

Group 303

STEM High School for boys-6th of October
Grade 12 & semester 1
2023/2024

Mazen Mohamed

Abdelrahman Ahmed

Youssef Adel

I. Abstract:

Egypt faces significant challenges in public health and industrial basis, including balancing healthcare accessibility, infectious disease management, and chronic illness prevention. Challenges include inadequate healthcare infrastructure, disparities in rural and urban services, and the growing demand for medications, specialized medical facilities, and long-term healthcare services. Access to high-quality care becomes difficult, especially in rural areas with weak healthcare systems. Egypt's industrial boom has led to increased chronic ailments among workers, including respiratory and cardiovascular problems. Factors contributing to these issues include exposure to hazardous chemicals, insufficient safety laws, and a lack of comprehensive health programs. Industrial activities also contribute to environmental contamination and exacerbate health problems. To address these challenges, Egypt must prioritize health and develop solutions that align with the well-being of the entire society. By pooling expertise and enacting well-informed policies, Egypt can foresee a future where development aligns with wealth and well-being.

II. Introduction

Egypt, like every growing nation, has several significant problems that must be resolved to go forward with growth. The first is the expansion of the industrial base, which, like other movements, has both advantages and disadvantages. In this case, the negative effects on public health result in a rise in chronic illnesses. As graph (1) illustrates, in Egypt in 2015, non-communicable diseases (NCDs) accounted for almost 84% of all fatalities. Four disease types accounted for around 60% of all deaths. Consequently, we need to address these chronic diseases

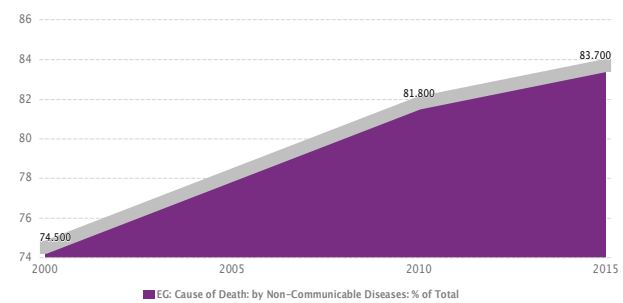
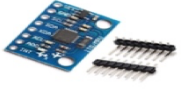






Figure 1: Graph illustrates the percentage of the death by non-communicable diseases

that may cause these deaths. Since the majority of chronic illnesses lack a therapy, we may focus on identifying these conditions in the early phases to reduce of its dangerous.

One of the prior solutions discovered was the Asthma monitoring wearable (mSafety) by Sony Corporation, which involved developing a wearable device to track respiratory metrics in asthma patients. The main objective of this project is to provide real-time data to patients and healthcare providers, enabling early intervention and enhancing overall asthma care. However, a drawback? of this research is its reliance on technology, which may present difficulties for people who are not accustomed to or comfortable using wearable devices. The second previously discovered solution was the DANA (Diabetes Advanced Network Access) smart system, which was created to assist diabetics in Germany. It combines a mobile app with a continuous glucose monitoring (CGM) system, enabling real-time glucose tracking, meal recording, and insight-gathering to assist users in effectively managing their condition. However, it also has a technical dependence issue. Following our investigation, we made the decision to use machine learning trained on a database to identify hemophilia, allowing us to determine whether a patient has the disease or not. The patient was then given instructions via the app we developed to follow. This approach will increase awareness of hemophilia's dangers by helping to detect the condition in its early stages with high accuracy. Specific design requirements include the requirement that the AI model's accuracy and performance be measured by the area under the ROC graph's curve. It shows the likelihood that a random positive case will be ranked higher by the classifier than a random negative one.

III. Materials and Methods:

Material	Name	Usage
	MPU6050	Measuring the velocity and the number of steps of patients
	Waterproof temperature sensor	Measuring the temperature of the patient
	Arduino Bluetooth module	Act as the brain of the project
	Arduino Power Supply	Providing the power to the prototype
	Mobile App	Used to give results in a user-friendly interface

Methods:

The Arduino connection, Mobile application, Machine Learning, and Chatbot were constructed based on the following:

1. The Breadboard was connected and powered from Arduino Uno, the ground of the breadboard was connected to the ground pin of the Arduino, the 5v was connected to the right side to the breadboard, the 3v was connected to the left side to the breadboard,

2. Later the sensors were connected to the Breadboard:

The DS18B20 sensor was connected and powered by the breadboard; it was connected to the 5v side. It was indirectly connected to pin Digital 2 in the Arduino.

