and the results show that an accuracy of 97.62 % and a loss of 6.41 % have been achieved. The ANN model deployed the cloud vital signs app whereby anomaly detection is performed.

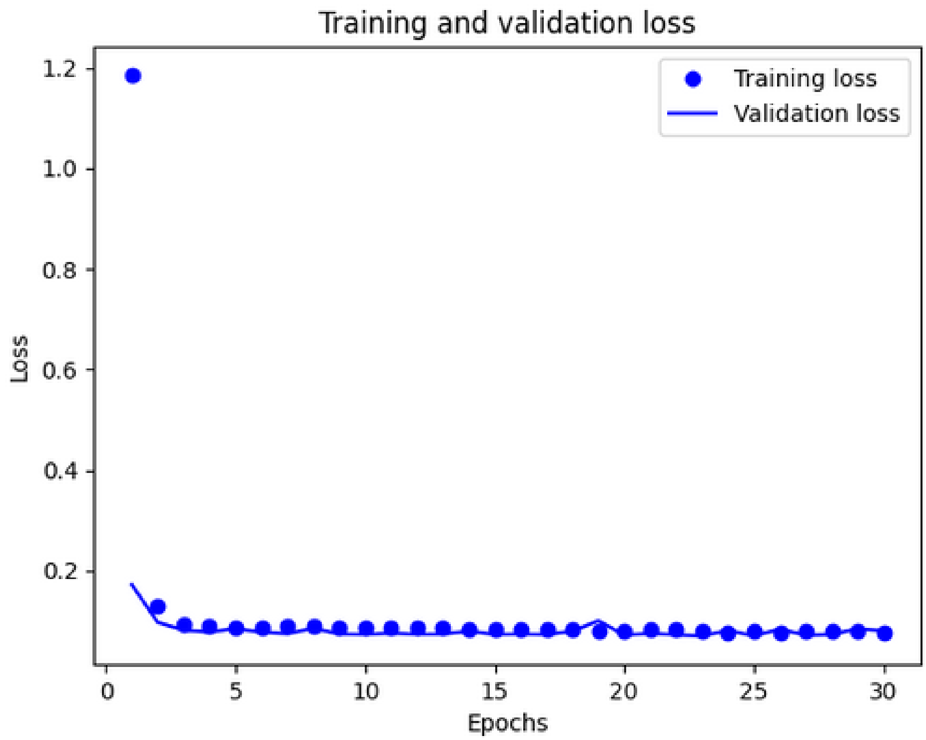


Figure 42: Training and validation loss

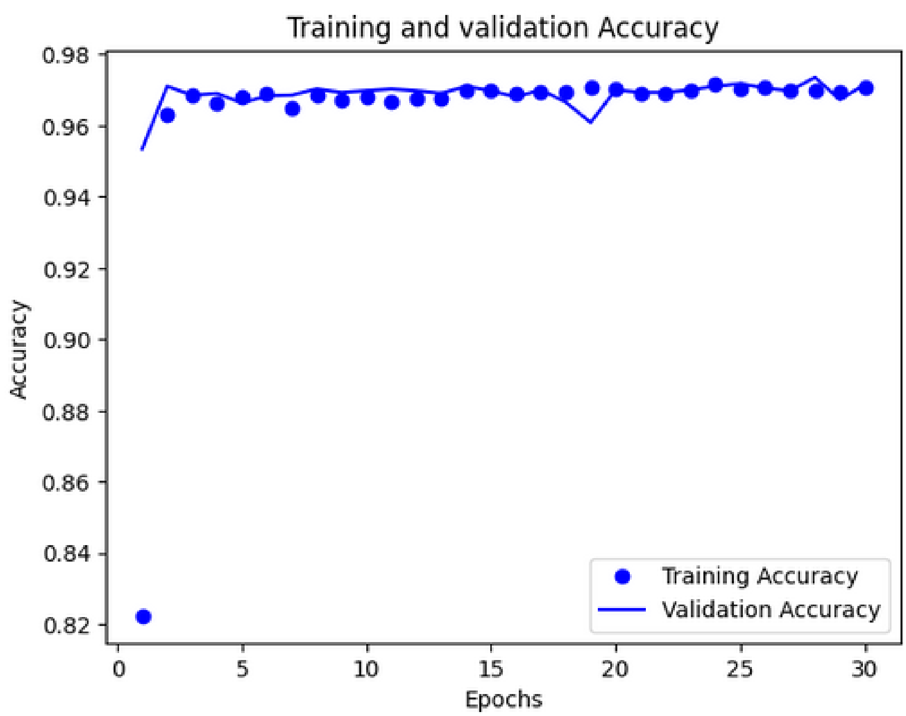


Figure 43: Training and validation accuracy

5.3 Edge based K-NN model training output and inferencing

5.3.1 K-NN model training output

After turning the value of K to get an optimum value. A good performance of the model was achieved. Figure 44 shows the model performance for K = 1 where by the accuracy was 97%. Figure 45 shows the model performance for K = 4 where by the accuracy increased to 100%.

	precision	recall	f1-score	support
0	0.96	0.97	0.97	1744
1	0.97	0.99	0.97	5820
accuracy			0.97	7564
macro avg	0.98	0.98	0.98	7564
weighted avg	0.99	0.99	0.99	7564

Figure 44: Model performance for K = 1

	precision	recall	f1-score	support
0	0.99	0.99	0.98	1744
1	1.00	0.99	0.99	5820
accuracy			0.99	7564
macro avg	0.98	0.99	0.99	7564
weighted avg	0.99	0.99	0.99	7564

Figure 45: Model performance for K = 4.

5.3.2 Inference on a virtual embedded board

Figure 46 shows the edge K-NN model inference on virtual embedded Proteus simulator. Due to the limited number of available sensor in Proteus, the sensor data was loaded on virtual SD card and the model inference results was displayed on LCD and virtual terminal. Figure 47 shows inference output from Proteus displayed on virtual terminal. Data loaded from SD card were put as an input of the model.

