UNIVERSITÀ DEL SALENTO

Facoltà di Ingegneria

Corso di Laurea in Ingegneria dell'Informazione

Documentazione di Progetto

per il corso

Internet of Things

Definizione e implementazione di un digital twin in ambito sanitario, focalizzato sullo scompenso cardiaco

Docente

Prof. Luigi Patrono

Tutor

Ing. Ilaria Sergi

Ing. Elisabetta De Giovanni

Studenti

Davide Cantoro

Michele Schilardi

Alberto Leo

Index

1 Introduction **2**
- 1.1 Digital Twin Healtcare . . . 2
  - 1.1.1 Dashboard . . . 3
- 1.2 Machine Learning . . . 4
- 1.3 Machine Learning in Healtcare . . . 5

2 Requirements Analysis **6**

3 State of the art **8**
- 3.1 Digital Twin . . . 8
- 3.2 Machine Learning and Artificial Intelligence . . . 8
  - 3.2.1 AI Act . . . 9

4 Technologies **10**
- 4.1 Rest API . . . 10
- 4.2 Python . . . 10
  - 4.2.1 scikit-learn . . . 11
  - 4.2.2 neurokit2 . . . 11
  - 4.2.3 Flask . . . 12
- 4.3 MongoDB . . . 12
- 4.4 Angular . . . 12
  - 4.4.1 CoreUI . . . 12

5 Proposed solution **13**
- 5.1 Data . . . 13
- 5.2 API . . . 14
- 5.3 Dashboard: Patients list . . . 15
- 5.4 Dashboard: Patient Digital Twin . . . 16
- 5.5 Machine Learning . . . 19
  - 5.5.1 Scheduling . . . 21
  - 5.5.2 AI Act . . . 21

6 Functional validation **23**
- 6.1 Machine Learning . . . 24

7 Conclusions and future works **25**

1 Introduction

In recent years, the importance of artificial intelligence, combined with iot devices in the healthcare sector, has been growing. In fact, the use of a digital twin offers an innovative approach in this area and, when used as a monitoring system, can be helpful in determining the medical condition of patients and facilitating quicker decision-making.

The use of digital twins in healthcare can be helpful in a number of ways, including managing resource allocation and keeping track of patients' conditions and reactions to therapies.

IoT devices are essential for the implementation of Digital Twin in this domain. IoT devices, such as fitness bands and chest bands, are able to capture a good amount of useful parameters to keep trace of the patient's statio cardiac condition, there are chest bands that can provide an ECG.

As time passes, these tools become less expensive and more accurate, this great improvement therefore allows a large amount of data to be quickly available. In addition to the economic advantage, these technologies are less invasive and more portable than the conventional techniques currently in use, and they also support remote control.

These fundamental presumptions led to the creation of the project, which aimed to define and implement a Digital Twin in healthcare with a particular focus on heart failure.

1.1 Digital Twin Healtcare

A Digital Twin consists of the virtualised digital transposition of a system, in the context of the project a system refers to a large number of patients.
But in order to accurately represent a patient's digital twin, a large number of sensors that can monitor the patient's condition are needed, which is where the Internet of Things (IoT) comes in.
The use of digital twins has the potential to revolutionize healthcare as they allow to facilitate the quick monitoring of many patients and the development of customized treatments based on their unique needs.

This is possible because of the Digital Twin, which makes it possible to track a variety of patient-related metrics, including specific pathology, family history and display of parameters in pseudo real time.
Personalised Medicine is one of the most exciting uses of digital twins in healthcare. The healthcare industry is shifting from a concept of medicine as patient care to one of prevention, in large part because of technologies like artificial intelligence.

A digital twin could be used to simulate an entire health care center. One could more effectively manage the use of hospital beds or the distribution of resources (human and/or material) by building a digital twin of the entire facility. This would result in cost optimization and general improvements to the overall effectiveness and quality of these facilities.

Another benefits of utilizing a Digital Twin in healthcare are the possibility to access data remotely, a requirement that first came out during COVID-19 and has since become standard for many realities.

1.1.1 Dashboard

A dashboard is a collection of charts and visualized data organized so that specific elements of interest for the context being studied are emphasized. A dashboard must provide quick access to significant data and information, regardless of its complexity or nature.
The Data in a dashboard are visualized using charts, reports, lists, and other visual elements.

A dashboard provides:

- a simple process for reading information
- quick access to information
- customized interface
- context-based data organization
- overall visualization of more information at once.

1.2 Machine Learning

Machine Learning (ML) is a sub-field of Artificial Intelligence (AI), it focuses on developing machine learning systems that learn from input data to make predictions or classifications.

There are two types of Machine Learning: supervised machine learning and unsupervised machine learning. The way the algorithm learns from the data input data distinguishes these two types.
Because it is simpler to obtain results immediately when the target variable is present, the most widely used Machine Learning algorithms is the supervised learning.

