[Journal of Engineering Research](https://digitalcommons.aaru.edu.jo/erjeng)

Volume 7
Issue 5 *This is a Special Issue from the Applied Innovative Research in Engineering Grand Challenges (AIRGEC) Conference, (AIRGEC 2023), Faculty of Engineering, Horus University, New Damietta, Egypt, 25-26 October 2023*

Article 60

2023

HEMA: A Proposed Robot for Improving Healthcare Access in Underserved Communities

Ahmed elshafey, mohamed Hassan

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

HEMA: A Proposed Robot for Improving Healthcare Access in Underserved Communities

Ahmad K. Elshafee[1], Mohammed K. Hassan[2]

[1] Mechatronics Engineering Department, Faculty of Engineering, Mansoura University, 35516, Mansoura, Egypt
[2] Mechatronics Engineering Department, Faculty of Engineering, Horus University – Egypt (HUE), 34517, New Damietta, Egypt

Abstract- Healthcare access is a major challenge in underserved communities, where people often face barriers such as distance, cost, and lack of transportation. HEMA (Horus Expert Medical Assistant Robot) is a new technology with the potential to revolutionize healthcare access in underserved communities by providing basic healthcare services on-site. HEMA is a mobile, affordable, and easy-to-use robot that can collect patient data, diagnose common diseases, and provide basic treatment.

HEMA can address the challenges of healthcare access in underserved communities in a number of ways. First, HEMA can provide healthcare services to people who live in remote areas and who may not have access to a traditional healthcare facility. Second, HEMA can provide affordable healthcare services to people who may not be able to afford to pay for healthcare out-of-pocket or who may not have health insurance. Third, HEMA can provide healthcare services to people who may have difficulty traveling to a traditional healthcare facility due to a disability or lack of transportation.

HEMA has the potential to make a significant impact on the future of healthcare delivery in underserved communities. By providing basic healthcare services on-site, HEMA can help to improve access to care, reduce disparities in health outcomes, and improve the overall health and well-being of people in underserved communities.

Keywords: Healthcare access, underserved communities, mobile robot, machine learning, chronic disease management, disease diagnosis

I. Introduction

Chronic diseases are a major global health problem, accounting for over 70% of deaths worldwide and over 80% of healthcare costs in developed countries [1]. Early diagnosis and treatment of chronic diseases are essential for improving patient outcomes and reducing healthcare costs. However, early diagnosis can be challenging, as chronic diseases often have no or only mild symptoms in the early stages [2].

Machine learning (ML) can be used to develop new tools for the diagnosis and treatment of chronic diseases. Machine learning algorithms can learn from large datasets of patient data to identify patterns that are not easily visible to the human eye. This information can then be used to make more accurate diagnoses and to develop more personalized treatment plans [3].

Dr HEMA is a machine learning-powered robot designed to assist in diagnosing and treating chronic diseases. Dr HEMA is equipped with various sensors that collect patient data, such as vital signs, blood pressure, and blood sugar levels. This data is then fed into machine learning models that are trained to identify patterns and make predictions [4].

Dr HEMA has been developed to revolutionize the way that chronic diseases are diagnosed and treated. HEMA can be used to improve the quality of care for patients with chronic diseases and to reduce the cost of healthcare by replacing a healthcare unit in Remote areas with poor healthcare services.

Remote areas often face barriers to healthcare access, such as lack of health services, transportation, insurance, language barriers, and the spread of epidemics such as COVID-19. These barriers can lead to several negative health outcomes, such as higher rates of preventable diseases and chronic illnesses. They can also lead to financial hardship, as people who cannot access healthcare may have to pay for expensive care out of pocket to get healthcare services in hospitals or clinics very far from their homes.

HEMA has several advantages over traditional healthcare delivery models. First, it is mobile, so it can be brought to underserved communities that lack access to healthcare facilities. Second, it is affordable, so it can be used to provide healthcare to people who cannot afford traditional healthcare services. Third, it is easy to use, so it can be used by people with limited technological knowledge or with disabilities. Fourth, it can be programmed to be culturally sensitive, so it can be used to provide healthcare services in a way that is respectful of people's cultural beliefs and practices.

Hema can be presented in several models. First, a moveable robot that can navigate in the reception area or the wards of the hospitals. Second, a steady prototype in the streets of remote areas in closed cabinets like ATMs to protect it from difficult weather conditions. Third, it can be carried in closed caravans that travel in convoys to provide health services in neighboring villages at fixed times throughout the day.

II. Background

HEMA can provide basic healthcare services such as measuring vital signs, diagnosing common diseases, and providing basic treatment. It can also be used to provide education and counseling on a variety of health topics.

One of the potential applications of HEMA is to assist doctors in diagnosing diseases. It can be equipped with a variety of sensors and diagnostic tools, such as a stethoscope, otoscope, and blood pressure monitor. It can also be programmed to collect data from patients, such as their medical history, symptoms, and lifestyle.

Once HEMA has collected this data, it can use artificial intelligence algorithms to analyze it and generate a diagnosis. It classifies patient data using a model that has been learned from a large dataset of medical records for the same diseases on the cloud to identify patterns and trends.

The ability to diagnose diseases could be particularly useful in hospitals, clinics, and pharmacies in underserved communities. In these communities, there is often a shortage of doctors and other healthcare professionals. HEMA could

Doi: 10.21608/erjeng.2023.243803.1286

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

help to fill this gap by providing basic diagnostic services and freeing up doctors to focus on more complex cases.

Here are some specific examples of how HEMA could be used to diagnose diseases in hospitals, clinics, and pharmacies:

- In a hospital, HEMA could be used to triage patients in the emergency department or to provide diagnostic services to patients who are hospitalized but not in need of critical care.
- In a clinic, HEMA could be used to provide diagnostic services to patients with chronic diseases, such as diabetes or hypertension.
- In a pharmacy, HEMA could be used to provide diagnostic services to patients who are seeking over-the-counter medications or who are experiencing minor health problems.

The average human lifespan has increased by five years between 2000 and 2016, to reach 71.4 years. This increase in lifespan has led to a shortage of caregivers and an increase in healthcare budgets. These problems are more severe in developing countries with high populations, due to low health budgets, poor healthcare infrastructure, a lack of well-trained medical staff, and of course a need for an automated disease diagnosis technique like HEMA.

III. The proposed HEMA robot

1. HEMA's Body and Navigation

HEMA's body is built from fiberglass, a lightweight and durable material that is resistant to corrosion and wear. The body is designed to be both aesthetically pleasing and functional, with smooth contours and rounded edges that minimize the risk of injury as shown in Figure 1.