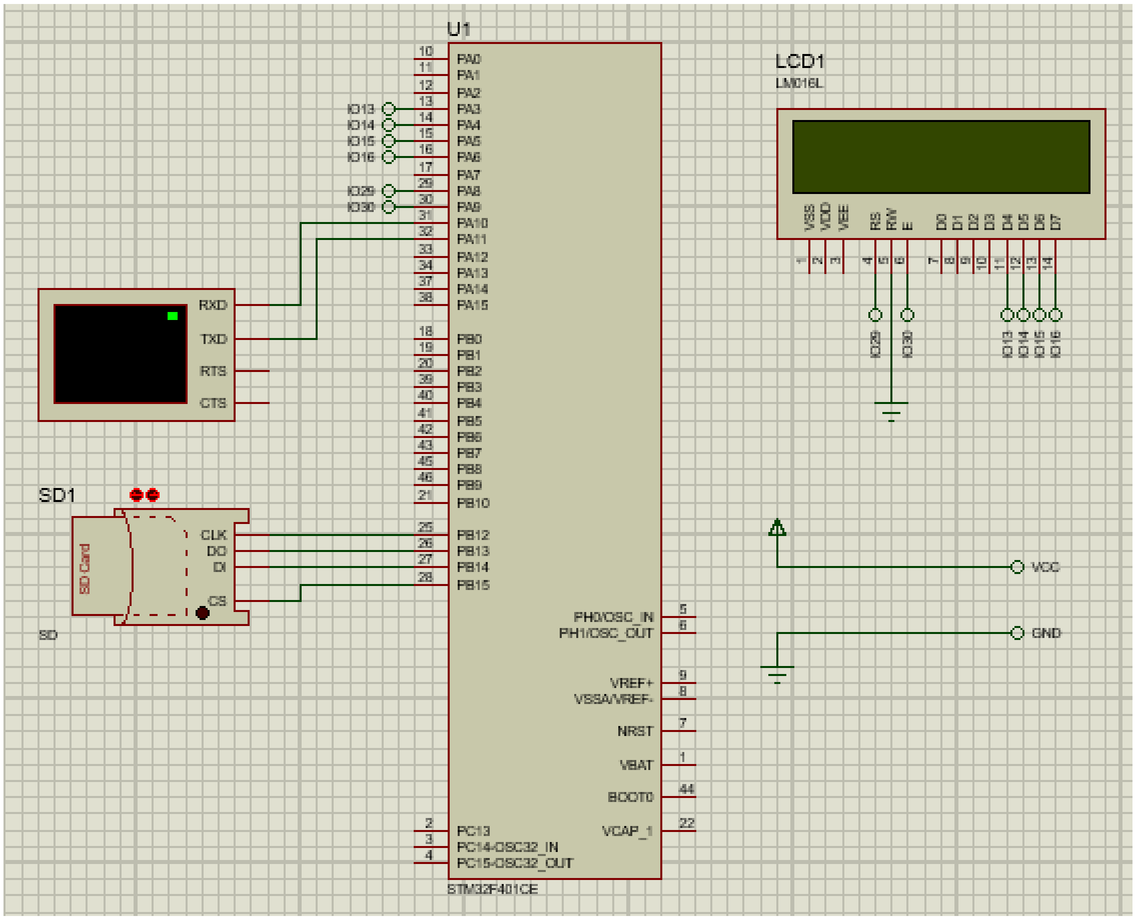


Figure 46: Proteus simulation

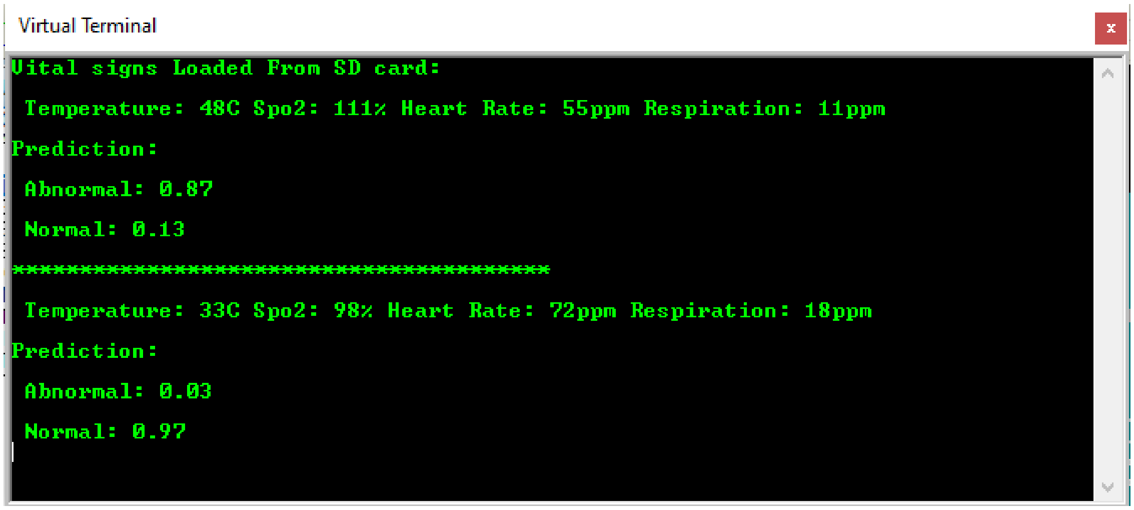


Figure 47: Inference Output on Proteus

5.3.3 Comparison of device performance with commercial available kit.

Device performance was referred from the existing device developed in [32] which was using the same sensor models for monitoring vital signs. In [32] it has been approved that used sensors provided same accuracy as the commercial kits. The existing commercial kit shown in the Figure 48 for body temperature monitoring was employed in validating and checking its coloration with the proposed MLX90614 contactless body temperature sensor shown in Figure 22. The Wellue Finger Oximeter shown in Figure 49 for measuring heart rate and oxygen saturation was employed to validate and check for coloration with the proposed pulse oximeter (MAX30100) shown in Figure 23. The results were presented in Figure 50, Figure 51 and Figure 52 which led to the conclusion that the proposed sensors perform at nearly the same accuracy as the commercial available sensor kits.