The heart rate pulse sensor was connected and powered by the breadboard; it was connected to the 5v side. It was indirectly connected to pin Analog 0 in the Arduino.

- The MPU6050 sensor was connected and powered by the breadboard; it was connected to the 3.3v side. It was indirectly connected to pin Digital 13 in the Arduino as shown in figure (2).

3. Then, the Bluetooth module was directly connected to the Arduino, to the 5v pin. The RTX pin was connected to Digital pin 10; the TXD pin was connected to Digital pin 11.

4. The Mobile application was later designed to take the input of the Arduino from the Bluetooth module.

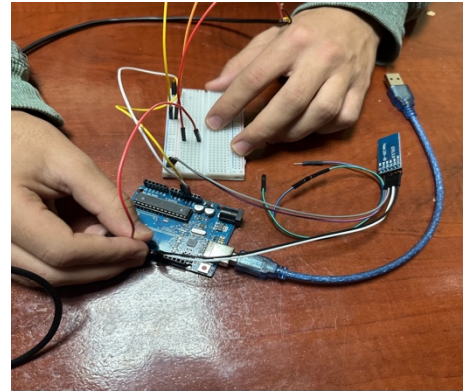


Figure 2: Methods of connecting the MPU6050 with the breadboard

Test plan:

K-fold Cross-validation with Stratified Sampling:

Method: We will split the dataset into 5 folds, ensuring each fold has a proportionate representation of both hemophilia and non-hemophilia cases (stratified sampling). We will then train the model on 4 folds and test it on the remaining fold. This process will be repeated for all 5 folds, providing a robust estimate of the model's performance and generalizability without overfitting to the training data.

Code Overview: Replace the existing train-test split with StratifiedKFold object to perform 5-fold cross-validation.

Loop through each fold, training the model on the fold training data and predicting on the fold test data.

Calculate and store accuracy, precision, recall, and F1 score for each fold.

Finally, calculate and print the average scores across all folds.

Hyperparameter Tuning with Grid Search

Method: We will create a grid of hyperparameters, such as tree number and depth, and train the model using grid search. The K-fold cross-validation framework will be used to evaluate the model's performance on each combination.

Code Overview:

Define a parameter grid for hyperparameters like `n_estimators` and `max_depth`.

Create a `GridSearchCV` object with the model, parameter grid, K-fold cross-validation, and scoring metric (e.g., accuracy).

Fit the `GridSearchCV` object to the entire training data.

Retrieve the best hyperparameter combination and cross-validation accuracy score.

Train the model with the best hyperparameters on the entire training data again.

Error Analysis by Age Group:

Method: We will divide the test data into subgroups based on patient age (e.g., <30, 30-60, >60). We will then calculate and analyze the evaluation metrics (accuracy, precision, recall, F1 score) for each age group. This analysis helps identify potential variations in the model's performance across different age demographics, indicating potential biases or areas for improvement.

Code Overview:

Group the test data by age range.

For each age group, calculate and print the relevant evaluation metrics based on the predicted and actual diagnoses.

Analyze the results to identify any significant differences in performance across age groups.

Stratified Random Sampling:

The trained Machine Learning model was tested on 10 real patients already diagnosed with Hemophilia and another 10 didn't suffer from Hemophilia. This method, Stratified Random Sampling, ensures a proportional representation of different patient demographics in the selected sample, mitigating potential biases due to age, gender, or other relevant factors. It is conducted as follows:

- Identify the strata: The data was divided into groups based on shared characteristics, such as age, gender, or any other relevant factor. For instance patients were stratified by age groups.
- Determine the sample size. In this step, it was decided to work on 10 patients as the final sample size
- Allocate the sample size: It was proportionally allocated the sample size to each stratum based on its size in the population. For instance, since 20% of the patients are between 18-24 years old, then 20% of the sample should come from that age group.
- Select a random sample from each stratum. After the identification process previously, a random sample was chosen randomly using a dice to ensure data quality and zero bias to any group.