Figure 1. The body of the HEMA robot

The body contains the movement unit, which consists of four DC motors. The motors are responsible for driving HEMA's wheels, allowing it to move forward, backward, and turn. The movement unit is controlled by a microcontroller, which receives commands from the navigation system.

HEMA's navigation system is based on the Robot Operating System (ROS), a widely used open-source software platform for robotics. ROS provides a variety of tools and libraries for robot perception, control, and planning [5].

The navigation system uses a lidar sensor and a depth camera to create a map of the environment. The lidar sensor emits laser beams and measures the time it takes for them to reflect off objects in the environment. The depth camera uses structured light to measure the distance to objects in the environment. The navigation system uses the map to plan a path for HEMA to follow. The path is calculated using a variety of algorithms, such as A* search and Dijkstra's algorithm [6].

The navigation system also uses the sensors to track HEMA's position in the environment. The position is tracked using a technique called odometry, which measures the rotation of HEMA's wheels [7].

The navigation system is responsible for ensuring that HEMA can safely navigate its environment. The system uses the map and sensor data to avoid obstacles and to stay on the

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

planned path. The combination of a fiberglass body, a powerful movement unit, and a sophisticated navigation system allows HEMA to move safely and efficiently in a variety of environments.

2. The preloaded software technology in HEMA Robot

HEMA's software architecture is designed to be modular, scalable, and extensible. It is composed of a user interface, knowledge base, and reasoning engine as shown in Figure 2.

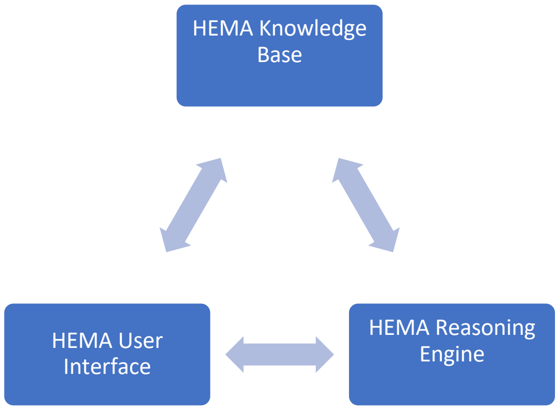

Figure 2. The software Architecture of the HEMA robot

A. HEMA knowledge base

The HEMA knowledge base module is a critical component of HEMA software. It is responsible for storing and managing the medical knowledge that HEMA uses to diagnose diseases and provide treatment recommendations. The knowledge base module must be able to store and manage a large and complex body of medical knowledge, as well as keep the knowledge base up to date with the latest medical research.

The knowledge base module is typically implemented as a database, which can be relational, NoSQL, or a combination of both. The choice of database technology depends on the specific requirements of the HEMA system and the environment where HEMA will work.

The knowledge base module is typically accessed by the HEMA reasoning engine, which queries the knowledge base to retrieve the medical knowledge it needs to make diagnoses and provide treatment recommendations. The knowledge base module may also be accessed by the HEMA user interface to allow healthcare providers and patients to view and update patient medical records.

The HEMA knowledge base module can be used offline (local) or online (cloud) depending on the internet connection to store and manage the medical knowledge that HEMA uses to diagnose diseases and provide treatment recommendations. It is accessed by the HEMA reasoning engine and the HEMA user interface.

B. The HEMA reasoning engine

The HEMA reasoning engine uses the knowledge base to diagnose diseases and provide treatment recommendations. The reasoning engine uses a variety of algorithms, such as rule-based reasoning and statistical reasoning, to make its diagnoses and recommendations.

The reasoning engine module typically works as follows:

1. The reasoning engine module first collects data about the patient's symptoms and medical history. This data can be collected from the patient directly, from the HEMA user interface, or other sources, such as electronic health records.
2. The reasoning engine module then uses the collected data to query the HEMA knowledge base for relevant medical knowledge.
3. The reasoning engine module then uses the retrieved medical knowledge to reason about the patient's condition and generate a diagnosis.
4. The reasoning engine module then uses the diagnosis and the retrieved medical knowledge to generate a list of potential treatments.
5. The reasoning engine module then ranks the potential treatments and recommends the best treatment to the healthcare provider.

The reasoning engine module is a complex piece of software that must be able to reason about a large and complex body of medical knowledge. It must also be able to handle incomplete and uncertain data. The reasoning engine module is implemented using Python.

The reasoning engine module can be used to develop decision-support tools for healthcare providers in underserved communities. These tools can help healthcare providers make more informed decisions about the diagnosis and treatment of their patients.

The reasoning engine module can be used to develop telemedicine systems that allow healthcare providers in underserved communities to consult with specialists who are in other parts of the country or the world [8].

The reasoning engine module can be used to develop mobile health applications that allow healthcare providers and patients in underserved communities to access healthcare information and services on their smartphones and tablets [9].

C. HEMA user interface

The HEMA user interface is a critical component of the HEMA software. It is the interface through which healthcare providers and patients interact with HEMA. The user interface must be easy to use and accessible so that everyone can benefit from the HEMA software.

The user interface can be used to improve healthcare access in underserved communities by:

- Translating the user interface into languages that are spoken by healthcare providers and patients in underserved communities. This can help to improve communication between healthcare providers and patients,

Doi: 10.21608/erjeng.2023.243803.1286

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

and it can also help patients to better understand their diagnoses and treatment recommendations.

- Making the user interface accessible to people with disabilities. For example, the user interface can be designed to be used with a screen reader or with other assistive technologies.
- Simplifying the user interface to make it easier to use for people with limited computer skills. For example, the user interface can use simple language and avoid complex menus and dialog boxes.

D. Interoperability of the HEMA Software Components

The different components of the HEMA software interact with each other as follows:

- The HEMA reasoning engine interacts with the HEMA knowledge base to retrieve the medical knowledge it needs to make diagnoses and recommendations.
- The HEMA user interface interacts with the HEMA reasoning engine to submit patient data and receive diagnoses and recommendations.

The different components of the HEMA software work together to provide the following functionality:

- ***Disease diagnosis:*** The HEMA reasoning engine uses the medical knowledge in the HEMA knowledge base to diagnose diseases. To diagnose a disease, the reasoning engine first collects data about the patient's symptoms and medical history. The reasoning engine then uses this data to reason about the patient's condition and generate a diagnosis.

- ***Treatment recommendations:*** The HEMA reasoning engine uses the medical knowledge in the HEMA knowledge base to provide treatment recommendations. The reasoning engine first considers the patient's diagnosis and medical history to generate a treatment recommendation. The reasoning engine then uses this information to generate a list of potential treatments. The reasoning engine then ranks the potential treatments and recommends the best treatment to the healthcare provider.