Figure 48: Commercial digital thermometer

Figure 49: The Wellue Finger Oximeter

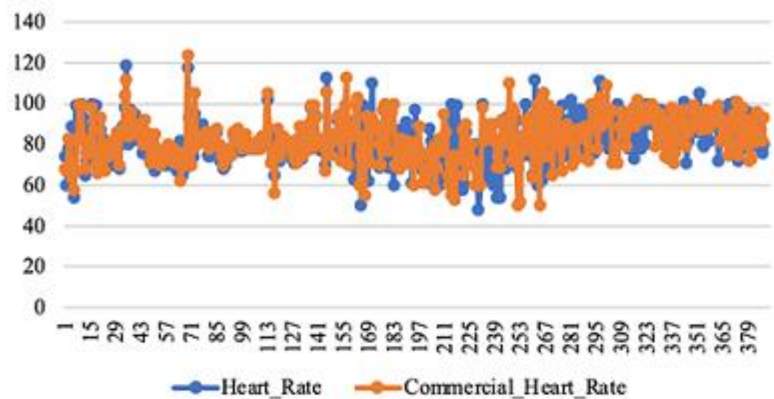


Figure 50: Heart beat rate values for the Wellue pulse oximeter and the

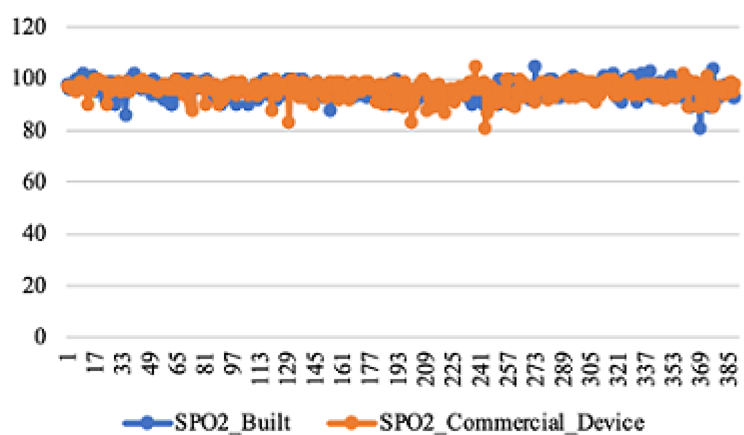


Figure 51: SPO2 values for the Wellue pulse oximeter and the developed system

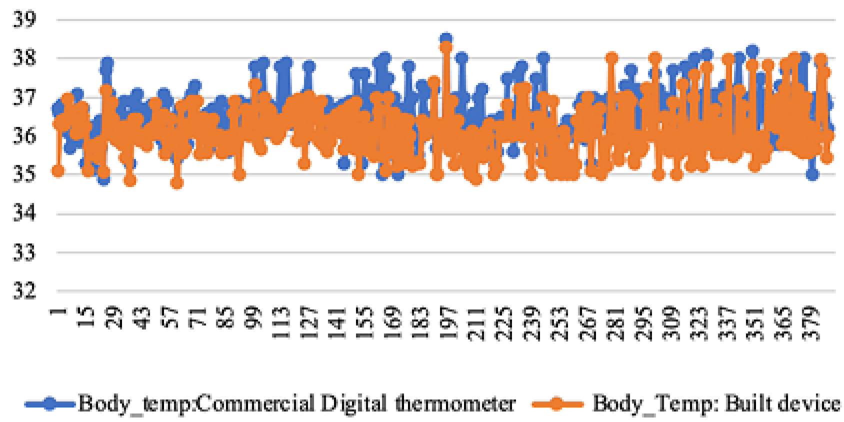


Figure 52: Temperature values of the digital thermometer and the developed

6 CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

As the doctor-patient ratio continues to decrease especially in developing countries, relying on doctors for real-time dissemination of medical data such as vital signs becomes difficult, thus potentially putting at risk patients that may need rapid intervention. Therefore, conversational AI assistants offer a complementary solution to relieve the workload of doctors in dealing with recurring requests such as interpreting patients' vital signs and providing recommendations in case of anomaly. This study presents a research-driven design and development of a cloud-based conversational AI platform trained to predict vital signs anomalies and provides recommendations from a living recommendation dataset created by physicians. The methodology of Conversation Driven-Development (CDD) has been adopted to ensure user interaction feedback is iteratively and progressively leveraged to reinforce the learning of the developed conversational AI assistant. The developed platform also provides a user interface for physicians to continuously produce a recommendation dataset of actions to take in case of an anomaly is detected. From the experimentation phase, the proposed platform shows great potential as a solution to manage and overcome the consequences of low physician-patient ratios, especially in developing countries. The next phases of this research will cover (1) the migration from a rule-based recommendation system towards an AI-based and (2) to develop of an advanced smart wearable vital sign device embedding both AI anomaly prediction, AI recommendation, and Conversational AI NLP for autonomous operation, especially in offline scenarios.