In supervised Machine Learning, a data scientist acts as a teacher and teaches the algorithm how to produce results, much like how one would teach a child to recognize various animal species based on pictures.
In fact, the algorithm can learn from a dataset that has already been labeled and produce a classification or prediction.

Unsupervised Machine Learning, on the other hand, uses a more autonomous approach. Without the need for prior labeling of the input dataset, the algorithm learns pattern identification on its own.

Because supervised machine learning was chosen for this project, it was necessary to select a suitable dataset that should have the presence of the target variables.

1.3 Machine Learning in Healtcare

In the fields of health and medicine, artificial intelligence now plays a significant role, machine learning is very helpful because it helps with the identification and/or prediction of diseases and pathologies.
Machine Learning enables the predicting and treatment of disease by providing medical imaging and diagnostics.
Machine Learning can also be used for discovering and developing new drugs or to better organise the management of resources.

In the context of heart failure, using Machine Learning it is in fact possible to go to a certain level of probability to identify the presence or absence of heart failure.

The project's goal with Machine Learning is to establish the basis for a Machine Learning model that is capable of recognizing people with heart issues.

2 Requirements Analysis

1. Aim and Objectives
    - help doctors identify patients who are at risk of developing heart failure.
    - Early detection, data visualization, and a user-friendly interface for medical staff are among the objectives.
    - The use of a digital twin that can track patient variables with a focus on heart failure. The requirement for "real time" in this digital twin is relaxed because it is sufficient for the data to be displayed periodically.
    - the availability of a machine learning algorithm that can identify heart failure.
2. Target
    - Medical professionals (doctors, nurses, etc.)
3. Functional Requirements
    - Multiple Dashboards: The application will have separate dashboards that display patient data and heart failure risk analysis.
    - Data Visualization: To present patient health trends, the application will make use of interactive graphs.
    - Patient Profiles: Individual patient profiles with pertinent medical data will be available to medical professionals.
    - Risk Assessment Algorithms: Using patient data, the application will implement sophisticated algorithms to evaluate the risk of heart failure.
4. Non-Functional Requirements
    - Performance: For data loading and analysis, the application will offer a responsive user interface with a fast response time.
    - Usability: Medical professionals with a range of technical skills will find the user interface to be simple to use and intuitive.

- Scalability: The application is going to be made to be able to handle more and more concurrent users.

5. User Experience and Interface Requirements

   - Dashboard Design: The application will feature crystal-clear dashboards that are aesthetically pleasing and contain pertinent patient data.
   - Data visualization: To assist medical professionals in understanding patient health trends, the application will make use of suitable data visualization techniques.

6. Scalability and performance

   - Performance Optimization: The application's performance will be enhanced to speed up response time.

3 State of the art

3.1 Digital Twin

A digital twin is a replica that makes it possible to simulate the state of a physical asset or system. The physical traits and changes in the patient's body are translated into the digital environment to create the patient's Digital Twin.
Similar to the development of engineering products, the creation of the digital twin of the human body involves more sophisticated and intricate procedures. The Digital Twin of the designed object can receive data from sensors with efficiency, but collecting information from people can be more expensive and time-consuming because it frequently requires blood tests, imaging systems, and medical scans.

There are no comprehensive explanations of a generic digital twin in the literature, let alone descriptions of implementations of this emerging technology in the practical field, despite the fact that it promises to bring about significant innovations in the field of medical support.

3.2 Machine Learning and Artificial Intelligence

Artificial Intelligence refers to techniques for sorting through vast amounts of data and identifying their patterns.
Machine learning is widely used in research, there are many laboratories that use it. It is used in particular to recognize and/or forecast diseases and pathologies, to predict and treat disease through the provision of medical imaging and diagnostics, to find and develop new drugs, as well as to better organize resource management.
It is also used to analyze gathered data in order to identify patterns and make predictions.

The digital twin in healtcare, in particular for heart failure prediction is an emerging field, there are not many articles about it.
The majority of articles discuss selecting machine learning algorithms and highlighting the value of such systems, but they do not discuss how these systems can interact with IoT systems, which, for instance, can using dashboards and manage multiple patients.

Some articles:

- Coorey, G., Figtree, G.A., Fletcher, D.F. et al. The health digital twin to tackle cardiovascular disease—a review of an emerging interdisciplinary field. npj Digit. Med. 5, 126 (2022).
- Choi, DJ., Park, J.J., Ali, T. et al. Artificial intelligence for the diagnosis of heart failure. npj Digit. Med. 3, 54 (2020).
- Tauben Averbuch and others, Applications of artificial intelligence and machine learning in heart failure, European Heart Journal - Digital Health, Volume 3, Issue 2, June 2022.

3.2.1 AI Act

Another deficit in the literature is the absence of articles on the AI Act, which is a relatively recent topic, actually realized this on June 14 2023.
The AI Act is a proposed law for Europe on Artificial Intelligence (AI) from a major regulator in Europe, making it the first law of its kind anywhere.
Applications of AI are divided into three risk categories by the law.