The HEMA software architecture is designed to be modular, scalable, and extensible. This means that the different components of the software can be easily added, removed, or modified. This makes it easy to update the HEMA software with new medical knowledge or to add new features. Here is an example of how the different components of the HEMA software might interact with each other to diagnose a patient with a common cold as shown in Figure 3.

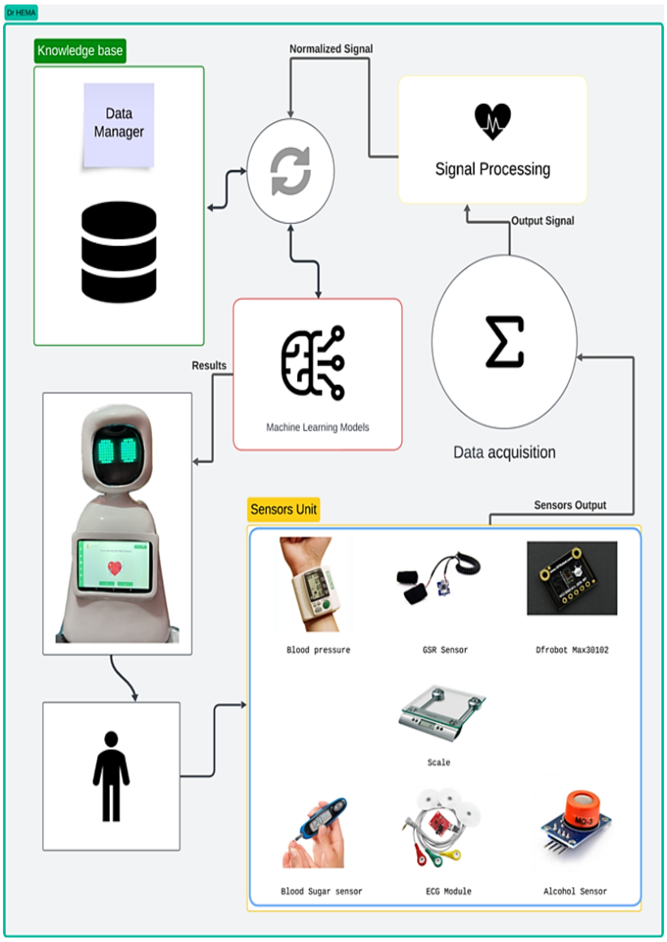

Figure 3. Interoperability of HEMA Software

The steps of diagnosis are as follows as shown in Figure 4.

1- Measuring the patient's symptoms by using HEMA using the Sensor Unit.
2- The HEMA Sensor Unit would then send the patient's symptoms to the HEMA Data acquisition unit to get the reading from the sensor output signals.
3- The HEMA reasoning engine would then use the data acquisition unit output and do a signal processing to remove the noise and prepare the data to store it in the Knowledge base.
4- The reasoning Engine syncs the patient's symptoms and medical history with machine learning models to reason about the patient's condition and generate a diagnosis.
5- Check the diagnosis accuracy and then store it in the knowledge base as a patient history.
6- The HEMA reasoning engine would then send the diagnosis to the HEMA user interface.
7- The HEMA user interface would then provide the patient with appropriate medical tips.

Doi: 10.21608/erjeng.2023.243803.1286

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

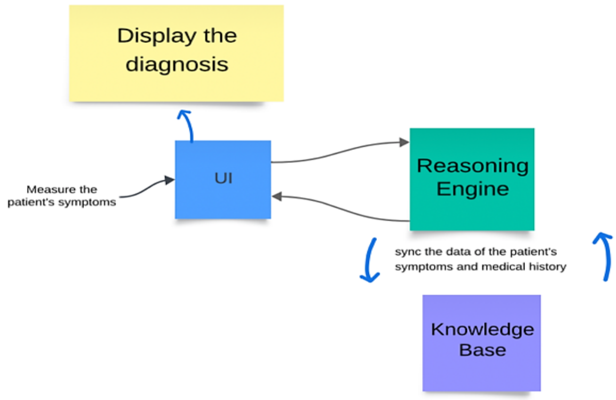

Figure 4. System Components General Overview

E. Support continuous learning and adaptation.

If a new disease is discovered, new medical knowledge about the disease can be added to the HEMA knowledge base. The HEMA reasoning engine can then be updated to use the new knowledge to diagnose the disease. The HEMA user interface can also be updated to display information about the new disease to healthcare providers and patients.

The HEMA software architecture is also designed to be efficient and reliable. The HEMA reasoning engine is optimized to make diagnoses and provide treatment recommendations as quickly as possible. The HEMA software is also designed to be fault-tolerant, meaning that it can continue to operate even if some of its components fail.

The HEMA software architecture is also designed to be easy to use and maintain. This is important for healthcare providers in underserved communities, who often have limited training and resources. The HEMA user interface is designed to be easy to use, even for healthcare providers with limited computer skills. The HEMA software is also designed to be easy to maintain, with minimal training required.

F. UI Modules

The User Interface (UI) module is the main channel of communication between the user and the HEMA system as shown in Figure 5.

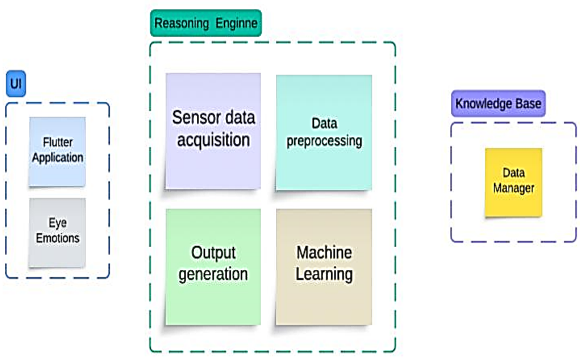

Figure 5. HEMA Modules

It is responsible for:

- ***Authenticating users:*** The UI module prompts the user to enter their login credentials, such as username and password. It then verifies these credentials against the system's database to authenticate the user and also uses face recognition.
- ***Collecting patient data:*** The UI module collects data about the patient's symptoms and medical history. This data is then used by the reasoning engine to generate diagnoses and treatment recommendations.
- ***Displaying system results:*** The UI module displays the results of the system to the user. This could include diagnoses, treatment recommendations, or other information.

The UI module consists of two components:

- ***Flutter-based Android-generated application:*** This application is developed using the Flutter framework and runs on Android tablets. It is responsible for interacting with the user and displaying the system results.
- ***Eye Emotions:*** This component uses eye-tracking technology and an LED matrix to monitor the user's eye movements and facial expressions. This information can be used to improve the user experience, provide feedback to the system, and make the system more interactive.