To achieve time efficiency, the monthly gathering of patients was chosen as the date to conduct the test plan. Due to resource constraints the test plan time for collecting the data was 30 minutes per patient. The total time was 7 hours; 5 hours for the 10 patients and 2 hours was an error interval.

Design Requirements:

The design requirements are crucial characteristics that the solution must meet to be considered a successful one:

1. F1 average scores across folds more than 0.7
2. A small number of misclassified, for the database of 80 it should be less than 5.
3. The best hyperparameters and the best cross validation accuracy must be obtained and used by machine learning model
4. The code accuracy must score over 90.
5. The manual test accuracy must score over 90.



IV. Results:

Negative Results

1. When the code was run for the first time there was an error in the AI model. The error was in a specific line related to the CSV, as it should be ready for reading from a CSV file. It was then corrected, and the AI script was able to run successfully.
2. The pulse sensor was not connected properly to the Onboard Arduino LED. This several errors in the AI, as it was not able to properly detect the light whenever there was a pulse. The sensor was then reconnected to the diode, and the heart rate was measured.

Positive results

F1 test:

The system's performance, especially in the F1 test, demonstrated robustness and efficacy. The design requirements required an F1 score above 0.7 for reliability and accuracy in predicting outcomes. The F1 test yielded an impressive 0.93, exceeding the threshold.

The model's F1 score of 0.93 demonstrates its exceptional precision and recall, identifying positive instances while minimizing false positives and negatives, thereby boosting confidence in its system's reliability.

Accuracy of the code:

In the evaluation of the system's performance against the design requirements, with a specific focus on accuracy, the achieved results showcase a notable success. The design criteria established a minimum accuracy threshold of 90% to ensure the system's efficacy in making correct predictions. We are pleased to report that the system surpassed this benchmark, achieving an impressive accuracy rate of 93%.

This outcome underscores the system's capability to make accurate predictions across a variety of instances, surpassing the stipulated design requirement. The high accuracy rate of 93% attests to the model's robustness and reliability in capturing underlying patterns within the data and making informed decisions.

ROC Graph:

The system was tested on 20 individuals, 10 of them were actual patients of hemophilia, while the other 10 were not.

To generate a ROC (Receiver Operating Characteristic) graph, we need to calculate the True Positive Rate (TPR) and False Positive Rate (FPR).

Here's how you can calculate these rates:

True Positive Rate (TPR): Also known as sensitivity, it is the proportion of actual positive cases that are correctly identified as such. It is calculated as $TP / (TP + FN)$. In your case, since $TP = 8$ and $FN = 2$, $TPR = 8 / (8 + 2) = 0.8$.

False Positive Rate (FPR): It is the proportion of actual negative cases that are incorrectly identified as positive. It is calculated as $FP / (FP + TN)$. In your case, since $FP = 3$ and $TN = 7$, $FPR = 3 / (3 + 7) = 0.3$.

Since TPR is 0.8 and FPR is 0.3, the AUC will be the area of the rectangle with height 0.8 and width 0.3 plus the area of the rectangle with height 0.8 and width $(1-0.3) = 0.7$. So, the AUC is 0.8.

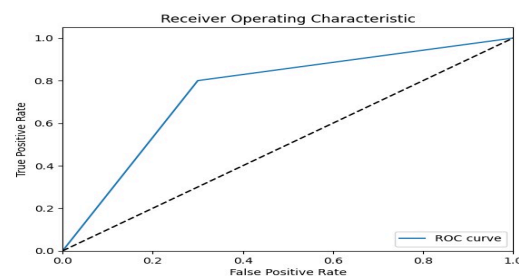
Here's the calculation:

$$AUC = (0.8 * 0.3) + (0.8 * 0.7) = 0.8$$

This means your classifier does a good job of distinguishing between the positive and negative classes. An AUC of 0.8 is considered good in real-world scenarios. It's always a good idea to double-checks your data and model when you get a high AUC score.

It's also important to consider other metrics as well when evaluating the performance of a classifier.

The following Graph represents the Receiver Operating Characteristic (ROC) curve.