First, it is forbidden to use programs and systems which present an unacceptable risk (like China's government-run social scoring).
Second, there are specific legal requirements that apply to high-risk applications, such as a tool that scans CVs and ranks job applicants.
Last but not least, applications that aren't explicitly forbidden or listed as high-risk remain largely unregulated.

4 Technologies

The technologies that were used and how they interacted to create the system architecture are shown in the illustration that follows.

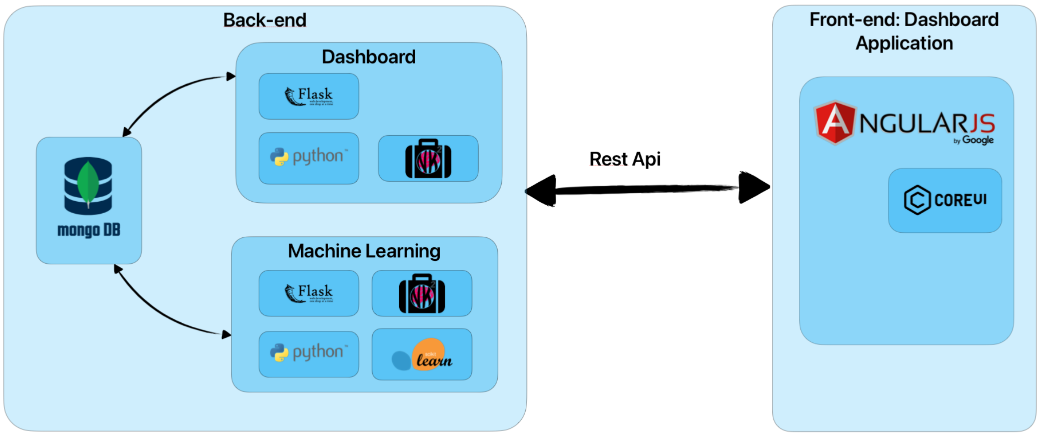

Figure 4.1: system architecture

4.1 Rest API

Utilizing HTTP requests, the REST API (Application Programming Interface) technology enables communication between software programs and/or web services. This enables the various components to be separated so that they can each act independently.

4.2 Python

Python was chosen for a number of reasons, one of which is that it is a very useful programming language in the area of rapid prototyping.

One of the most important Python libraries for scientific computing is NumPy (np). It is used to carry out operations on multidimensional arrays and matrices efficiently.
Pandas (pd): Pandas is a library that offers adaptable and quick data structures for analyzing and manipulating data. It is employed to manage and load patient data, most likely from a file or an external source.
Schedule: Task scheduling in Python is made simple with the help of the

library called Schedule. It is helpful for carrying out specific tasks at predetermined times or on a regular basis.
Requests: This Python HTTP/1.1 library allows you to send requests over HTTP. It allows to communicate with web APIs, download files from the Internet, and make HTTP requests to send data.

4.2.1 scikit-learn

Scikit-learn (or sklearn) is a machine learning library for Python.
It has a variety of classification, regression, and clustering algorithms and has been designed to integrate with numerical and scientific libraries like NumPy and SciPy.

4.2.2 neurokit2

NeuroKit2 (ok nk2) is a toolbox for physiological signal processing.
Electrocardiography (ECG), photoplethysmography (PPG), electrodermal activity (EDA), respiratory (RSP), electromyography (EMG), and electrooculography (EOG) signals are used as tools to work with cardiac activity with nk2.
It makes it possible to calculate respiratory and heart rate variability (HRV) metrics.

Additionally, it uses a number of different algorithms, including an effective internal R-peak detector, to find R-peaks and other QRS waves.
It supports microstates and frequency band analysis for neurophysiological signals like EEG.

Neurokit2 cite:

- Makowski, D., Pham, T., Lau, Z. J., Brammer, J. C., Lespinasse, F., Pham, H., Schölzel, C., & Chen, S. A. (2021). NeuroKit2: A Python toolbox for neurophysiological signal processing. Behavior Research Methods, 53(4), 1689-1696. https://doi.org/10.3758/s13428-020-01516-y

4.2.3 Flask

Flusk is a simple and lightweight micro-framework to develop web applications and RESTful APIs .

4.3 MongoDB

MongoDB is a NoSQL database. MongoDB is a popular document-based database widely used in big data analysis.It offers more flexibility than conventional relational databases and is used to handle large volumes of data and workload-intensive applications.

4.4 Angular

Angular is a framework for web application development developed by Google. By offering a structured approach, it makes the development of challenging single-page applications (SPA) easier.

One of Angular's key characteristics is its component-based architecture, which enables the reuse and modularity of these components. Data binding and templating, two features of Angular that make it easier to create dynamic user interfaces, are both useful.