List of References

[1] Brekke IJ, Puntervoll LH, Pedersen PB, Kellett J, Brabrand M. The value of vital sign trends in predicting and monitoring clinical deterioration: A systematic review. PLoS One. 2019 Jan 15;14(1):e0210875. doi: 10.1371/journal.pone.0210875. PMID: 30645637; PMCID: PMC6333367.

[2] Mirela Prgomet, Magnolia Cardona-Morrell, Margaret Nicholson, Rebecca Lake, Janet Long, Johanna Westbrook, Jeffrey Braithwaite, Ken Hillman, Vital signs monitoring on general wards: clinical staff perceptions of current practices and the planned introduction of continuous monitoring technology, International Journal for Quality in Health Care, Volume 28, Issue 4, September 2016, Pages 515–521, https://doi.org/10.1093/intqhc/mzw062.

[3] Gao, T., Greenspan, D., Welsh, M., Juang, R.R. and Alm, A., 2006, January. Vital signs monitoring and patient tracking over a wireless network. In 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference (pp. 102-105). IEEE.

[4] Hong, W., Earnest, A., Sultana, P., Koh, Z., Shahidah, N. and Ong, M.E.H., 2013. How accurate are vital signs in predicting clinical outcomes in critically ill emergency department patients? European Journal of Emergency Medicine, 20(1), pp.27-32.

[5] Simbawa, J.H., Jawhari, A.A., Almutairi, F., Almahmoudi, A., Alshammrani, B., Qashqari, R. and Alattas, I., 2021. The Association Between Abnormal Vital Signs and Mortality in the Emergency Department. Cureus, 13(12).

[6] Sheth, A., Yip, H.Y. and Shekarpour, S., 2019. Extending patient-chatbot experience with internet-of-things and background knowledge: case studies with healthcare applications. IEEE intelligent systems, 34(4), pp.24-30.

[7] Kar, R. and Haldar, R., 2016. Applying chatbots to the internet of things: Opportunities and architectural elements. arXiv preprint arXiv:1611.03799.

[8] Dere, G., 2021. Biomedical applications with using embedded systems. Data acquisition-recent advances and applications in biomedical engineering.

[9] Syed Misbahuddin, Junaid Ahmed Zubairi, Abdul Rahman Alahdal, Muhammad Arshad Malik, "IoT-Based Ambulatory Vital Signs Data Transfer System", Journal of Computer

Networks and Communications, vol. 2018, Article ID 4071474, 8 pages, 2018. https://doi.org/10.1155/2018/4071474.

[10] Wenxi Chen, D. Wei, Xin Zhu, M. Uchida, S. Ding and M. Cohen, "A mobile phone-based wearable vital signs monitoring system," The Fifth International Conference on Computer and Information Technology (CIT'05), 2005, pp. 950-955, doi: 10.1109/CIT.2005.19

[11] Munem, A.S.A., Croock, M.S.: Health monitoring system for ambulance emergency system using wireless sensor network. Eng. Technol. J. 34, 2877–2890 (2016)

[12] Youssef Ali Amer, A., Wouters, F., Vranken, J., de Korte-de Boer, D., Smit-Fun, V., Duflot, P., Beaupain, M.H., Vandervoort, P., Luca, S., Aerts, J.M. and Vanrumste, B., 2020. Vital signs prediction and early warning score calculation based on continuous monitoring of hospitalized patients using wearable technology. Sensors, 20(22), p.659.

[13] Battineni, G.; Chintalapudi, N.; Amenta, F. AI Chatbot Design during an Epidemic like the Novel Coronavirus. Healthcare 2020,8,154

[14] Lee, C.O., Lee, M., Han, D., Jung, S. and Cho, J., 2008, July. A framework for personalized healthcare service recommendation. In HealthCom 2008- 10th International Conference on e-health Networking, Applications and Services (pp. 90-95). IEEE.