V. Analysis

Hemophilia:

Hemophilia is quite common in Egypt, impacting many people and creating major obstacles in their day-to-day life. According to the World Federation of Hemophilia (WFH), a sizable number of people struggle with this inherited bleeding illness. The effects of hemophilia on those who have it are significant, since it can lead to joint injury, persistent discomfort, and disability. In some areas, access to specialized therapy may be restricted, which highlights the need for better healthcare administration and diagnostics. Our research aims to improve the overall quality of life for hemophilia patients in Egypt by addressing this demand by giving timely therapies and a tool for early identification. Our approach is informed by data-driven insights from credible organizations like as WFH and the Egyptian Ministry of Health. This allows us to connect our efforts with worldwide programs aimed at advancing hemophilia care that is customized to meet the unique requirements of the Egyptian community.

AI Machine learning model:

The code that was used is a Python script that involves the use of an AI machine learning model, specifically referred to as Random Forest Classifier from the Scikit-learn library. Here's a breakdown of what the code is doing:

The code specifies the features and the target variable used for training the AI model. In this case, the features include various health metrics, and the target variable is the diagnosis. This is shown in the following line:

```
# Define features and target variables
features = ["Heart Rate (bpm)", "Body Temperature (°C)", "Average Steps per Day", "Average Velocity (m/s)", "Age"]
target = "Diagnosis"
```

Following that, the script defines parameters for the Random Forest Classifier AI model to test it, such as the number of trees in the forest (estimators), the maximum depth of each tree (maxdepth), and a random seed (random state). These parameters are referred to as hyper-parameters, and they are responsible for testing the grid search abilities of the machine learning model. Lastly, the script defines data for a real patient and uses the trained model to predict the diagnosis for this patient. The result is printed based on whether the predicted diagnosis is "Hemophilia" or "Non-Hemophilia."

In summary, the code reads a dataset, trains a Random Forest Classifier model on health metrics data, evaluates the model's performance, and then predicts the diagnosis for a new patient using the trained model. The script showcases the application of machine learning for diagnosing Hemophilia based on health data.

Arduino:

The Arduino code uses a C++ script that is responsible of the communication between the sensors and each other and is connected to the machine learning AI model for integration of data.

Introducing sensors and communication:

The code includes various libraries for interfacing with sensors and communication. These include libraries for the MPU6050 accelerometer, Wire library for I2C communication, Software Serial for Bluetooth communication, One Wire and Dallas Temperature for interfacing with DS18B20 temperature sensor.

Defining analog pins:

This section defines pin assignments for different sensors and components. PULSE_PIN is assigned to an analog pin for the pulse sensor, LED_BUILTIN is set to pin 13 for an onboard LED, MPU6050_ADDR is the I2C address of the MPU6050 accelerometer, and DS18B20_PIN is the

pin for the DS18B20 temperature sensor.

Bluetooth connection with AI:

A Software Serial instance named Bluetooth is created for Bluetooth communication with the AI model, using pins 10 for RX and 11 for TX. It is illustrated below.

```
#include <SoftwareSerial.h>
```

```
Software Serial Bluetooth(10, 11); // RX, TX
```

Loop function:

In this loop function as shown in figure (3), it reads the analog signal from the pulse sensor and detects pulses based on signal peaks. It also controls the onboard LED based on pulse detection. Temperature data is read from the DS18B20 sensor, and MPU6050 accelerometer data is updated.

The acquired data (pulse rate and body temperature) is then sent over Bluetooth to a Python script, and the same data is printed to the Serial Monitor for debugging purposes. The pulse detection flag is reset, and there's a delay of 1000 milliseconds (1 second) before the next iteration of the loop.

```
void loop() {  
  // Read pulse sensor  
  int signal = analogRead(PULSE_PIN);  
  
  // Detect pulse peak  
  if (signal > peakSignal) {  
    peakSignal = signal;  
  } else if (signal < peakSignal && signal < previousSignal) {  
    if (peakSignal - signal > threshold) {  
      pulseDetected = true;  
      pulseRate++; // Increment pulse count  
    }  
    peakSignal = 0;  
  }  
}
```

Figure 3: Loop function

MPU6050:

The MPU6050 embodies a dual-sensor package, integrating both a three-axis MEMS accelerometer and a three-axis MEMS gyroscope. The accelerometer utilizes piezoelectric crystals, generating voltage proportional to linear acceleration along the X, Y, and Z axes. This allows for monitoring gait patterns, joint movements, and sudden changes in posture. The gyroscope, composed of vibrating structures, detects changes in angular velocity across the axes, capturing subtle rotations and twists.

