Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Mohamed Khider - BISKRA
Faculty of Exact Sciences, Natural and Life Sciences
Computer Science Department

Order N° : IA_StartUp 13/M2/2023

Thesis

Submitted in fulfillment of the requirements for the Masters degree in

Computer science

Option : **Artificial Intelligence**

On the forecasting of heartbeat signal and body temperature using IoT devices and ML techniques

---

By :
ZOHRA DAKHIA

Graduating on 25/06/2023 in front of the following committee of juries:

Abdelmoutia Telli	MCA	President
Houcine Belouaar	MCA	Supervisor
Sihem Sahli	MAA	Examiner

Session 2022-2023

بسم الله الرحمن الرحيم

Contents

List of Figures

List of Tables

Acknowledgements

First and foremost, I am immensely grateful to Allah, the Most Gracious, the Most Merciful, for His blessings, guidance, and unwavering support throughout this journey.

I am deeply appreciative of my supervisor, ***Dr. Belouaar Houcine***, for his exceptional mentorship, invaluable insights, and continuous support. His expertise, patience, and dedication to my success have been instrumental in shaping this thesis.

I would like to extend my sincere thanks to the Laboratory of Computer Science and Artificial Intelligence ***LINFI*** for providing me with a stimulating research environment, access to cutting-edge resources, all the devices needed, and opportunities for collaboration. The unwavering support and encouragement from the LINFI team have been crucial in the completion of this thesis.

I am grateful to ***Prof. Kahloul Laid*** for his valuable feedback, constructive criticism, and encouragement. His expertise and guidance have significantly contributed to the quality of my research work.

I would also like to express my appreciation to all the faculty members and staff of the Computer Science Department at University Mohamed Khider Biskra for their assistance, encouragement, and academic guidance during my studies. Their knowledge and expertise have been invaluable in shaping my academic and research pursuits.

I would like to express my gratitude to ***Dr. Moussi abdelkarim*** and all the members of the Entrepreneurship House for their valuable assistance in preparing my startup thesis.

I am deeply thankful to my family and friends for their unwavering support, prayers, and motivation. Their belief in my abilities and constant encouragement have been a driving force behind my academic achievements.

Lastly, I would like to acknowledge the contributions of all other individuals, colleagues, and peers who have provided assistance, advice, or support during my research journey. Your contributions have been recognized and deeply appreciated.

Dedication

This work is dedicated to my beloved parents, who have been my pillars of strength and the driving force behind my accomplishments. Your endless love, guidance, and unwavering support have been invaluable to me, and I am grateful for the sacrifices you have made for my sake.

Mom ***Debabeche Sabrina***, your unconditional love, care, and wisdom have shaped me into the person I am today. Your unwavering belief in me and your constant encouragement have been a beacon of light during challenging times. Thank you for your unwavering dedication to my well-being and success.

Dad ***Azzedine***, your unwavering support, guidance, and encouragement have been instrumental in my journey. Your belief in my abilities and your constant motivation have inspired me to reach for the stars. Thank you for being my role model and for instilling in me the values of perseverance, determination, and hard work.

To my sisters, ***Fedoua***, ***Nihal***, ***Ranime***, ***Elena***, and ***Nourssin***, who have always been there for me with their unconditional love and unwavering support. Your presence in my life has been a source of strength and inspiration, and I am grateful for the bond we share.

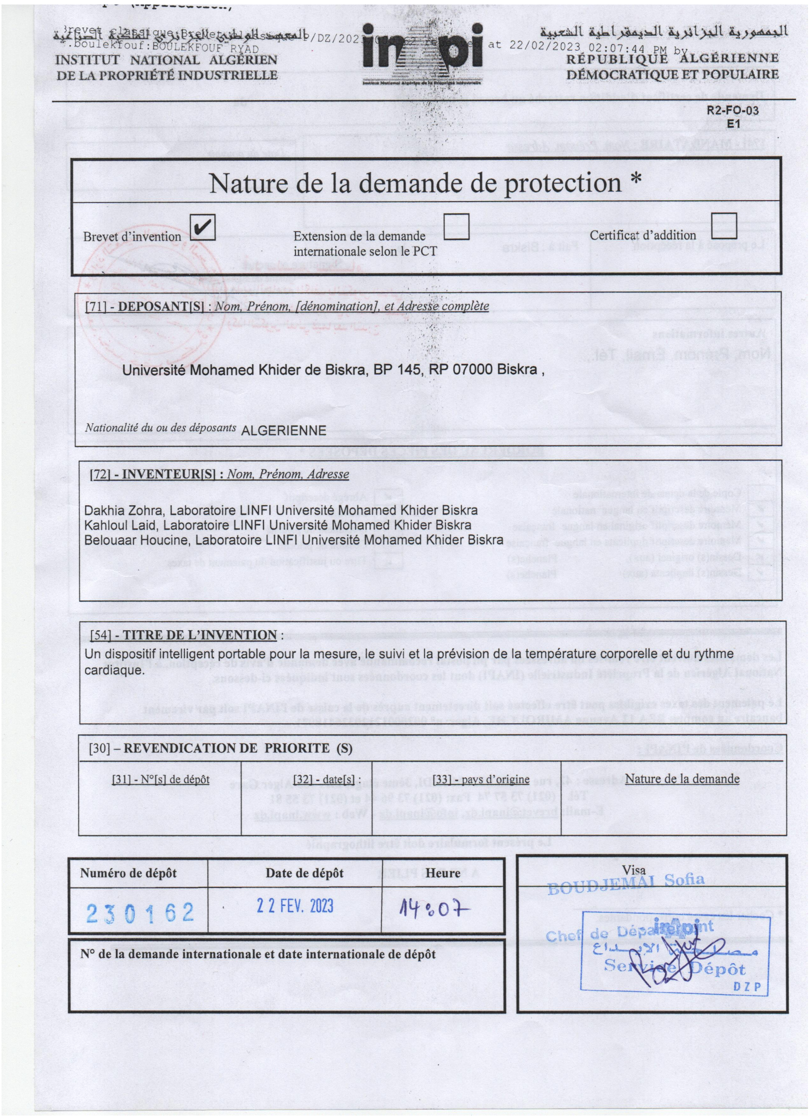
To my best friends, ***Nouioua Meriem***, ***Nouioua Soumia***, ***Bellounar Hanan***, ***Saadi Amani***, ***Saouli Nour*** and ***Atir Ikram***, who have stood by me through thick and thin, providing unwavering support, encouragement, and friendship. Your presence in my life has brought joy, laughter, and countless cherished memories.

To my comrade ***Salem Torkia***, ***Riguet Rym***, ***Djelloul Ines***, ***Medjeneh Sabrina***, ***Dendouga Manel***, ***Mechlek Samia***, ***Othmane Amani***, ***Baisi Fadia*** and ***Tengouri Nada*** , who have been my constant companions on this academic journey, sharing in the challenges and triumphs. Your friendship and camaraderie have made the journey more meaningful and enjoyable.

I dedicate this work to all of you, as a token of my deepest appreciation for your unwavering dedication, love, and support. Thank you for being an integral part of my life and for enriching my journey with your presence.

Patent

We are happy to report that our ground-breaking smart wearable device now has a patent number. This patent acknowledges the originality and distinctiveness of our gadget, which measures and forecasts vital indications including heartbeat and body temperature. Our gadget will revolutionize how vital signs are tracked and analyzed for better healthcare results, and we are eager to keep improving it.


INSTITUT NATIONAL ALGÉRIEN DE LA PROPRIÉTÉ INDUSTRIELLE

inapi

الجمهورية الجزائرية الديمقراطية الشعبية
RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

at 22/02/2023 02:07:44 PM by

R2-FO-03
E1

Nature de la demande de protection *

Brevet d'invention ✔ Extension de la demande internationale selon le PCT ☐ Certificat d'addition ☐

[71] - DEPOSANT[S] : *Nom, Prénom, [dénomination], et Adresse complète*

Université Mohamed Khider de Biskra, BP 145, RP 07000 Biskra ,

Nationalité du ou des déposants ALGERIENNE

[72] - INVENTEUR[S] : *Nom, Prénom, Adresse*

Dakhia Zohra, Laboratoire LINFI Université Mohamed Khider Biskra
Kahloul Laid, Laboratoire LINFI Université Mohamed Khider Biskra
Belouaar Houcine, Laboratoire LINFI Université Mohamed Khider Biskra

[54] - TITRE DE L'INVENTION :
Un dispositif intelligent portable pour la mesure, le suivi et la prévision de la température corporelle et du rythme cardiaque.

[30] – REVENDICATION DE PRIORITE (S)

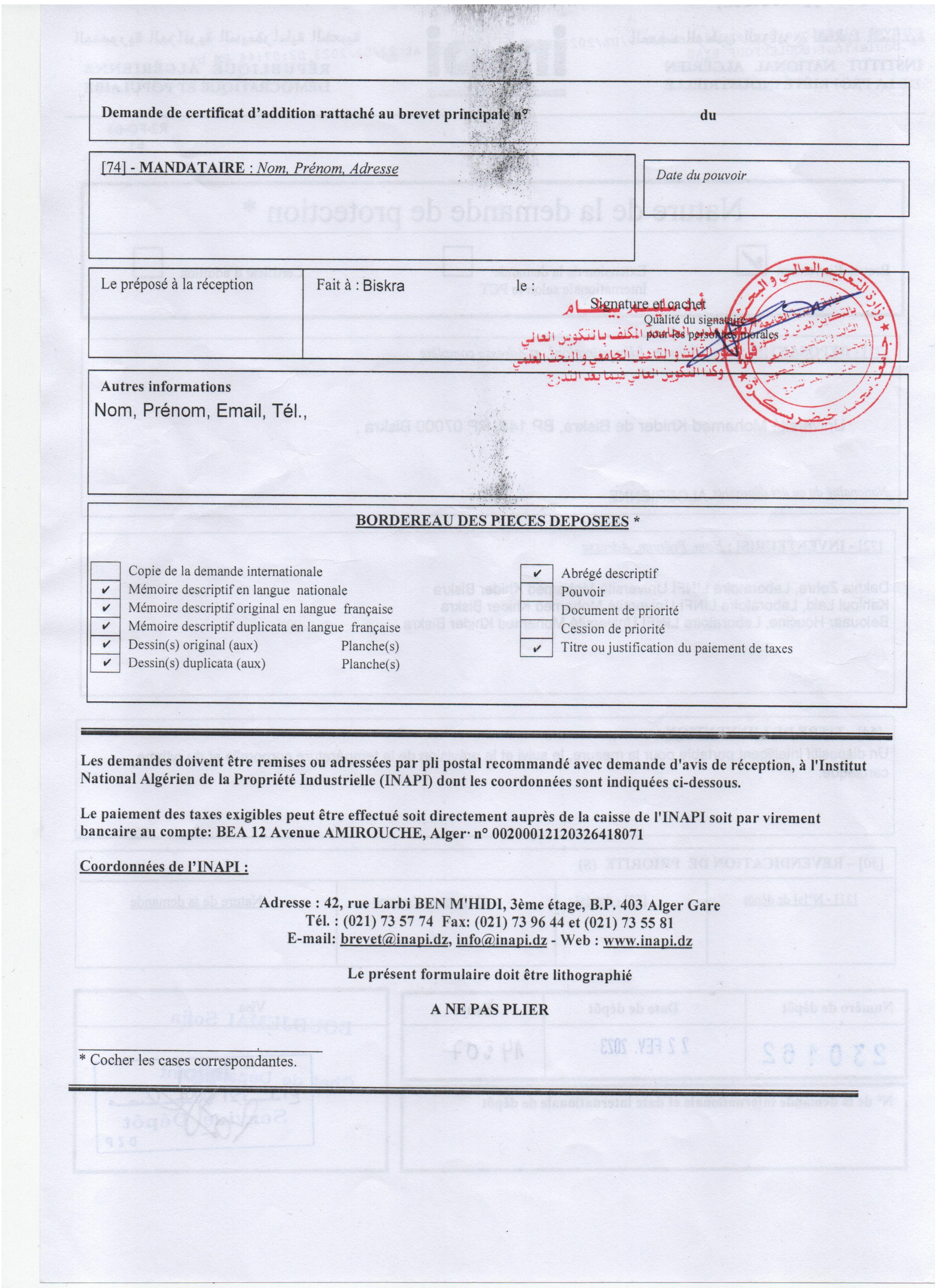
[31] - N°[s] de dépôt	[32] - date[s] :	[33] - pays d'origine	Nature de la demande

Numéro de dépôt	Date de dépôt	Heure
230162	22 FEV. 2023	14:07

N° de la demande internationale et date internationale de dépôt

Visa
BOUDJEMAI Sofia
Chef de Département Service Dépôt
inapi DZP

Figure 1: Patent


Demande de certificat d'addition rattaché au brevet principale n° du

[74] - MANDATAIRE : *Nom, Prénom, Adresse*

Date du pouvoir

Le préposé à la réception	Fait à : Biskra	le :
		Signature et cachet Qualité du signataire pour les personnes morales

Autres informations

Nom, Prénom, Email, Tél.,

BORDEREAU DES PIECES DEPOSEES *

☐	Copie de la demande internationale	✔	Abrégé descriptif
✔	Mémoire descriptif en langue nationale	☐	Pouvoir
✔	Mémoire descriptif original en langue française	☐	Document de priorité
✔	Mémoire descriptif duplicata en langue française	☐	Cession de priorité
✔	Dessin(s) original (aux) Planche(s)	✔	Titre ou justification du paiement de taxes
✔	Dessin(s) duplicata (aux) Planche(s)		

Les demandes doivent être remises ou adressées par pli postal recommandé avec demande d'avis de réception, à l'Institut National Algérien de la Propriété Industrielle (INAPI) dont les coordonnées sont indiquées ci-dessous.

Le paiement des taxes exigibles peut être effectué soit directement auprès de la caisse de l'INAPI soit par virement bancaire au compte: BEA 12 Avenue AMIROUCHE, Alger· n° 00200012120326418071

Coordonnées de l'INAPI :

Adresse : 42, rue Larbi BEN M'HIDI, 3ème étage, B.P. 403 Alger Gare
Tél. : (021) 73 57 74 Fax: (021) 73 96 44 et (021) 73 55 81
E-mail: brevet@inapi.dz, info@inapi.dz - Web : www.inapi.dz

Le présent formulaire doit être lithographié

A NE PAS PLIER

* Cocher les cases correspondantes.

Figure 2: Patent

Conferences

As part of my academic journey, I had the opportunity to present my project in :

- I participated in the esteemed First School on AI techniques for healthcare TIASM, an international conference dedicated to advancing research in AI techniques for healthcare on 22/06/2023 in biskra.
- I participated in the scientific week of Artificial intelligence at the service of society, an national conference dedicated to advancing research in AI techniques for society on 16,17,18,19 april 2023 in biskra.
- I participated in the competition for the best innovative project, a national competition on 15,16,17 November 2023 in biskra.

Submitted Papers

- Zohra Dakhia, Houcine Belouaar, Laid Kahloul, "Smart Wearable Health Surveillance System using IoMT and Artificial Intelligence Techniques", Submitted to The International Conference on Intelligent Data Science Technologies and Applications (IEEE) Kuwait, March. 2023.
- Zohra Dakhia, Laid Kahloul, Houcine Belouaar, "Enhancing Vital Sign Prediction Accuracy using Long Short-Term Memory (LSTM) Networks: A Comparative Analysis and Experimental Evaluation", International Conference on Data Science and Information Technology (IEEE) China, may. 2023.

Abstract

There are now more opportunities in the healthcare development thanks to the confluence of Internet of Things (IoT) and artificial intelligence (AI) technology. In this project, the development of a smart wearable system that makes use of IoT and AI to track and forecast vital signs, notably body temperature and heartbeat, is the main objective. The system uses the NodeMcu micro-controller and non-invasive sensors, like the LM35 temperature sensor and KY-039 heartbeat sensor, to record and send real-time physiological data. Regression models and Long-Short-Term-Memory (LSTM) networks are two advanced AI processing methods that are used to evaluate trends in the acquired data and forecast future vital sign readings. The monitoring data and predicted findings are conveniently accessible to healthcare practitioners through a Web-based application, enabling quick patient treatments. Because of the system's architecture, scalability and data analytics are made possible through safe storage and seamless integration with cloud-based systems. With the help of IoT and AI, this study shows how preemptive treatments and data-driven decision-making have the potential to improve patient outcomes and improve healthcare monitoring.

Key words: Forecasting, Artificial intelligence, Machine learning, Internet of things, Regression methods, Long-Short-Term-Memory, Wearable devices.

Résumé

Il y a maintenant plus d'opportunités dans le secteur de la santé grâce à la confluence de l'Internet des objets (IdO) et de l'intelligence artificielle (IA). Dans ce projet, le développement d'un système portable intelligent qui utilise l'IdO et l'IA pour suivre et prévoir les signes vitaux, notamment la température corporelle et le rythme cardiaque, est l'objectif principal. Le système utilise le micro-contrôleur NodeMcu et des capteurs non invasifs, comme le capteur de température LM35 et le capteur de battement de cœur KY-039, pour enregistrer et envoyer des données physiologiques en temps réel. Les modèles de régression et les réseaux de mémoire à court terme (MLTC) sont deux méthodes avancées de traitement de l'IA qui sont utilisées pour évaluer les tendances des données acquises et prévoir les futures lectures des signes vitaux. Les données de surveillance et les résultats prévus sont facilement accessibles aux professionnels de la santé grâce à une application Web qui permet de traiter rapidement les patients.

En raison de l'architecture du système, l'évolutivité et l'analyse des données sont rendues possibles grâce au stockage sécurisé et à l'intégration transparente avec les systèmes infonuagiques. Avec l'aide de l'IdO et de l'IA, cette étude montre comment les traitements préventifs et la prise de décisions fondées sur les données peuvent améliorer les résultats pour les patients et le suivi des soins de santé.

Mots clés: Prévision, Intelligence artificielle, Apprentissage automatique, Internet des objets, Méthodes de régression, Mémoire à court terme, Dispositifs portables.

ملخص

هناك الآن مزيد من الفرص في تطور الرعاية الصحية بفضل تلاقي تكنولوجيا إنترنت الأشياء والذكاء الاصطناعي. في هذا المشروع، يكمن الهدف الرئيسي في تطوير نظام قابل للارتداء ذكي يستخدم إنترنت الأشياء والذكاء الاصطناعي لتتبع وتوقع علامات حيوية، بصفة خاصة درجة حرارة الجسم ومعدل ضربات القلب. يستخدم النظام متحكم NodeMcu ومستشعرات غير جراحية، مثل مستشعر درجة الحرارة LM35 ومستشعر نبضات القلب KY-039 لتسجيل وإرسال البيانات الفسيولوجية في الوقت الحقيقي. تعتبر نماذج الانحدار وشبكات الذاكرة القصيرة والطويلة المدى طريقتين متقدمتين للمعالجة يتم استخدامهما لتقييم الاتجاهات في البيانات المكتسبة وتوقع قراءات العلامات الحيوية المستقبلية. البيانات المراقبة والنتائج المتوقعة متاحة بسهولة لممارسي الرعاية الصحية من خلال تطبيق ويب، مما يتيح معالجة سريعة للمرضى.
بفضل بنية النظام، يتم تمكين قابلية التوسعة وتحليل البيانات من خلال التخزين الآمن والتكامل السلس مع أنظمة قائمة على السحابة. بمساعدة إنترنت الأشياء و الذكاء الاصطناعي، توضح هذه الدراسة كيف أن العلاجات الاحترازية واتخاذ القرارات القائمة على البيانات لديها القدرة على تحسين نتائج المرضى وتحسين رصد الرعاية الصحية.

الكلمات المفتاحية:
التنبؤ، الذكاء الاصطناعي، التعلم الآلي، إنترنت الأشياء، طرق الانحدار، الذاكرة طويلة المدى، الأجهزة القابلة للارتداء.

CHAPTER 01

GENERAL INTRODUCTION

Chapter 1

General introduction

1.1 Introduction

Under the modern world, people's health and wellbeing are continuously under danger from a wide range of illnesses and medical issues [73]. The requirement for efficient monitoring and management of health hazards has been further highlighted by the introduction of contagious diseases like the COVID-19 pandemic. Furthermore, conditions like elevated body temperatures and unexpected cardiac crises pose serious risks to people's general health.

Particularly in light of the COVID-19 pandemic, there is an increasing demand for trustworthy and effective monitoring systems in the field of infectious disease services. These systems are essential in tracking patient's vital signs over time, such as their heartbeat and body temperature, which enables early diagnosis of anomalies and rapid medical measures [31]. These systems improve patient care, reduce the danger of cross-infection, and maximize the use of healthcare resources by supplying real-time data and enabling remote monitoring [131]. For enhancing patient outcomes, assuring prompt treatments, and successfully managing the difficulties caused by infectious diseases, improved monitoring systems must be implemented in infectious disease services [41].

High body temperatures have a number of negative health consequences, including the risk of infectious infections as well as developmental defects in children [46]. Infants and young children who have elevated body temperatures are more likely to experience disorders like febrile seizures, which are convulsions brought on by a high fever [63]. Even though these seizures are frequently brief and they can nevertheless have lasting effects.

Additionally, unexpected cardiac crises among athletes on the field emphasize the crucial significance of keeping an eye on heart health and acting quickly on any signals of concern [53]. Athletes who engage in vigorous exercise are more likely to have abrupt cardiac events like cardiac arrest or heart failure [113]. Early detection of warning indicators and prompt medical attention are essential for maximizing survival chances and reducing potential long-term consequences [122].

We intend to equip everyone, especially children and athletes, with a proactive tool for monitoring their vital signs [105], including body temperature and heartbeat, by addressing these health concerns through the development of a smart wearable gadget and a thorough health surveillance system. We aim to improve the safety, well-being, and overall health outcomes of people in varied circumstances by early abnormality identification and prompt notifications.

1.2 Problematic

The shortcomings of conventional techniques in accurately measuring and managing people's health give rise to the need for a health monitoring system that tracks and forecasts both body temperature and heartbeat. Manual measures and sporadic monitoring may not give a thorough picture of health issues, resulting in postponed interventions and unsatisfactory results. Traditional methods fall short of accurately capturing real-time changes in situations where continuous monitoring is essential, such as during physical activity or for people with chronic diseases. Therefore, a thorough health surveillance system that combines continuous monitoring and predictive capabilities is necessary to enable real-time monitoring, early abnormality detection, and proactive healthcare management, ultimately enhancing overall health outcomes.

1.3 The work's purpose

Our proposed solution addresses the need for an innovative health surveillance system by introducing a smart wearable device and a connected Website. This system empowers doctors to predict the future conditions of their supervised patients and take necessary actions to ensure their well-being. The wearable device, equipped with sensors, collects vital data and transmits it to the cloud, enabling doctors to monitor multiple patients simultaneously. By leveraging artificial intelligence techniques, the system can forecast both body temperature and heartbeat signals, providing valuable insights into the patient's future state. This proactive approach enables doctors to take preventive measures and intervene early, ultimately saving lives and improving patient outcomes.

1.4 Research Requirements

Before we can accomplish this research, we need to understand:

- Artificial Intelligence (AI) Techniques: Learn in-depth AI principles and methods, especially those related to machine learning, deep learning, data analysis, and predictive modeling. This entails researching several AI techniques, including regression, classification, and time series analysis, in order to accurately forecast future health problems based on the gathered data.

- Internet of Things (IoT) Technologies: Learn about cloud computing, data connection protocols, sensor integration, and other IoT ideas and technologies. Investigate various IoT frameworks and platforms that enable smooth data collection, transmission, and storage from the wearable system to the cloud-based system.

- Sensor Technology and Selection: Examine and compare several sensor technologies that can be used to measure heartbeat and body temperature. To ensure accurate and dependable data gathering, be aware of the features, constraints, and calibration needs of different sensors.
- Plan a strategy to accomplish our major objective.

1.5 Structure of the dissertation

The dissertation is divided into four chapters, each of which focuses on a different component of the research. An summary of the chapters is given below:

- **Chapter 2 : Background** dives into the research's history and offers a thorough analysis of the pertinent works of literature and thoughts on the subject. It presents the underlying theories and technological principles that support the research, setting a strong foundation for the following chapters.
- **Chapter 3 : Design and implementation** The process of system design is the main topic. It outlines the process followed in developing the health surveillance system, including the decision-making process used in choosing hardware and software components, the incorporation of IoT technologies, and the tools and methods applied in the implementation process. The chapter gives a thorough overview of the system architecture and the justification for design choices.
- **Chapter 4 : Experimentation and results** Provides the findings from the research. The effectiveness of the system as a whole, the performance of the predictive models, and the precision of the data analysis methods are all included in the evaluation of the developed system. Additionally, the chapter analyzes any remarkable results or insights and compares the findings to relevant studies.
- **Chapter 5 : Conclusion and perspective** It provides an overview of the research's major conclusions and contributions. The study's implications and potential effects on the field of healthcare surveillance are covered in this chapter. It also suggests potential paths for additional study and advancement in this field.

CHAPTER 02

BACKGROUND

Chapter 2

Background

2.1 Introduction

The fields of artificial intelligence (AI) and the internet of things (IoT) are both expanding quickly and have enormous potential for improving many facets of human activity, including healthcare [119]. AI has already been used successfully in the healthcare industry to enhance diagnosis, anticipate health issues, and offer individualized treatment strategies [125]. By enabling remote monitoring and the gathering of real-time data, IoT has also been very successful in the medical industry. Further enhancing healthcare outcomes, lowering costs, and boosting efficiency are all possible with the integration of AI and IoT [23].

This chapter attempts to give readers background information about our project's main focus, which is the prediction of body temperature and heartbeat. We'll talk about AI and IoT techniques. We'll also speak of the benefits and drawbacks of the various measurement techniques now in use. after that we discuss the idea of sensors and the Internet of Things (IoT), a system that enables intelligent object-to-object communication. We'll also go over some of the advantages of IoT and explain how it may be used in the healthcare sector. Finally, we will quickly discuss microcontrollers and their function in our project.

2.2 Healthcare

2.2.1 Definition

After doing a lot of research, I discovered numerous definitions and meanings for the term "healthcare." Various sources have provided the following basic definitions.

The Cambridge dictionary definition is :
"Healthcare is the collection of medical services that a nation or organization offers to treat the sick both physically and mentally" [10].

The just-in-time (JIT) definition created by nurses is :
"Providing the Best Patient Experience Possible by Treating the Right Patients in the Right Ways" [149].

2.2.2 Types of health care

Three main categories of healthcare exist:

Primary Care

For the majority of people, primary care represents their initial point of contact. GPs, dentists, and neighborhood pharmacists are a few professionals who work in primary care. General Practitioner is referred to as GP.

General practitioners are also referred to as family doctors or primary care doctors.

Primary care providers include nurse practitioners and some other categories of nurses [19].

Secondary care

Patients may see a secondary care professional after seeing a primary care physician. This person might be an expert.

For instance, you might first visit your GP if you experience minor chest pains. The GP might then recommend a cardiologist to you. Working in secondary care is the cardiologist.

Secondary care specialists include psychiatrists, neurologists, and endocrinologists.

The patient is frequently referred to a secondary care professional by the primary care provider. In a hospital, secondary care cases predominate over primary care cases [55].

Tertiary care

Assume the cardiologist determines that the patient who is experiencing minor chest pains requires heart surgery. A hospital will be the setting for the procedure. The patient will stay in a hospital. The procedure and hospital stay are both considered to be tertiary care [77].

2.3 Artificial Intelligence

2.3.1 Definition

Artificial intelligence (AI) is defined as a branch of science and engineering devoted to the computational analysis of what is commonly referred to as intelligent behavior and the development of artifacts exhibiting such behavior [98]. Also is an umbrella term that suggests using a computer to simulate intelligent behavior with little to no human involvement. The creation of robots is widely acknowledged as the beginning of artificial intelligence [72].

Artificial intelligence is viewed with skepticism by many. They don't comprehend how computers can "learn" and make wise choices. Nevertheless, anyone can understand how AI works.

The two most crucial ideas that enable AI are machine learning and deep learning. Despite the frequent confusion between the two terms, they refer to two fundamentally distinct techniques, each with their own domains of application.

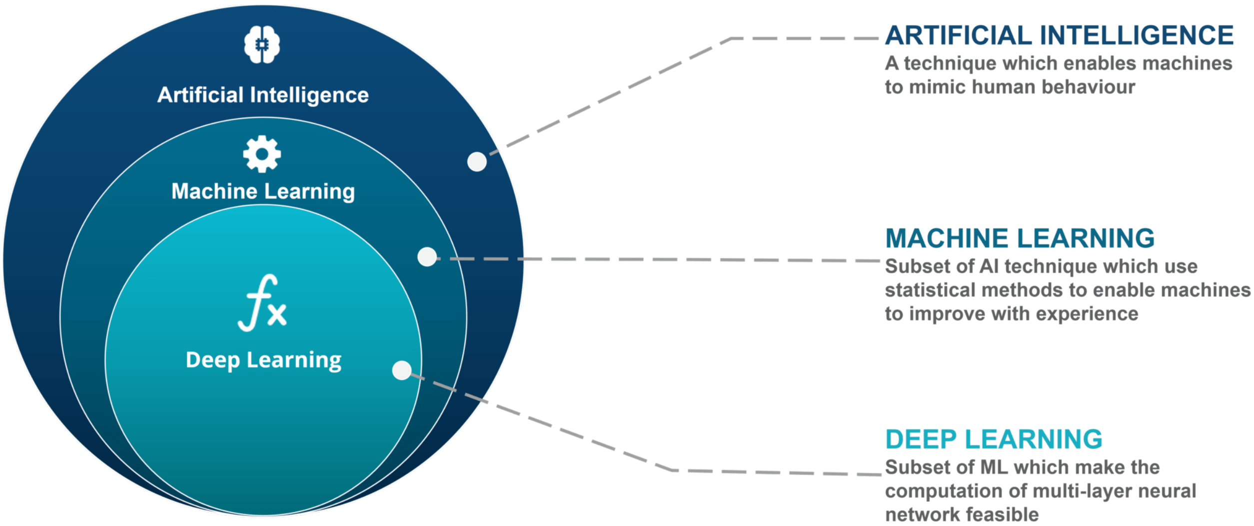


Figure 2.1: Artificial Intelligence [5]

2.3.2 Machine learning

Definition

Machine learning (ML) is typically referred to as the most well-liked newest technologies in the fourth industrial revolution and gives systems the ability to learn and improve from experience automatically without being specifically programmed [158]. teaches computers how to handle data more effectively [110].