4.4.1 CoreUI

CoreUI is an advanced front-end technology designed to make it easier to develope web application that use the Angular framework.
It offers a thorough selection of predefined UI elements that make it simple and to use to create graphical user interfaces. The wide range of pre-built components offered by CoreUI reduces the time required for deployment. Additionally, CoreUi's responsive adaptation of pages for each device and viewing mode guarantees a consistent user experience across desktop and mobile platforms.

5 Proposed solution

Two dashboards make up the suggested method for implementing a digital twin in healthcare.

The first dashboard shows a summary of all patients, is important the presence of a parameter indicating the probability that the patient is suffering from Heart Failure, this parameter is given by M.L. algorithm.
The second dashboard make possible to see all the patient's vital parameters, this dashboard is the effective digital twin.
Additionally, a machine learning algorithm that can predict whether a patient will experience heart failure was used.

An application's backend, which manages data processing and coordination between the frontend and database, is a key component. In this instance, the data of patients with suspected heart disease is stored in the backend using the Python language and the MongoDB connection module. It also includes a logistic regression model that has been previously trained to predict and categorize patients based on whether they have heart disease or not. Meanwhile, Angular and CoreUI were used for the frontend.

In addition, On each page of the site it is possible to download the csv of the dataset by simply clicking on the respective button.

5.1 Data

It was crucial to create the database used as the basis for both the digital twin and the machine learning algorithm because there wasn't a starting database available.

An ECG was produced with the help of neurokit2.
For the spO2 and heart rate data, after a thorough research, useful datasets were identified, and using their relative frequencies it was possible to generate measurements. These measurements were then incorporated into the digital twin.

The remaining data was based on the 'Heart Disease' dataset, one of the most popular datasets in this area.

Data on patients with heart failure diagnoses are managed by the Heart-Failure database. The patient and ECG collections are both present in the database. The Patients collection keeps track of patient data, including the outcomes of an machine learning algorithm that determine whether heart failure is present and how likely it will be. The ECG collection contains records of ECG signals for each patient.

5.2 API

The backend transmits data from each patient using the Rest API.
This enables the separation of all logic between the backend and the frontend, improving the system's eventual evolution.


```
_id: ObjectId('64b111266a188c7c51986df5')
name: "Blake Zabala"
age: 46
sex: 0
cp: 1
trestbps: 105
chol: 204
fbs: 0
restecg: 1
thalach: 172
exang: 0
oldpeak: 0
slope: 2
ca: 0
thal: 2
▸ heart_history: Array (1440)
▸ spo2_history: Array (1440)
target: 1
PBS: 98.27
id: 1
```


Figure 5.1: Patient API response

5.3 Dashboard: Patients list

The first dashboard will contain a multi-patient view (patients list).

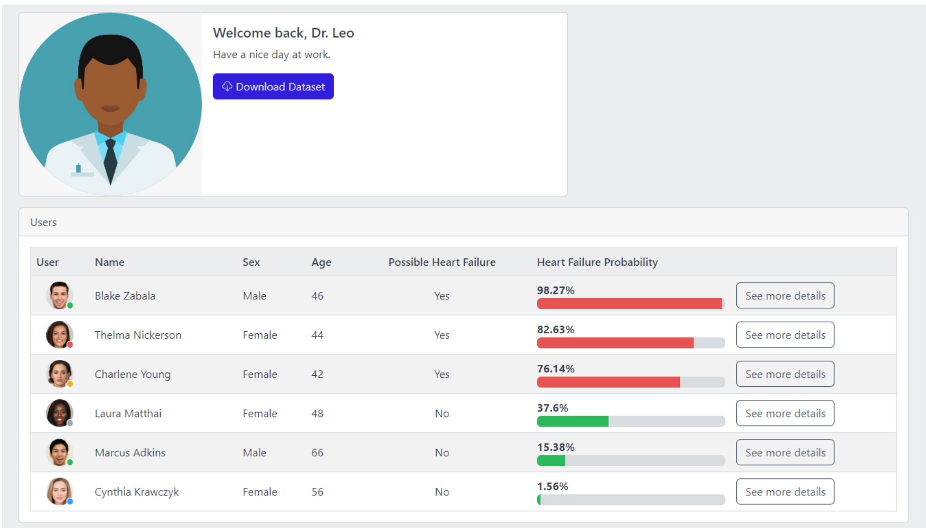

Figure 5.2: Dashboard Patients list

The main dashboard's purpose is to give the doctor a quick overview of the patients being treated in his or her department in terms of their health. Each patient's name, age, gender, and chest pain are provided in the dashboard.

The doctor is immediately informed of each patient's risk of presenting with heart failure and the percentage likelihood that this will happen after receiving their personal information. The doctor also has access to each patient's individual dashboard, which contains all the necessary information.

A table was used to create the dashboard that contains all of the patients, with each field being contained in a column that is labeled. The doctor can switch the display mode to the patient-specific dashboard using the button in the last column. In actuality, when a user clicks, a routing process is started that directs them back to their unique page.