The MPU6050 contains a piezoelectric crystal is sandwiched between the two electrodes. When a mechanical deformation takes place, it generates charge and hence it acts as a capacitor. A voltage is developed across the electrodes of the transducer which can be measured and calibrated with the deforming force to directly measure the mechanical deforming force. Figure (4) below shows a simple piezoelectric transducer.

The mechanical force can be

calculated by the following equation:
$$F = \frac{\epsilon_0 * A * V}{d * k}$$

Thus, by measuring the value of voltage across the electrodes of piezoelectric transducer, we can find the value of mechanical force. Hence, mechanical force converted into electrical signal which is the sole requirement of the accelerometer used in the system.

Figure 4: simple piezoelectric transducer.

Pulse Sensor:

Photoplethysmography (PPG) The pulse sensor utilizes the principles of PPG to detect the heart rate and translate into measurable data. The sensor is nestled on the fingertip, wrist, or earlobes since all of them have a rich arterial source

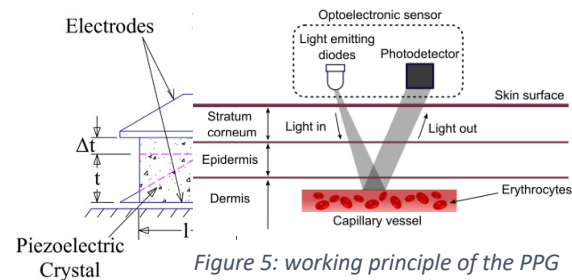


Figure 5: working principle of the PPG sensor

and are relatively easy to attach a sensor. The sensor emits light rays from diodes, specifically infrared and red electromagnetic rays, which penetrates the skin and blood vessels, and then reflects onto a photodetector. Figure (5) illustrates the working principle of the PPG sensor. However, pulsating blood flow momentarily alters light absorption, causing fluctuations in the reflected light. That's because volume changes are caused by pressure changes in blood vessels, which occur throughout the cardiac cycle.

The sensor's working mechanism depends on these fluctuations, detecting and interpreting them as heart rate variations. The onboard LED, often referred to as LED_BUILTIN, is a light-emitting diode located on the Arduino board itself. In the system it is used as an indicator to visually represent the detection of a pulse. When a pulse is detected, the LED will be

turned on.

DS18B20 Temperature Sensor:

The DS18B20 is a digital temperature sensor that communicates over a one-wire bus. It provides accurate temperature readings with a resolution of up to 12 bits. This seemingly simple sensor utilizes established scientific principles and a robust operational mechanism to accurately record temperature fluctuations.

Silicon Bandgap: The Core Technology:

The DS18B20 leverages the intrinsic relationship between silicon's electrical conductivity and temperature, known as the silicon bandgap principle. As temperature increases, the bandgap energy narrows, allowing more electrons to conduct electricity. The sensor exploits this phenomenon by incorporating two integrated diodes with distinct doping concentrations. This strategic design creates a temperature-sensitive "bandgap diode" and a stable "reference diode."

Voltage Variations: Translating Temperature Changes:

- ❖ When a constant current flows through both diodes, their voltage drops differ due to the varying bandgap energies. This differential voltage, directly proportional to the temperature change, becomes the raw data captured by the DS18B20. This voltage variation serves as the sensor's primary method of communicating temperature changes.

VI. Conclusion:

In summation, our project represents a critical response to the significant prevalence of hemophilia in Egypt, as estimated by reputable sources such as the World Federation of Hemophilia (WFH). Hemophilia's far-reaching impact on individuals, manifesting in chronic pain, joint damage, and disability, highlights the urgency of improved diagnostics and healthcare management in the country. Recognizing the potential limitations in access to specialized treatment in certain regions, our innovative project emerges as a vital solution, aiming not only to detect

hemophilia early but also to provide timely interventions that can substantially improve the lives of those affected. By aligning our efforts with global initiatives and incorporating data-driven insights from esteemed organizations like WFH and the Ministry of Health in Egypt, our project aspires to contribute to the broader global health agenda while addressing the unique challenges faced by hemophilia patients in Egypt. In this comprehensive approach, we envisage our project not just as a technological innovation but as a catalyst for positive change in the landscape of hemophilia care within the Egyptian healthcare system.