The different stages of machine learning

Predictive marketing, industrial maintenance, facial and voice recognition... Machine Learning applications are now becoming more common in organizations. At the crossroads of statistics, artificial intelligence and computing, this technology involves programming algorithms to allow computers to learn for themselves. The goal of a machine learning project to develop effective learning models from large data sets (datasets). To achieve this, a specific process is recommended. So here are the steps:

- Identify your company's needs and objectives

  Before embarking on building a viable learning model, it remains essential to know why the machine learning solution must be implemented. Machine learning projects are costly and time-consuming. By setting quantifiable objectives, it is possible, on the one hand, to establish a framework and, on the other hand, to judge whether the project is a success or not [52]. At this stage, it is a matter of knowing precisely the business problem to be solved: once the purpose of the project is determined, you are able to indicate what types of data to collect, what results (output data) to expect, and even the type of model to use (supervised, unsupervised, reinforcing learning, etc.) [88].

- Collect the data needed

  The quality and quantity of the data have a direct impact on the efficiency of the resulting model [80]. To develop their ability to accumulate knowledge and make decisions independently, machines need to consume a large amount of information: the more numerous and reliable it is, the more accurate and tailored the result will be to the needs of the company [173]. It is therefore essential to gather data according to the objectives defined in the previous step. Do you collect from multiple data sources? Integrate them by merging different databases [181].

- Prepare data

  A successful learning model relies above all on quality data: it is therefore necessary to pre-process the data collected in order to extract its full potential [138]. Data incorrectly annotated, data not available, duplicates, inconsistent or superfluous information... Data integration can create a number of complications within your data warehouse. This third step aims to clean up and standardize (make comparable) raw data, and even improve it with other sources. The goal? To make this type of data consistent and usable by algorithms [35]. If you are handling confidential data, it is at this point in the process that you should consider anonymizing or pseudonymizing it to ensure compliance with the GDPR [24].

- Determine the right model

  The data is now ready to be used. The next phase: choose the right algorithm to address the initial problem. k-means, random forest, decision tree... There are different models developed by data scientists to address different problems and levels of complexity [163]. Beyond opting for the right model, it is necessary to correctly program the algorithms to obtain accurate results and relevant forecasts: we must then play on the hyperparameters, adjustment variables to control the model drive process [185].

- Train and evaluate the model

  Among all the steps of machine learning, the training test remains the most characteristic phase of machine learning. Fed with data, the model is trained over time to gradually improve its ability to react to a given situation, solve a complex problem or perform a task [67]. For this learning phase, it is recommended to use training data (also called "training set"). All the information collected is often too cumbersome and resource-intensive: it is then enough to select a part of the dataset (sampling) in order to train the model more effectively and to perfect its predictions

[162]. Just make sure you choose a sample that is representative of your data or you risk creating a bias.

- Test and deploy the model

  Time for practice: this last stage of machine learning tends to confront the model with the reality of the field [150]. In this test phase, we use the other part of the data, the test dataset [93]. This subset of information refines the model with scenarios or data that the computer has not yet experienced in the training phase. This allows you to evaluate the model's performance in the context of your business [161].

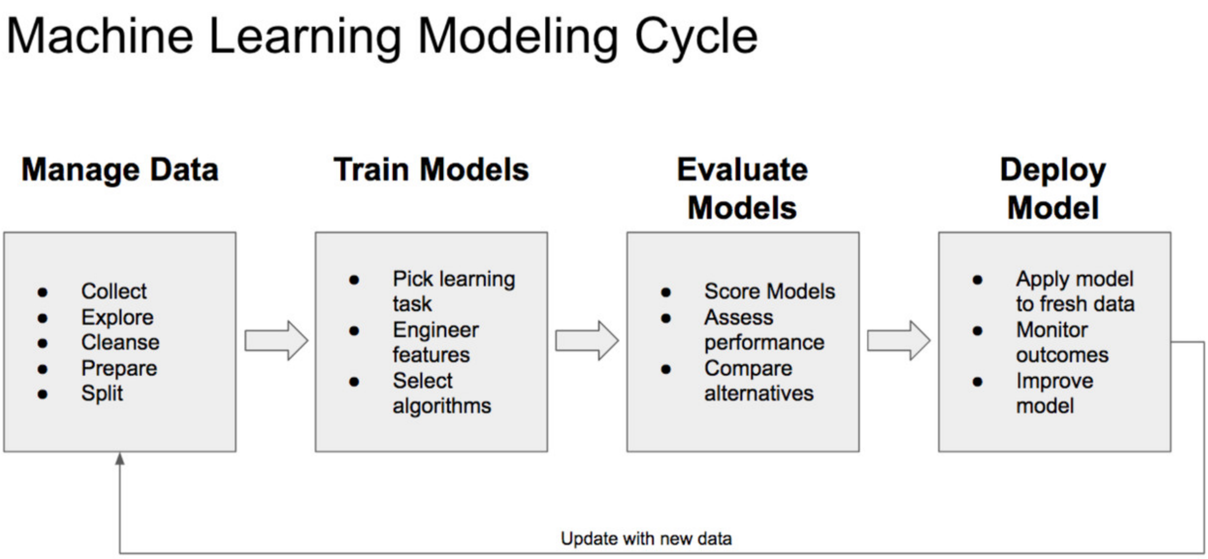


Figure 2.2: Machine Learning (ML) Modeling Cycle [102]

Types of Machine Learning

Network computing units self-organize using a learning algorithm, which is adaptive, to achieve the desired behavior. In machine learning, predictions are made using samples of target behaviors or historical data collected during data observations. Algorithms for machine learning are categorized as [89] :

- Supervised learning : where a function is built by the algorithm to map inputs to desired results [151]. The learner then contrasts its actual response with the desired response and modifies its internal memory to increase the likelihood that the next time it receives the same input, it will respond appropriately [27]. There are two main types of supervised learning: regression and classification.

  – In regression, the goal is to learn a continuous function that can be used to predict a real-valued output for any new input [139].

- In classification, the goal is to learn a discrete function that can be used to assign each new input to one of a set of predetermined classes [56].

Some of the most popular supervised learning algorithms include:

- Linear Regression
- Logistic Regression
- Decision Trees
- Support Vector Machines

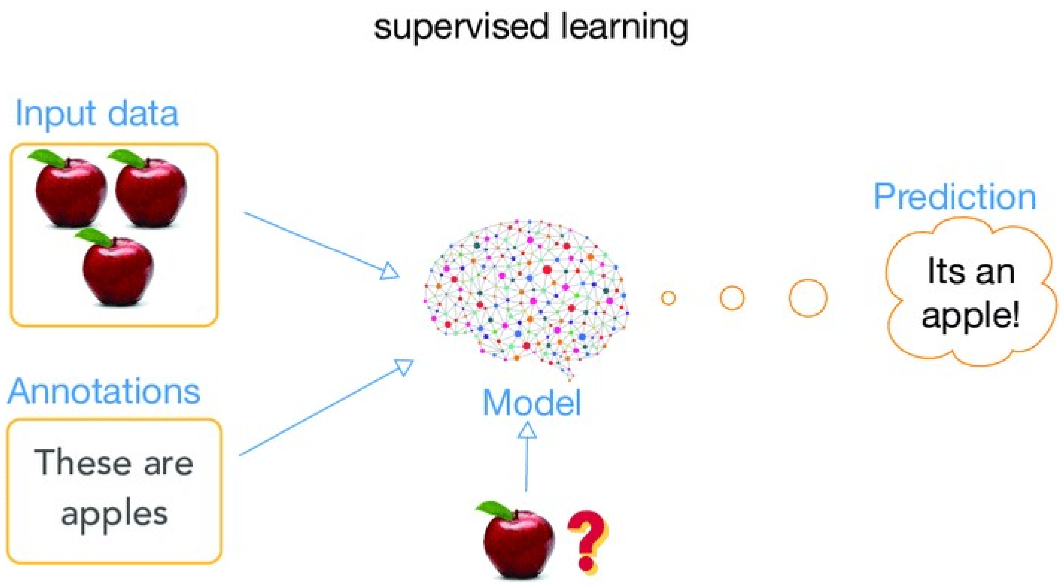


Figure 2.3: Supervised learning [70]

- Unsupervised learning : which models a set of inputs using (clustering, dimensionality reduction, recommender systems, self organizing learning, etc.). There are no intended results (no any labeled examples). The environment offers no feedback to the learner [51].

  There are many different types of unsupervised learning, but some of the most commonly used are Clustering, dimensionality reduction, and association mining.

  - Clustering: In clustering data is grouped together based on similarity [66]. This can be done in a number of ways, but the most common approach is to use algorithms that group data points together that are close together in terms of some distance metric (e.g. Euclidean distance) [130]. Clustering is often used to group data points that are not linearly separable, and as such it can be seen as a way of finding structure in data. Clustering algorithms are also often used as a pre-processing step for other machine learning tasks such as classification and regression [140].
  - Dimensionality Reduction: It is the process of reducing the number of features in a data set. This can be done for a variety of reasons, including simplifying

the data set, reducing the storage requirements, or increasing the computational efficiency [167]. There are two main types of dimensionality reduction: supervised and unsupervised [64]. Supervised methods involve training a model on a labelled data set, while unsupervised methods do not require labels. One common unsupervised method is principal component analysis (PCA), which finds the directions that maximize the variance in the data [123]. Dimensionality reduction can be an effective way to improve the performance of machine learning algorithms, and it is an important tool in data science [58].

– Association mining: Association rules are typically represented as if-then statements, where the if part represents a condition and the then part represents the outcome [76]. For example, an association rule might state that if a customer buys a product A, then they are also likely to buy product B. Association rules can be used to predict future events or to make recommendations [86]. For example, if a customer buys product A, the system might recommend product B. Association rule learning is a powerful tool for making sense of large datasets and can be used for a variety of applications [159].

There are a variety of unsupervised machine learning algorithms available, each with its own advantages and disadvantages. Some of the most well-known algorithms include:

– k-means clustering
– Gaussian mixture models
– Hierarchical clustering
– Self-organizing maps
– Support vector machines
– Neural networks
– Dimensionality reduction
– Anomaly detection
– Generative adversarial networks

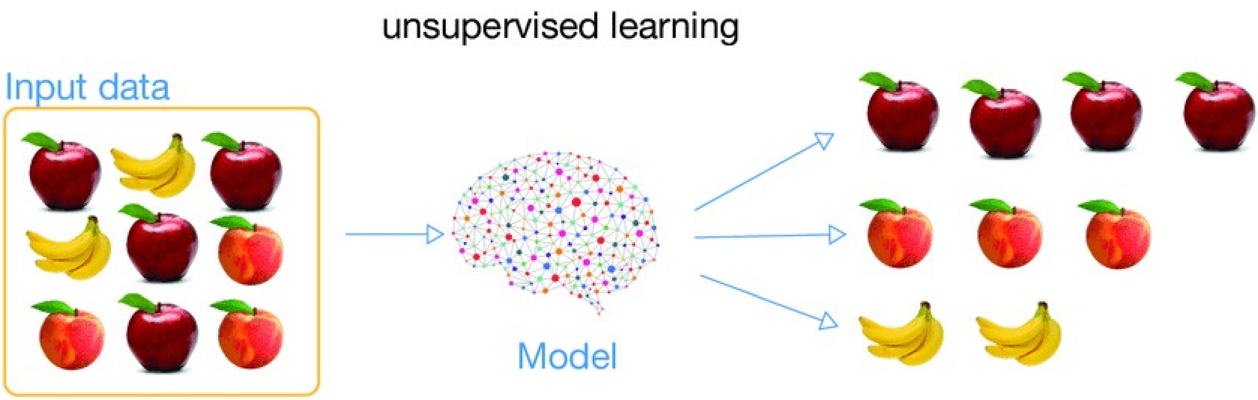


Figure 2.4: Unsupervised learning [164]

- Semi-supervised learning : where the algorithm generates examples with and without labels, a special function [168]

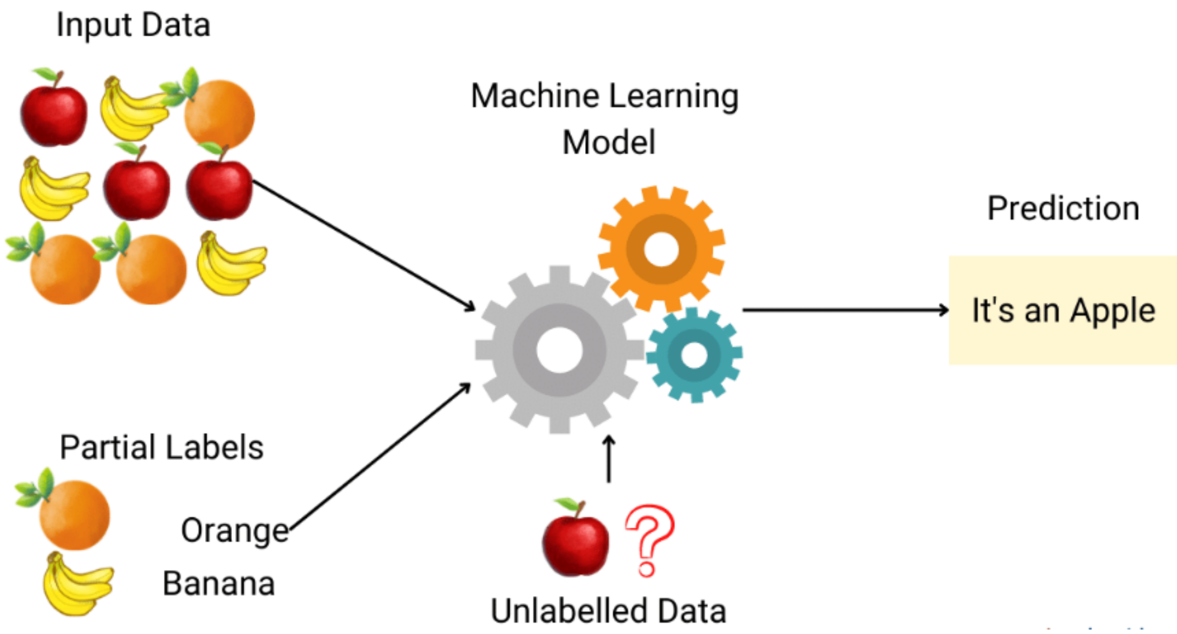


Figure 2.5: Semi-supervised learning [177]

- Reinforcement learning : is interaction with the environment while learning [124]. Feedback is given to the learner regarding the suitability of their response [87]. Types of reinforcement learning There are three main types of reinforcement learning: positive reinforcement, negative reinforcement, and punishment [14].
  - Positive reinforcement occurs when a behaviour is rewarded in order to increase the likelihood of that behaviour being repeated [103].
  - Negative reinforcement occurs when a behaviour is rewarded in order to decrease the likelihood of that behaviour being repeated [40].
  - Punishment occurs when a behaviour is punished in order to decrease the likelihood of that behaviour being repeated [187].

  The most popular Reinforcement Learning algorithms:

  - Q-learning
  - SARSA
  - TD learning

  These algorithms are used to solve problems such as finding the shortest path between two points or learning how to play a game such as chess. More about reinforcement Learning algorithms on machine learning algorithms post.

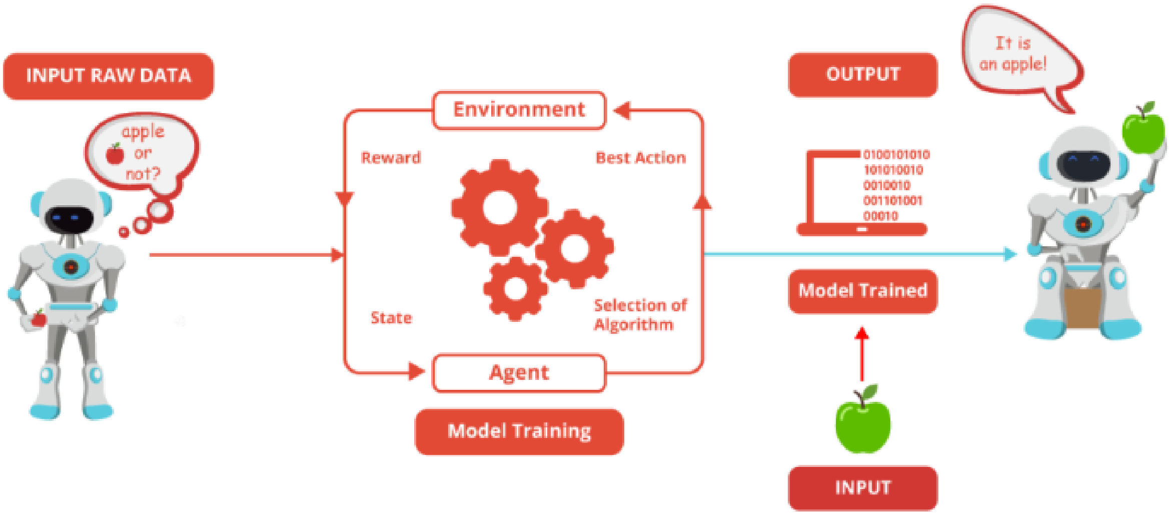


Figure 2.6: Reinforcement learning [15]

- Transfer learning : is a method that makes use of previously learned information and applies it to new information to enhance the performance of a machine learning model [71]. A model can be transferred to a new dataset and adjusted to perform better, for instance, if it was trained on data from a similar but different dataset. Transfer learning is frequently employed when there is not enough data to create a model from scratch [82]. Additionally, it can be used to prevent overfitting, which occurs when a model too closely resembles the training data and struggles to generalize to new data [34]. Transfer learning has the potential to significantly enhance the performance of machine learning models when used properly [47].

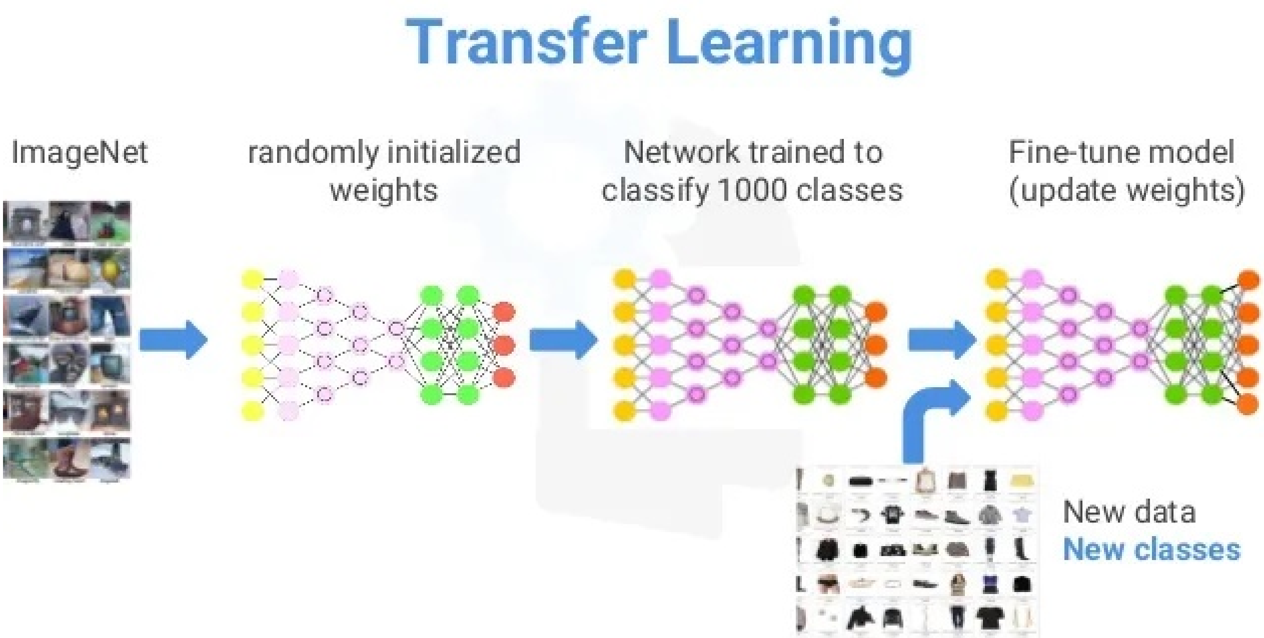


Figure 2.7: Transfer learning [18]

2.3.3 Deep learning

Deep Learning is a subset of machine learning techniques built on the acquisition of data models [99]. It is designed to help computers learn and recognize meaningful representations or patterns from massive amounts of data. It draws inspiration from the structure and operation of the human brain [175]. A fascinating new trend in machine learning has recently emerged thanks to deep learning. The classical neural network (NN) literature serves as a solid theoretical foundation for deep learning [153].Deep learning algorithms are particularly effective in tasks like image recognition, natural language processing, speech recognition, and many others because they can automatically find intricate patterns, hierarchical representations, and complex relationships within the data [65].

Deep learning can be used to solve regression situations where the objective is to predict a continuous value or set of continuous values. Deep learning regression problems entail teaching neural networks the relationship between input features and desired outcomes so they can predict the future based on unobserved data. Many regression techniques are frequently employed in deep learning. Here are a few illustrations:

- **Feedforward Neural Networks (FNNs):** The most basic type of neural network is the FNN, which consists of an input layer, one or more hidden layers, and an output layer [133]. The output layer commonly employs a linear activation function to generate continuous predictions in regression problems [96].
- **Convolutional Neural Networks (CNNs):** CNNs can be used to solve regression issues but are mostly employed for image-related jobs [37]. Regression predictions can be made using CNNs using convolutional layers, pooling layers, and fully connected layers, which can be used to learn hierarchical representations from input data [148].
- **Recurrent Neural Networks (RNNs):** RNNs are made to manage sequential data by preserving hidden states that encapsulate temporal dependencies [114]. RNNs perform sequence processing and produce predictions during regression tasks. Regression tasks frequently employ variants to capture longer-term dependencies, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) [90].
- **Deep Belief Networks (DBNs):** DBNs are made up of stacked layers of Autoencoders or Restricted Boltzmann Machines (RBMs) [188]. They are able to recognize intricate patterns in regression tests and can learn hierarchical data representations [61].
- **Transformer Networks:** Transformers are increasingly used in jobs involving natural language processing, but they can also be applied to regression. Because they are based on a self-attention process, transformers may recognize dependencies between input parts [183]. They are adaptable for regression issues and have demonstrated success in sequence-to-sequence activities [48].
- **Variational Autoencoders (VAEs):** VAEs are generative models with regression capabilities. They are able to create fresh samples and learn latent representations of the data [38]. VAEs can produce continuous forecasts and capture the underlying data distribution in regression tasks [39].

2.3.4 The differences between the Machine Learning and the Deep Learning

The algorithms are what differentiate the Machine Learning and Deep Learning approaches [92]. A variety of unique algorithms were proposed by machine learning. Examples include SVM, linear regression, decision trees, random forests, etc. All of these methods don't need as much data as deep learning does [106]. The Neural Network is the only algorithm that Deep Learning provides (NN). Although there are many different NNs, the fundamental idea is the same [97]. The distinction between the ML and the DL is shown in 2.8.

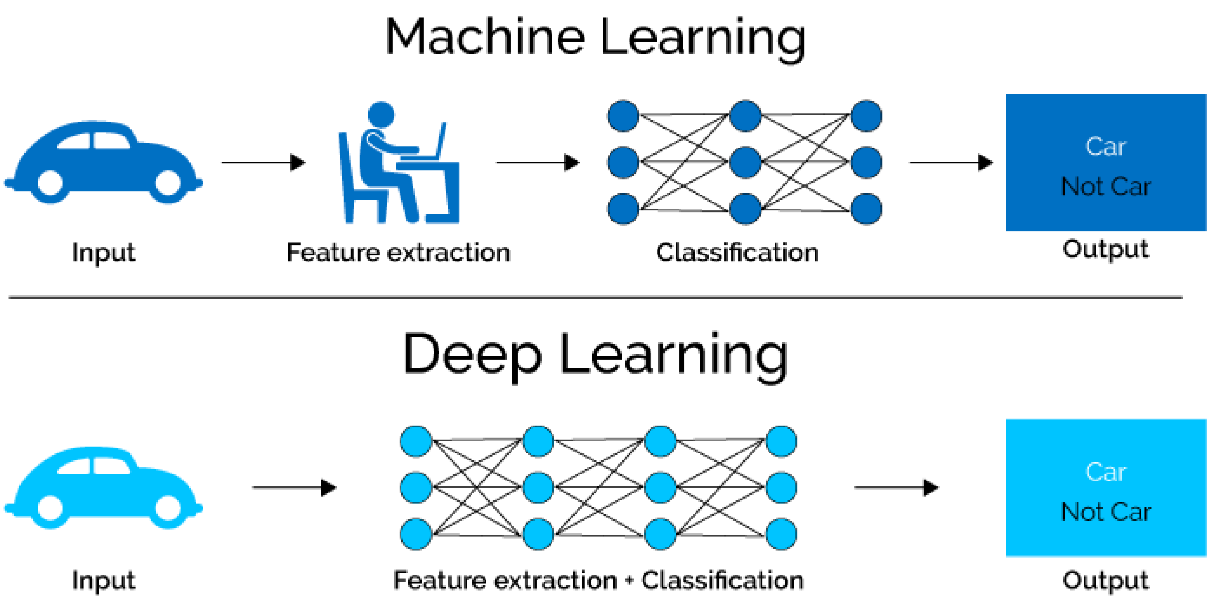


Figure 2.8: Machine Learning Vs Deep Learning[13]

2.3.5 AI in Healthcare

AI in healthcare is a catch-all phrase used to refer to the use of machine learning (ML) algorithms and other cognitive technologies in healthcare settings. AI can be defined simply as the ability of computers and other machines to mimic human cognition and to learn, think, and make decisions or take actions [42]. Therefore, AI in healthcare refers to the application of machines to analyze and take action on medical data, frequently with the aim of forecasting a specific outcome [60].

The use of ML and other cognitive disciplines for medical diagnosis is a significant application of AI in healthcare. AI can assist doctors and other healthcare professionals in providing more precise diagnoses and treatment plans by using patient data and other information. By analyzing big data to create better preventive care recommendations for patients, AI can also assist in making healthcare more proactive and predictive.

Some application of AI in healthcare

- Managing Medical Records and Other Data
- Treatment Design
- Virtual Nurses
- Digital Consultation
- Precision Medicine

Artificial intelligence and early body temperature and heartbeat detection

The treatment prognosis in many critical illness cases depends on how quickly diseases are identified. The ideal scenario is for symptoms to appear before long enough for us to realize there is a problem and have plenty of time to seek medical attention [166]. The early warning signs of some diseases, however, are absent, and we frequently hear of instances where these signals arrive too late [184].

Furthermore, it's possible that not many people frequently visit medical specialists. For some people, the wait time to see a doctor may also be a problem [132]. Here, AI algorithms can aid in performing initial screenings to detect minute details that might indicate underlying problems and then referring them to the experts [174]. Artificial intelligence has made a contribution by using two of its technological advancements, prediction and forecasting, to early detect body temperature and heart beat.

2.4 Internet of Things

2.4.1 Definition

The term "Internet of Things" generally describes situations where network connectivity and computing capability are extended to objects, sensors, and common household items that are not typically thought of as computers, enabling these devices to generate, exchange, and consume data with little to no human intervention [154]. The Internet of Things (IoT) is made up of things, sensor devices, communication infrastructure, computational and processing units that could be stored in the cloud, and systems for making decisions and triggering actions[20]. Over the internet, the sensors send the computational and processing unit object-specific data. Smart services can be designed using a variety of different sensors[127]. However, there isn't a solitary, inclusive definition.

2.4.2 IoT architecture and technologies

The four key layers that make up the IoT architecture define all of the functionalities of IoT systems. These layers include the network layer, the middleware layer, the application layer, and the business layer [95]. The perception layer, which includes physical devices like sensors, RFID chips, barcodes, etc., is the foundation of the Internet of Things architecture. and other tangible items linked to an IoT network. These devices gather data and transmit it to the network layer [49]. The information is transmitted from the perception layer to the information processing system using the network layer as a transmission medium. This information transmission may use any wired or wireless technology, including 3G/4G, Wi-Fi, Bluetooth, and others. Middleware layer is the layer below that. This layer's primary responsibility is to process the data obtained from the network layer and make decisions based on the outcomes of ubiquitous computing [62]. The application layer uses this processed data after that for worldwide device management. A business layer that controls the overall Internet of Things system, its applications, and services is present on top of the architecture. The business layer further uses this knowledge to plan future goals and strategies by visualizing the data and statistics it receives from the application layer [100].