5.4 Dashboard: Patient Digital Twin

Some identifying components from the earlier dashboard, including the photo, name, age, gender, and likelihood of having a heart attack, are first included in this more comprehensive dashboard. Additionally, details about the potential existence of chest pain and the highest heart rate measurement are reported. The final field has been further divided into two sections for completeness: on the one hand, the highest historical value ever recorded in the database, and on the other, the value recorded over a given time period, in accordance with the display option selected by the doctor.

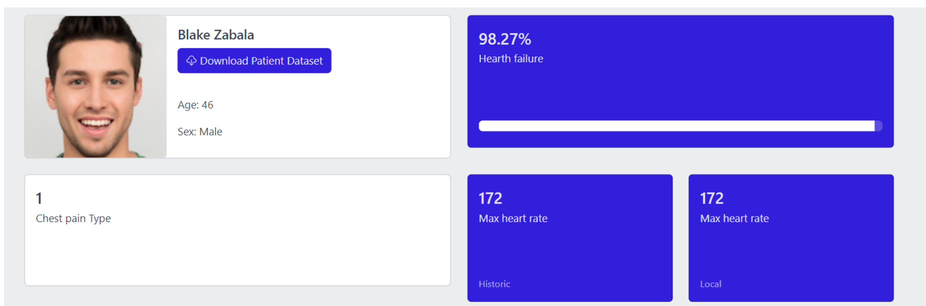

Figure 5.3: Patient general metric

The patient's heartbeat values, which were recorded every minute, are displayed on a graph. The default display is set to 5 hours, meaning that the patient's values recorded during this time are displayed. However, a function was implemented that allows the doctor to change how the values are displayed in the graph. The doctor has the option to adjust this time limit at any time to 1 hour or 12 hours.

On the other hand, the second graph shows the spO2 measurements taken for each patient. The same logic that was used to create the previous graph was also used to create this one.

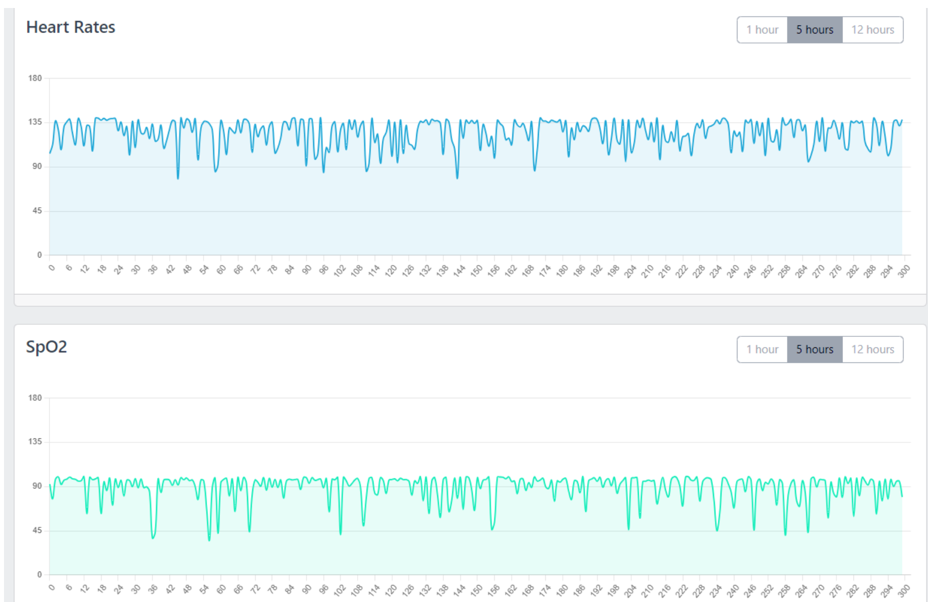

Figure 5.4: Patient Heart Rate and SpO2

The patient's ECG is be viewed on a carousel. As you scroll through the images, the same graph is visible, but with important features like the R, P, Q, S, and T peaks highlighted.