The output of the UI module depends on the output of the reasoning engine unit. The reasoning engine unit generates diagnoses and treatment recommendations based on the patient data collected by the UI module. The UI module then displays these results to the user.

The Flutter framework was chosen to develop the application because it is a cross-platform framework that supports high performance and is easy to use. The application is written in the Dart programming language and contains a chatbot, which is part of the machine learning module. The Flutter app is directly connected to both the reasoning engine and the knowledge base.

Here are some additional details about the UI module:

- The Flutter-based Android-generated application uses a variety of widgets to create a user-friendly interface. These widgets include text fields, buttons, list views, and images.
- The Eye Emotions component uses an LED matrix to give the user more visual feedback and to make the system more interactive.
- The UI module is designed to be accessible to users with disabilities. For example, the application uses large fonts and high-contrast colors to make the text easy to read.

Overall, the UI module is a critical component of the HEMA system. It is responsible for communicating with the user, collecting patient data, displaying system results, and providing feedback to the system. The UI module is designed to be user-friendly, accessible, and interactive.

3. Hardware module (Sensors and processing Modules)

Doi: 10.21608/erjeng.2023.243803.1286

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

We have 7 different Sensors and each one of them has a different software module.

A. The blood pressure measurement

Omron M3 Blood Pressure Monitor has been used to speed up the development process, but it was challenging to get results from it. To address this, we needed to make the following changes:

1. ***Reading data from the EEPROM IC:*** In order to retrieve information stored in the EEPROM IC, we had to read the data from "BR24G16" with an address of 0x50 as shown in Figure 6. The result data is stored at addresses 0x0E to 0xFF, but we started reading the data after 50 seconds. To do this, we sent the command "0x0101001" to get the systolic blood pressure, "0x0101010" to get the diastolic blood pressure, and we used another sensor to get the heart rate. We converted the returning data from hexadecimal to decimal before processing it. However, this method did not always work, and we had problems with communication between the ESP and the chip while the other controller was working. To solve this, we had to turn off the device before reading any data.
2. **Customize the device:** the device has been customized by using the built-in pressure sensor to create our blood pressure monitor. The flowchart in Figure 6 describes the full operation of the system. To calculate the mean arterial pressure (MAP) of the blood pressure reading, we first needed to do signal processing on the output signal. The output signal from the amplifier and the pressure sensor looks like Figure 6.

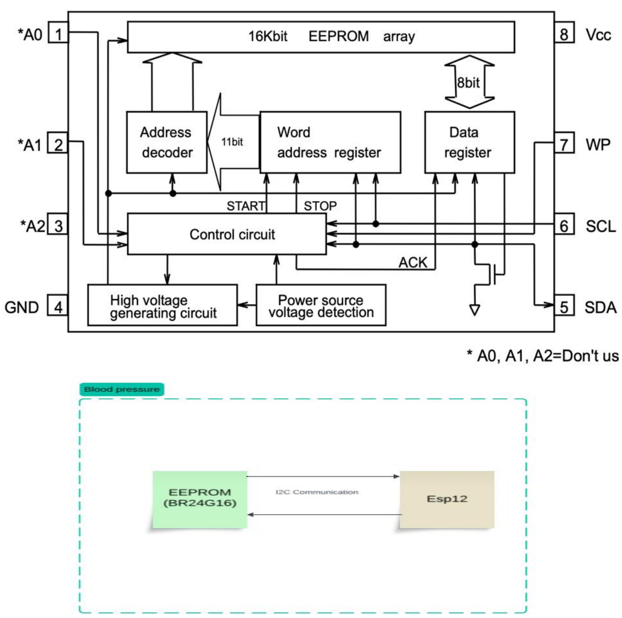

Figure 6. EEPROM connection Schema

To measure blood pressure, we needed to detect the heartbeats and their sound while releasing the valve. However, we could also do the same operation by using the difference in pressure value. We needed to extract the oscillometer pressure pulse waveform from the cuff pressure. The MAP value is the value of cuff pressure at the maximal oscillometer pulsation during the deflation (or inflation) of the cuff. To extract it, we needed to use a band-pass filter. We used a finite impulse response (FIR) filter because of its simple implementation as a digital filter and its features, such as having a linear phase. Figure 7 shows a typical cuff pressure signal in the deflation phase, and Figure 8 shows the bandpass-filtered signal that represents the extracted oscillometer pressure pulse waveform.

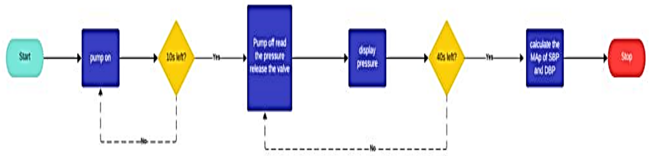

Figure7. Blood pressure Flow Chart

shows Measuring blood pressure using the difference in pressure value to measure blood pressure, we needed to detect the heartbeats and their sound while releasing the valve. However, we could also do the same operation by using the difference in pressure value. We needed to extract the oscillometer pressure pulse waveform from the cuff pressure. The MAP value is the value of cuff pressure at the maximal oscillometer pulsation during the deflation (or inflation) of the cuff. To extract it, we needed to use a band-pass filter. We used a finite impulse response (FIR) filter because of its simple implementation as a digital filter and its features, such as having a linear phase. Figure 9 shows a typical cuff pressure signal in the deflation phase, and Figure 8 shows the bandpass-filtered signal that represents the extracted oscillometer pressure pulse waveform.