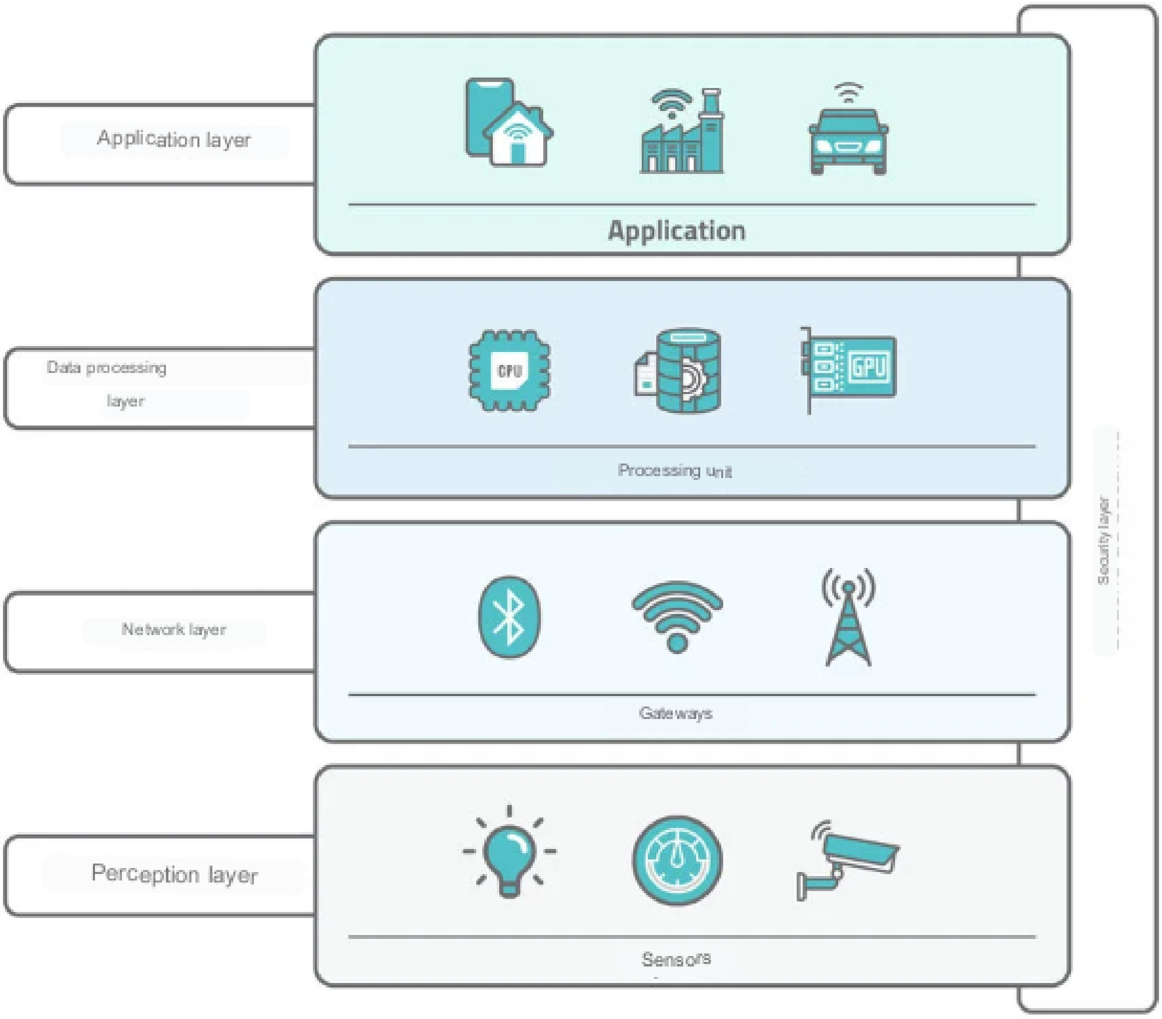


Figure 2.9: Four-layer IoT architecture [81]

2.4.3 IoT in healthcare

Applications for medical differentiation and diagnosis

- Fonseca et al. [118] offered to work on creating intelligent living environments for chronic multimorbid patients receiving home healthcare. The main obstacle to creating adequate indications of medical knowledge, or ontologies, in the healthcare industry is the "Knowledge Acquisition Bottleneck." In this study, machine learning that is supervised or predictive is employed. In this method, input-to-output mappings are a component of the systems, and they are given a training set, which is a set of input-output pairs with labels. Since it improves the quality of life for patients with comorbidities while lowering costs, this method helps to usher in a new era of caregiving amenities. The present model's effectiveness and functionality cannot be verified by data, which is a problem.

- Sandstrom et al. [157] tated that smartphone users could share their sensor data with others in the cloud, like that provided by IBM for the Internet of Things and Emotion Sense. Three areas are greatly impacted by smartphones: ambulatory assessment, behavior monitoring, understanding, and outcome prediction. A correlation between smartphone sensor data and individual health can be created using deep learning techniques. Deep-stacked Auto Encoders (SAE) first divide the data into sections in order to categorize the softmax layer and elicit its features. The relationship between divided sensor data and health is then created quantitatively. Finally, a simulation is developed to validate the performance of the suggested technique. SAE holds a plain format, small computation (without forcing GPU), and excellent performance, which are advantages of this approach (regularizing weight and sparsity). More sensor data genres from wearables and smartphones should be examined in the future, though.

2.4.4 IoT and early body temperature and heartbeat detection

Nowadays, working as a caregiver in a hospital is considered as one of the most difficult tasks as it requires a constant vigilance especially in infectious diseases services, Approaching to patients is very perilous and harmful, taking as best example, COVID-19 crisis where permanent observation is crucial [115].
So the sensors of iot and all other composant help doctors to track the state of a patient at distance espcialy in those two factors doctor can track the body temperature of considred patient and also same for the heartbeat [169].

2.4.5 IoT as a software

To give computers and applications real-time data, IoT software manages data gathering and communication [101]. Real-time analytics and artificial intelligence techniques are used to power complete IoT solutions, which process and present data to consumers as useful information.

Arduino

Open-source software called Arduino IDE enables users to write and download code in a real-time working environment. This code will then be kept in the cloud, so those seeking more redundancy frequently utilize it. Any Arduino software board is entirely compatible with the system [4].

Arduino Software (IDE)
Coding and uploading to the board without an Internet connection is simple using the offline IDE.

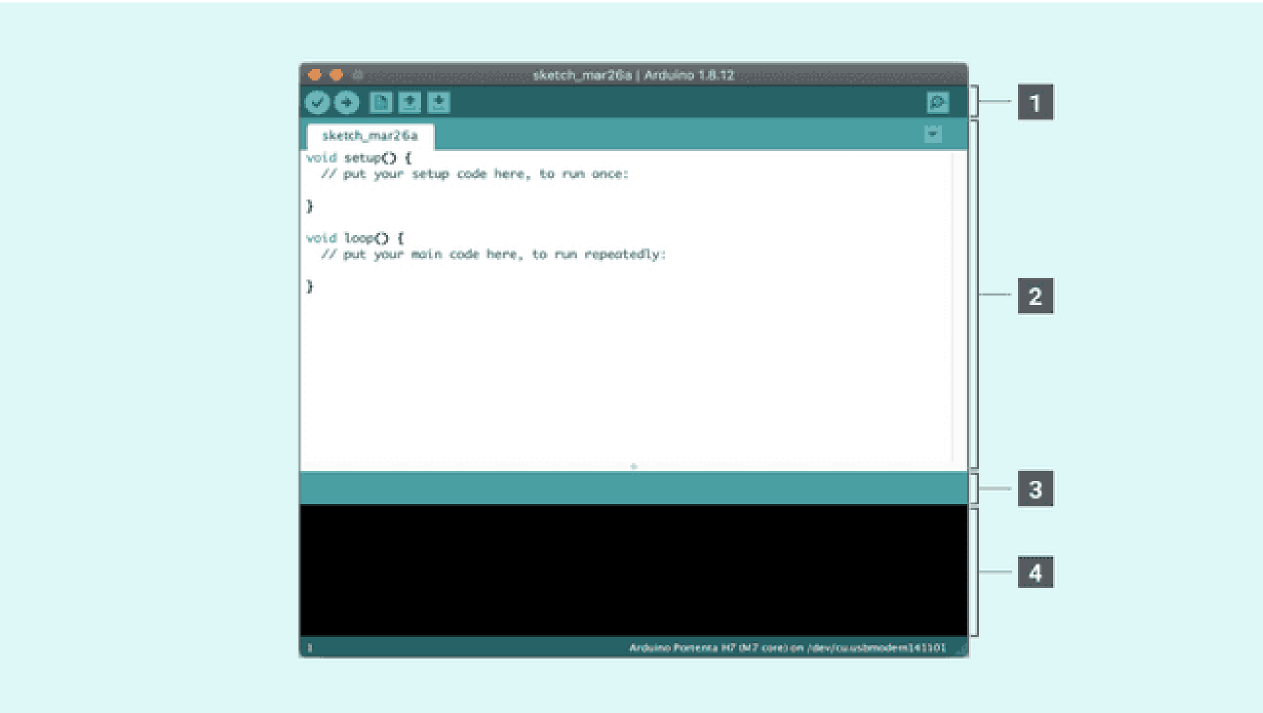


Figure 2.10: The Arduino Software IDE [84]

1. A menu system and a toolbar with buttons for typical operations. You may create, open, and save sketches, validate and upload programs, open the serial monitor, and more using the toolbar buttons.
2. The message section indicates faults and provides feedback while storing and exporting.
3. The text editor you use to write code.
4. The text console shows text generated by the Arduino Software (IDE), along with detailed error warnings and other details.

2.4.6 IoT as a hardware

Sensors

A sensor serves as an input device that outputs (signals) in relation to a particular physical quantity (input) [2].

Various kinds of sensors

The list of numerous sensor types that are frequently used in various applications may be seen below. The purpose of each of these sensors is to measure a particular physical characteristic, such as temperature, resistance, capacitance, conduction, heat transfer, etc.

1. Temperature Sensor
2. Proximity Sensor
3. Accelerometer
4. IR Sensor (Infrared Sensor)
5. Pressure Sensor
6. Light Sensor
7. Ultrasonic Sensor
8. Smoke, Gas and Alcohol Sensor
9. Touch Sensor
10. Color Sensor
11. Humidity Sensor
12. Position Sensor
13. Magnetic Sensor (Hall Effect Sensor)
14. Microphone (Sound Sensor)
15. Tilt Sensor
16. Flow and Level Sensor
17. PIR Sensor
18. Touch Sensor
19. Strain and Weight Sensor

Temperature sensor

The Temperature Sensor is among the most prevalent and well-liked sensors. A temperature sensor measures temperature changes by sensing the temperature, as the name suggests.

Figure 2.11: Lm35 temperature sensor [129]

Figure 2.12: NTC temperature sensor [94]

There are numerous types of temperature sensors, including Thermistors, Thermocouples, RTDs (Resistive Temperature Devices), Temperature Sensor ICs (such the LM35 and DS18B20), and Thermistors.

Analog or digital temperature sensors are available. When the temperature of an analog temperature sensor varies, its physical properties, such as resistance or voltage, also change. A traditional analog temperature sensor is the LM35.

A discrete digital value is produced by the digital temperature sensor (usually, some numerical data after converting analog value to digital value). Simple Digital Temperature Sensor (DS18B20).

HeartBeat sensor

Heart beat sensors are designed to give digital output heart beat when a finger is placed on it.

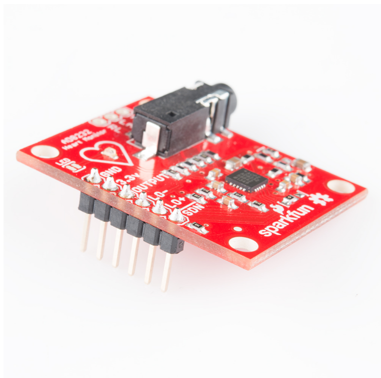


Figure 2.13: AD8232 heartBeat sensor [69]

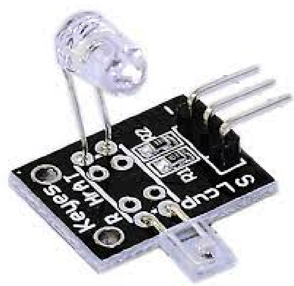


Figure 2.14: Ky39 heartBeat sensor [156]

2.4.7 Micro-Controller

is used to retrieve data values that the sensors have detected. The device has a Wi-Fi module built in that enables wireless transmission with the cloud online server, as depicted in the Figure 2.15.

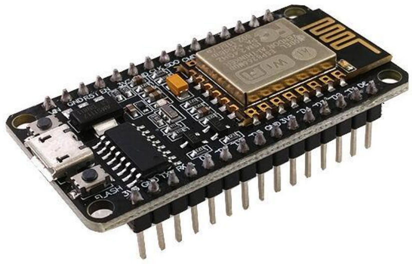


Figure 2.15: NodeMcu [147]

2.5 Historical Uses of IoT and AI for Detecting Body Temperature and Heartbeat

The healthcare sector has recently seen a transformation thanks to the convergence of Internet of Things (IoT) and Artificial Intelligence (AI) technologies, especially in the areas of body temperature and heartbeat monitoring. This section presents a historical overview of IoT and AI applications for tracking and analyzing data on body temperature and heartbeat, demonstrating the developments and contributions in this area.

2.5.1 Smart Beds and Bedding

IoT-enabled beds and bedding with sensors that keep an eye on your heartbeat and body temperature while you sleep. These methods offer information about the quality of sleep and can be used to spot potential sleep-related disorders. Examples include the Pod Pro from Eight Sleep and the SleepIQ from Sleep Number.

Figure 2.16: Smart Bed [26]

2.5.2 AI-based Fever Detection

To detect increased body temperatures, AI algorithms are used with thermal imaging or infrared cameras. For early detection of people with fever, which may indicate an underlying ailment, these devices have been employed in public places, airports, and healthcare institutions.

Figure 2.17: Fever Detection [78]

2.5.3 Sports and Fitness Monitoring

AI algorithms and IoT-enabled wearables are utilized to track heartbeat and body temperature during physical activity. Athletes and fitness enthusiasts can receive real-time feedback from these devices to improve performance and avoid overheating or overexertion.

Figure 2.18: Sports Monitoring [165]

2.5.4 Smart Clothing and Textiles

IoT-enabled textiles and apparel that have sensors built in to measure body temperature and heartbeat. Particularly in situations when conventional wearable sensors would not be practical or comfortable, these smart materials can offer continuous, unobtrusive monitoring. Smart socks, shirts, and patches that capture and transmit data for analysis using AI algorithms are a few examples.

Figure 2.19: Smart Clothes [29]

2.6 What's the difference between prediction and forecasting

2.6.1 Forecasting

Forecasting is the process of analyzing and clarifying a future state with regard to any operation that is being carried out. In order to predict facts for future events, this process weighs both the information from the past and the information from the present [182]. In a nutshell, forecasting is the process of anticipating future trends and their effects on the organization. A forecast is a calculation or estimation that determines the outcome of a future event using information from previous events and current trends [170].

Health Forecasting

Health forecasting is a cutting-edge field of forecasting and a useful tool for foreseeing future health-related events or circumstances, such as the demand for healthcare services [32]. By providing health service providers with advance notice so they can take the proper mitigating actions to reduce risks and manage demand, it facilitates preventive medicine and health care intervention strategies. For the prediction of particular health conditions or situations, health forecasting requires trustworthy data, information, and suitable analytical tools [91]. Numerous techniques have frequently been used to forecast general or specific health conditions because there is no one approach to health forecasting. However, the range of frequently used health forecasting methods/approaches does not yet have any designated health forecasting horizons (time frames). Additionally, the fundamental concepts of health forecasting have not been sufficiently described to direct the process [44].

Forecasting health principles

The forecasting of health is based on four main principles:

- measuring error and uncertainty.
- the main thing.
- accuracy and the nature of data aggregation.
- the future of health projections.

2.6.2 Prediction

A prediction is a claim made to explain a likely result or upcoming occurrence. Its root words are Pre, which means before, and dicer, which means to say in English. In spite of the uncertainty, businesses and governments use predictions made by experts to steer through uncertain projects [21].

Health predicting

Modeling of patient illness and care processes, which naturally have long-term temporal dependencies, is required for personalized predictive medicine. Electronic medical records contain episodic and erratic healthcare observations [142]. Quantitative prediction tools that correctly forecast a disease's occurrence, its prognosis or course, or a person's likelihood to respond to a specific treatment, have a number of benefits [186], These tools:

- allowing patients and their families to make better-informed decisions about prevention and treatment (for example, weighing the potential side effects of a preventative regimen against the likelihood that a particular outcome will occur) [85].
- aid medical professionals in precisely tailoring care by organizing prevention and treatment [54].
- aid medical systems in allocating funds to patients who face the greatest risk of a particular outcome [36].

In conclusion, forecasting primarily focuses on projecting future values or trends within a defined time frame, whereas prediction is a broader word that incorporates the estimation of future occurrences or outcomes. Forecasting is frequently utilized in fields including economics, weather forecasting, demand forecasting, and stock market predictions and typically requires time series analysis techniques.

2.6.3 Clarification of health projections

Forecasting the state of one's health or the onset of a disease entails warning people of impending events. It is also a type of preventive medicine or preventive care that involves public health planning and aims to make it easier for populations to access medical services. Health forecasting has frequently been used to predict admissions, daily hospital attendance, and trips to the emergency room [30]. Because of the way they are used in different fields, there are some key terms in forecasting that deserve to be noted [145]. Across many fields of study, the term "prediction" is primarily used to refer to an opinion-based speculation without any overt causal assumptions. Although they are occasionally used interchangeably and without definition, prognosis and prediction can mean different things in the literature on health forecasting [25]. Prognosis refers to a forecast of outcomes under no intervention, whereas prediction refers to a forecast of health outcomes linked to some type of intervention in the area of health. Another closely related idea that is well-known in the literature on disease surveillance is syndromic surveillance [144].

2.7 Related work

Numerous studies have investigated various strategies in the field of vital sign prediction. For predicting future body temperature, Khadija et al. [112] used a random forest model, the Random forest is an ensemble learning method that combines multiple decision trees to make predictions. The study likely involved training the random forest model on historical data, including factors such as previous body temperature readings, environmental conditions, and other relevant variables to forecast future body temperature accurately. Another study [33] used a statistical model to predict both heartbeat and body temperature, Statistical models can encompass a wide range of techniques, including linear regression, time series analysis, or other statistical approaches. The study likely involved analyzing historical data and identifying patterns and correlations between heartbeat, body temperature, and other factors to develop a predictive model. Additionally, ARIMA (Autoregressive Integrated Moving Average) model were used in a different study [172], ARIMA models are widely used in time series forecasting. These models capture the autocorrelation and seasonality of the data and make predictions based on past values and differencing operations. The study likely involved applying ARIMA models to historical data of vital signs, such as heartbeat or body temperature, to forecast future values. Deep neural networks have also been used in the context of predicting vital signs [116], Deep neural networks are a type of machine learning model inspired by the structure of the human brain. They consist of multiple layers of interconnected neurons that learn hierarchical representations of the input data. The study likely involved training deep neural networks on large amounts of vital sign data, possibly including time series information, to predict future values of vital signs such as heartbeat or body temperature.

2.8 Conclusion

In this chapter, we have discussed the concepts of prediction and forecasting and their applications in the field of healthcare, defined the subtle differences between the two, and seen the difficulties in health forecasting. We have also presented the most widely used algorithms and their applications.

Additionally, we discussed how to measure body temperature and heartbeat, introduced temperature and heart rate sensors by defining and categorizing them. We have also provided a brief overview of IoT technology before presenting the microcontrollers required for IoT applications. In the next chapter, We'll see how the ideas presented in this chapter's introduction can be used to develop a health surveillance monitoring system and employ it to non-invasively forecast future changes in body temperature and heartbeat.

CHAPTER 03

DESIGN AND IMPLEMENTATION

Chapter 3

Design and implementation

3.1 Introduction

Artificial intelligence initiatives have increased in popularity and success as a result of technological advancement and the availability of more data. Remote health monitoring is one area where this has had a significant influence. We have created a wearable device that can remotely track a patient's heart rate and body temperature in response to the expanding demand for such technology. In particular, it will be launched for the first time in Algeria, where it will be used to monitor vulnerable individuals and forecast their future health status. This device will play a vital role in the healthcare sector.

This chapter will give a general overview of our project, outlining the parts needed to construct the wearable gadget and the suggested architectural layout. We will also introduce the program that is used to plot features for body temperature and heartbeat data on a Web interface. Readers will have a thorough knowledge of the project's components and the user interface needed to communicate with the wearable device by the end of this chapter.

3.2 Design

Our proposed system is devided in two parts : **Hardware part** and **Software part**. The present project's goal is to create a smart device that patients can wear (a wearable device). The gadget consists of a micro-controller, a multiplexer, a WIFI module, a temperature sensor, a heartbeat sensor, and so forth. The measures will then be used to anticipate the future heart rate and body temperature of monitored patients using machine and deep learning technology. The measurements will be uploaded to a cloud server, examined in real-time, and accessible remotely by doctors or guards. Because of this, the suggested system is created as shown in 3.1:

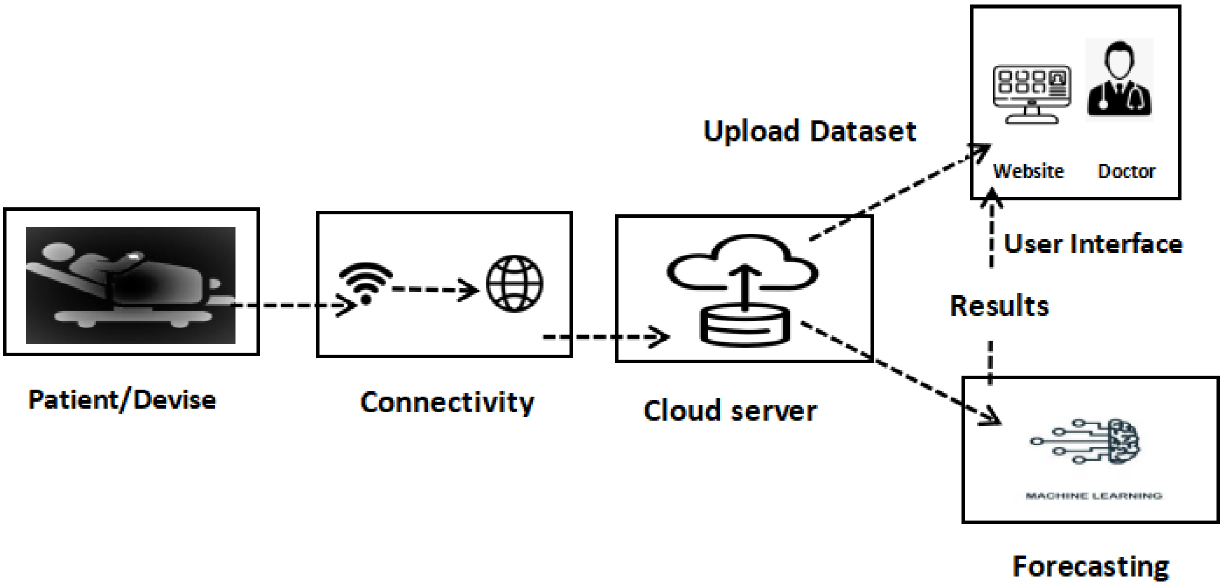


Figure 3.1: General architecture

3.2.1 Hardware part

The hardware part should be made up of the parts listed below:

- Temperature sensor;
- HeartBeat sensor;
- a microcontroller to extract data values detected by sensors;
- a Wi-Fi module that permits the device and cloud to communicate wirelessly
- a Breadboard to connect the components;
- 3d design to made the cover;
- a battery;
- a cover to shield the previously mentioned elements.

We have selected the following components for our device:

- Lm35 as a temperature sensor;

Figure 3.2: Lm35 temperature sensor

- ky39 as a heartbeat sensor;

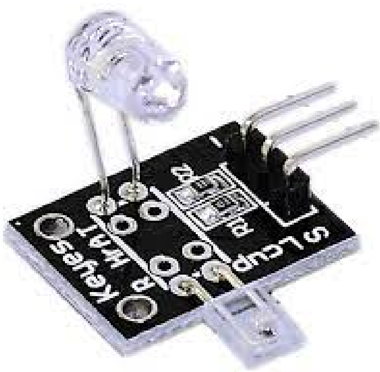


Figure 3.3: ky39 heartbeat sensor

- NodeMcu as microcontroller;

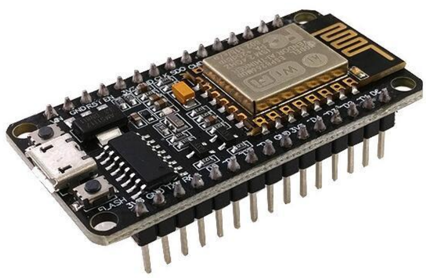


Figure 3.4: NodeMcu Microcontroller

- a Breadboard;

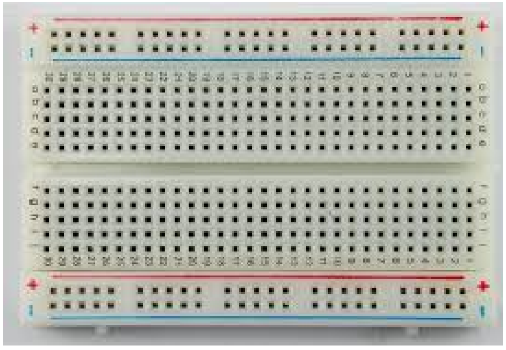


Figure 3.5: BreadBoard

- Power supply as a battery.

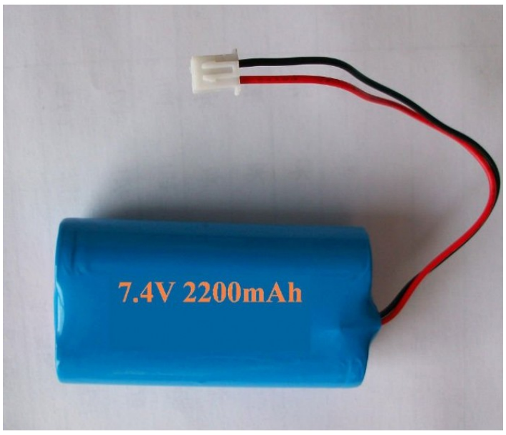


Figure 3.6: Power supply

Hardware architecture

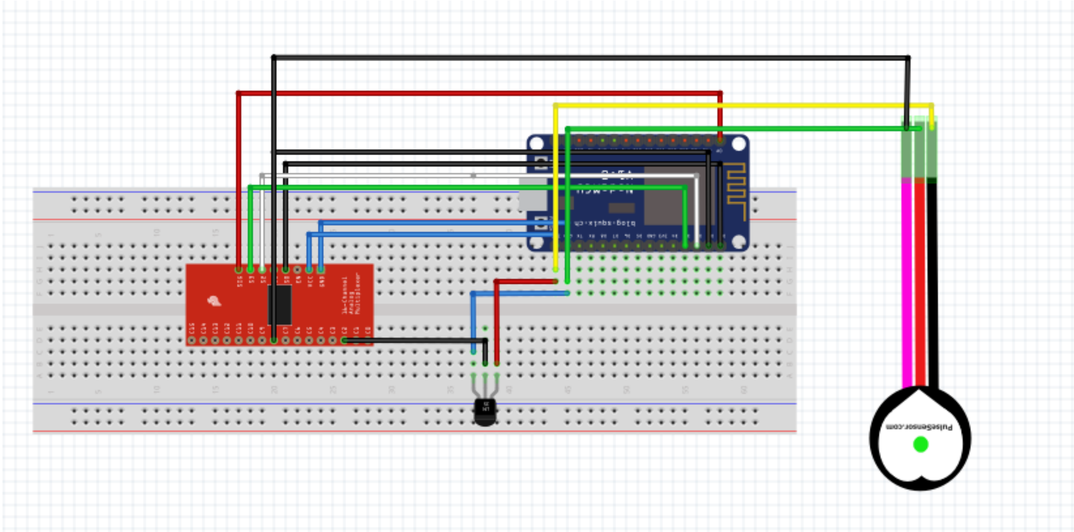


Figure 3.7: Hardware system design

The circuit in Figure 3.7 is depicted using the "Fritzing" program https://fritzing.org/. The actual parts are now connected as follows:

- The Vcc of the LM35 is connected to 3v of the ESP8266.
- The Out pin of LM35 is connected to the C2 of the multiplexer as an analogue pin.
- The GND of LM35 is connected to the GND of the ESP8266.
- The Vcc of the Ky039 is connected to 3v of the ESP8266.
- The Out pin of Ky039 is connected to the C8 of the multiplexer as an analogue pin.
- The GND of Ky039 is connected to the GND of the ESP8266.

3.2.2 Software part

Datasets Collection and storage

In order to use the models, we used Google Colab https://colab.research.google.com/ The future values of the body's core temperature and heartbeat are predicted during the regression process.

Two different data sets have been used to train our model (Skin Temperature dataset https://data.world/laurie/skin-temperature and Heartbeat dataset https://ecg.mit.edu/time-series/). They are imported it into Python code to train our model and choose the best model due to the MSE.

In order to achieve the flexibility of gathering, storing, and having real-time access to data, we will set up a channel in the open source Internet of Things application "ThingSpeak" https://thingspeak.com/ Figure 3.9 It is a data platform and application programming interface (API) where we can transport and store data from sensors to the cloud using the WiFi module built into the NodeMCU. The architecture of data collection and storage is in the Figure 3.8

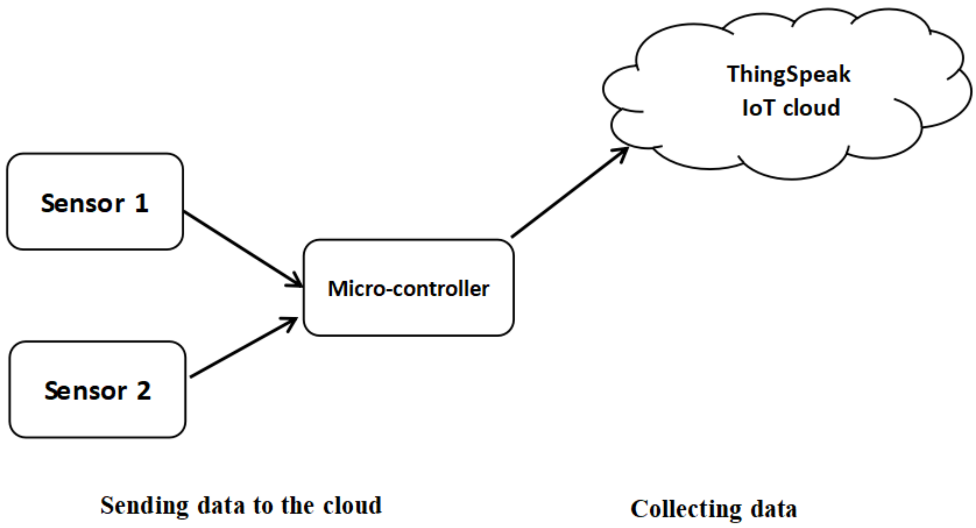


Figure 3.8: Architecture of data collection, storage

Figure 3.8 represents: Instant measurement of the two metrics and send them by wifi to store them in the ThingSpeak cloud as shown in the Figure 3.9.