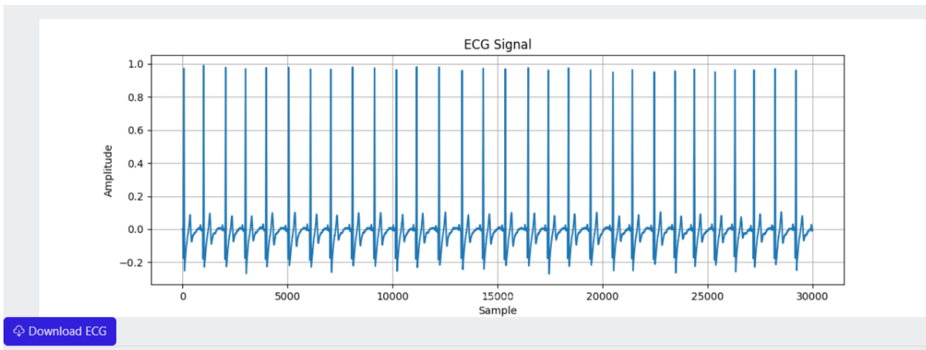

Figure 5.5: ECG signal

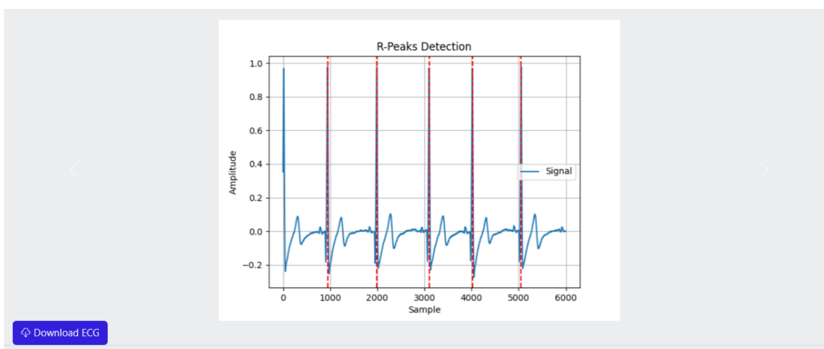

Figure 5.6: ECG signal with R peak

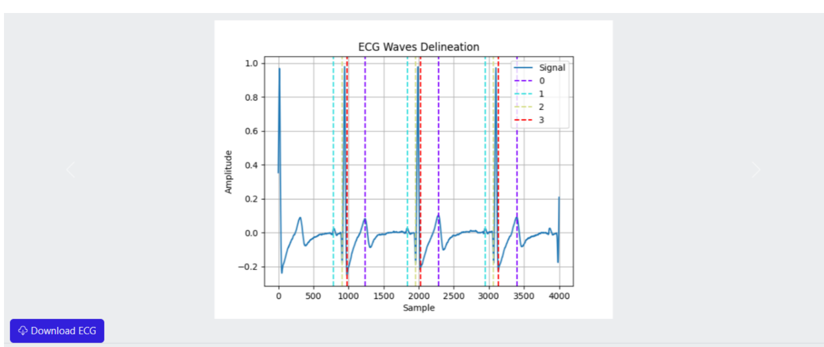

Figure 5.7: ECG signal with P, Q, S, T peak

5.5 Machine Learning

The technique of Machine was applied to bring attention to patients with heart issues.

The dataset used to train the Machine Learning algorithms was taken from Keggle and is the most widely used dataset in the context of Heart Disease. The dataset selection was crucial, the presence of the target variable made it possible to use the Supervised Learning.

After a Data Visualisation e Preparation step of the dataset, Six classification algorithms were selected: Logistic Regression, Decision Tree Classifier, Random Forest Classifier, Support Vector Machines, K-Neighbors Classifier e Naive Bayes.

The k-fold cross validation, a resampling technique, was used to train the Machine Learning models.

```
#k-fold cross validation

# Classification models
models_simplest = []
models_simplest.append(('LR', LogisticRegression(solver='lbfgs', max_iter=5000)))
models_simplest.append(('DTC', DecisionTreeClassifier()))
models_simplest.append(('RFC', RandomForestClassifier(n_estimators=100)))
models_simplest.append(('SVM', SVC(gamma='scale')))
models_simplest.append(('KNN', KNeighborsClassifier()))
models_simplest.append(('NB', GaussianNB()))

# evaluate accuracy for each model
results = []
names = []
for name, model in models_simplest:
    kfold = model_selection.KFold(n_splits=10)
    cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold, scoring="accuracy")
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)
```

```
LR: 0.742667 (0.061446)
DTC: 0.688333 (0.106667)
RFC: 0.759333 (0.040731)
SVM: 0.701167 (0.045228)
KNN: 0.730333 (0.049810)
NB: 0.734667 (0.086871)
```

Figure 5.8: K-Fold Cross Validation

After the models were trained, Tune Hyperparameters was used to identify the best model parameters in order to enhance their performance, repeating the K-Fold Cross Validation again.

```
[ ]
# Logistic Regression

# Create the model
logreg = LogisticRegression(random_state=0, max_iter=5000, solver='liblinear')

#Parameters for grid search
solvers = ['newton-cg', 'lbfgs', 'liblinear']
penalty = ['l2']
c_values = [100, 10, 1.0, 0.1, 0.01]
param_grid = dict(solver=solvers,penalty=penalty,C=c_values)

grid = GridSearchCV(estimator=logreg, param_grid=param_grid, cv=5, scoring="accuracy", verbose=1, n_jobs=4)

# Run grid search on the training data
grid.fit(X_train,Y_train)

# print score
print("Accuracy: ",grid.best_score_)
print("Model: ",grid.best_estimator_)
print("Best parameters: ",grid.best_params_)

LogisticRegression_simplest_model = grid.best_estimator_
```

```
Fitting 5 folds for each of 15 candidates, totalling 75 fits
Accuracy:  0.7469387755102042
Model:  LogisticRegression(C=100, max_iter=5000, random_state=0, solver='newton-cg')
Best parameters:  {'C': 100, 'penalty': 'l2', 'solver': 'newton-cg'}
```