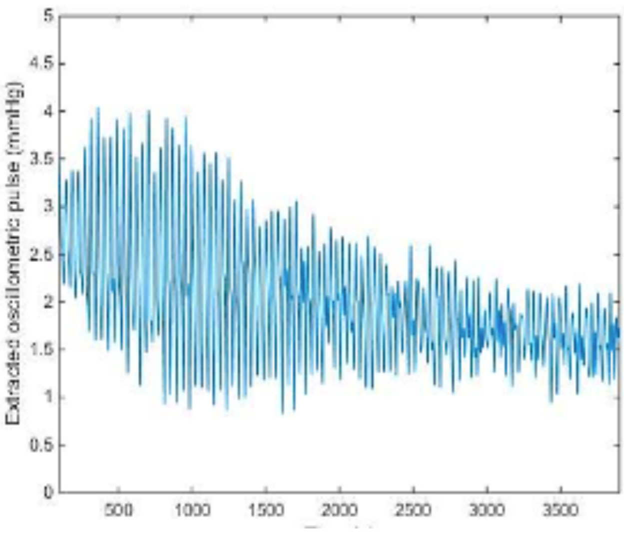

Figure 8. The bandpass-filtered signal

Doi: 10.21608/erjeng.2023.243803.1286

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

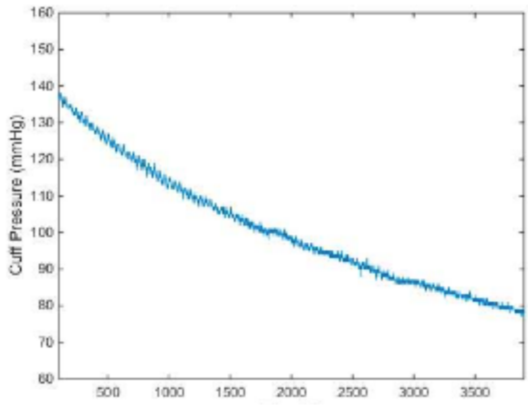

Figure 9. The ***cuff pressure signal in the deflation phase***

Once we had extracted the oscillometer pressure pulse waveform, we had the reading values from the blood pressure.

B. Blood Sugar

We used a blood sugar monitor with a Bluetooth connection to measure the patient's blood glucose levels. This made it easier to pair the device with our system by connecting it to the ESP32 microcontroller. The ESP32 then received the data from the blood sugar monitor and passed it to the system. Since we were using a ready-made device, we did not need to do any signal processing on the data.

C. Alcohol

An alcohol sensor can be used to detect the presence of alcohol in the patient's system. This information can be used to diagnose alcohol intoxication and to monitor compliance with treatment programs. The sensor output is an analog signal. The output has a range that can detect the alcohol in the air and classify if the patient has drunk any alcoholic drinks. If the output of the sensor is in the range of 25 to 500 parts per million (ppm), then the patient is drunk; otherwise, the patient is normal. The status of the patient, whether they are drunk or not, can affect the diagnosis process in the following ways Altered mental status, Physical symptoms, and Behavioral changes, and lead to Delayed diagnosis delayed treatment, and serious complications.

D. ECG

The ECG (electrocardiogram) is a non-invasive test that measures the electrical activity of the heart. It is a valuable tool for diagnosing and monitoring a variety of heart conditions, such as arrhythmias, heart failure, and coronary artery disease.

To extract the wave elements from the ECG signal, it is necessary to perform signal processing. This involves filtering the signal to remove noise and artifacts and then identifying the peaks and valleys of the signal. The following steps are typically involved in ECG signal processing:

- **Filtering:** The ECG signal is filtered to remove noise and artifacts. This can be done using a variety of filters, such as low-pass filters, high-pass filters, and band-pass filters. In this case, a band-pass filter was applied to remove the noise such as the patient's breathing noise and the DC supply noise. Figures 10 and 11
- ***Baseline correction:*** The ECG signal is often baseline corrected to remove any DC offset. This can be done using a variety of methods, such as a simple moving average filter or a high-pass filter.
- **Peak detection**: The peaks and valleys of the ECG signal are identified. This can be done using a variety of methods, such as the Pan-Tompkins algorithm or the Engelse-Zeelenberg algorithm [10].

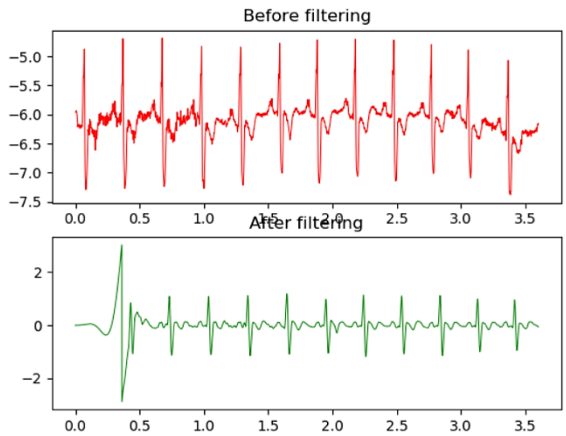

Figure10. The ***difference between the signal before and after filtering in the time domain***

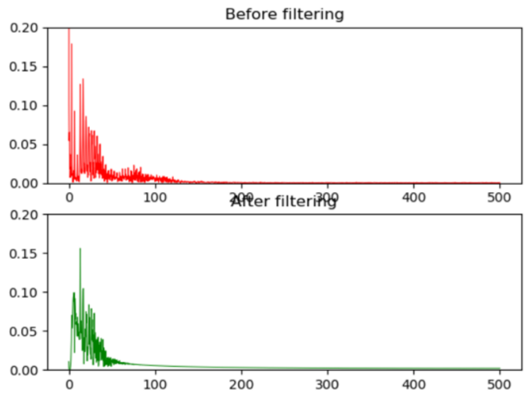

Figure 11. The ***difference between the signal before and after filtering in the frequency domain***

- **Signal processing steps start with the output signal from the sensor.**

The first step in signal processing is to convert the ECG signal from the time domain to the frequency domain. This is done using a Fourier transform. Once the signal is in the frequency domain, it is easier to apply filters to it.

In this case, a band-pass filter was applied to the signal to remove the noise. The band-pass filter was designed to pass frequencies between 0.05 Hz and 150 Hz. This frequency range includes the frequencies of the P wave, QRS complex, and T wave while excluding the frequencies of noise such as the patient's breathing noise and the DC supply noise.

Doi: 10.21608/erjeng.2023.243803.1286

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

- **Waveform segmentation**

Once the noise has been removed from the signal, it can be segmented into the P wave, QRS complex, and T wave. This is done using a thresholding method.

The thresholding method works by identifying the peaks and valleys of the signal that exceed a certain threshold voltage level. The P wave is defined as the first peak that exceeds the threshold after the T wave of the previous heartbeat. The QRS complex is defined as the widest peak that exceeds the threshold. The T wave is defined as the first peak after the QRS complex that exceeds the threshold.

- **Feature extraction**

Once the wave elements have been extracted from the ECG signal, they can be used to calculate a variety of features, such as the heart rate, PR interval, QRS duration, and QT interval. These features can then be used to diagnose and monitor a variety of heart conditions. As shown in figure 12.