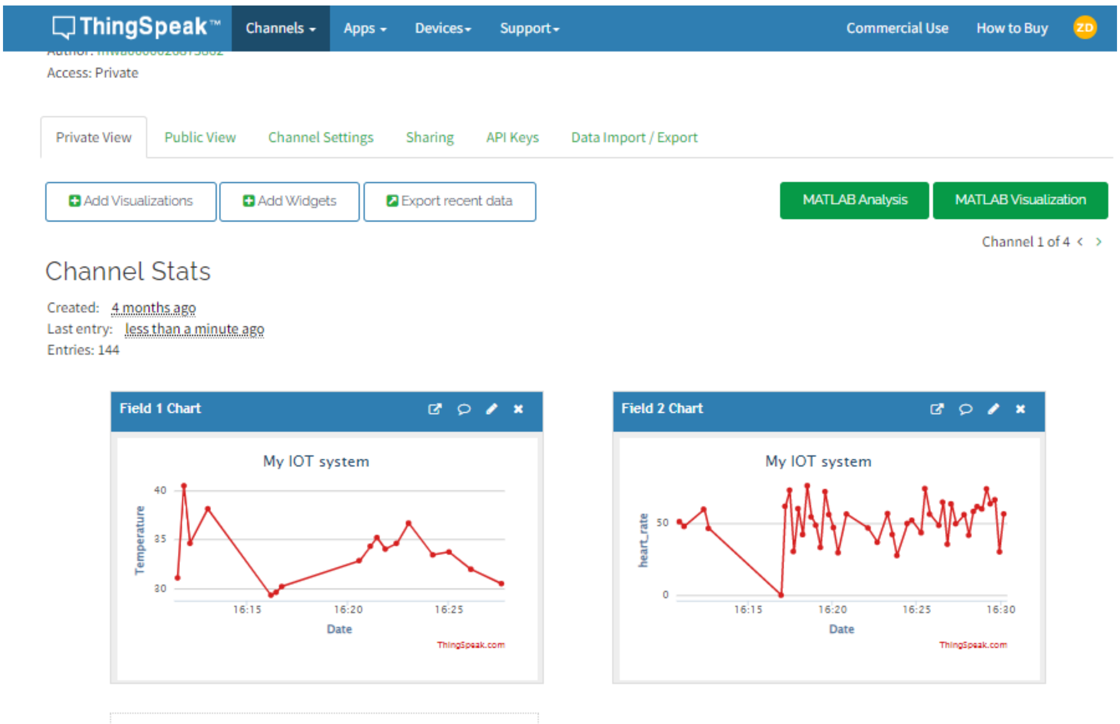


Figure 3.9: ThingSpeak

The reason behind choosing cloud capacity for our extend is to discover sufficient capacity space to hold the information we are securing. Other than putting away information in a inaccessible database, in our venture will permit the specialist who is checking his understanding to get to and visualize modern information from anyplace and anytime and not as it were from his office.

Once we sign into ThingSpeak, we are going need to make a channel where to store our information secretly, visualize and recover them as a time-series dataset.

First method using machine learning

To predict future body temperature and heart rate based on historical values (time series dataset), we have proposed an architecture based on machine learning model depicted in Figure 3.10. After collecting the data sets using our monitoring device and storing them in the cloud, we fetch them to feed the suitable algorithm to train it, then we get the values predictive value of body temperature and heartbeat in the following hours. displayed in the user interface. For our project, to predict both body temperature and heartbeat based on time series data, we will use the following regression methods as they are used to predict the item variables target on continuous scale:

1. ***Linear Regression:*** it satisfies a model based on linear coefficients. $w = (w_1, \ldots, w_p)$ the residuals between the observed targets in the data set and those for which the linear approximation should be reduced as much as possible [108].

2. ***Ridge regression:*** it is a method of calculating coefficients of multiple regression models when the independent variables are closely correlated with each other [117].

3. ***Decision tree regressor:*** It is used to modify a sine curve by including a noise observation. It learns local linear regressions that resemble sinusoidal curves. The decision tree learns noise and extremely specific details from the training data when the maximum tree depth is set too high [176].

4. ***Random forest regressor:*** It is a meta estimator that applies the mean to adjust several decision tree classifiers to different subsamples of the data set in order to increase prediction accuracy and reduce redundancy [43].

5. ***K-Neighbours regressor:*** The target variable is predicted by local interpolation of the targets connected to K nearest neighbors in the training dataset [152].

6. ***LASSO regressor:*** The absolute shrinkage and minimum regression of the selection operator prevent its coefficients from exploding, unlike conventional high-dimensional linear regression. The ability to LASSO regression to perform variable selection can be very useful when there are many variables [179].

7. ***Gradient boosting regressor:*** By optimizing any differentiable loss functions, it builds models incrementally and generalizes them [128].

8. ***Adaboost regressor:*** A regressor is to begin with fitted to the beginning information set within the versatile boosting metaestimator, powerless learners are at that point altered to favor occurrences where the earlier classifiers were inaccurate [180].

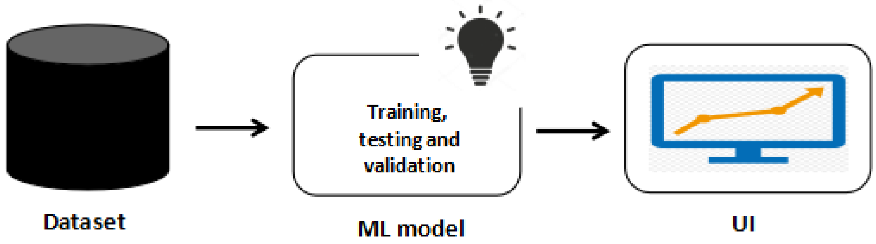


Figure 3.10: Forecasting process with ML

Second method using deep learning

Due to its capacity to recognize complex patterns, manage high-dimensional data, perform end-to-end learning, manage long-term dependencies, robustness to noisy data, and scalability, deep learning is frequently favoured over classic machine learning approaches for forecasting. Deep learning models are particularly suited for predicting because they include numerous layers and hierarchical representations that enable them to automatically learn complex relationships in the data. They have the ability to handle high-dimensional and sequential data, extract pertinent features, and even perform good prediction in the presence of noise. Deep learning models can benefit from parallel processing and scale effectively with huge datasets. While the decision between deep learning and conventional machine learning is influenced by a number of variables, deep learning's advantages make it a potent tool for forecasting tasks. So we have proposed an architecture based on deep learning exactly LSTM model depicted in Figure 3.11 to train the model for vital signs datasets (train the model for body temperature dataset and train it for heartbeat dataset) and predict those two metrics.

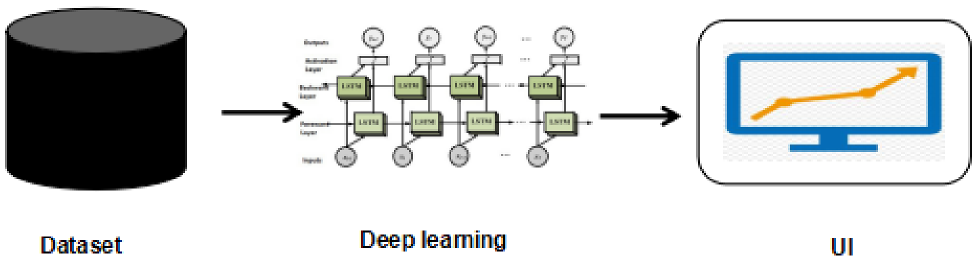


Figure 3.11: Forecasting process using DL

- **LSTM (Long Short-Term Memory):** is a unique kind of recurrent neural network (RNN) architecture made to handle the vanishing gradient issue and successfully capture long-term dependencies in sequential input [74]. It introduces a memory cell with an input gate, a forget gate, and an output gate as its three primary parts. The LSTM can selectively learn, retain, and forget information over lengthy sequences because to these gates, which control the flow of information into and out of the memory cell [79]

The output gate decides how much of the cell's content should be used to compute the output of the LSTM at each time step, while the input gate controls how much fresh information should be stored in the cell [126]. The forget gate decides which information should be removed from the cell. In tasks involving sequential data, such as natural language processing, speech recognition, and time series prediction, LSTM is particularly effective because of its ability to acquire and disseminate pertinent information over extended sequences [50].

First, the output of an LSTM is fundamentally dependent on three factors at any one time :

- The network's present long-term memory is referred to as the cell state.
- The prior concealed state, or the output at the previous instant.
- The input information for the present time step.

The information in a sequence of data enters, is stored in, and exits the network using a series of 'gates' that are used by LSTMs. A typical LSTM has three gates: an output gate, an input gate, and a forget gate. Each of these gates is a separate neural network and may be thought of as a filter [75].

In the discussion that follows, we'll use the following diagram Figure 3.12 to represent an LSTM cell. Consider navigating the diagrams from left to right as you look at them.

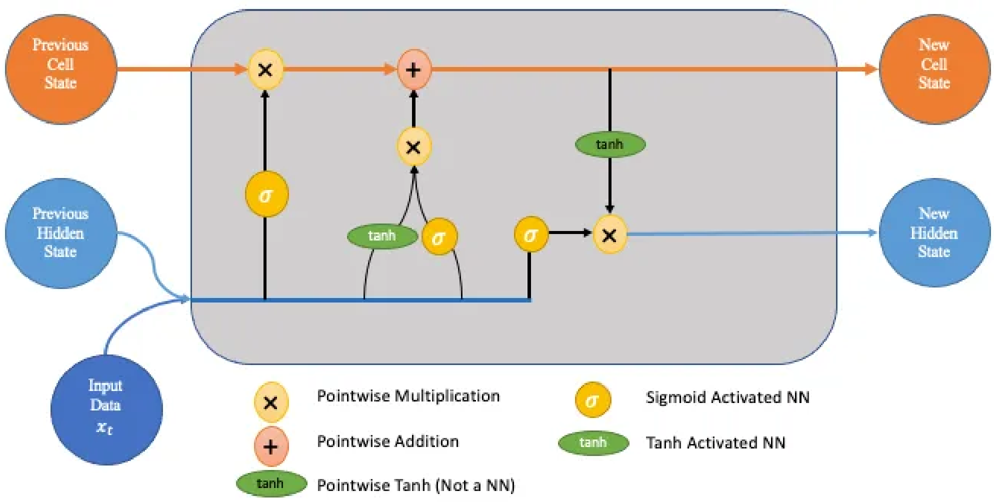


Figure 3.12: Cell LSTM Diagram [146]

How are LSTM networks implemented ?

Step 01:

The forget gate is the initial stage of the procedure. In this step, we will determine which pieces of the cell state—the network's long-term memory—are relevant in light of both the prior hidden state and the fresh incoming data.

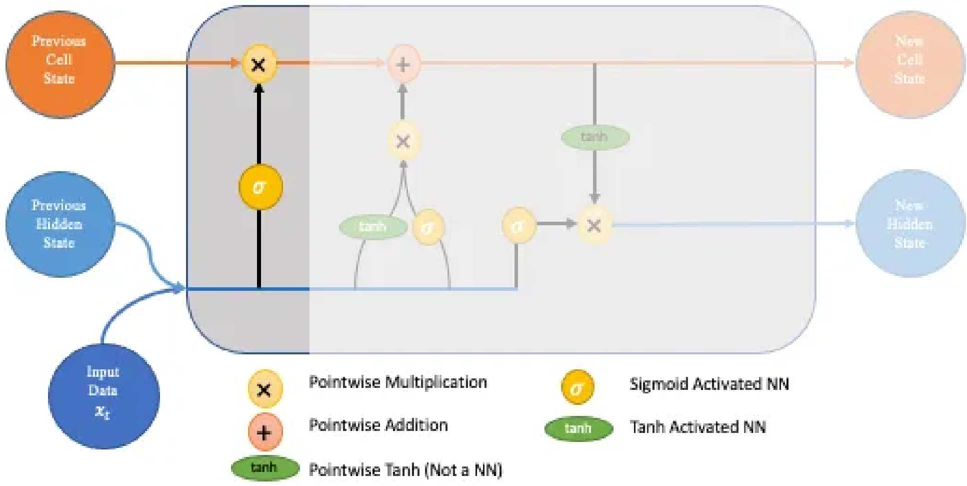


Figure 3.13: Forget Gate

In order to accomplish this, a neural network is fed both the new input data and the previous hidden state. This network creates a vector with each member falling inside the range [0,1] (this is verified by sigmoid activation). This network (inside the forget gate) is trained to output a value that is roughly close to 0 when an input component is regarded irrelevant and a value that is roughly close to 1 when it is considered important. Consider each component of this vector as a filter or sieve that lets through more data as the value comes closer to 1 [28].

The preceding cell state is pointwise multiplied with these output values before being transferred upward. The components of the cell state that the forget gate network has determined to be irrelevant will be multiplied by a value near to 0 as a result of this pointwise multiplication, which means they will have less of an impact on the subsequent steps [104].

In summary, based on the previous concealed state and the new data point in the sequence, the forget gate determines which parts of the long-term memory should now be forgotten (have less weight).

Step 02:

The new memory network and the input gate are involved in the following step. This step's objective is to decide what new information, in light of the prior concealed state and the incoming input data, has to be added to the network's long-term memory (cell state) [75].

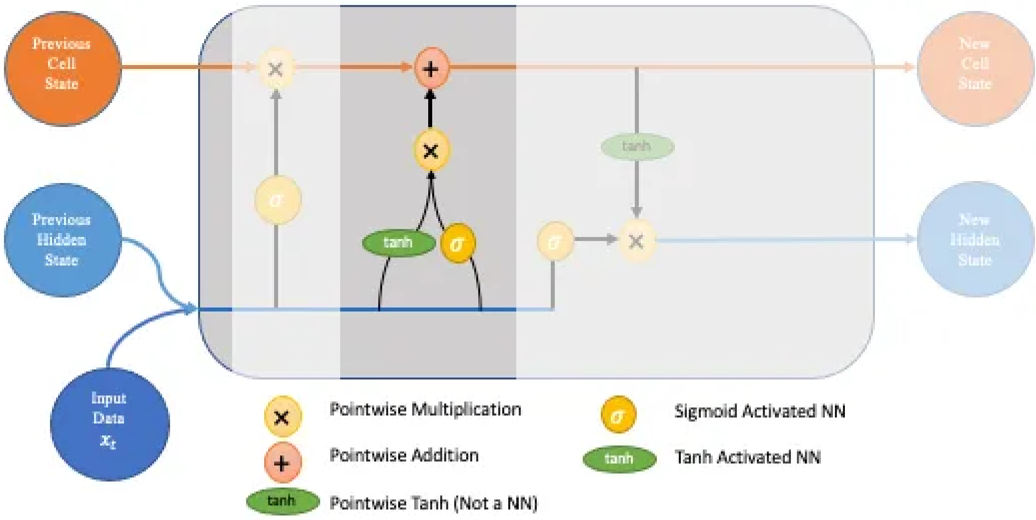


Figure 3.14: Input Gate

The old hidden state and the fresh input data are the same inputs for both the new memory network and the input gate, which are both independent neural networks. It is important to note that the inputs used here are the same as those used by the forget gate [109].

- The new memory network is a tanh-activated neural network that has mastered the art of fusing the prior hidden state with fresh input data to produce a "new memory update vector." Given the context from the prior concealed state, this vector essentially contains information from the new input data. Given the new information, this vector indicates how much to update each part of the network's long-term memory (cell state).

  As you can see, we are using a tanh because its values fall between [-1,1] and might be negative. If we want to lessen the influence of a component on the cell state, we must allow for negative values here.

- However, there is a significant flaw in step 1 above, when we build the new memory vector: it doesn't properly determine whether the new input data is even worthwhile remembering. The input gate is useful in this situation. The input gate, which is a sigmoid activated network, serves as a filter by determining which parts of the "new memory vector" should be kept. Due to the sigmoid activation, this network will produce a vector of values in the [0,1] range, allowing it to function as a filter through pointwise multiplication. An output close to zero indicates that we do not want to update that component of the cell state, much to what we observed with the forget gate.

- The results of parts 1 and 2 are multiplied in points. This results in the regulation and, if necessary, the setting of the magnitude of new information we decided on in part 2 to 0. The network's long-term memory is updated after the combined vector that results is added to the cell state.

Step 03:

We can proceed to the output gate, which determines the new hidden state, when we have finished updating the network's long-term memory. We'll employ three factors—the recently updated cell state, the previous hidden state, and the fresh input data—to make this determination [57].

One could believe that all we need to do is output the updated cell state, but it would be like asking someone to tell you everything they've ever learned about stocks when all they're asked is whether they think they'll go up or down tomorrow [178]!

In the same way that we did with the forget gate network, we establish a filter, the output gate, to stop this from happening. Since we want the filter property obtained from outputs in [0,1], the inputs are the same (prior hidden state and new data), and the activation is likewise sigmoid [160].

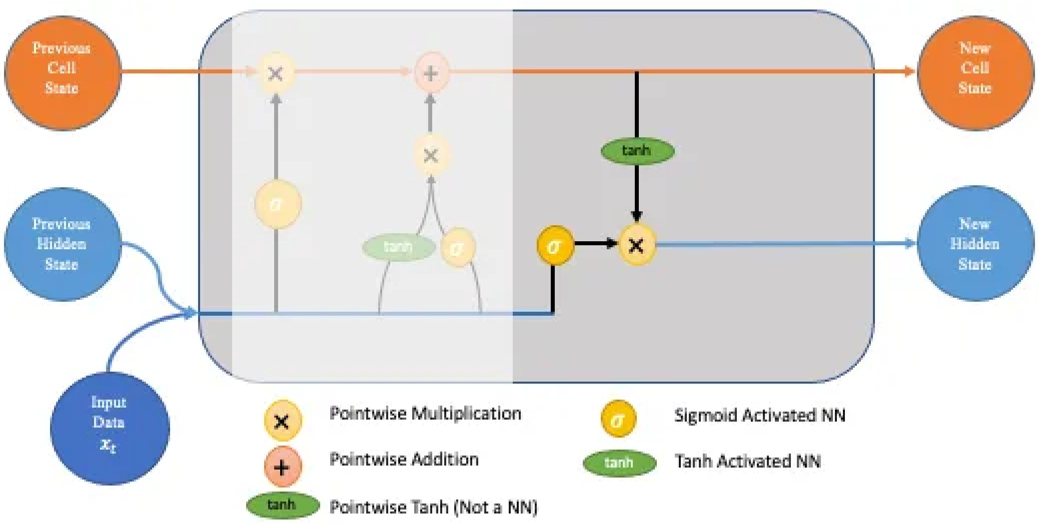


Figure 3.15: Output Gate

We want to apply this filter to the recently modified cell state, as was already indicated. This guarantees that only pertinent data is output and saved in the new concealed state. However, in order to force the values within the range [-1,1], we send the cell state through a tanh before applying the filter [22].

This last step's detailed procedure is as follows:

- Apply the tanh function pointwise to the existing cell state to get the compressed cell state, which is now located in [-1,1].
- To obtain the filter vector, run the prior hidden state and the current input data through the sigmoid activated neural network.
- Apply this filter vector via pointwise multiplication to the compressed cell state and output the new hidden state.

There are methods for evaluating models, but choosing the best model to use is not arbitrary. We can take advantage of the regression models we already use. calculate strategies R^2 and root mean square error (RMSE) to evaluate the performance of each model.

- $RMSE$: is the standard deviation of the forecast or prediction error. $RMSE$ is defined as follows [136].

  $$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(\frac{d_i - f_i}{\sigma_i}\right)^2}.$$

  Where n represents the overall sample size for the data, and $(d_i - f_i)$ is the difference between the true value and the predicted value. for the i^{th} test specimens.

- R^2 :is a statistical measure of the proportion of variance of a dependent variable in a regression model that can be explained by an independent variable or a set of independent variables [135].

3.2.3 The 3D design

The 3D model of the wearable device we created, which is represented as a watch in the Figure 3.16.

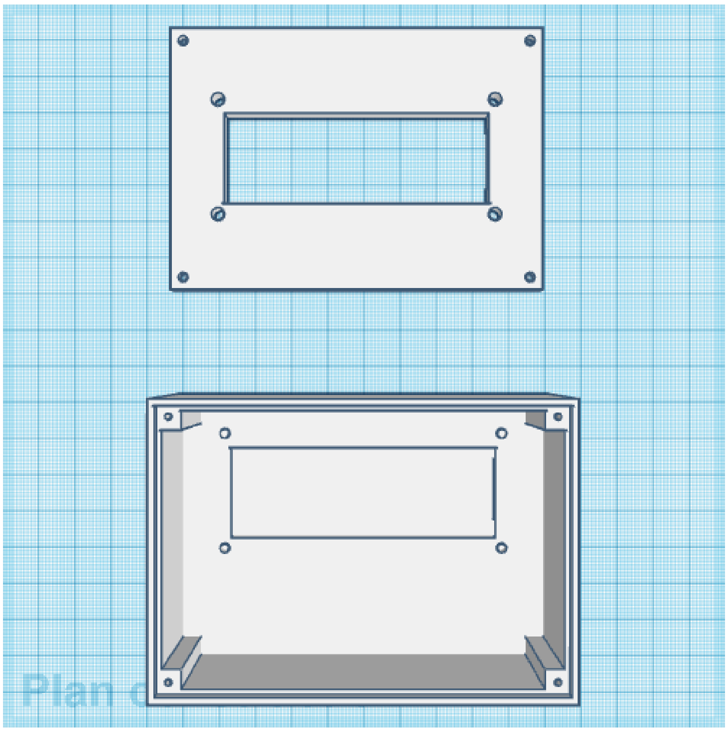


Figure 3.16: The 3D design

3.3 Diagrams

3.3.1 Diagrams of our system

This part is the conception by diagrams phase of our system:

Use case diagram

The use case diagram Figure 3.17 displays the system's capabilities, including website management, patient monitoring, state forecasting, access to patient and doctor information, body temperature and heartbeat measurement, user authentication and login within the Website, involving roles like Admin, Doctor, Patient, and the Wearable Device.

Figure 3.17: Use case system diagram

Sequence diagram

The flow diagram Figure 3.18 shows how the wearable device (Smartwatch) interacts with the Website. It shows actions like taking body temperature and heartbeat readings, transmitting data, user identification and login, and accessing personal data and wearable data.

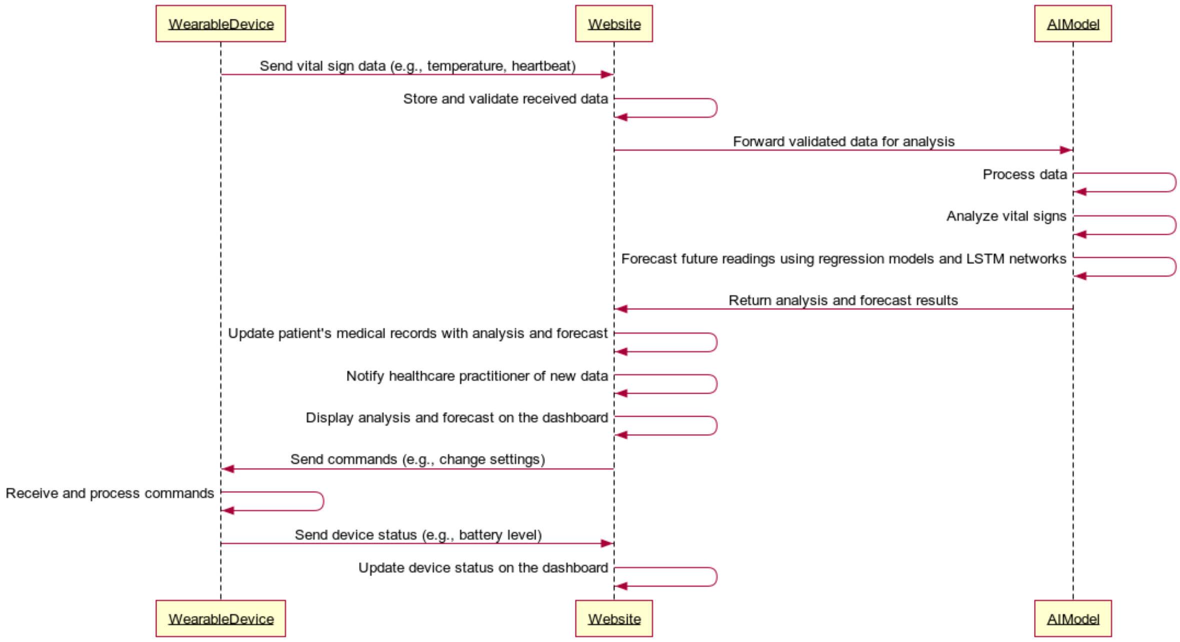


Figure 3.18: Sequence diagram

Class diagram

This class diagram Figure 3.19 shows the various system parts, such as the Wearable Device class for sensing body temperature and heart rate, the Website class for user interaction, and the subclasses for the Admin, Doctor, and Patient roles with their respective functionalities.

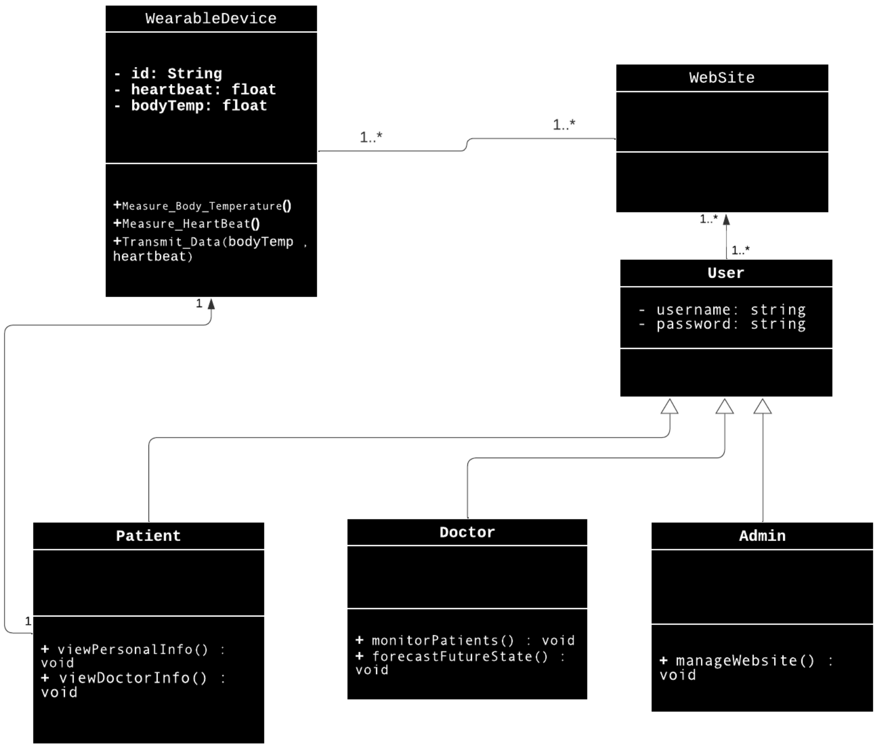


Figure 3.19: Class diagram

3.3.2 Diagrams of the Website

This part is the conception by diagrams phase of the Website:

Use case diagram

Use case diagrams Figure 3.20 outline a Website's general capabilities. In use case diagrams, use cases and actors only describe what the system does and how actors use it. The inner workings of the system. They are used to provide a broad overview of a software system's functional activity.