Figure 5.9: Tune Hyperparameters

```
Default model               Tuned model             Improvement
LR: 0.742667 (0.061446)     0.809333 (0.061446)     0.06666666666666665
DTC: 0.688333 (0.106667)    0.763167 (0.106667)     0.07483333333333353
RFC: 0.759333 (0.040731)    0.801000 (0.040731)     0.04166666666666652
SVM: 0.701167 (0.045228)    0.805167 (0.045228)     0.10400000000000009
KNN: 0.730333 (0.049810)    0.809333 (0.049810)     0.07899999999999996
NB: 0.734667 (0.086871)     0.801000 (0.086871)     0.06633333333333336
```

Figure 5.10: Tune Improvement

To evaluate the best model, precision was used as a metric; It is crucial that the model be extremely accurate and perform very few errors when it comes to making predictions about whether a patient has heart disease or not. Here is where precision is crucial.

Accuracy, Prediction, Missclassification Rate

```
Accuracy: 0.8032786885245902
Precision: 0.8205128205128205
Misclassification Rate: 0.19672131147540983
```

Classification Report

	precision	recall	f1-score	support
0	0.83	0.83	0.83	30
1	0.84	0.84	0.84	31
accuracy			0.84	61
macro avg	0.84	0.84	0.84	61
weighted avg	0.84	0.84	0.84	61

Confusion Matrix, False Positive, False Negative

```
Confusion Matrix
    [[25  5]
     [ 5 26]]

False Positive - Type I error
    There are  5  False Positive with rate of  0.16666666666666666

False Negative - Type II error
    There are  5  False Negative with rate of  0.16129032258064516
```

Figure 5.11: Metrics

The model chosen for the implementation of the Digital Twin was logistic regression (which has a precision of 83.87%).

5.5.1 Scheduling

Every day at 23:00, the backend schedules the execution of the function to predict the probability that a patient will have heart failure. This is done using the "schedule" library. As a result, daily predictions can be made automatically, keeping the data up-to-date.

5.5.2 AI Act

The AI Act uses a hierarchical subdivision of risks to classify AIs. Each hierarchy has its own constraints of implementation that must be respected.

Unacceptable risk: Prohibited. They are AIs that "deploy subliminal techniques beyond a person's consciousness in order to distort a person's behaviour in a manner that is likely to cause physical or psychological harm"
High risk: Permitted subject to compliance with AI requirements and ex-ante conformity assessment. As for Biometric identification and categorisa-

tion of natural persons: "AI systems intended to be used for the 'real-time' and 'post' remote biometric identification of natural persons;"
Minimal or no risk: Permitted with no restrictions. E.g. Spam filters, Video games

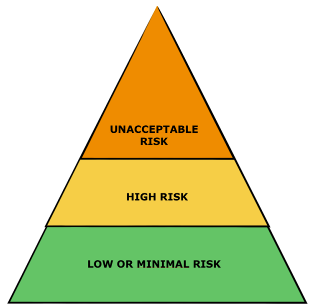

Figure 5.12: Ai Act Risk

Low Risk Example

- AI system that predicts 30-day hospital readmissions for patients with coronary heart disease based on Electronic Medical Record (EMR) data intended for hospital management to plan their intensive care (IC) capacity
- AI system that selects patients who meet specific medical and biomarker criteria from EMR data intended to recruit participants for participation in a clinical trial

High Risk Example

- Application for consumers used to assess skin lesions intended to provide a risk estimate and medical advice
- AI system which monitors patient side effects to selective serotonin reuptake inhibitors intended to make dose adaption recommendations to physicians

Following these guidelines and considering the previous examples, the submitted proposal falls into low or minimal risk. This system would classify as a medical device as it is used for the prediction of disease. However, it is not used for the specific characteristics it falls into low or minimal risk.

6 Functional validation

Utilizing manual tests, functional validation was performed through by comparing application results to the functional specification. In order to cover every scenario, both cardiac patients and healthy patients were used in the tests.

The system exposes the Rest API, so by interrogating it, it is possible to assess its and check the porperly operating.


```
{
    "_id": "64b111266a188c7c51986df5",
    "name": "Blake Zabala",
    "age": 46,
    "sex": 0,
    "cp": 1,
    "trestbps": 105,
    "chol": 204,
    "fbs": 0,
    "restecg": 1,
    "thalach": 172,
    "exang": 0,
    "oldpeak": 0.0,
    "slope": 2,
    "ca": 0,
    "thal": 2,
    "heart_history": [⋯
    ],
    "spo2_history": [⋯
    ],
    "target": 1,
    "PBS": 98.27,
    "id": "1",
    "max720": 172.0,
    "max300": 139.8,
    "max60": 139.8
}
```