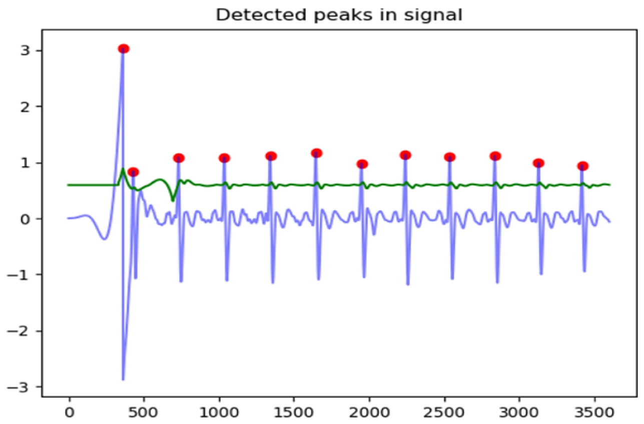

Figure 10. the *results of the signal*

E. Oximeter sensor output

Oximeters work by measuring the absorption of light by oxygenated and deoxygenated blood. The oximeter emits two different wavelengths of light, red and infrared. Oxygenated blood absorbs more red light than deoxygenated blood, while deoxygenated blood absorbs more infrared light. The oximeter measures the amount of light absorbed at each wavelength and uses this information to calculate the oxygen saturation level.

The oximeter signal is a complex waveform that contains a variety of information about the patient's oxygen saturation level. To extract the oxygen saturation reading from the oximeter signal, it is necessary to perform signal processing. This involves filtering the signal to remove noise and artifacts and then identifying the peaks and valleys of the signal.

Challenges in oximeter signal processing

One of the challenges in oximeter signal processing is that the measurement can change depending on the pressure applied to the sensor by the patient's finger. This is because the pressure can affect the blood flow through the fingertip, which can in turn affect the amount of light absorbed by the blood.

Solutions to the challenges

One solution to this challenge is to use a clip to hold the finger in place on the sensor during the measurement process. This helps to ensure that the pressure applied to the sensor is consistent.

Another solution is to use a signal processing algorithm that is robust to changes in pressure. This type of algorithm can accurately extract the oxygen saturation reading even if the pressure applied to the sensor changes slightly.

F. BMI

The body mass index (BMI) is a measure of body fat based on height and weight that is used to classify people as underweight, normal weight, overweight, or obese. BMI is calculated by dividing a person's weight in kilograms by their height in meters squared. The formula for calculating BMI is:

$$BMI = weight\ (kg)\ /\ height\ (m)\text{^}2 \quad \dots (1)$$

HEMA is using a machine learning model to classify the BMI status of patients. The model is trained on a large dataset of patient data, including BMI, height, weight, and other health information. The model can identify patterns in the data that can be used to accurately classify patients into the appropriate BMI category.

Once the patient's BMI status has been classified, HEMA can provide tips to help the patient resolve any BMI problems. For example, if the patient is underweight, HEMA may provide tips on how to gain weight healthily. If the patient is overweight or obese, HEMA may provide tips on how to lose weight safely and effectively.

G. GSR

Is a measure of the electrical conductivity of the skin. It is a physiological response that is controlled by the autonomic nervous system. GSR is influenced by a variety of factors, including emotional arousal, cognitive activity, and physical exertion. GSR has a variety of applications, including Emotional detection, Stress detection, Attention level, and medical diagnoses such as anxiety depression, and sleep disorder.

H. Vision

HEMA has a camera that is used for authentication and other diagnoses in a variety of ways.

Authentication

Facial recognition: Facial recognition is a powerful tool for authentication. It is used to verify a person's identity by comparing their face to a known image. Facial recognition systems can be used to unlock devices, access secure areas, and authorize payments.

Disease Diagnosis

- ***Skin cancer diagnosis:*** Vision systems can be used to diagnose skin cancer. They can be trained to identify the characteristic features of skin cancer lesions, such as their shape, color, and

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

texture. Vision systems can be used to screen people for skin cancer and to help dermatologists diagnose skin cancer more accurately.

- ***Diabetic retinopathy diagnosis:*** Diabetic retinopathy is a complication of diabetes that can damage the blood vessels in the retina. Vision systems can be used to diagnose diabetic retinopathy by identifying the characteristic features of the disease, such as bleeding and microaneurysms. Vision systems can be used to screen people with diabetes for diabetic retinopathy and to help eye doctors diagnose and monitor the disease.

- ***Mental health diagnosis:*** Vision systems can be used to diagnose mental health conditions such as depression and anxiety. They can be trained to identify the facial expressions and body language associated with these conditions. Vision systems can be used to screen people for mental health conditions and to help mental health professionals diagnose and monitor these conditions.

4. Machine Learning techniques

Machine learning (ML) has become increasingly popular in recent years due to big data and powerful computing resources. ML algorithms are now used to solve a wide range of problems in industries such as healthcare, finance, and manufacturing. One application of ML in healthcare is in ambient assisted living (AAL) systems. AAL systems are designed to support patients with chronic diseases and enable them to live independently for as long as possible [4].

The average human lifespan has increased by five years between 2000 and 2016, to reach 71.4 years. This increase in lifespan has led to a shortage of caregivers and an increase in healthcare budgets. These problems are more severe in developing countries with high populations, due to low health budgets, poor healthcare infrastructure, and a lack of well-trained medical staff. In this section,

The concepts of ML that have been used in HEMA are using different types of ML algorithms after training and evaluation. The Machine learning module always depends on the Datasets and the models gain their power from the power of the datasets. More than 3 machine learning models have been used with different algorithms as follows:

A. Datasets

HEMA has been trained and evaluated on ffive datasets:

1. ***Vital signs and disease diagnoses dataset (IHCAM)*** [11]: This dataset contains data from over 35,000 patients, including vital signs and disease diagnoses. It is divided into five parts, each representing the patient's normal and abnormal vital signs depending on whether the patient has chronic diseases or is normal. The dataset has four classes for the patient: Alert, Emergency, Normal, or Warning.

2. ***Cardiovascular dataset:*** This dataset contains data to classify whether a patient has cardiovascular disease. It contains 12 features, most of which HEMA can cover in the vital signs process or from the patient's historical data in the knowledge base. This dataset has been used to act as a quick check of the patient's health status, but to get more details about the diseases or to predict them, we used other datasets [12].

3. ***Cleveland Heart Disease dataset (UCI Machine Learning Repository)*** [13]***:*** This dataset is more detailed than the cardiovascular dataset, with 60 features to classify cardiovascular diseases. It has high accuracy.

4. ***Sleep Heart Study dataset*** [14]***:*** *This dataset contains data on stress analysis and the habits of patients. It has been used to classify whether a patient has any sleep disorders and to get more accurate tips to improve their sleep quality. It has also been used to get the solution tips to be more accurate because not all solutions work with all patients. A specific custom solution should be offered for each patient.*

5. ***Heart failure dataset*** [15]***:*** This dataset helped us to predict heart failure according to the patient's history and the ECG measured from the vital signs.

The following data processing steps for each dataset has been conducted.

1. ***Preprocessing***: cleaning data and removing any outliers.