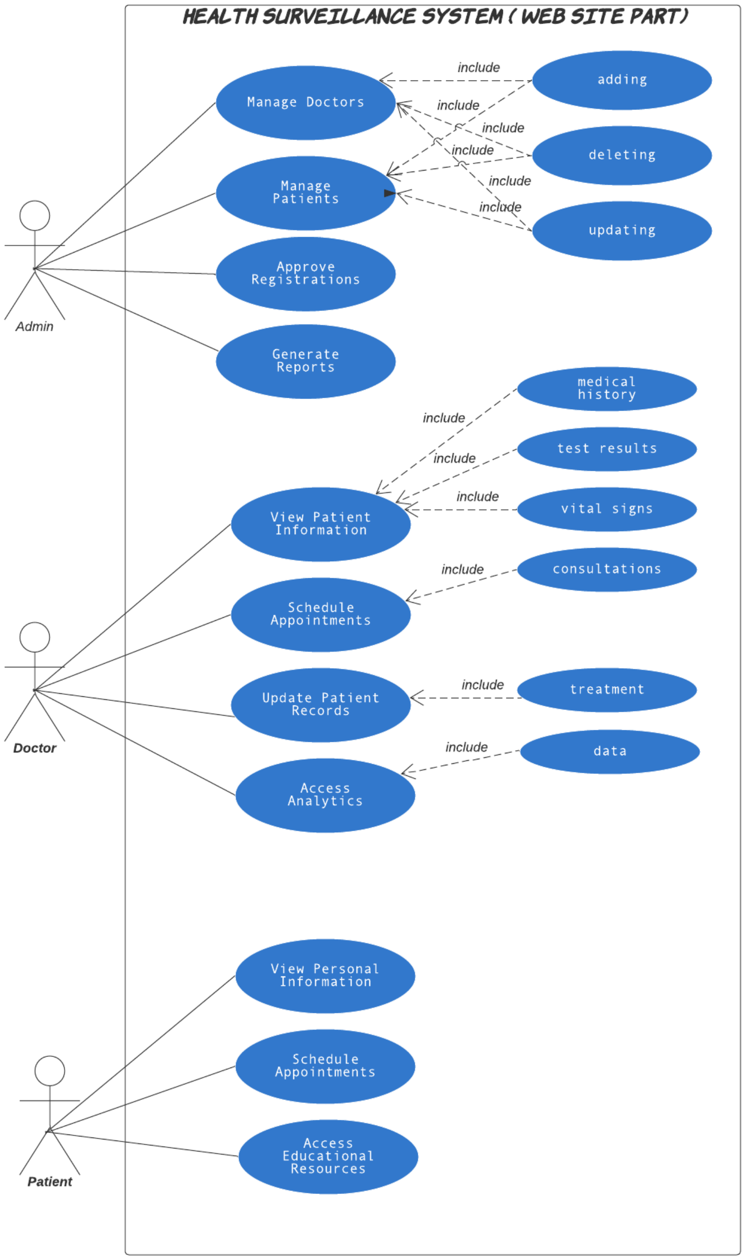


Figure 3.20: Use case Website diagram

Sequence diagram

A sequence diagram is a visual depiction of the timing of messages transmitted to or received from a system. A dynamic picture of how various actors interact with the system is shown in the sequence diagrams that follow: In practice, the system and the actors engage in a variety of interactions. In this report, we will simply highlight the crucial interactions:

Authentication sequence diagram :

The authentication diagram is identical for all actors, it can be presented as following Figure 3.21:

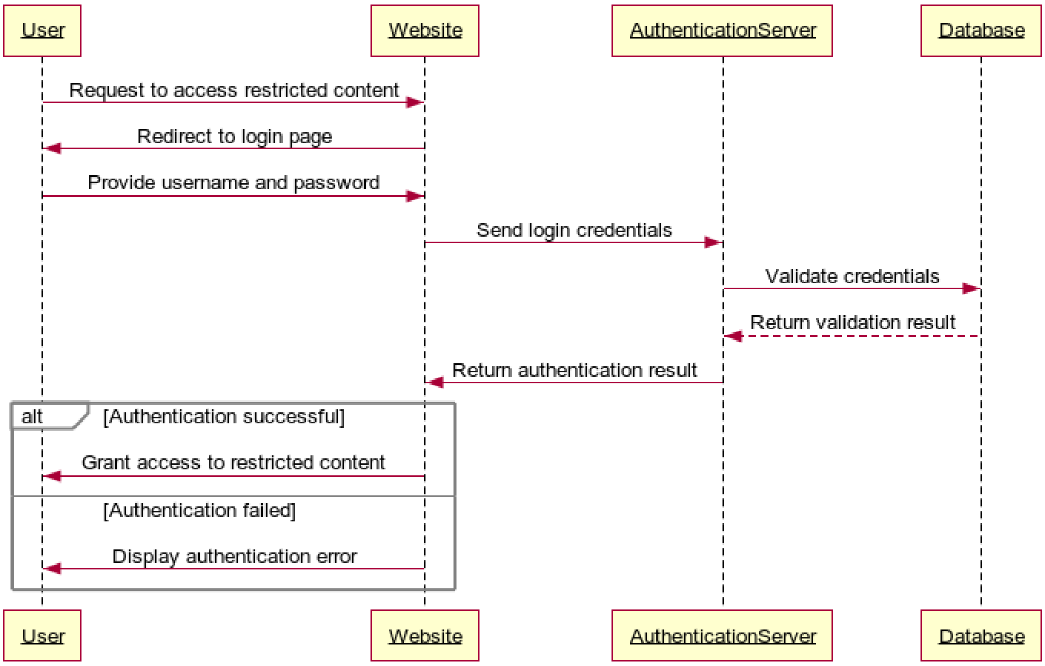


Figure 3.21: Authentication sequence diagram

Sequence diagram relationship between admin and system:

The role of the admin is as follows Figure 3.22

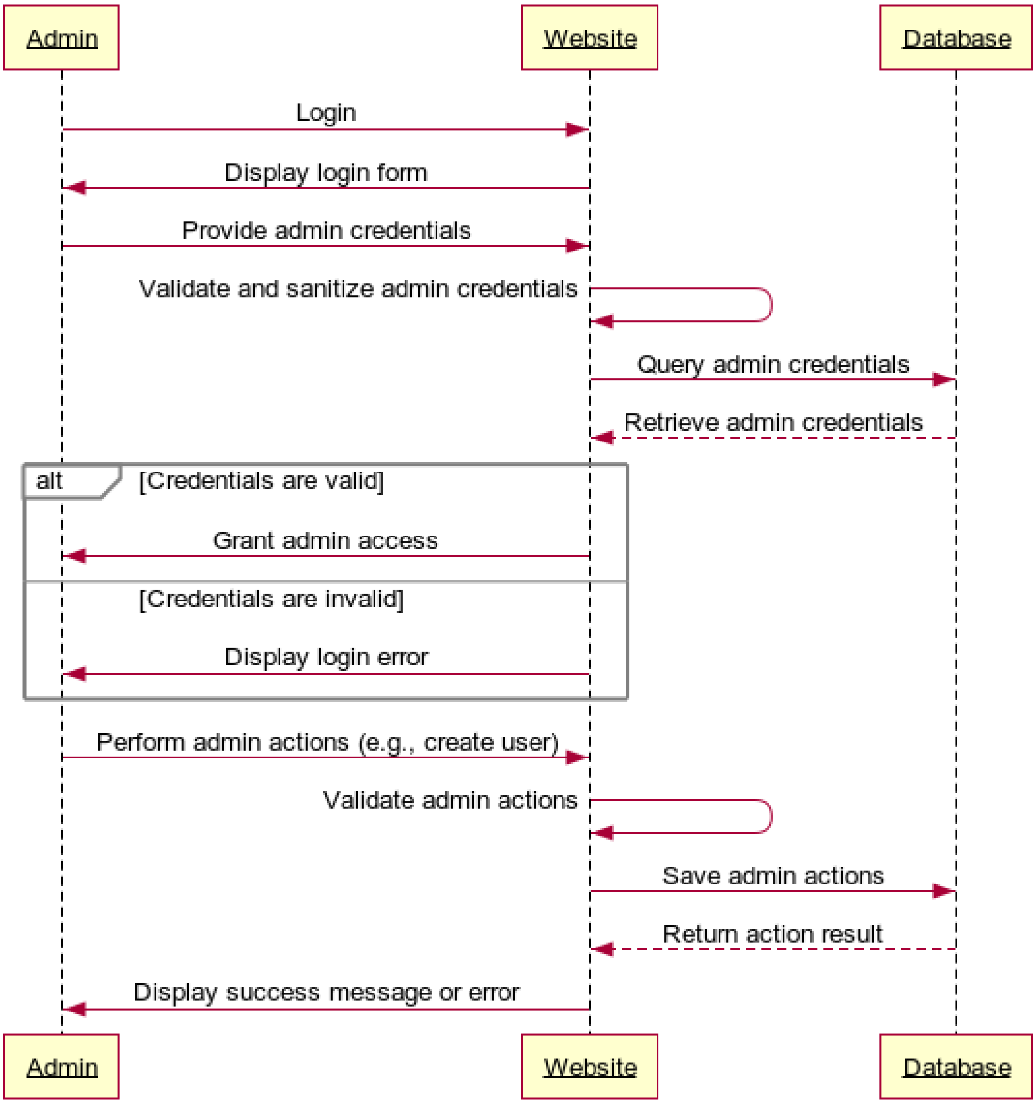


Figure 3.22: Sequence diagram relationship between admin and system

Sequence diagram relationship between doctor and system :

The benefit of the doctor in the site is as follows Figure 3.23

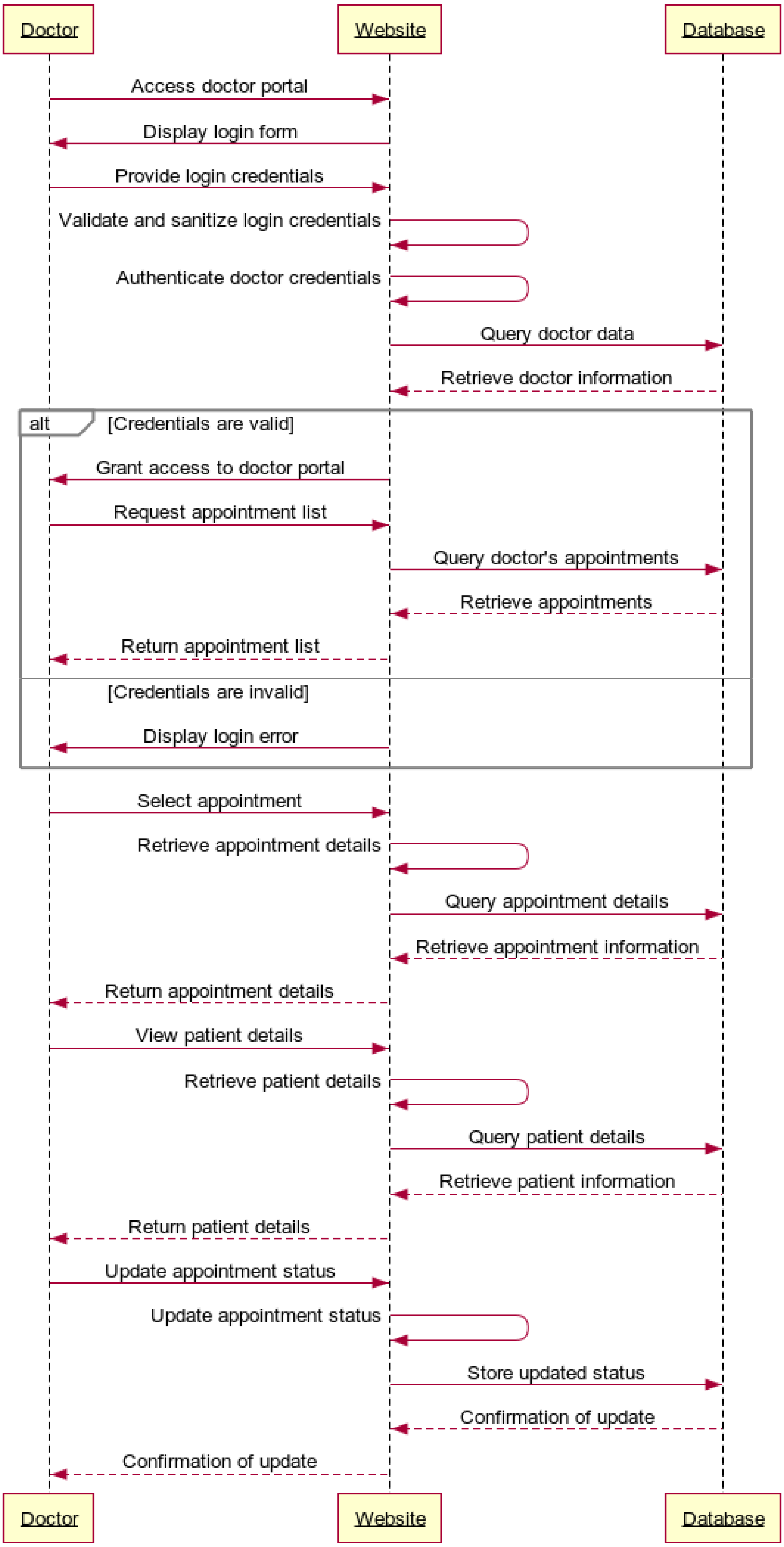


Figure 3.23: Sequence diagram relationship between doctor and system

Sequence diagram relationship between patient and system :

The benefit of the patient in the site is as follows Figure 3.24

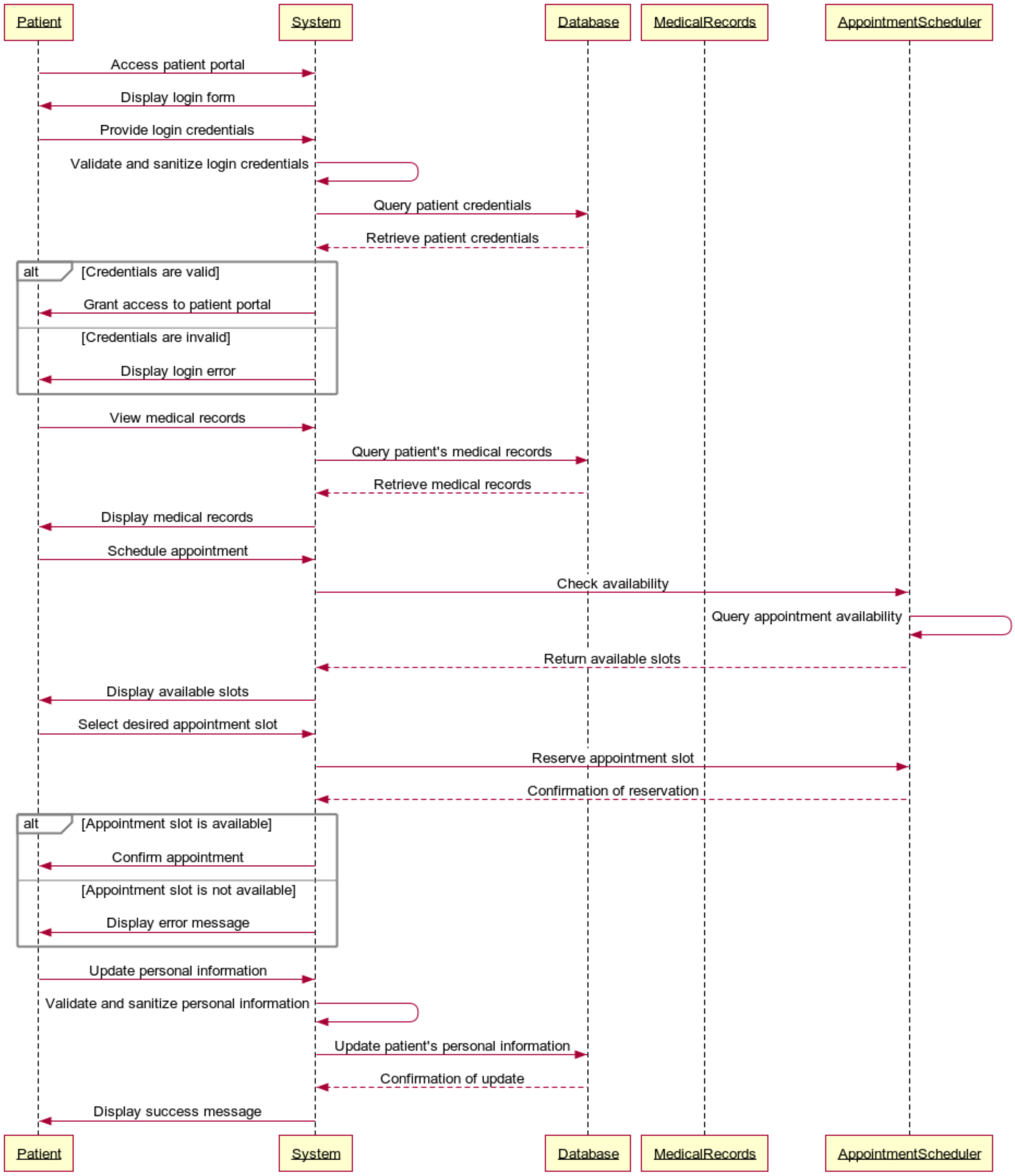


Figure 3.24: Sequence diagram relationship between patient and system

Sequence diagram relationship between doctor and patient:

The relationship between doctor and patient as follows Figure 3.25

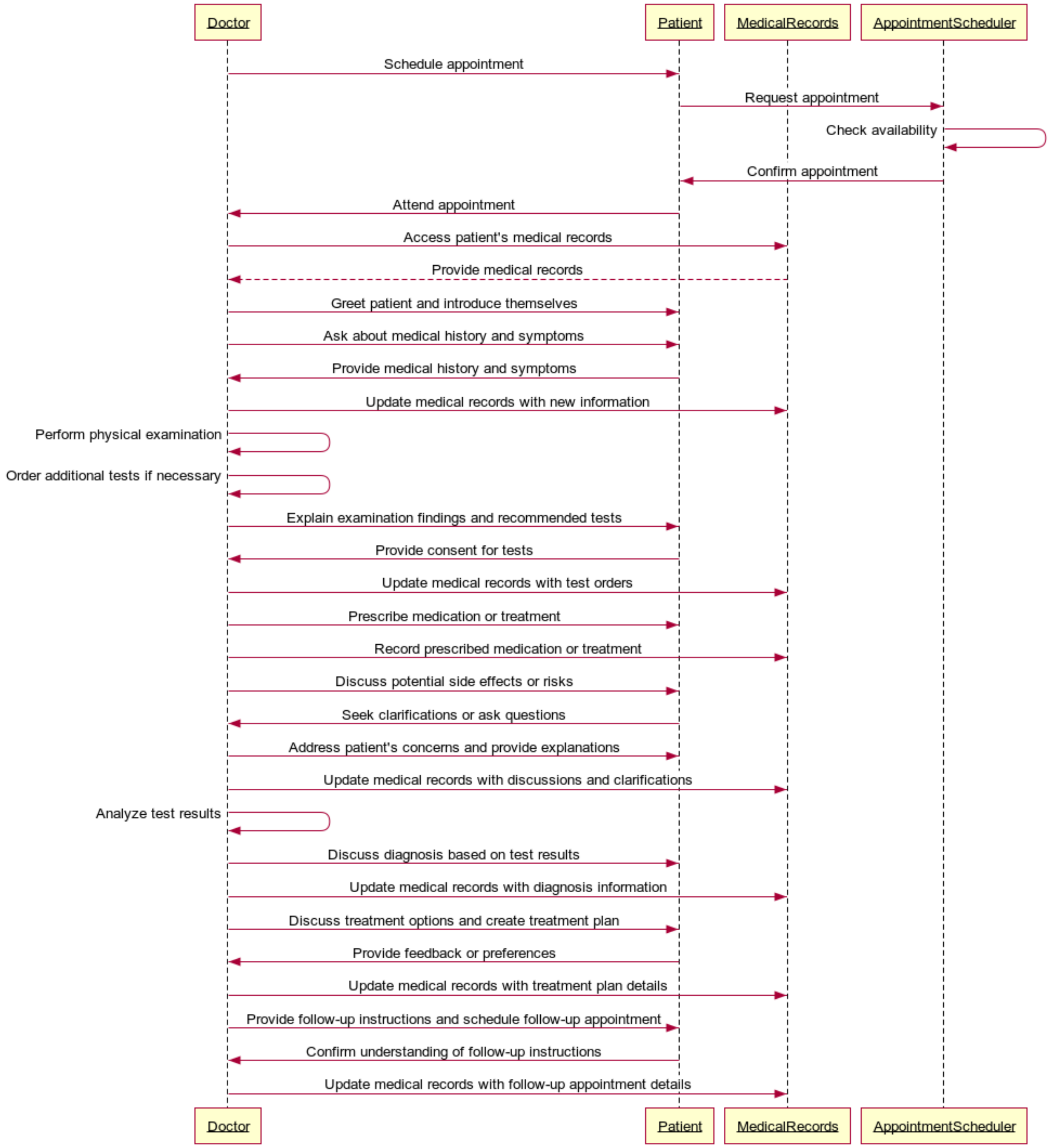


Figure 3.25: Sequence diagram relationship between doctor and patient

Class diagram

This Figure 3.26 shows the structure of the Website, using the class diagram.

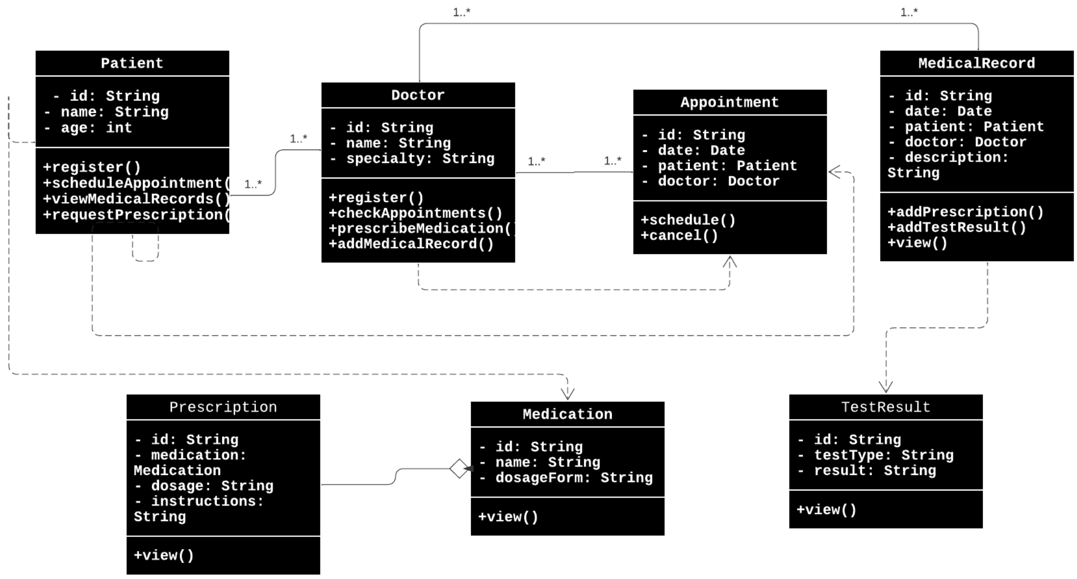


Figure 3.26: Class diagram

3.4 Implementation

3.4.1 Languages and tools for development

In this section, we will cover the tools and platforms that we used in our system, beginning with hardware tools and then moving on to software tools.

Hardware tools

Our system was built on a graphics station running Windows 64-bit, an Intel(R) Core(TM) i7-7500U CPU @ 2.70 GHz, 2.90 GHz, and 16 GB of RAM.

Anycubic i3 Mega 3D printer: The Anycubic i3 Mega 3D printer is a fairly standard machine, modeled after the Prusa i3. Its print volume is not "huge" and is limited to prints up to 210 x 210 x 205 mm, while its resolution can be reduced to 50 microns. The FDM printer is compatible with PLA, ABS, HIPS, PETG and wood fiber with a diameter of 1.75 mm for a 0.4 mm nozzle that can reach 260°C. The heater bed can be mounted at 110°C to print objects from an SD card (via the monitor screen) or via a USB cable (control from a computer) [3].

Software tools

1. ***Python:*** Python is an object-oriented, high-level, interpreted, open-source programming language with dynamic semantics that comes with a large number of supporting libraries. Python is used for machine learning, data analysis, and even design [1].
2. ***Google Colab:*** Google Research created Colaboratory, often known as "Colab". Colab allows anyone to develop and run any Python code of their choice through the web. It is a setting that is particularly suitable to education, data analysis, and machine learning [8].
3. ***Django:*** High-level Python web framework Django promotes quick development and streamlined, practical design. It was created by seasoned programmers and handles a lot of the hassle associated with web development, freeing you up to concentrate on building your app without having to invent the wheel. It is open source and free [59].
4. ***ThingSpeak:*** With the help of the IoT analytics tool ThingSpeak, you can gather, visualize, and examine real-time data streams online. Data sent by your devices to ThingSpeak is instantly visualized by ThingSpeak. You can undertake online analysis and analyze data as it comes in with the option to run MATLAB code in ThingSpeak. For IoT systems that need analytics, ThingSpeak is frequently used for prototyping and proof-of-concept systems [16].
5. ***Fritzing:*** Fritzing is free circuit modeling software that can be installed on almost any computer. The good thing about this is that it doesn't use a lot of resources and can run on any computer, the default version of Fritzing doesn't have NodeMcu model v3, which means we have to download and install it's handmade [9].

6. ***Tinkercad:*** Tinkercad is a loose on line three-D modeling application that runs in an internet browser recognised for its simplicity and simplicity of use [17].

7. ***HTML:*** HyperText Markup Language, or HTML, enables online users to construct and organize sections, paragraphs, and links using elements, tags, and attributes [83].

8. ***CSS:*** A style sheet language called CSS is used to describe how a page produced in a markup language like HTML is presented [68].

9. ***Bootstrap:*** We use Bootstrap, a free and open-source web development platform, for the design. By providing a set of vocabulary for template designs, it is intended to simplify the process of creating responsive, mobile-first websites [107].
10. ***SQLite:*** C programming language database engine. It is a library that programmers incorporate into their apps rather than a stand-alone application. It so belongs to the group of embedded databases [137].
11. ***Lucidchart:*** With the use of the web-based diagramming tool Lucidchart, users may improve organizational structures, processes, and systems by working together visually to create, edit, and share charts and diagrams [155].
12. ***Modelio:*** A tool for building, documenting, and visualizing models of various systems, including software, business processes, and more is called Modelio. It enables users to develop, analyze, and simulate models by supporting well-known modeling methodologies including UML, BPMN, and SysML. Modelio helps with requirements capture, software component design, and code creation, among other tasks, which makes it easier to create and manage complicated systems [121].

3.4.2 Hardware realisation

We would normally connect the actual components based on the connections shown in the software after modeling our circuit in Fritzing . A well-liked tool for designing circuit diagrams and PCB layouts is Fritzing, which gives a visual depiction of the circuit's parts and connections.

Depending on our particular configuration, we would normally utilize wires, jumper cables, or breadboard connectors to link the elements in the actual circuit. We may carefully connect the proper pins or terminals of the components to create the required electrical connections by following the connections shown in the Fritzing software's circuit diagram Figure 3.7.

The physical result of the previously mentioned hardware connection operation is depicted graphically in Figure "connection." It offers a clear and thorough picture of the finished circuit by showcasing the real components and their connections. The image provides a visual guide that enables a more thorough comprehension of the physical configuration and layout of the associated gear. The specific positioning of the parts and their corresponding connections may be seen by looking at Figure 3.27 which can help with troubleshooting, analysis, or additional adjustments if needed.

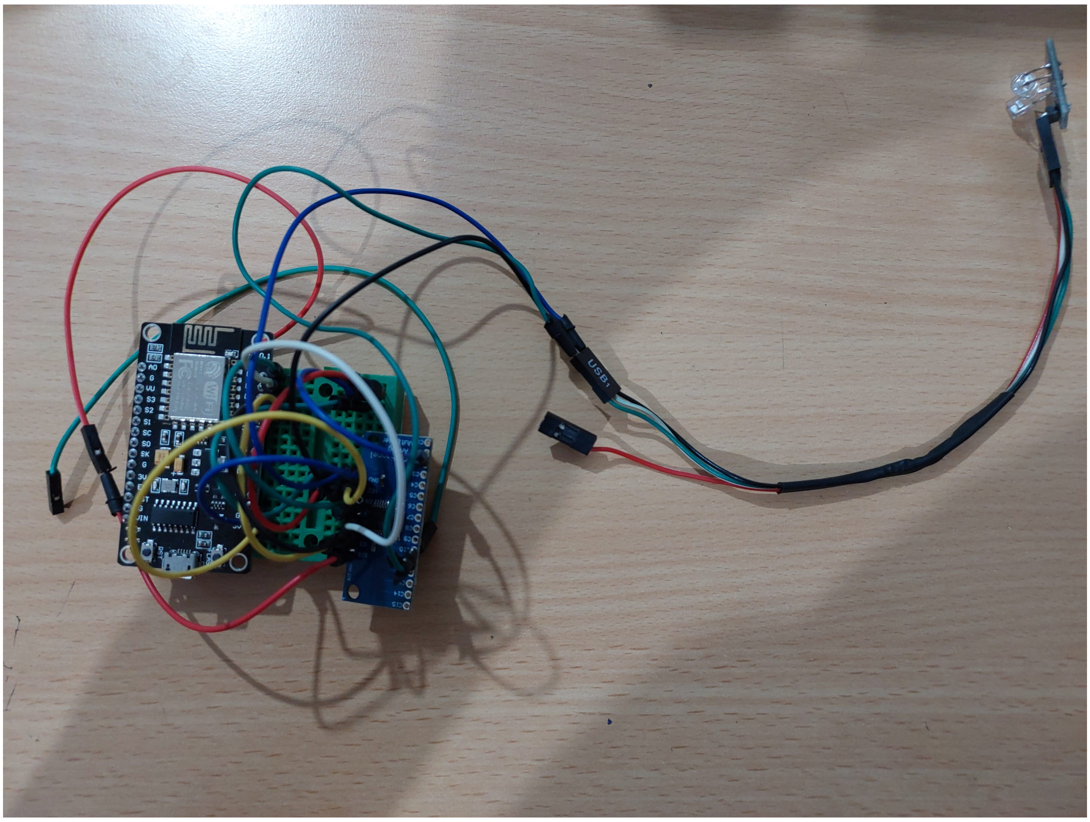


Figure 3.27: First connection of the hardware

3.4.3 Software realisation

Sketch for Arduino

We programmed our device using ArduinoIDE https://www.arduino.cc/en/software. We've employed a microcontroller (NodeMcu) with an integrated Wi-Fi module.

In order to use the Wi-Fi library, which was created utilizing the naming conventions and general functionality of the ArduinoWiFi library, we first had to call the ESP8266.h library.

include <ESP8266WiFi.h>

after that we had to call the thingspeak library :

include "*ThingSpeak.h*"

The next lines of code were added after that:

- **unsigned long Channel_ID = 1876612;:** Our Channel ID.
- **const char * myWriteAPIKey = "Z7SS3N2TRI2CSMIB";:** Our write API key.
- **char ssid[] = "Tenda_5E4F80";:** The name of the network to which we were logged in is this (Gateway).
- **char pass[] = "laid2021";:** The gateway password must also be written down, of course.
- **const int Field_Number_1 = 1;**
- **const int Field_Number_2 = 2;**
- **define S0 D0;:** Assign Multiplexer pin S0 connect to pin D0 of NodeMCU.
- **define S1 D1;:** Assign Multiplexer pin S1 connect to pin D1 of NodeMCU.
- **define S2 D2;:** Assign Multiplexer pin S2 connect to pin D2 of NodeMCU.
- **define S3 D3;:** Assign Multiplexer pin S3 connect to pin D3 of NodeMCU.
- **define SIG A0;:** Assign SIG pin as Analog output for all 16 channels of Multiplexer to pin A0 of NodeMCU.

The figure 3.28 represents the declaration part of the variables used. This is a block of code written in the C++ programming language. Here are some of the things that are happening in this code:

- Libraries are being imported at the top, specifically the OneWire library and the ThingSpeak library.
- Constants are being defined, such as the channel ID and the write API key for ThingSpeak, as well as the field numbers for the two fields that will be written to.
- The code is setting up WiFi connectivity by defining the SSID and password for the network to which the NodeMCU will be connected.
- The code is defining the pins that will be used to control a multiplexer and the analog input pin that will be used to read values from the multiplexer.
- The code is defining variables that will hold the values read from each of the 16 channels on the multiplexer, with different data types depending on the precision needed for the sensor value outputs.

- The code uses the OneWire library to communicate with the multiplexer and read values from each of its channels.
- The code uses the ThingSpeak library to send the values read from the multiplexer to a ThingSpeak channel via WiFi.

```
my_code
#include <OneWire.h>
#include "ThingSpeak.h"
OneWire oneWire(ONE_WIRE_BUS);
unsigned long Channel_ID = 1876612;
const char * myWriteAPIKey = "Z7SS3N2TRI2CSMIB";
char ssid[] = "Tenda_5E4F80";
char pass[] = "laid2021" ;
const int Field_Number_1 = 1;
const int Field_Number_2 = 2;
float resolution=3.3/1023;
String value = "";
float value_1, value_2;
int x, y;
WiFiClient  client;
```

Figure 3.28: Declaration part

Both setup() Figure 3.29 and loop() Figure 3.30, Figure3.31, Figure3.32, which are needed functions, must be used for the Arduino sketch to correctly compile.