Figure 6.1: Patient API response

```
{
    "_id": "64b9e70c5635a48b3188754e",
    "patient_id": "64b111266a188c7c51986df5",
    "ecg_signal": [⋯
    ],
    "id": 1,
    "name": "Blake Zabala",
    "ECG_Image": "iVBORw0KGgoAAAANSUhEUgAABLAAAAGQCAYAAAC
        +tZleAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjcuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/
        bCgiHAAAACXBIWXMAAA9hAAAPYQGoP6dpAAEAAElEQVR4nOydd5xU1fn/
        P7OzjYWlSe8gCnYUAmIvFN1EYxITv8aoIcYkJhoTfimSKEaNmhhrrFFRY4tGY9TIiiCCgPSy9N4WFnZhd9nepv+mLl3zr1zp
        +zOPc8Z7n3erxcvZmZn7nPPuac85znP8xxPKBQKgWEYhmEYhmEYhmEYhmEylCzVN8AwDMMwDMMwDMMwDMMwiWADFsMwDMMwDMMwDMMwDMMwDJPRsAGLY
        UwDMMwDMMwDMMwDMNkNGzAYhiGYRiGYRiGYRiGYRiGYTIaNmAxDMMwDMMwDMMwDMMwGQ0bsBiGYRiGYRiGYRiGYZiMhg1YDMMwDMMwDMMwDMMwDMMwTEbDBi
        DFsMwDMMwDMMwDMMwDJPRsAGLYRiGYRiGYRiGYRiGyWjYgMUwDMMwDMMwDMMwDMNkNGzAYhiGYRiGYRiGYRiGYTIaNmAxDMMwDMMwDMMwDMMwGQ0
```


Figure 6.2: Patient ECG API response

6.1 Machine Learning

A variety of metrics were used to evaluate the performance of the machine learning algorithm, including accuracy, precision, missclassification rate, confusion matrix, false positive rate, and false negative rate.

Accuracy, Prediction, Missclassification Rate

```
Accuracy: 0.8032786885245902
Precision: 0.8205128205128205
Misclassification Rate: 0.19672131147540983
```

Classification Report

	precision	recall	f1-score	support
0	0.83	0.83	0.83	30
1	0.84	0.84	0.84	31
accuracy			0.84	61
macro avg	0.84	0.84	0.84	61
weighted avg	0.84	0.84	0.84	61

Confusion Matrix, False Positive, False Negative

```
Confusion Matrix
    [[25  5]
     [ 5 26]]

False Positive - Type I error
    There are  5  False Positive with rate of  0.16666666666666666

False Negative - Type II error
    There are  5  False Negative with rate of  0.16129032258064516
```

Figure 6.3: Machine Learning report

7 Conclusions and future works

The use of IoT devices is essential because it enables data collection, such as smart bands and heart bands.
Measures like blood oxygenation or historical heart rate are helpful in understanding the patient's physical condition and can be provided by the suggested solution for developing a digital twin in healthcare.
Additionally, the ecg is very helpful in the field of heart failure because it gives a pseudo-real-time overview of the patient and allows for the identification of the R, P, Q, S, and T peaks.

A simple user interface for interacting with the system is provided by the Flask server. With Flask, a REST API architecture that interacts with the database and returns JSON responses can be efficiently developed and managed.

It is also crucial to have the Heart Failure Possibility index because it provides an index that indicates the likelihood that the patient has heart failure. Doing this, Six Machine Learning models were trained and to evaluate the best model, precision was used as a metric.

It is crucial that the model be extremely accurate and commit very few errors when it comes to making predictions about whether a patient has heart disease or not. The use of precision is crucial. The model chosen for the implementation of the Digital Twin was logistic regression (which has a precision of 83.87%). The precision percentage achieved by the model still has potential for improvement, although it is still good. This is because the original dataset had a limited amount of observations.

Overall, the other models also had good results.
The difference found between the Reduced Dataset and the Full Dataset is slight, as the selected features are among the most impactful.
The removal of duplicate observations had a huge impact on the model, improving the final model.
The full documentation on this experiments can be found on git-hub: link

The REST API provides an easy and efficient way to communicating with the system. Retrieval of patient information, ECG information, and relevant images is possible tanks to REST API.

Given the ongoing growth of the field of digital twins and machine learning in healthcare, this implementation has a wide range of potential future developments.

- patient app, with a page to enrich the data collected by the bands and a messaging system to remind the patient to enter daily data (if necessary)
- implementation of advanced machine learning algorithms, like neural network.
- doctor's app allowing the modification and/or enrichment of data
- doctor's app that gives the possibility to upload treatments related to the individual patient
- implementing advanced patient management: assigning a referring doctor, treatments ...
- introduce more parameters in the digital twin
- automating the collection of more parameters with IoT systems
- see how different coding languages other than Python impact the M.L. algorithm (e.g. Mojo)
- the API does not currently require authentication, so appropriate security measures must be put to use.