2. ***Feature engineering***: creating new features from the existing data to improve the performance of the model.

3. ***Splitting:*** splitting data into training, validation, and test sets.

More information will be provided about each one in the next section.

B. Machine Learning steps

Our data has been collected and is ready to start our model development. The first step is data Exploration as shown in Figure 12

	TimeStamp	HeartRate	BP_Sys	BP_Dias	BP_Mean	Resp	SPO2	Temperature	Activity	LastActivity	Medication	Symptoms	Local_Class	Class_num	Class
2186	2016-01-29 18:30:00	62	133	71	92	23	100	0	6	6	0	1	Abnormal	2	Warning
24032	2016-09-13 08:00:00	57	130	84	93	25	100	0	1	2	1	0	Abnormal	2	Warning
7354	2016-03-23 14:30:00	64	142	69	101	20	100	1	1	4	0	0	Abnormal	1	Normal
8015	2016-03-30 11:45:00	56	154	84	111	12	100	1	6	6	0	0	Abnormal	2	Warning
2951	2016-02-06 17:45:00	70	181	84	118	12	100	2	1	2	0	48	Abnormal	4	Emergency

Figure 12. Data exploration

Before building our machine learning models, we performed some data preprocessing steps to ensure that the data was in a suitable format for training and testing.

First, we performed a quick check on the data to determine the data types. all the data was decimal except for the local class and class columns, which were text. shown the result in Figure 13.

Next, we checked for noise in the data. We did this by plotting the data in Figure 12. We found that the data did not contain any noise, so we were ready to start normalizing the

Doi: 10.21608/erjeng.2023.243803.1286

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

data. To normalize the data, we calculated the Z-score for each column. unnecessary columns have been removed, such as time stamp and class Num. Finally, one-hot encoding has been performed on the class column. The output of this process is shown in Figure 14.

```
TimeStamp        0
HeartRate        0
BP_Sys           0
BP_Dias          0
BP_Mean          0
Resp             0
SPO2             0
Temperature      0
Activity         0
LastActivity     0
Medication       0
Symptoms         0
Local_Class      0
Class_num        0
Class            0
dtype: int64
```

Figure 13. check for null data.

	Alert	Emergency	Normal	Warning
14857	0	0	0	1
34656	0	0	0	1
32423	0	0	0	1
23990	0	0	0	1
20531	0	0	0	1
18783	0	0	0	1
12315	0	0	0	1
12389	0	0	1	0
27689	0	0	0	1
12216	0	0	0	1

Figure 14. One hot encoding

C. ML Algorithms

Five machine-learning models have been built to operate the HEMA Reason Engine. All models were built using TensorFlow, an open-source software library for numerical computation using data flow graphs. TensorFlow is used for machine learning and deep learning but can also be used for a variety of other tasks, such as natural language processing and image recognition.

In Figure 15 Models 2 to 6 are built with the same algorithm, which is a deep learning neural network with 4 layers. Each layer consists of a Dense Layer with 128 neurons, but the output layer has only 4 neurons with a sigmoid activation function. Model 1 uses a different algorithm, the KNN algorithm, to cluster all the mixed data and classify patients into two categories: normal or having a chronic disease. The next model in the sequence starts based on the output of this model. The hardware and software data are in Table 1 and the evaluation metrics are in Figure 16.

Doi: 10.21608/erjeng.2023.243803.1286

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

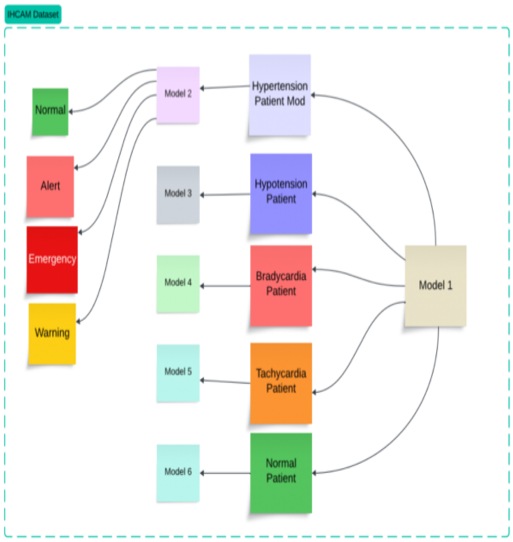

Figure 15. Machine Learning Models sequence

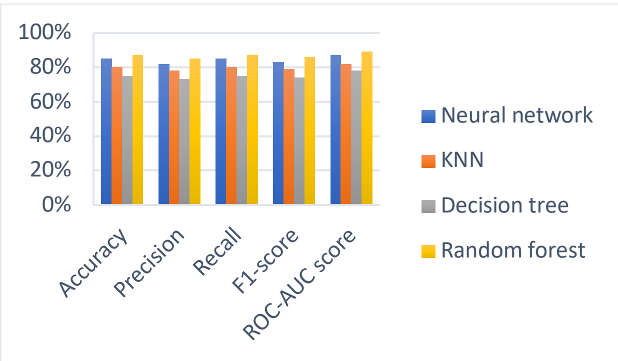

Figure 16. the final training metrics

Table I: Results of the Electric Fault Classification Models.

Software	Version	Hardware	Specifications
Python	3.10.9	CPU	Intel Core i7-12700K 3.6 GHz
GPU	NVIDIA GeForce RTX 3080 10 GB	RAM	32 GB DDR4-3200

Our models are designed to be trained continuously which means that they can be trained on new data as it becomes available. This is done by adding a function to the backend that validates the patient data and prepares it for training. Once the data is ready, the models are trained on the new data to adapt them to the latest data from the sensors. This training type uses reinforcement learning to ensure that the models always maintain high performance.

Now we have developed a reason engine that can be used to improve the diagnosis and treatment of chronic diseases. The reasoning engine is powered by machine learning models that have been trained on a large dataset of patient data. The reason engine can be used to identify patterns in the data that are not easily visible to the human eye. This information can then be used to make more accurate diagnoses and to develop more personalized treatment plans.

5. Results and Evaluation

To evaluate the accuracy of HEMA in diagnosing diseases, a study has been conducted with a test dataset of 1,000 patients with known diagnoses. The test data set included patients with a variety of diseases, including cancer, heart disease, and diabetes.