The first function to run in the sketch is the void setup() Figure 3.29, which runs just once. Typically, it has statements that configure the pin modes.

```
my_code

void setup() {

    pinMode(S0,OUTPUT);
    pinMode(S1,OUTPUT);
    pinMode(S2,OUTPUT);
    pinMode(S3,OUTPUT);
    pinMode(SIG, INPUT);


  Serial.begin(115200);
  DS18B20.begin();
    WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);
  internet();
  Serial.println("Testing Dual Sensor data");
}
```

Figure 3.29: Setup function

This is the setup function that initializes the NodeMCU board and sets up various components and libraries. Here are some of the things that are happening in this code:

- The code sets the pins for the multiplexer as outputs and the SIG pin as an input.
- The code initializes the serial communication at a baud rate of 115200 for debugging and output purposes.
- The DS18B20 sensor is being initialized, which is a digital temperature sensor that uses the OneWire protocol.
- The code sets the WiFi mode to station mode and begins the ThingSpeak library using the WiFi client that was defined earlier in the code.
- The internet function is called, which is likely a user-defined function that handles the connection to the WiFi network.
- Finally, a message is printed to the serial monitor to indicate that the code is testing dual sensor data.

```
my_code

void loop() {
  internet();
  digitalWrite(S0,LOW); digitalWrite(S1,HIGH); digitalWrite(S2,LOW); digitalWrite(S3,LOW);
  float     vout=analogRead(SIG);
  vout=(vout*300)/1023;
     ThingSpeak.writeField(Channel_ID, Field_Number_1, vout, myWriteAPIKey);
   digitalWrite(S0,LOW); digitalWrite(S1,HIGH); digitalWrite(S2,LOW); digitalWrite(S3,HIGH);
   float reads[samp_siz], sum;
   long int now, ptr;
   float last, reader, start;
   float first, second, third, before, print_value;
   bool rising;
   int rise_count;
   int n;
   long int last_beat;
   for (int i = 0; i < samp_siz; i++)
     reads[i] = 0;
   sum = 0;
   ptr = 0;
   while(1)
   {
       n = 0;
     start = millis();
     reader = 0.;
     do
     {
       reader += analogRead(SIG);
       n++;
```

Figure 3.30: Loop part 1

```
my_code
    now = millis();
  }
  while (now < start + 20);
  reader /= n;  // we got an average

  sum -= reads[ptr];
  sum += reader;
  reads[ptr] = reader;
  last = sum / samp_siz;

  if (last > before)
  {
    rise_count++;
    if (!rising && rise_count > rise_threshold)
    {

      rising = true;
      first = millis() - last_beat;
      last_beat = millis();

      print_value = 60000. / (0.4 * first + 0.3 * second + 0.3 * third);
      Serial.print(print_value);
    ThingSpeak.writeField(Channel_ID, Field_Number_2, print_value, myWriteAPIKey);
     Serial.print('\n');
      third = second;
      second = first;
    }
  }
  else
```

Figure 3.31: Loop part 2

```
  else
  {
    // Ok, the curve is falling
    rising = false;
    rise_count = 0;
  }
  before = last;
  ptr++;
  ptr %= samp_siz;
}
 Serial.print("Sensor 2 : ");Serial.println(sensor2);
 Serial.print("Sensor 10 : ");Serial.println(print_value);
  delay(1000);

}
```

Figure 3.32: Loop part 3

This code appears to be designed to read data from a light-dependent resistor (LDR) connected to an Arduino board, calculate the frequency of the signal generated by the LDR, and send this data to the ThingSpeak IoT platform for further analysis.

The setup() function initializes the pins used by the LDR and sets up the serial communication and WiFi connection.

The internet() function attempts to connect to a WiFi network, and if the connection fails, it will keep attempting to connect every 5 seconds until a connection is established.

The loop() function contains the main program logic. It starts by calling internet() to ensure that a WiFi connection is established.

Then, it sets the pin configuration for the LDR, reads the voltage value from the LDR, converts it to a voltage level, and sends it to the ThingSpeak platform using the ThingSpeak.writeField() function.

Next, it sets the LDR pin configuration for frequency measurement and starts a loop to measure the frequency of the signal generated by the LDR. It takes multiple readings of the LDR voltage and calculates the average voltage level. It then checks whether the voltage level is rising or falling and determines the frequency of the signal based on the time between rising edges of the signal. Finally, it sends the frequency data to ThingSpeak using the ThingSpeak.writeField() function.

The loop then prints the values of the two sensors to the serial monitor and waits for 1 second before repeating the process.

Following the data's storage in ThingSpeak, we exported it as a CSV file to be used in forecasting.

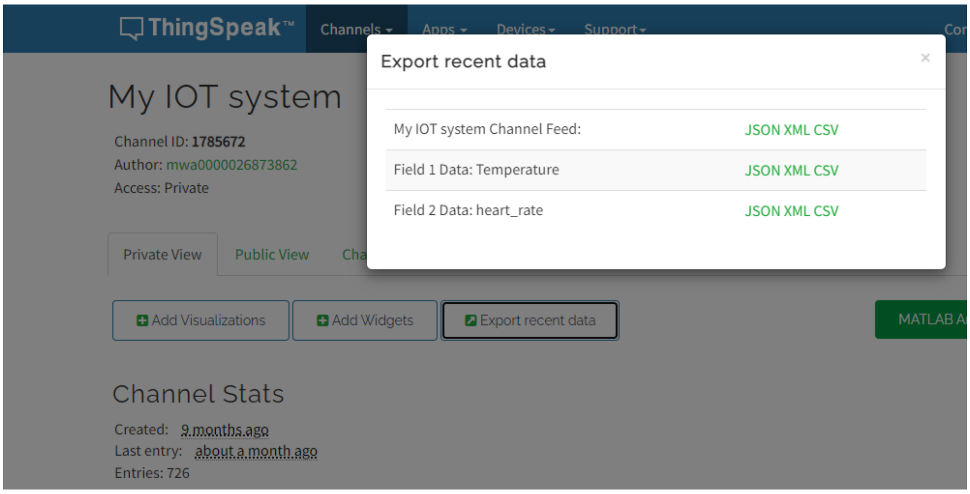


Figure 3.33: Exporting the ThingSpeak's time-series dataset

Implementation of regression models

Machine learning method

1 Define Problem.

Now the body temperature and heartbeat will be predicted using machine learning.

To do this, we implemented the models using Google Colab https://colab.research.google.com. We started by importing the required packages, which are shown in Figure 3.34.

```
[ ]  from sklearn.preprocessing import StandardScaler
     import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     from sklearn.linear_model import LinearRegression
     from sklearn.metrics import mean_squared_error, r2_score
     from sklearn.model_selection import train_test_split
     from sklearn import metrics
     from sklearn.linear_model import Ridge
     from sklearn.preprocessing import scale
     from sklearn.linear_model import Ridge, RidgeCV, Lasso, LassoCV
     from sklearn import datasets
     from sklearn.ensemble import GradientBoostingRegressor
     import seaborn as sns
     from numpy import asarray
     from numpy import mean
     from numpy import std
     from sklearn.datasets import make_regression
     from xgboost import XGBRegressor
     from sklearn.model_selection import cross_val_score
     from sklearn.model_selection import RepeatedKFold
     from lightgbm import LGBMRegressor
     from catboost import CatBoostRegressor
     from sklearn.ensemble import RandomForestRegressor
     from sklearn.tree import DecisionTreeRegressor
     from sklearn.neighbors import KNeighborsRegressor
     from sklearn.ensemble import AdaBoostRegressor
```

Figure 3.34: Necessary packages for machine learning models

It seems that we are importing several Python libraries related to data analysis and machine learning. Here is a brief overview of each library:

- ***pandas:*** a library for data manipulation and analysis. It provides data structures such as DataFrames for working with structured data.
- ***numpy:*** a library for numerical computing in Python. It provides arrays and functions for manipulating them.
- ***matplotlib:*** a plotting library for creating visualizations in Python.
- ***sklearn:*** a library for machine learning in Python. It provides a variety of algorithms for tasks such as classification, regression, clustering, and dimensionality reduction.
- ***seaborn:*** a library for data visualization based on matplotlib. It provides a high-level interface for creating informative and attractive statistical graphics.
- ***xgboost:*** an implementation of the gradient boosting algorithm for regression and classification problems.
- ***lightgbm:*** a gradient boosting framework that uses tree-based learning algorithms.

- ***catboost:*** a gradient boosting library that uses an ordered boosting algorithm.
- ***randomforestregressor:*** a decision tree-based ensemble algorithm for regression.
- ***decisiontreeregressor:*** a decision tree-based algorithm for regression.
- ***kneighborsregressor:*** a k-nearest neighbor algorithm for regression.
- ***adaboostregressor:*** an implementation of the AdaBoost algorithm for regression problems.

2 Prepare Data

We have downloaded two datasets the first for **body temperature** from https://data.world/laurie/skin-temperature and the second for **heart beat** from https://ecg.mit.edu/time-series/, a cloud-native data catalogue, and integrated it into Python code to train our model.

```
dataset = pd.read_csv('/content/drive/MyDrive/Datasets/Heart_Beat_Dataset.csv')
x = dataset['time'].values.reshape(-1, 1)
y = dataset['heartBeat'].values.reshape(-1, 1)
```

Figure 3.35: Heart beat Dataset import

```
from typing_extensions import dataclass_transform
data = pd.read_excel("/content/drive/MyDrive/Datasets/Dataset_Temperature.xlsx", na_values=" None")
```

Figure 3.36: Body temperature Dataset import

At this phase, we utilized a function that divides arrays or matrices into random train and test subsets, Figure 3.37 Figure 3.38 The imported dataset is divided into training and forecasting sets, with 50% of the measurements picked for training the forecasting models and the remaining 50% aimed for forecasting.

```
x_train,val_x,y_train,val_y= train_test_split(x,y,test_size=0.5, random_state=1)
```

Figure 3.37: Splitting heart beat dataset

```
x_train,val_x,y_train,val_y= train_test_split(x1,y1,test_size=0.5, random_state=1)
```

Figure 3.38: Splitting body temperature dataset

3 Training step

```
names=['LinearRegression','Ridge','Lasso','KNeighborsRegressor','DecisionTreeRegressor','RandomForestRegressor','GradientBoostingRegressor',
       'AdaBoostRegressor', 'CatBoostRegressor' 'LGBMRegressor', 'XGBRegressor']
models=[LinearRegression(),Ridge(),Lasso(),KNeighborsRegressor(),DecisionTreeRegressor(),RandomForestRegressor(),GradientBoostingRegressor(),
        AdaBoostRegressor(), CatBoostRegressor(), LGBMRegressor(),  XGBRegressor()]
Model=[]
RMSE=[]
R_sq=[]
Score =[]
a = -1
for name,model in zip(names,models):

  model.fit(x_train.reshape(-1, 1),y_train.reshape(-1, 1))
  Y_prd=model.predict(val_x.reshape(-1, 1))
  Model.append(name)
 # Score.append( model.score(x,val_Y))

  RMSE.append(np.sqrt(mean_squared_error(val_y.reshape(-1, 1),Y_prd.reshape(-1, 1))))
  a=r2_score(val_y.reshape(-1, 1),Y_prd.reshape(-1, 1))
  if a > 0 :
     R_sq.append(a)
  else :
     R_sq.append(-1*a)

  evaluation=pd.DataFrame({
                        'Model': Model,
                        'RMSE': RMSE,
                        'R_sq': R_sq
                        #'Score' :Score
                       })

print("following are the testing scores:")
print(evaluation)
```

Figure 3.39: Heart beat model performances

```
names=['LinearRegression','Ridge','Lasso','KNeighborsRegressor','DecisionTreeRegressor','RandomForestRegressor','GradientBoostingRegressor',
       'AdaBoostRegressor', 'CatBoostRegressor' 'LGBMRegressor', 'XGBRegressor']
models=[LinearRegression(),Ridge(),Lasso(),KNeighborsRegressor(),DecisionTreeRegressor(),RandomForestRegressor(),GradientBoostingRegressor(),
        AdaBoostRegressor(), CatBoostRegressor(), LGBMRegressor(),  XGBRegressor()]
Model=[]
RMSE=[]
R_sq=[]
Score =[]
a = -1
for name,model in zip(names,models):

  model.fit(x_train,y_train)
  Y_prd=model.predict(val_x)
  Model.append(name)
 # Score.append( model.score(x,val_Y))

  RMSE.append(np.sqrt(mean_squared_error(val_y,Y_prd)))
  a=r2_score(val_y,Y_prd)
  if a > 0 :
     R_sq.append(a)
  else :
     R_sq.append(-1*a)

  evaluation=pd.DataFrame({
                        'Model': Model,
                        'RMSE': RMSE,
                        'R_sq': R_sq
                        #'Score' :Score
                       })

print("following are the testing scores:")
print(evaluation)
```

Figure 3.40: Body temperature model performances

This is a Python code that evaluates the performance of several regression models on a validation dataset. Here's what it does step-by-step:

- Defines a list of regression model names and a list of corresponding model instances.
- Initializes empty lists to store the model names and evaluation metrics (RMSE and R-squared).

- Loops through each model in the list of model instances and fits it to the training data (x_train and y_train).
- Predicts the target variable for the validation data (val_x) using the fitted model and calculates the RMSE and R-squared.
- Appends the model name and evaluation metrics to the corresponding lists.
- Creates a pandas DataFrame to display the evaluation results. Prints the evaluation results.

4 Evaluate Algorithms.

The performance metrics for several machine learning models during the training phase for both body temperature and heartbeat datasets are provided in Table 3.1 and Table 3.2. The measures include R-squared ($R2$) score, Mean Square Error (MSE) and Root Mean Square Error (RMSE). The models include gradient boosting, adaboost, CatBoostRegressorLGBMRegressor, and XGBRegressor. They also contain linear regression, ridge regression, lasso regression, k-neighbors regressor, decision tree regressor, random forest regressor, and gradient boosting regressor.

Table 3.1: **Body temperature testing Scores**

Model	RMSE	MSE	R-squared
Linear Regression	0.254985	0.065	0.956732
Ridge Regression	0.254586	0.0648	0.950618
Lasso Regression	0.236307	0.0558	0.680573
K-Nearest Neighbors Regressor	0.201935	0.0407	0.227227
Decision Tree Regressor	0.313045	0.0979	1.949285
Random Forest Regressor	**0.176501**	**0.0311**	**0.062440**
Gradient Boosting Regressor	0.209001	0.0436	0.314625
AdaBoost Regressor	0.190277	0.0362	0.089626
CatBoostRegressorLGBMRegressor	0.188707	0.0356	0.071716
XGBRegressor	0.196732	0.0387	0.164804

The testing results for various machine learning models for predicting body temperature are displayed in the table below, labeled Table 3.1 Based on their propensity to calculate body temperature with reasonable accuracy from the supplied data, these models were assessed. RMSE (Root Mean Square Error) and R-squared are two evaluation metrics that are employed. Lower numbers suggest greater accuracy in the measurement of RMSE, which is the average difference between anticipated and measured body temperatures. The amount of the body temperature variation that the models can account for is expressed as a percentage using R-squared. **The Random Forest Regressor was the model that stood out among the others with the lowest RMSE, demonstrating its superior predictive ability.**
Figure 3.41 displays the training results.

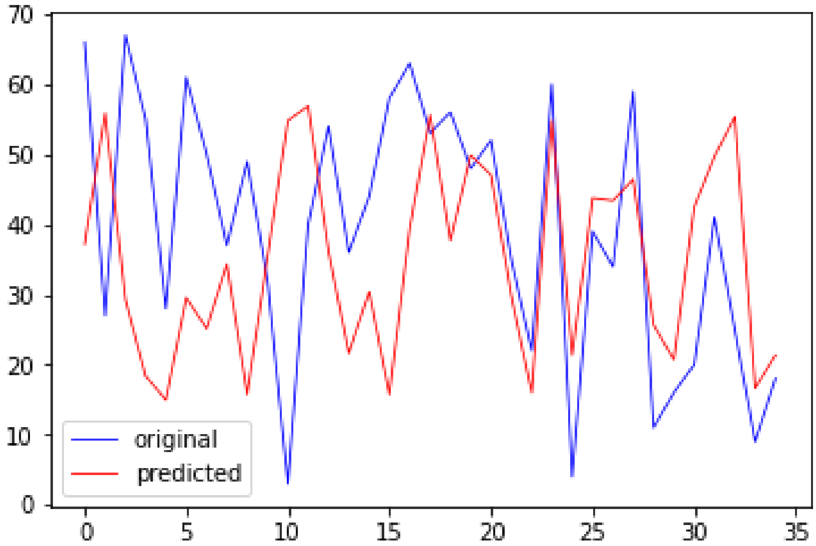


Figure 3.41: Results of the training (Body Temperature)

Table 3.2: **Heartbeat testing Scores**

Model	RMSE	MSE	R-squared
Linear Regression	0.216365	0.04681	0.013779
Ridge Regression	0.216365	0.04681	0.013779
Lasso Regression	0.216362	0.04681	0.013807
K-Nearest Neighbors Regressor	0.219234	0,0480	0.012545
Decision Tree Regressor	0.276081	0.0762	0.605726
Random Forest Regressor	0.232242	0.0539	0.136268
Gradient Boosting Regressor	0.215827	0.0465	0.018675
AdaBoost Regressor	0.214580	0.0460	0.029988
CatBoostRegressorLGBMRegressor	0.219439	0.0481	0.014441
XGBRegressor	0.213387	0,0455	0.040739

The testing results of different machine learning models for predicting heartbeat are shown in the table 3.2 that is provided. Based on how well these models performed in properly calculating heartbeat using the supplied data, they were evaluated. Root Mean Square Error (RMSE), Mean Square Error (MSE) and R-squared are the evaluation metrics taken into account. The average difference between the predicted and real heartbeat values is quantified by the RMSE measure, with smaller values indicating higher accuracy. R-squared, on the other hand, quantifies the percentage of the variance in heartbeat that the models are able to account for. **The XGBRegressor, out of all the models examined, had the lowest RMSE score, indicating that it had greater predictive performance for predicting heartbeats.**
Figure 3.42 displays the training results.

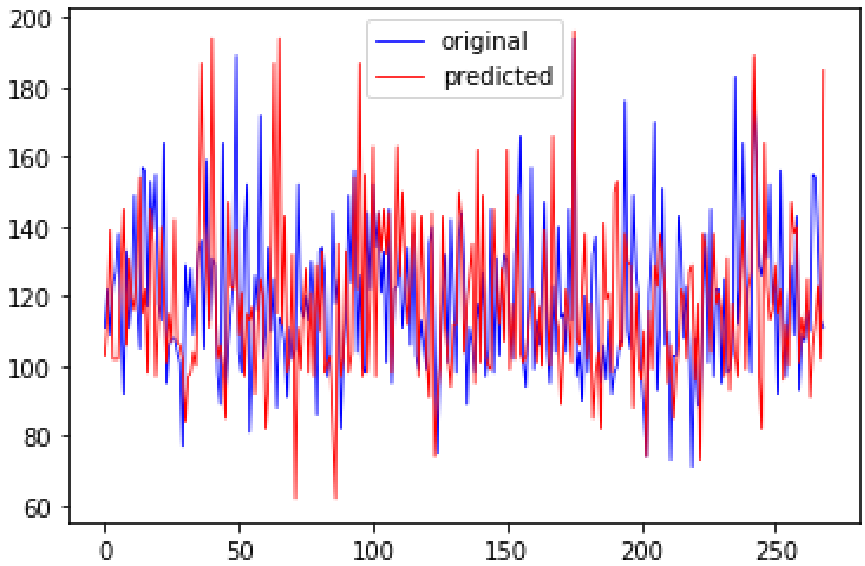


Figure 3.42: Results of the training (Heartbeat)

Deep learning method

1 Define Problem.

Now, deep learning will be used to forecast body temperature and heartbeat. The proposed model architecture design in the Figure 3.43

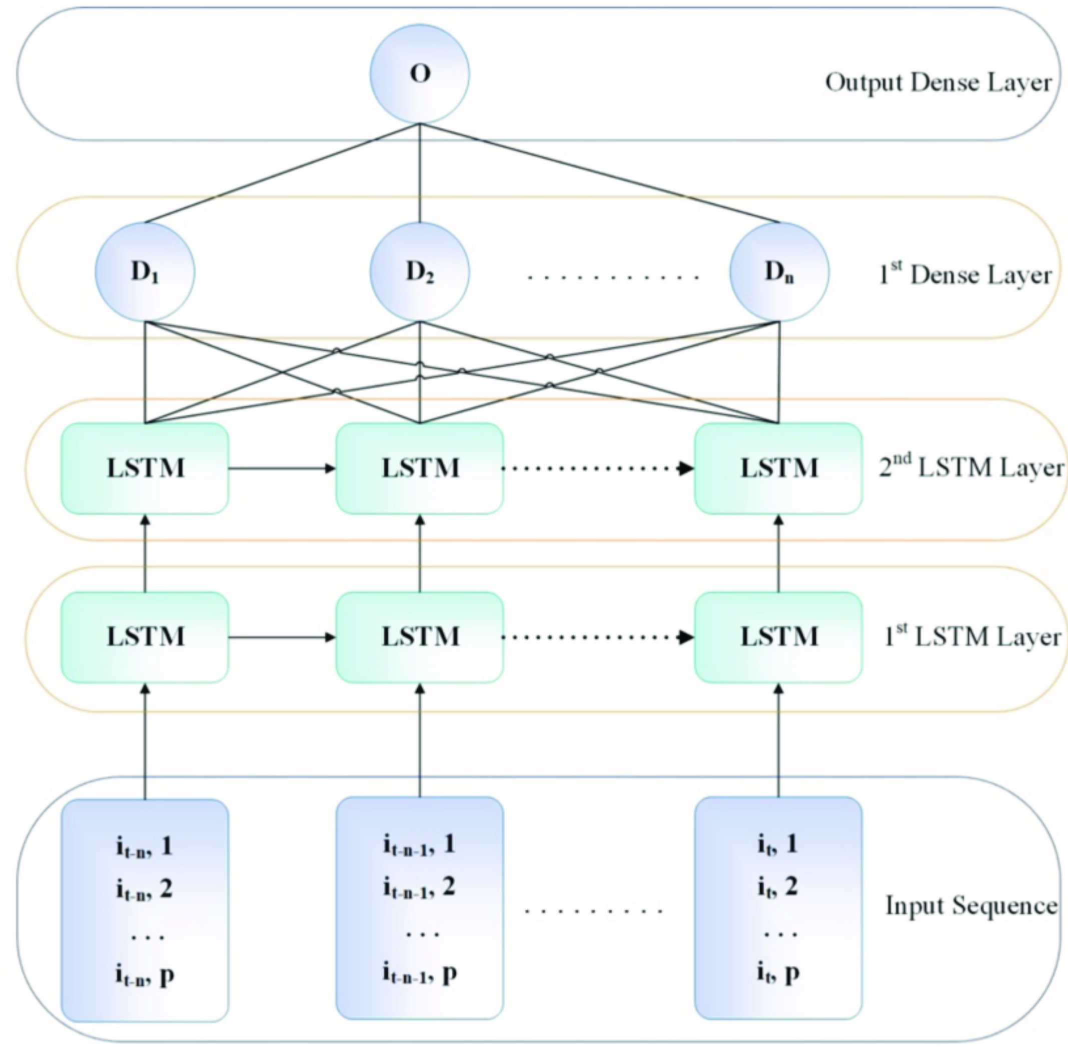


Figure 3.43: Proposed Model Architecture

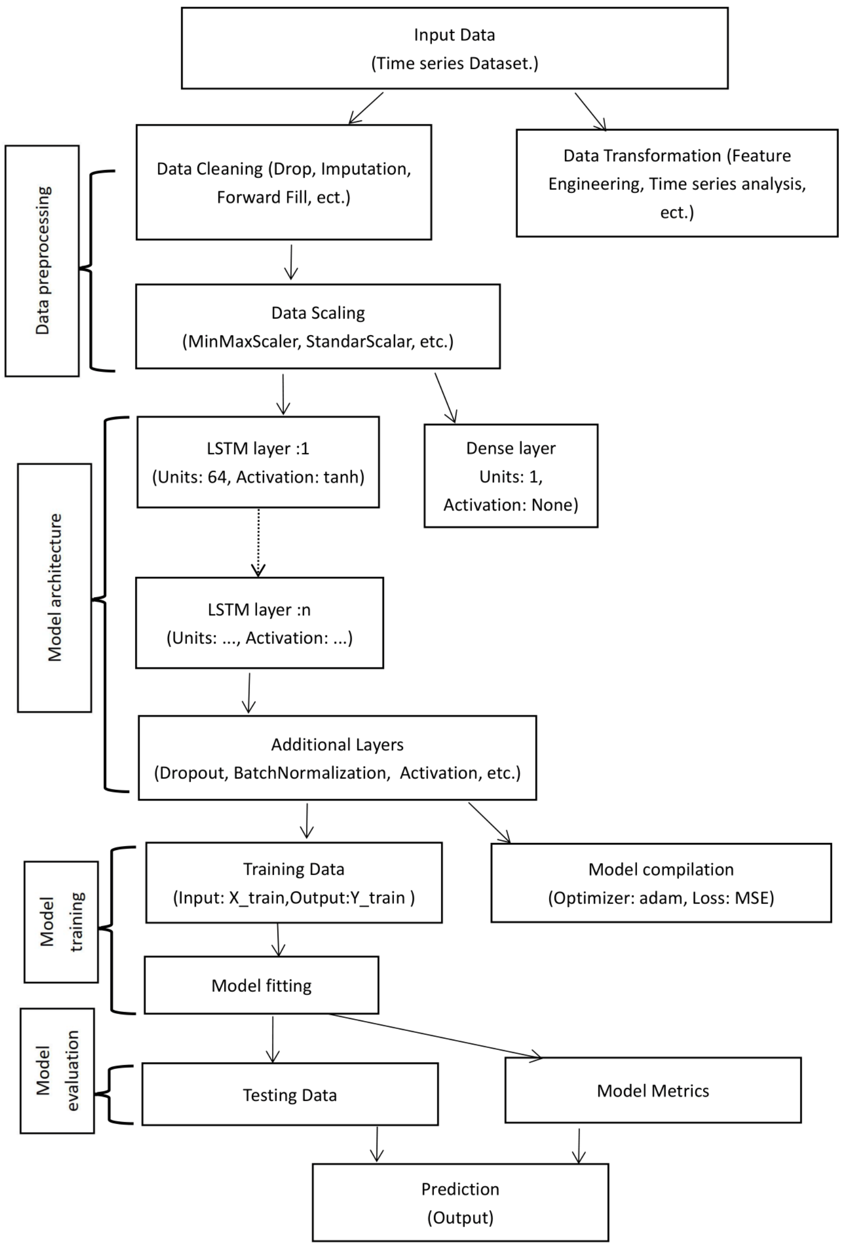


Figure 3.44: Execution steps

To do this, we implemented the model using Google Colab https://colab.research.google.com. We started by importing the required packages, which are shown in Figure 3.45.

```
import numpy as np
import pandas as pd
from keras.models import Sequential
from keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler
```

Figure 3.45: Necessary packages for deep learning model

- ***LSTM and Dense from keras.layers:*** These several layer types can be included in a neural network model. Long Short-Term Memory, or LSTM, is a recurrent neural network layer that is frequently applied to sequence data. A layer that is densely connected.
- ***MinMaxScaler from sklearn.preprocessing:*** It is used to scale data within a predetermined range (often 0 to 1).

2 Prepare Data

In this step we read files from drive and reshapes the 1D array into a 2D array with a single column.

```
data = pd.read_csv('/content/drive/MyDrive/Datasets/Heart_Beat_Dataset.csv', index_col='date')
scaler = MinMaxScaler(feature_range=(0, 1))
print(data.values)
scaled_data = scaler.fit_transform(data.values.reshape(-1, 1))
```

Figure 3.46: Heartbeat Dataset import

```
data = pd.read_csv('/content/drive/MyDrive/Dataset/temp.csv')
values = data['skin_temp'].values.reshape(-1,1)
```

Figure 3.47: Body temperature Dataset import

Splitting the data into training and test sets :

```
train_size = int(len(scaled_data) * 0.7)
train_data = scaled_data[:train_size]
test_data = scaled_data[train_size:]
```

Figure 3.48: Splitting heartbeat dataset

```
train_size = int(len(values) * 0.7)
train_data, test_data = values[:train_size,:], values[train_size:len(values),:]
```

Figure 3.49: Splitting Body temperature dataset

3 Training step

The training of Long Short-Term Memory (LSTM) networks for forecasting vital signs, with an emphasis on the prediction of the heartbeat signal and body temperature, is our ground-breaking contribution to the implementation phase. To the best of our knowledge, this is the first time LSTM has been used to predict vital signs. We intend to capture the complex temporal correlations and long-term patterns contained in vital sign data by using the capabilities of LSTM, providing precise and trustworthy forecasts. Our implementation makes use of IoT and AI technologies to incorporate real-time data collection from sensors, ensuring ongoing and ongoing monitoring of vital indicators. We illustrate the efficacy and promise of our LSTM-based strategy in enhancing healthcare outcomes and promoting proactive patient care through comprehensive experimentation and evaluation.

```
# Create the input and output sequences for the LSTM model
def create_sequences(data, seq_length):
    X, y = [], []
    for i in range(len(data) - seq_length):
        X.append(data[i:i + seq_length])
        y.append(data[i + seq_length])
    return np.array(X), np.array(y)

seq_length = 30
X_train, y_train = create_sequences(train_data, seq_length)
X_test, y_test = create_sequences(test_data, seq_length)

# Build the LSTM model
model = Sequential()
model.add(LSTM(units=64, input_shape=(seq_length, 1)))
model.add(Dense(units=1))
model.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
model.fit(X_train, y_train, epochs=50, batch_size=32)
```

Figure 3.50: Heart beat model training.

```
# Prepare the data for LSTM model
def create_dataset(dataset, look_back=1):
    dataX, dataY = [], []
    for i in range(len(dataset)-look_back):
        a = dataset[i:(i+look_back), 0]
        dataX.append(a)
        dataY.append(dataset[i + look_back, 0])
    return np.array(dataX), np.array(dataY)

look_back = 5
trainX, trainY = create_dataset(train_data, look_back)
testX, testY = create_dataset(test_data, look_back)

# Build the LSTM model
from keras.models import Sequential
from keras.layers import Dense, LSTM

model = Sequential()
model.add(LSTM(units=64, input_shape=(look_back, 1)))
model.add(Dense(units=1))
model.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
model.fit(trainX, trainY, epochs=50, batch_size=32)
```

Figure 3.51: Body temperature model training.

The function **'create_sequences'** takes in the data and **'seq_length'** as input and creates input-output sequences for the LSTM model. It iterates over the data and creates sequences of length **'seq_length'**. The input sequences X are formed by slicing the data from index **'i'** to **'i + seq_length'**, and the corresponding output values y are the data at index **'i +seq_length'**.

After that we initialize an LSTM model using the Sequential class. The model consists of an LSTM layer with 64 units and an input shape of **'(seq_length, 1)'**, where **'seq_length'** is the length of the input sequences and 1 indicates the number of features in each input data point. A dense layer with 1 unit is added after the LSTM layer. The model is compiled with the Adam optimizer and mean squared error (MSE) loss function.

And we train the LSTM model using the fit method. It takes **'X_train'** and **'y_train'** as inputs and trains the model for 50 epochs with a batch size of 32. The model learns to minimize the mean squared error loss between the predicted output and the actual output during training.