[15] Chen, J., Li, K., Rong, H., Bilal, K., Yang, N. and Li, K., 2018. A disease diagnosis and treatment recommendation system based on big data mining and cloud computing. Information Sciences, 435, pp.124-149.

[16] Zaki, W.M.A.W., Shakhih, M.F.M., Ramlee, M.H., Ab Wahab, A.: Smart medical chatbot with integrated contactless vital sign monitor. J. Phys. Conf. Ser. 1372, 012025 (2019).

[17] Kalla, D. and Samiuddin, V., 2020. Chatbot for medical treatment using NLTK Lib. IOSR J. Comput. Eng, 22.

[18] Shifa, G., Sabreen, S., Bano, S.T. and Fakih, A.H., 2020. Self-diagnosis medical chat-bot using artificial intelligence. Easychair Preprint No. 2736.

[19] Sivaraj, K., Jeyabalasuntharam, K., Ganeshan, H., Nagendran, K., Alosious, J. and Tharmaseelan, J., 2021. Medibot: End to end voice based AI medical chatbot with a smart watch.

[20] Ko, S.Q., Kumar, S.K., Jacob, J., Hooi, B.M., Soo, M., Nashi, N., Cruz, M.T.D., Yeo, A.W., Wong, Z.X., Smitasin, N. and Lum, L., 2022. Technology-Enabled Virtual Ward for COVID Management of the Elderly and Immunocompromised in Singapore: A Descriptive Cohort.

[21] https://www.cprime.com/resources/what-is-agile-what-is-scrum/ (ACCESSED ON 10th, October 2022).

[22] Human vital signs | Kaggle Accessed on 23rd, June 04:42 PM

Nano 33 BLE Sense | Arduino Documentation | Arduino Documentation Accessed on 18th, November 05:30 PM.

[23] https://docs.arduino.cc/hardware/nano-33-ble-sense Accessed on 18th, November 04:42 PM

[24] MLX90614 Non-Contact IR Temperature Sensor Pinout, Datasheet, Equivalents & Specs (components101.com)

[25] Datasheet, M., 2009. Pulse oximeter and heart-rate sensor IC for wearable health.

[26] https://components101.com/sensors/melexis-mlx90614-contact-less-ir-temperature-sensor Accessed on 18th, November 05:30 PM.

[27] Monochrome 0.96" OLED Display (SSD1306)

https://components101.com/displays/oled-display-ssd1306 Accessed on 17th, November 05:00 PM.

[28] Active Passive Buzzer

https://components101.com/misc/buzzer-pinout-working-datasheet Accessed on 18th, November 05:30 PM.

[29] RGB LED Diode Lights

https://components101.com/diodes/rgb-led-pinout-configuration-circuit-datasheet Accessed on 18th, November 05:30 PM.

[30] SIM800L GSM Module Pinout, Datasheet, Equivalent, Circuit, and Specifications

https://components101.com/wireless/sim800l-gsm-module-pinout-datasheet-equivalent-circuit-specs

[31] Alan Nichol. Conversation-driven development.

https://blog.rasa.com/conversation-driven-development-a-better-approach-to-building-ai-assistants/ , Accessed on: October, 2022.

[32] E. Twahirwa, K. Mtonga, K. Jayavel, P. Bamurigire, W. Kasakula. Technology Empowered Healthcare Response in Pandemics: An IoT-based Covid-19 Test and Isolate System. The 10th IEEE International Conference on E-Health and Bioengineering - EHB 2022.

[33] 14500-1200mah-3-7-v-lithium-ion-rechargeable-battery-com45 https://www.faranux.com/product/14500-1200mah-3-7-v-lithium-ion-rechargeable-battery-com45/ , Accessed on: November, 2022.

Appendices

Plagiarism check report using Turnitin plagiarism tool

Title: Developing an IoT-based Conversational AI Recommender Assistant for Vital Sign Predicted Anomalies

ORIGINALITY REPORT

15%	12%	9%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Rwanda Student Paper	1%
2	Submitted to Limerick Institute of Technology Student Paper	1%