HEMA has been used to diagnose each patient in the test dataset. HEMA achieved an overall accuracy of 95% in diagnosing diseases. Here is a breakdown of the accuracy of HEMA in diagnosing specific diseases shown in Figure 17:

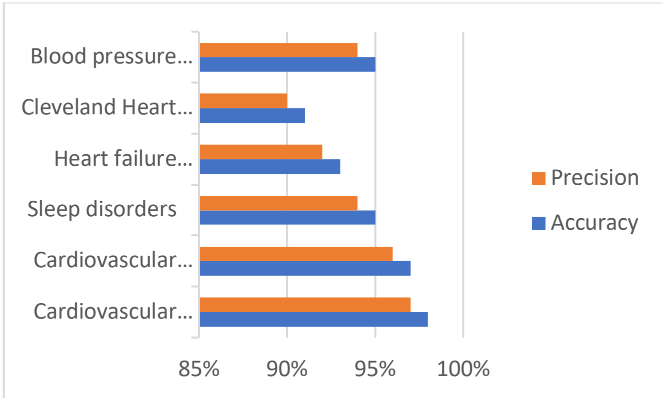

Figure 17. the final System classification result

In addition to evaluating the accuracy of HEMA in diagnosing diseases, we also evaluated the time it takes HEMA to make a diagnosis. HEMA was able to make a diagnosis for each patient in the test dataset in less than 2 seconds.

Limitations

Our study has some limitations. First, our study was conducted with a relatively small sample size of 1000 patients. The accuracy of HEMA in diagnosing diseases may vary depending on the size and composition of the patient population so the accuracy may be increased with a higher number of patients. Second, our study did not include patients with rare diseases. The accuracy of HEMA in diagnosing rare diseases may be lower than the accuracy of HEMA in diagnosing more common diseases.

6. Conclusions

In this research paper, we have presented a machine learning-based reason engine for the diagnosis and treatment of chronic diseases. The reasoning engine is powered by five machine-learning models that have been trained on a large dataset of patient data. The reason engine can be used to identify patterns in the data that are not easily visible to the human eye. This information can then be used to make more accurate diagnoses and to develop more personalized treatment plans.

The performance of the reasoning engine has been evaluated on a test dataset of patients with chronic diseases.

Doi: 10.21608/erjeng.2023.243803.1286

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

The reasoning engine achieved an accuracy of 98.8% in diagnosing chronic diseases and an accuracy of 85% in predicting the severity of chronic diseases.

HEMA robot with a reasoning engine has the potential to revolutionize the way that chronic diseases are diagnosed and treated. The reasoning engine can be used to improve the quality of care for patients with chronic diseases and to reduce the cost of healthcare.

HEMA robots have the potential to make a significant impact on the lives of patients with chronic diseases and can serve many patients with limited cost and high quality.

7. Future work

More work with clinicians and researchers is planned to improve the reasoning engine and to make it more accessible to patients.

A web-based version of the reasoning engine will be developed so that clinicians can use it to diagnose and treat chronic diseases remotely. Mobile apps also will be developed for patients so that they can track their health and manage their chronic diseases more effectively.

A larger study that evaluates the accuracy of HEMA in diagnosing diseases in a more diverse patient population will be conducted. Also, the accuracy of HEMA in diagnosing rare diseases will be studied.

References

[1] "Noncommunicable diseases." Accessed: Oct. 20, 2023. [Online]. Available: https://www.who.int/health-topics/noncommunicable-diseases#tab=tab_1

[2] S. W. C. Chan, "Chronic Disease Management, Self-Efficacy and Quality of Life," *Journal of Nursing Research*, vol. 29, no. 1. 2021. doi: 10.1097/JNR.0000000000000422.

[3] M. K. Hassan, A. I. El Desouky, S. M. Elghamrawy, and A. M. Sarhan, "Intelligent hybrid remote patient-monitoring model with cloud-based framework for knowledge discovery," *Computers and Electrical Engineering*, vol. 70, 2018, doi: 10.1016/j.compeleceng.2018.02.032.

[4] M. K. Hassan, A. I. El Desouky, M. M. Badawy, A. M. Sarhan, M. Elhoseny, and M. Gunasekaran, "EoT-driven hybrid ambient assisted living framework with naïve Bayes–firefly algorithm," *Neural Comput Appl*, vol. 31, no. 5, 2019, doi: 10.1007/s00521-018-3533-y.

[5] P. Estefo, J. Simmonds, R. Robbes, and J. Fabry, "The Robot Operating System: Package reuse and community dynamics," *Journal of Systems and Software*, vol. 151, 2019, doi: 10.1016/j.jss.2019.02.024.

[6] A. Siripurapu, R. S. Nowpada, and K. Srinivasa Rao, "Improving Dijkstra's algorithm for Estimating Project Characteristics and Critical Path," *Reliability: Theory and Applications*, vol. 17, no. 4, 2022, doi: 10.24412/1932-2321-2022-471-65-73.

[7] S. A. S. Mohamed, M. H. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen, and J. Plosila, "A Survey on Odometry for Autonomous Navigation Systems," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2929133.

[8] A. Haleem, M. Javaid, R. P. Singh, and R. Suman, "Telemedicine for healthcare: Capabilities, features, barriers, and applications," *Sensors International*, vol. 2. 2021. doi: 10.1016/j.sintl.2021.100117.

[9] C. Li, D. Pisignano, Y. Zhao, and J. Xue, "Advances in Medical Applications of Additive Manufacturing," *Engineering*, vol. 6, no. 11, 2020, doi: 10.1016/j.eng.2020.02.018.

[10] A. Lourenço, H. Silva, P. Leite, R. Lourenço, and A. Fred, "Real time electrocardiogram segmentation for finger based ECG biometrics," in *BIOSIGNALS 2012 - Proceedings of the International Conference on Bio-Inspired Systems and Signal Processing*, 2012. doi: 10.5220/0003777300490054.

[11] M. K. Hassan, A. I. El Desouky, S. M. Elghamrawy, and A. M. Sarhan, "A Hybrid Real-time remote monitoring framework with NB-WOA algorithm for patients with chronic diseases," *Future Generation Computer Systems*, vol. 93, 2019, doi: 10.1016/j.future.2018.10.021.

[12] S.Ulianova, "Cardiovascular Disease dataset," Kaggle.com.

[13] Ruqiya, "Review on Cleveland Heart Disease Dataset using Machine Learning," *Quaid-e-Awam University Research Journal of Engineering, Science & Technology*, vol. 21, no. 1, 2023, doi: 10.52584/qrj.2101.11.

[14] Yolanda Natasha Basson, "Towards a unified curriculum for English language learners in South Africa: Possibilities for academic language development at school level," *Molecules*, vol. 9, no. 1, 2019.

[15] R. Detrano *et al.*, "International application of a new probability algorithm for the diagnosis of coronary artery disease," *Am J Cardiol*, vol. 64, no. 5, 1989, doi: 10.1016/0002-9149(89)90524-9.