4 Evaluation step.

```python
# Evaluate the model on the testing data
test_loss = model.evaluate(X_test, y_test)
print('Test loss:', test_loss)

# Make predictions with the model
predictions = model.predict(X_test)

# Invert the scaling to get the actual values
predictions = scaler.inverse_transform(predictions)
y_test = scaler.inverse_transform(y_test)

# Visualize the predictions and actual values
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = [4, 4]
plt.xlabel('Time')
plt.ylabel('Heart rate')
plt.plot(predictions, label='Predictions')
plt.plot(y_test, label='Actual values')
plt.legend()
plt.show()
```

Figure 3.52: Heartbeat model evaluation

```python
# Evaluate the model on the testing data
test_loss = model.evaluate(testX, testY)
print('Test loss:', test_loss)

# Make predictions on test data
predictions = model.predict(testX)

# Scale the predictions back to original scale
predictions = scaler.inverse_transform(predictions)

# Plot the predicted values against the actual values
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = [4, 4]

plt.plot(predictions, label='Predictions')
plt.plot(y_test, label='Actual values')

plt.xlabel('Time')
plt.ylabel('Skin Temperature')
plt.legend(['Actual Values', 'Predicted Values'])

plt.show()
```

Figure 3.53: Body temperature model evaluation

In our work, the Mean Squared Error (MSE) for heartbeat prediction using the LSTM (Long Short-Term Memory) method was **0.0015** the results of prediction is showing in the Figure 3.54, and for body temperature prediction, it was **0.0197** the results of prediction is showing in the Figure 3.55.Since Mean Squared Error (MSE) and Root Mean Square Error (RMSE) are mathematically related, we may calculate the RMSE values using the MSE results you gave. According to the MSE values given, it seems that the machine learning strategy did not do as well as the deep learning approach for predicting heartbeat and body temperature. The lower MSE values with LSTM indicate that it was more accurate in identifying patterns and forecasting the situation of these vital indicators in the future.

The LSTM model would therefore be a good option for forecasting the future condition of vital indicators like heartbeat and body temperature given its superior performance versus the machine learning approach. Time-series prediction problems benefit from LSTM's capacity to recognize long-term dependencies and patterns in sequential data.

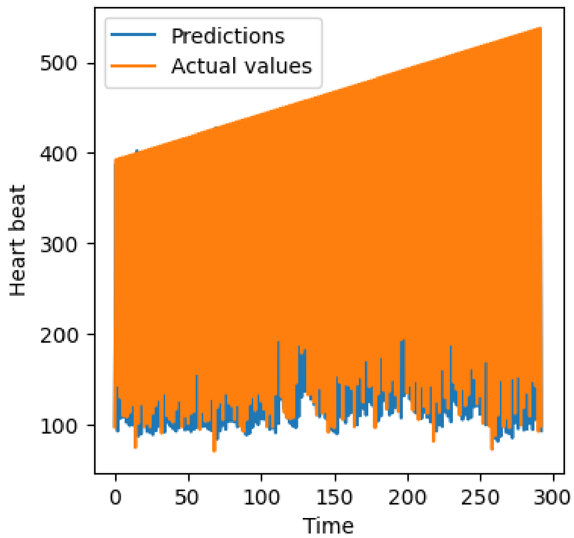


Figure 3.54: Prediction results (Heartbeat)

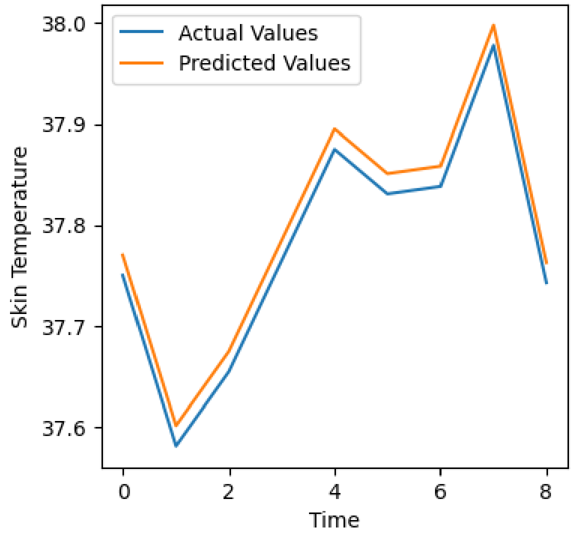


Figure 3.55: Prediction results (Body temperature)

We selected to use LSTM for prediction based on the encouraging outcomes from training LSTM models on vital sign datasets, highlighting our contribution in implementing and enhancing LSTM technique for improved accuracy and reliability in vital sign forecasting.

Integrating the Model into the Website

We will talk about incorporating a regression model into our website in this part. This integration tries to accurately forecast patient's future states. We can estimate patient outcomes and conditions based on pertinent data by integrating this potent model into our website.

django code (views)

```
import numpy as np
import pandas as pd
from keras.models import Sequential
from keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler
@login_required(login_url='doctorlogin')
@user_passes_test(is_doctor)
def my_view(request):
    # Read CSV file
    df = pd.read_csv('hospital/Dataset_Temperature.csv')
    data = df.to_dict('records')
    context = {'data': data}
    X = df['entry_id'].values.reshape(-1, 1)
    Y = df['field1']
    Y=Y.values.reshape(-1, 1)
    model = Sequential()
    model.add(LSTM(units=64, input_shape=(look_back, 1)))
    model.add(Dense(units=1))
    model.compile(optimizer='adam', loss='mean_squared_error')
    model.fit(trainX, trainY, epochs=50, batch_size=32)
    predictions = model.predict(X)
    print(Y.flatten())
    print(predictions)
    mlr_diffH = pd.DataFrame({'Actual value': Y.flatten(), 'Predicted value':
    predictions})
    mlr_diffH
    x_ax = range(len(Y))
    plt.xlabel('Time')
    plt.ylabel('Body Temperature')
    plt.plot(x_ax, predictions, lw=0.8, color="red", label="predicted")
    return render(request, 'hospital/my_view.html', context, plt.show())
```

Figure 3.56: Function for body temperature

```
@login_required(login_url='doctorlogin')
@user_passes_test(is_doctor)
def doctor_review_patent(request):

    df = pd.read_csv('hospital/Heart_Beat_Dataset.csv')
    data = df.to_dict('records')
    context = {'data': data}
    X = df['time'].values.reshape(-1, 1)
    Y = df['heartBeat']
    Y=Y.values.reshape(-1, 1)
    model = Sequential()
    model.add(LSTM(units=64, input_shape=(look_back, 1)))
    model.add(Dense(units=1))
    model.compile(optimizer='adam', loss='mean_squared_error')
    model.fit(trainX, trainY, epochs=50, batch_size=32)
    predictions = model.predict(X)
    print(Y.flatten())
    print(predictions)
    mlr_diffH = pd.DataFrame({'Actual value': Y.flatten(), 'Predicted value':
    predictions})
    mlr_diffH
        x_ax = range(len(Y))
    plt.xlabel('Time')
    plt.ylabel('HeartBeat')
    plt.plot(x_ax, predictions, lw=0.8, color="red", label="predicted")

    return render(request, 'hospital/doctor_review_patent.html', context, plt.show())
```

Figure 3.57: Function for heartbeat

Django views Figure 3.56 and Figure 3.57 are essential in the integration of a regression model into a website that forecasts a patient's future state. They manage HTTP requests, retrieve patient data, ready it for predictions using the regression model, apply the results to update the database, and present the expected state in the website's user interface. By ensuring smooth communication between the model and the website, this integration improves user experience overall while giving consumers useful insights.

django code (URLs)

```
urlpatterns +=[
    path('doctor-dashboard', views.doctor_dashboard_view,name='doctor-dashboard'),

    path('doctor-patient', views.doctor_patient_view,name='doctor-patient'),

      path('doctor_review_patent/', views.doctor_review_patent,
      name='doctor_review_patent'),

    path('my_view/', views.my_view, name='my_view'),
    path('doctor-view-patient', views.doctor_view_patient_view,name='doctor-view-
    patient'),
    path('doctor-view-discharge-
    patient',views.doctor_view_discharge_patient_view,name='doctor-view-discharge-
    patient'),

    path('doctor-appointment', views.doctor_appointment_view,name='doctor-
    appointment'),
    path('doctor-view-appointment', views.doctor_view_appointment_view,name='doctor-
    view-appointment'),
    path('doctor-delete-
    appointment',views.doctor_delete_appointment_view,name='doctor-delete-
    appointment'),
    path('delete-appointment/<int:pk>', views.delete_appointment_view,name='delete-
    appointment'),

      path('doctor-approve-patient', views.doctor_approve_patient_view, name='doctor-
      approve-patient'),
    path('approve-patient2/<int:pk>/', views.approve_patient2_view, name='approve-
    patient2'),
    path('reject-patient2/<int:pk>/', views.reject_patient2_view, name='reject-
    patient2'),

]
```

Figure 3.58: URLs

URL routing Figure 3.58 is crucial for integrating a regression model into a website that uses Django to forecast a patient's future state. Incoming requests are mapped to the appropriate views using specified URL patterns provided in the Django project's URL setup. Django makes sure that the right view is called whenever a user reaches the prediction endpoint by setting up the necessary URL patterns and tying them to view functions or classes that execute the prediction logic. As a result, the model and website can work together seamlessly, making it possible to anticipate the patient's future state with accuracy using the patient ID that has been provided.

django code (HTML)

```
  <div class="panel panel-primary">
    <div class="panel-heading">
      <h6 class="panel-title">Your Total Patient List</h6>
    </div>
    <table class="table table-hover" id="dev-table">
      <thead>
        <tr>
          <th>Name</th>
          <th>Profile Picture</th>
          <th>Symptoms</th>
          <th>Contact</th>
          <th>Address</th>

          <th>Heart Beat</th>
          <th>Body Temperature</th>
          <th>Cloud</th>
    </tr>
      </thead>
      {% for p in patients %}
      <tr>
        <td> {{p.get_name}}</td>
        <td> <img src="{% static p.profile_pic.url %}" alt="Profile Pic" height="40px"
        width="40px" /></td>
        <td>{{p.symptoms}}</td>
        <td>{{p.mobile}}</td>
        <td>{{p.address}}</td>

        <td>   <a href="/doctor_review_patent"> Show </a> </td>
        <td>   <a href="/my_view"> Show </a> </td>
           <td>   <a href="https://thingspeak.com/login?skipSSOCheck=true"> More </a>
           </td>
        </tr>
      {% endfor %}
    </table>
  </div>
</div>
```

Figure 3.59: Doctor view patient

In a Django online application, the HTML code for the doctor's view Figure 3.59 of patient data normally consists of a clean, organized layout. In order to convey the pertinent patient information, it has elements like tables, headings, and form fields. A patient's ID, name, age, medical history, and current condition may all be seen in the table. Furthermore, buttons or links can be included to allow the doctor to obtain more in-depth data or carry out tasks, such examining the patient's anticipated future state. The HTML code makes sure that patient data is presented in an understandable and structured way, enabling doctors to quickly study and evaluate the patient's condition and make decisions based on the available information.

```
<html>
<head>
    <title>Heart Beat Data</title>
</head>
<body>
    <h1>Heart Beat Data</h1>

    <table>
        <thead>
            <tr>
                <th>ID</th>
                 <th></th>
                 <th></th>  <th></th>
                <th>Time</th>
                <th></th>
                 <th></th>  <th></th>
                <th>Heart Beat</th>
            </tr>
        </thead>
        <tbody>
            {% for row in data %}
            <tr>
                <td>{{ row.time }}</td>
                    <th></th>
                 <th></th>  <th></th>
                <td>{{ row.date }}</td>
                    <th></th>
                  <th></th>  <th></th>
                <td>{{ row.heartBeat }}</td>
            </tr>
            {% endfor %}
        </tbody>
    </table>
</body>
</html>
```

Figure 3.60: HTML code for heartbeat values

```
<html>
<head>
    <title>Body Temperature Data</title>
</head>
<body>
    <h1>Body Temperature Data</h1>

    <table>
        <thead>
            <tr>
                <th>Time</th>
                       <th></th>
                 <th></th>  <th></th>
                <th>ID</th>
                       <th></th>
                 <th></th>  <th></th>
                <th>Body Temperature</th>
            </tr>
        </thead>
        <tbody>
            {% for row in data %}
            <tr>
                <td>{{ row.created_at }}</td>
                       <th></th>
                 <th></th>  <th></th>
                <td>{{ row.entry_id }}</td>
                       <th></th>
                 <th></th>  <th></th>
                <td>{{ row.field1 }}</td>
            </tr>
            {% endfor %}
        </tbody>
    </table>
</body>
</html>
```

Figure 3.61: HTML code for body temperature values

The HTML page for temperature forecasting Figure 3.61 displays both historical data and expected values, whereas the HTML page for heartbeat Figure 3.60 forecasting displays both historical data and predicted values.

3.4.4 Prototype realisation

Printing the model

Preparation, printing, and post-processing are the three key steps in the 3D printing process using an Atmega printer. Using slicing software, the 3D model is created and transformed into machine-readable instructions during the preparation stage. To guarantee proper printing, the printer has been calibrated and adjusted. The printer uses heated nozzles and filament deposition to carry out the instructions layer by layer during printing. To ensure successful printing, the process must be closely watched. The intended outcome is then achieved by performing any necessary finishing touches, such as support removal or surface refinement, during the post-processing stage. The main processes in using an Atmega printer for 3D printing are summarized in this abstracted overview. the following figures Figure 3.62, Figure 3.63 shows the step of printing:

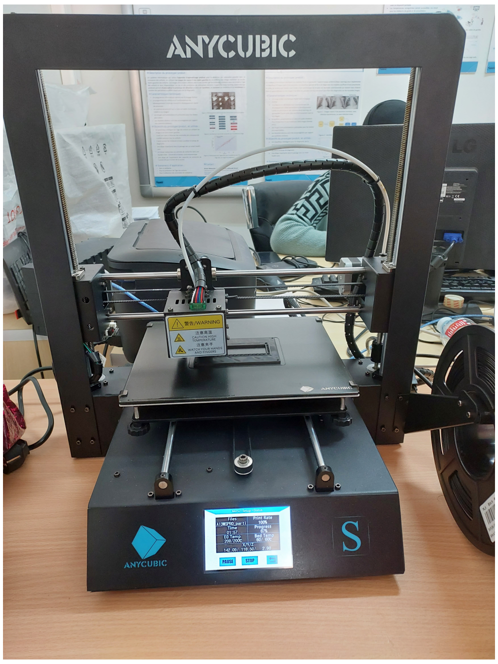


Figure 3.62: First part

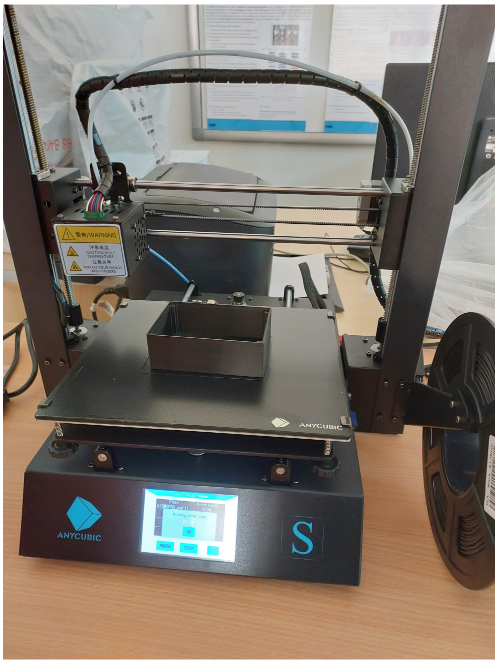


Figure 3.63: Second part

Figure 3.64 shows the time needed to print the wearable device.

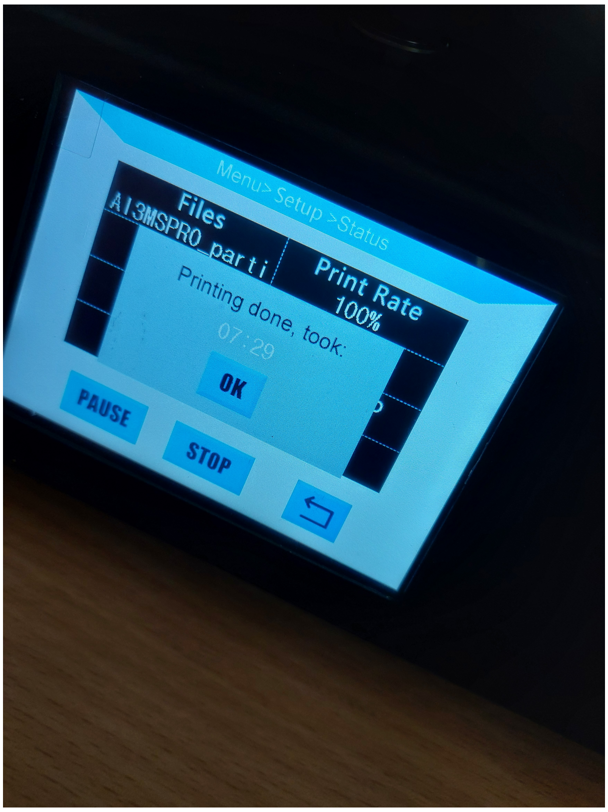


Figure 3.64: Time of printing

Figure 3.65 shows printing result.

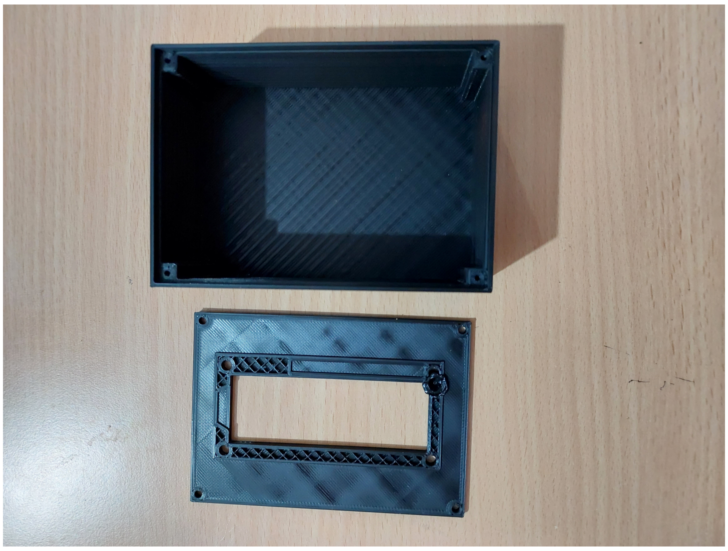


Figure 3.65: The prototype

Figure 3.66 shows the final result.

Figure 3.66: The final result

3.5 Conclusion

This chapter has presented a clear explanation of the concept behind our project as well as step-by-step instructions on the tools and techniques needed to realize it in both the hardware and software components. We have also seen how we developed our wearable device and gathered data. subsequently saved and retrieved as a time-series data set. Then we saw how regression methods was implemented, how models were run and compared, and how the best model was selected based on how well it fit the dataset and produced the best results.

The next chapter will demonstrate how to implement the same procedure using our datasets.

CHAPTER 04

EXPERIMENTATION AND RESULTS

Chapter 4

Experimentation and results

4.1 Introduction

In this chapter, we'll finish our project by designing the website and wearable technology that will help us succeed. We will be able to gather time-series data on body temperature and heartbeat using the wearable device, which we will utilize to predict future values. The website will offer a user interface for displaying and analyzing the gathered data, enabling us to derive useful insights. We will have successfully realized our project by the end of this chapter.

4.2 Obtained results

4.2.1 The hardware part

Hardware details:

The wearable smart device has sensors that can assess heart beat and body temperature. The gadget has a 1.3-inch OLED screen with a 240 by 240 pixel resolution.

The device can connect to the web application using wireless wifi technology. For everyday use and use while participating in sports, the device is made to be waterproof and dustproof.

Basic hardware specifications:

- accurate temperature sensor
- Heart beat monitor for precise heart beat determination
- OLED screen for data display that is crisp and clear Wifi technology to access the web application
- Design that is dust- and water-resistant battery with a long life and simple charging.

Figure 4.1: The wearable device

4.2.2 The software part

Logo

The Figure 4.2 below represents the logo featured on our Website.

Figure 4.2: The system's logo

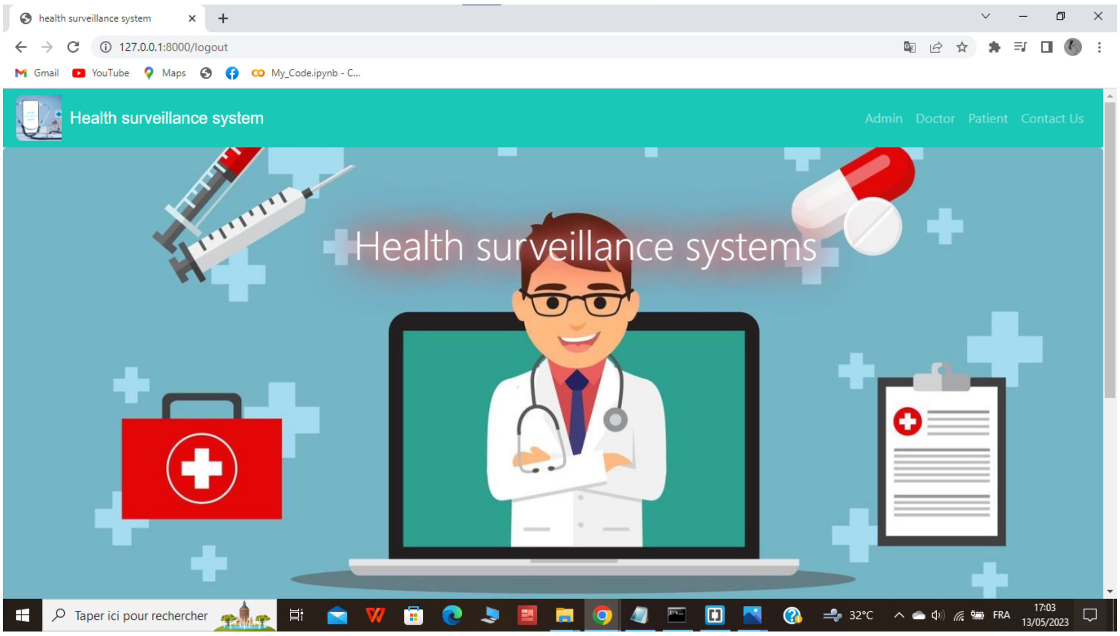


Figure 4.3: Home page

A health surveillance system's home page Figure 4.3 has login links for doctors, patients, and administrators so they can access their individual accounts. For users to contact system support personnel, it also has a contact us feature. The website could give a brief overview of the main functions and offerings of the system, like telehealth visits and remote patient monitoring. Users can manage their health and wellbeing with the assistance of their healthcare providers by visiting the home page, which acts as the main entry point to the system's services.

1-Admin Panel

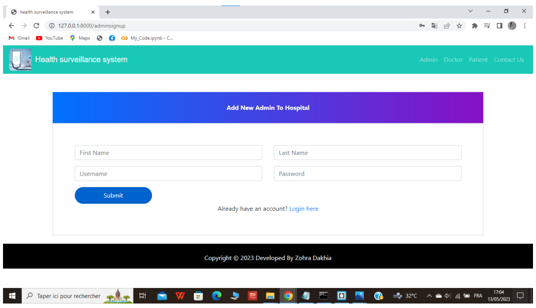


Figure 4.4: Sign Up

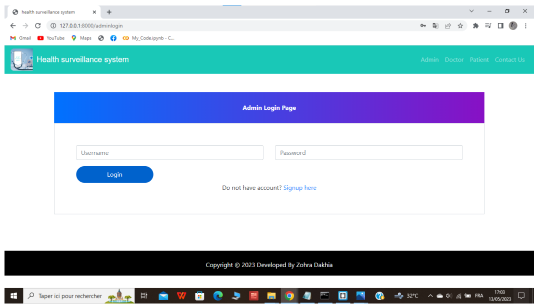


Figure 4.5: Login

The admin sign up Figure 4.4 and login Figure 4.5 pages are designed to ensure that only authorized personnel can access and manage the health surveillance system. The login page requires administrators to enter their login credentials before accessing their personalized dashboard, while the sign up page enables new administrators to create their account and obtain access to the system [7]. Together, these pages offer a secure and effective way for system administrators to manage the system and collect data.

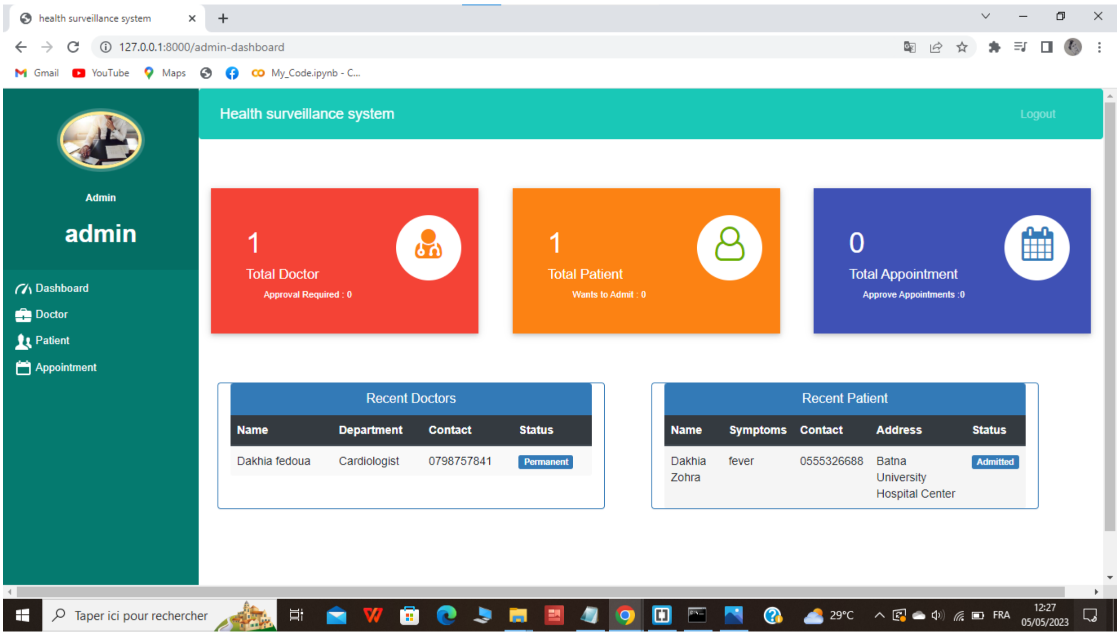


Figure 4.6: Admin dashboard

An administrator can handle numerous system components on the admin dashboard page Figure 4.6 of our Website, which offers a quick overview of the system's operations. User administration, system preferences, reports, assistance support, and analytics are some of the dashboard's key features. Administrators can effectively administer the system and make sure that healthcare practitioners and patients have access to the resources they require by using the dashboard, which gives them quick access to crucial data and functions.

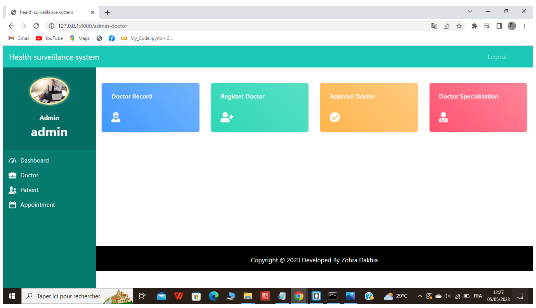


Figure 4.7: Doctors administration

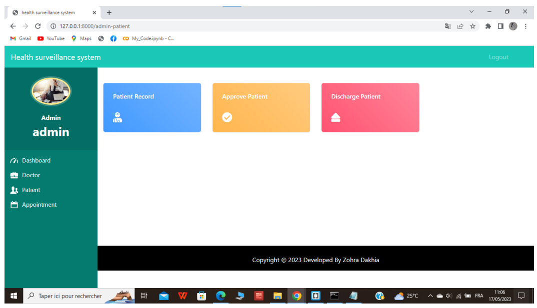


Figure 4.8: Patients administration

In our Website, the administrator (both doctors and patients) is in charge of managing user accounts, authorizing requests, updating data, and customizing system settings, among other administrative tasks. The administrator might also have access to more complex functions, like the ability to create reports and support system users. By overseeing system operations, the administrator contributes to ensuring that the system is running smoothly and that patients and healthcare professionals have access to the tools they need to maintain their health and wellness.

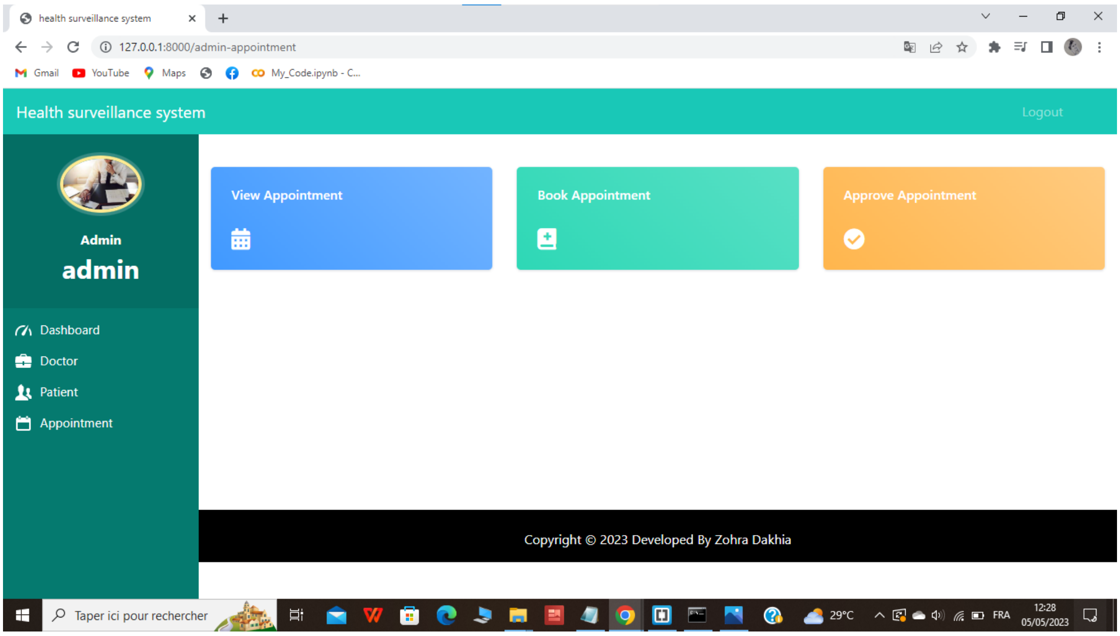


Figure 4.9: Admin appointment

An important tool for the administrator of our Website is the appointment management page Figure 4.9. The administrator can set up, postpone, or cancel appointments using this overview of forthcoming appointments. The health monitoring system ensures that doctors and patients may properly manage their health and wellness by effectively scheduling appointments.