VII. Recommendations:

To enhance the accuracy and efficiency of detecting hemophilia disease, consider incorporating advanced sensors that can provide more detailed and comprehensive data. Here are some recommendations for sensors that can be used in conjunction with or as alternatives to the MPU6050 accelerometer and gyroscope and pulse rate kits:

Smart fabric technology:

For our hemophilia detection project, smart fabric technology offers a creative solution that is transforming the way patient data is gathered and tracked. Through the seamless integration of electrical components and sophisticated materials into textiles, smart fabrics allow for the direct monitoring of physiological data while wearing apparel. With this method, patients may wear wearables with greater comfort and wearability and real-time data transfer to our mobile app is made possible without the need for separate wearable devices. Temperature, pressure, and biometric data are all captured by smart textiles, which are versatile enough to handle a variety of sensors and provide a comprehensive approach to hemophilia diagnosis. Smart textiles offer a comprehensive solution, improving the practicality and user experience in the continuous monitoring and detection of hemophilia, thanks to their user-friendly design and possibility for multi-sensor integration.

MPU9250:

- ❖ The MPU9250 as shown in figure (6) is an advanced 9-axis motion sensor that surpasses its predecessor, the MPU6050, by incorporating a 3-axis magnetometer alongside the 3-axis accelerometer and 3-axis gyroscope. This integration provides a more comprehensive and precise measurement of motion and orientation in three-dimensional space. The additional magnetometer allows the MPU9250 to detect the Earth's magnetic field, enabling accurate heading determination and orientation tracking. This makes the MPU9250 particularly beneficial in applications where precise

motion and spatial orientation are critical, such as robotics, drone navigation, and wearable devices. The enhanced capabilities of the MPU9250 contribute to more accurate and detailed data capture, making it a superior choice for projects that demand advanced motion tracking and orientation sensing in real-life application

Figure 7: MPU9250



Electrocardiogram:

One important suggestion for the suggested capstone project on chronic illness detection is to include Electrocardiogram (ECG) sensors to gauge the heart's electrical activity. To obtain important insights into heart health—a factor that is especially important in hemophilia cases where possible cardiovascular problems may occur—

ECG data is an essential component. The integration of ECG sensors into the monitoring system allows the project to provide a more thorough picture of the patient's cardiovascular health. The aforementioned inclusion facilitates

the prompt identification of irregularities, guaranteeing prompt intervention and customized healthcare approaches designed to tackle possible cardiac problems linked to hemophilia as shown in the following figure (7)

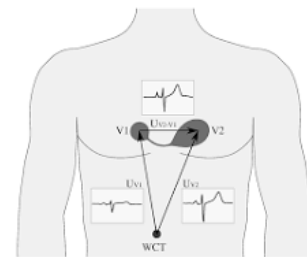


Figure 6: Electrocardiogram

Raspberry Pi:

The Raspberry Pi as shown in the following figure (8), a versatile single board computer, stands as an ideal recommendation for the hemophilia detection project designed to operate on a large scale in real-life applications. Renowned for its substantial processing power, GPIO flexibility, and extensive community support, the Raspberry Pi offers a robust platform for seamlessly integrating sensors like MPU6050, PPG, and a waterproof temperature sensor. Its regulated power supply, efficient power management capabilities, and compatibility with battery backups ensure stability and continuity of operation. Additionally, the Raspberry Pi's low-noise design, temperature resilience, and modularity make it well-suited for scalable deployment. The Raspberry Pi empowers the project with the computational prowess needed for implementing Random Forest AI, providing a comprehensive solution for accurate hemophilia detection while facilitating future expansions and optimizations.



Figure 8: The Raspberry Pi

VIII. Literature cited:

- ❖ James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An introduction to statistical learning: with applications in R
- Lillicrap, D., & Blanchette, V. S. (Eds.). (2018). Hemostasis and thrombosis: Basic principles and clinical practice (6th ed.). Wolters Kluwer. Chapter 52
- Kadir, R. A. (2016). Hemophilia: A guide for patients and families (3rd ed.)
- Python for Scientists and Engineers by Hans Petter Langtangen (2017)
- <https://github.com/scikit-learn/scikit-learn>