2-Doctor Panel

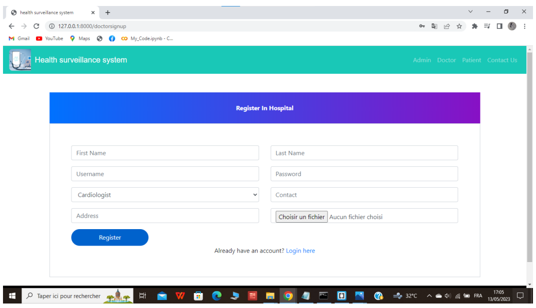


Figure 4.10: Sign Up

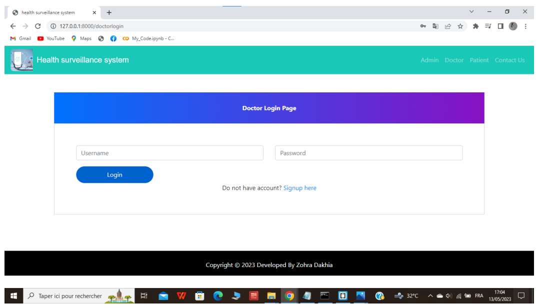


Figure 4.11: Login

In our Website, the login and signup pages for doctors are crucial components that offer secure access to the system and permit doctors to manage their information. Doctors must input their username and password on the login screen Figure 4.11, which may also have extra security measures to guard against unauthorized system access.

Doctors can make an account and gain access to the system by entering their personal and professional information on the sign up page Figure 4.10. They can log in to the system and manage their information after creating an account, including changing their profile, seeing patient data, and making appointments.

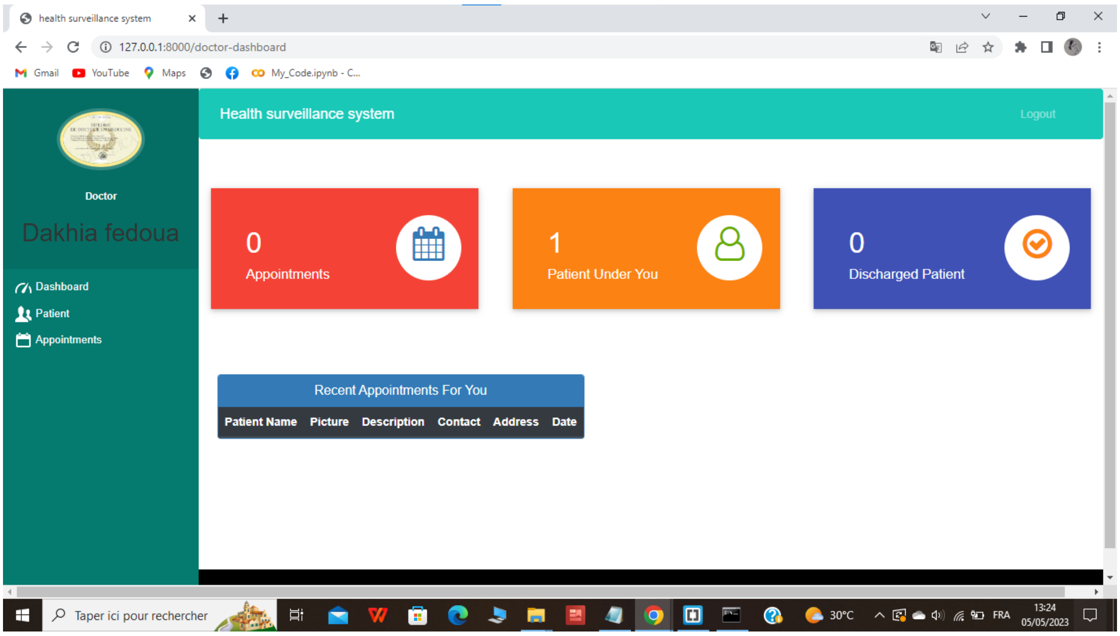


Figure 4.12: Doctor dashboard

In our Website the doctor dashboard is created to be an intuitive and user-friendly interface that gives clinicians instant access to crucial patient data. Doctors can more effectively and efficiently manage their patients' treatment by using the dashboard.

Typically, the dashboard has a patient list function that enables clinicians to view a list of all of their patients along with their medical histories. This feature gives clinicians a summary of the health state of their patients, including their medical background, current prescriptions, and other crucial data.

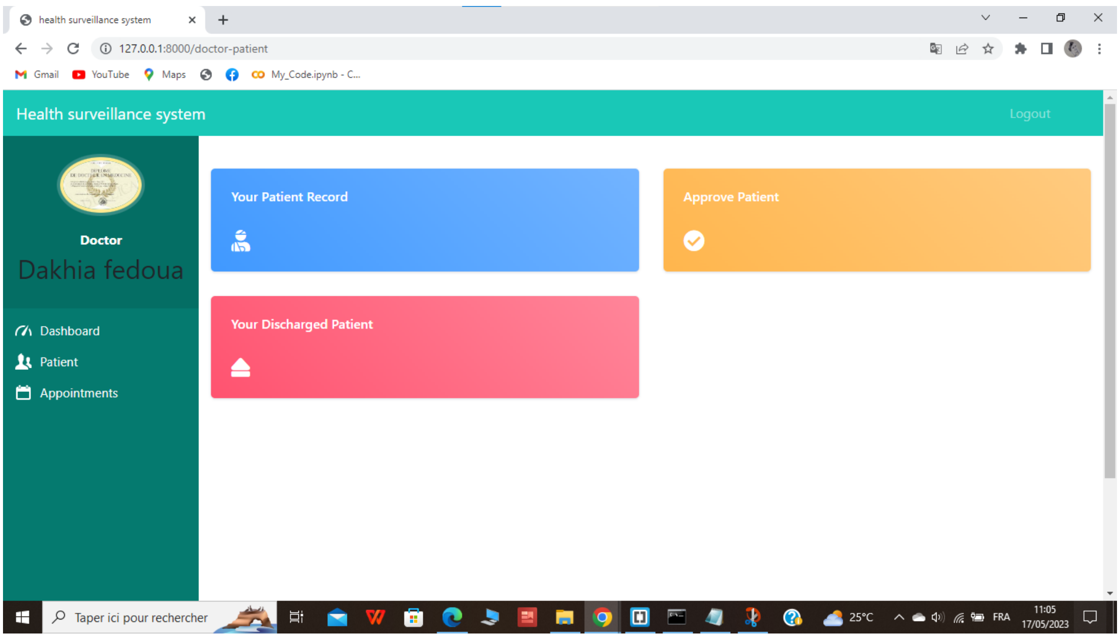


Figure 4.13: Patients state

Our Website's patient states page provides one button for accessing to real-time vital sign data, historical trends, and alternatives for accessing patient records and other button to releasing the patient.

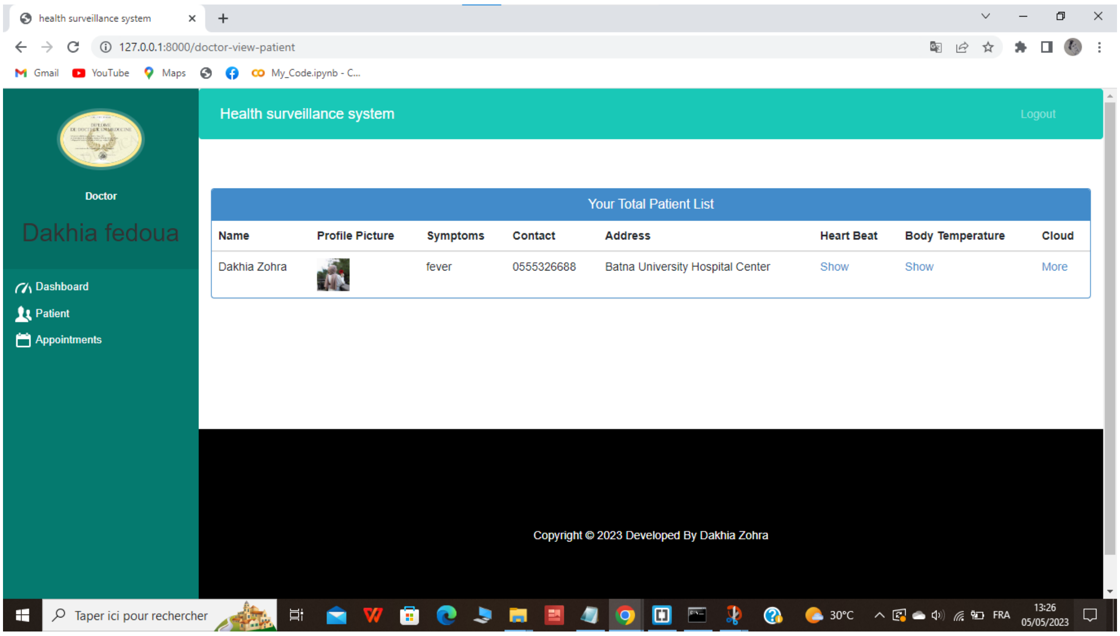


Figure 4.14: Patients monotoring

In our website, a patient monitoring interface often contains the output of sensors or wearable gadgets that can gather real-time data on a patient's health state, such as heart rate and body temperature. The technology receives the data wirelessly and processes it before sharing it with healthcare professionals and storing it for later use.

Doctors can check the health data of patients in real-time, track changes over time, and also estimate the future state of the considered patient thanks to an integration between the patient monitoring system and the patient dashboard. Healthcare professionals can also access the data to remotely monitor patients, spot any health issues, and take appropriate action.

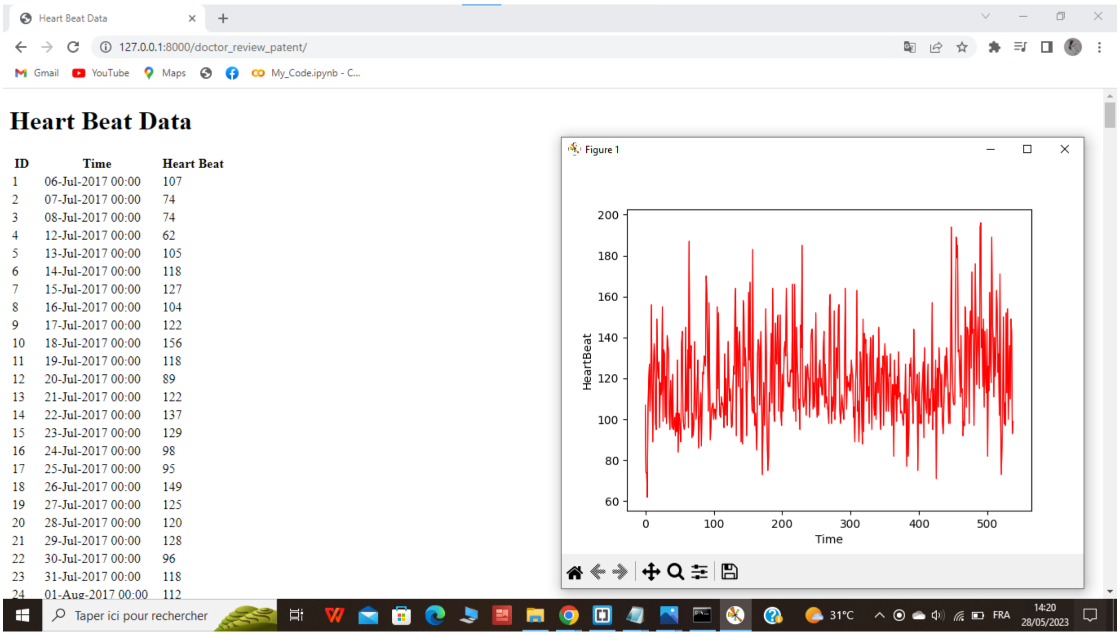


Figure 4.15: Heartbeat data

Two crucial elements of a patient monitoring system in a health surveillance system are the measurement and predicting of heartbeat data.

Heartbeat data measuring is employing sensors or wearable technology to gather real-time information on a patient's heartbeat and other vital indications. The patient monitoring system receives the data after which it may be examined and used to spot any irregularities or changes in the patient's heartbeat.

On the other hand, heartbeat data forecasting uses LSTM to make predictions about future heartbeat patterns based on existing data. This can assist medical professionals in spotting potential health issues before they arise and taking preventative measures to avoid them. The patient monitoring system often employs a combination of historical data and real-time data from the patient's present condition to properly estimate heart rate data. The data is analyzed using AI techniques to find patterns that can be utilized to forecast future heart rate trends. Healthcare professionals can then utilize these forecasts to decide on patient care with more knowledge. as showing in the Figure 4.15.

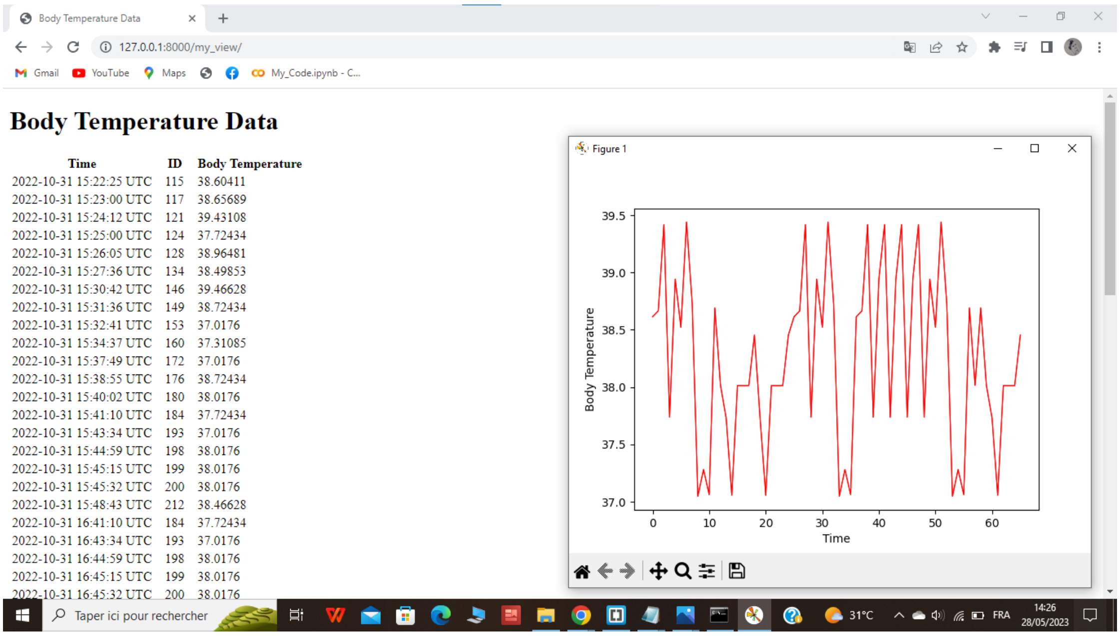


Figure 4.16: Body temperature data

An important part of a patient monitoring system in our Website is anticipating and measuring body temperature.

Body temperature data measurement entails employing sensors or wearable technology to gather information in real-time about a patient's body temperature. The information is then sent to the patient monitoring system, where it may be examined and used to spot any anomalies or temperature changes in the patient.

On the other hand, body temperature data forecasting uses LSTM to make predictions about future body temperature patterns based on existing data. This can assist medical professionals in spotting potential health issues before they arise and taking preventative measures to avoid them. The patient monitoring system often employs a combination of historical data and real-time data from the patient's present condition to properly estimate body temperature data. The data is analyzed using AI techniques to find trends that can be utilized to forecast future body temperature patterns. Healthcare professionals can then utilize these forecasts to decide on patient care with more knowledge. as showing in the Figure 4.16.

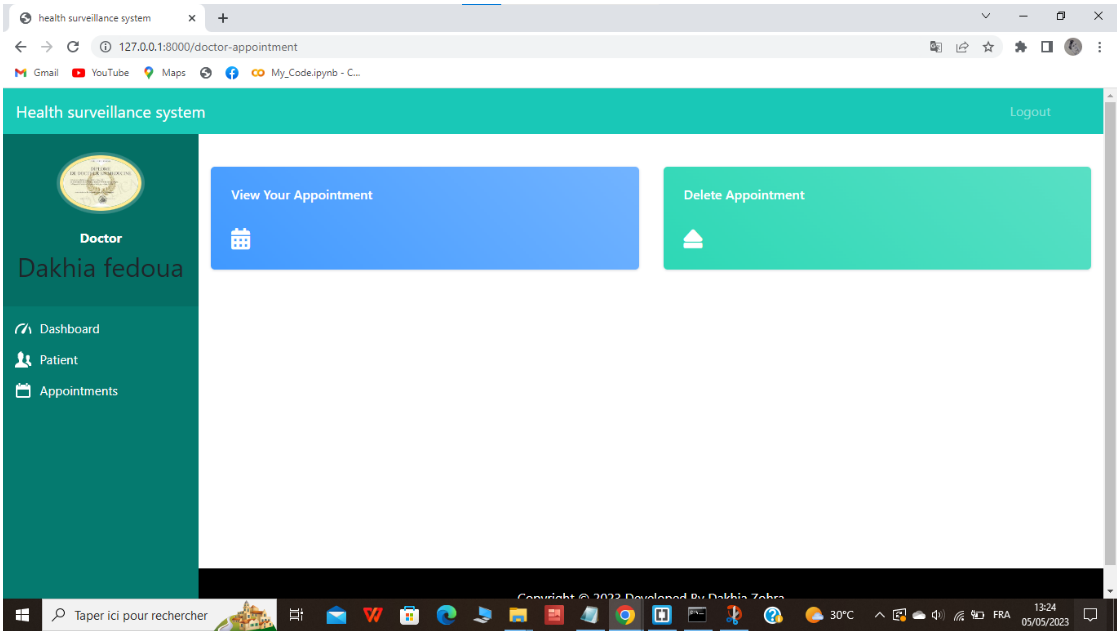


Figure 4.17: Doctor appointment

The appointment scheduling Figure 4.17 tool is a critical component in our Website, allowing doctors to efficiently manage patient appointments. The tool includes a calendar view of all scheduled appointments for a specific time period, as well as features like patient check-in, waitlist management, and appointment status tracking. By utilizing these features, doctors can better organize their time and provide improved patient care, resulting in reduced wait times, increased patient satisfaction, and a more effective use of resources.

3-Patient Panel

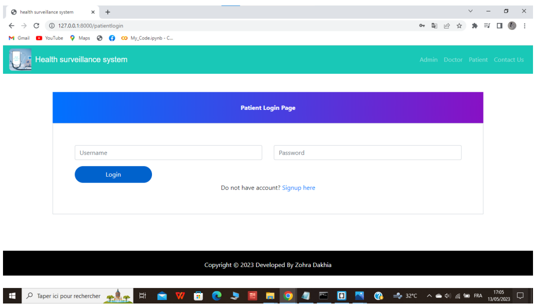


Figure 4.18: Sign Up

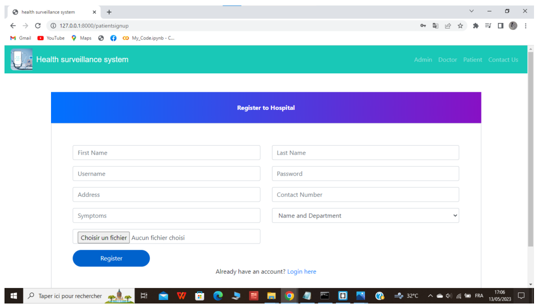


Figure 4.19: Login

In our Website, the login and signup pages for patients are crucial components that offer secure access to the system and permit patients to control their information. Patients must input their username and password on the login page Figure 4.19, which may also have extra security measures to guard against illegal system access.

Patients can create an account and gain access to the system by entering their personal and medical information on the sign up page Figure 4.18. They can control their information in the system after creating an account, including changing their profile, checking their medical records, and making appointments.

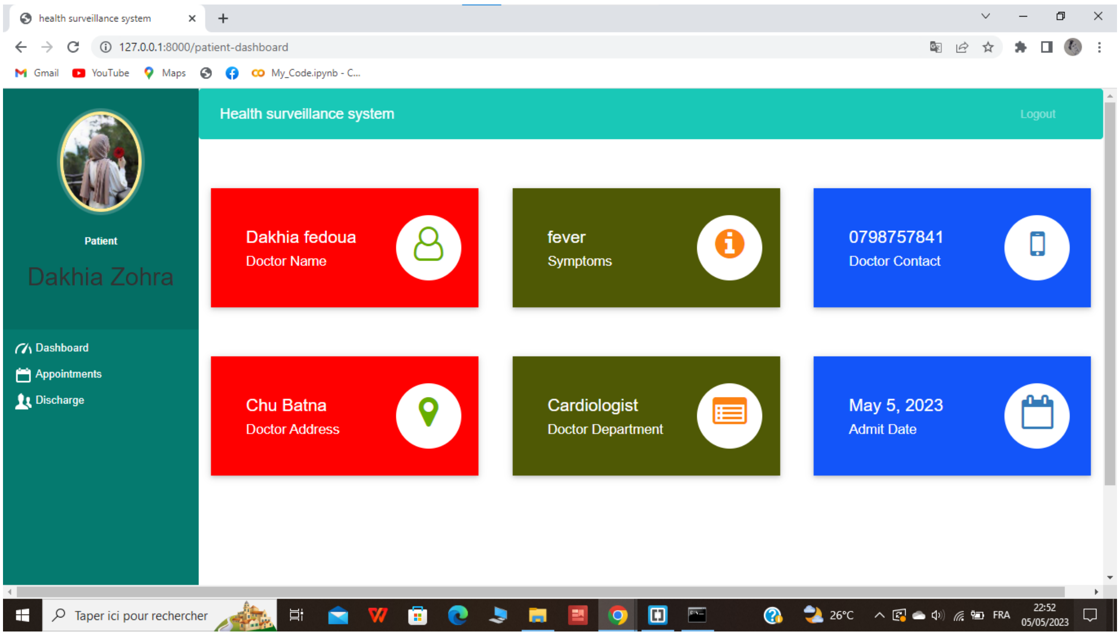


Figure 4.20: Patient dashboard

Security and privacy features are an essential component of a patient dashboard. The dashboard should be made to comply with legal standards for data privacy and security, and patients' private health information should be kept confidential and secure. The sharing of a patient's health information with other healthcare professionals or family members may also be under the patient's control. So as showing in the Figure 4.20 the patient can retrieve his medical information.

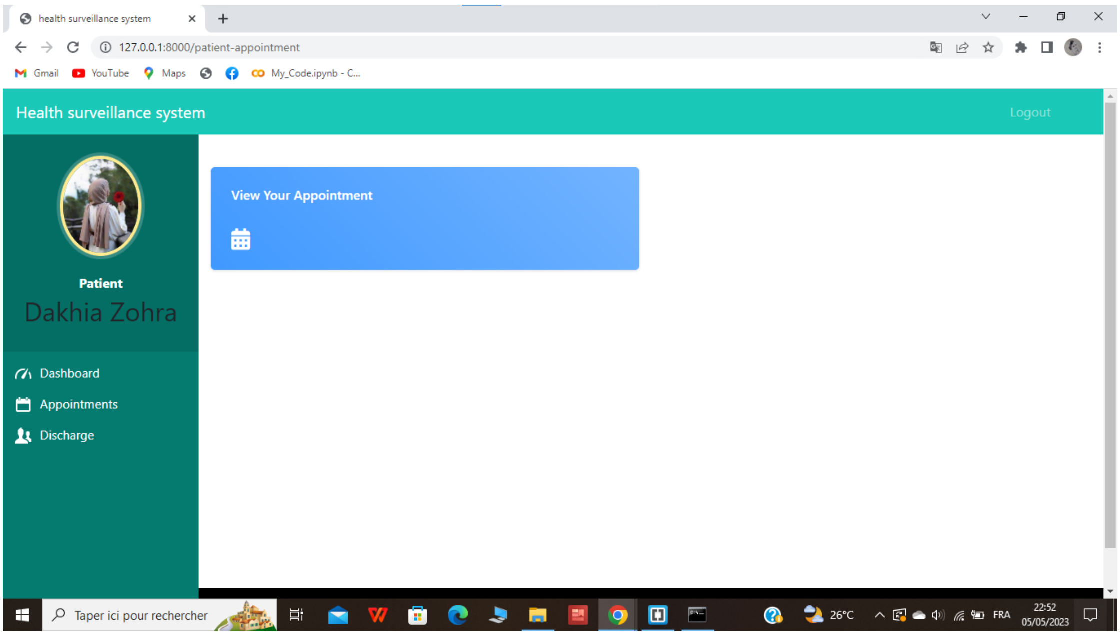


Figure 4.21: Patient appointment

The patient appointment page Figure 4.21 is a crucial component of our Website since it gives patients the ability to book, change, or cancel appointments with healthcare professionals while also giving doctors a full picture of their daily schedules.

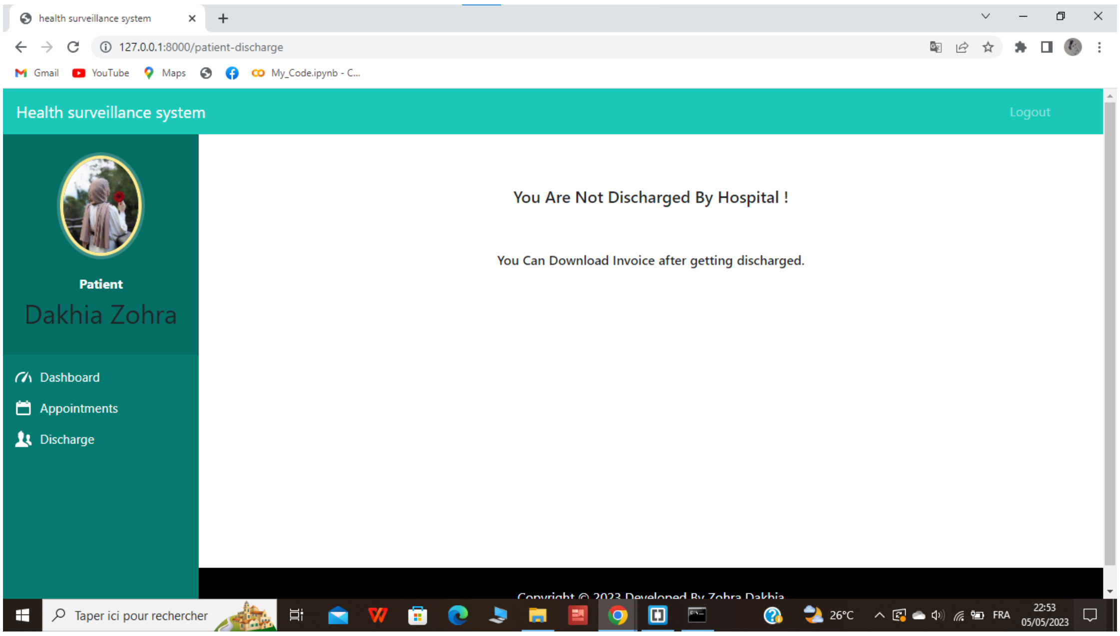


Figure 4.22: Patient discharge

A crucial component of our website that enables healthcare practitioners to effectively manage patient discharges and maintain correct record-keeping is the patient discharger page Figure 4.22.

4-About us Panel

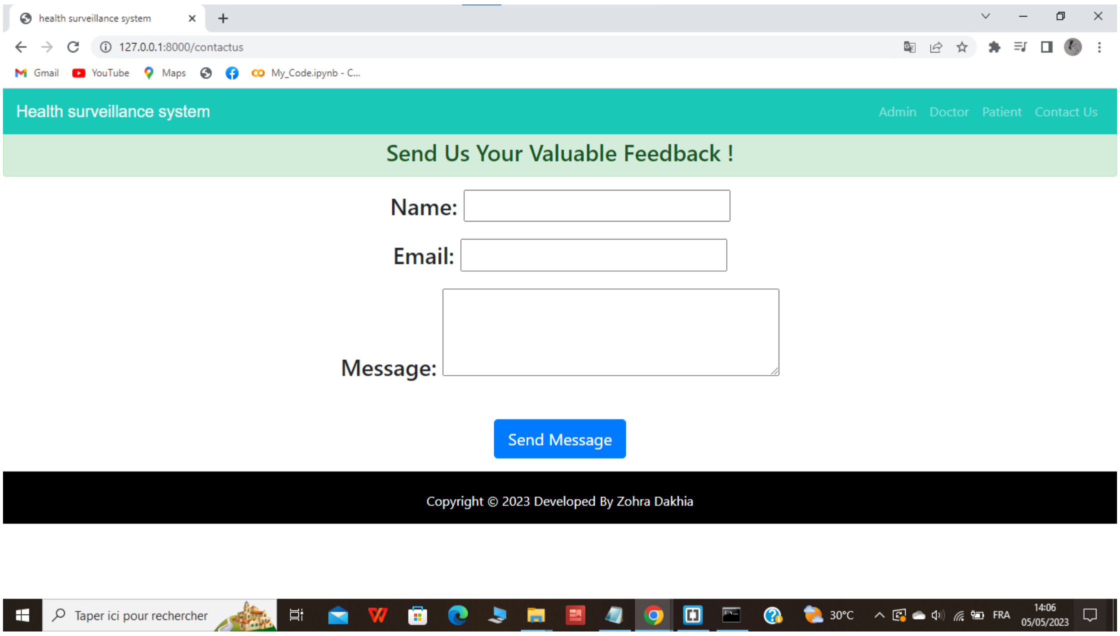


Figure 4.23: About Us

Users can contact healthcare providers through message on the "About Us" page Figure 4.23 with questions and for support, as well as information about our Website and its creators.

4.3 Comparaison with related work

As a way to conclude this section, Table 4.1 contrasts our assertion with one from a comparable work. As comparative metrics, we used RMSE and the total number of problems.

Method	MSE (HeartBeat)	MSE (Body Temperature)	work
LSTM (Proposed)	0.0015	0.0197	our work
Statistical Model	0.0123	0.0256	[33]
Deep Neural Network	0.0052	0.0214	[116]
ARIMA	0.0098	0.0231	[172]
Random Forest Regressor	\	0,0706	[112]

Table 4.1: Comparison of Different Approaches for the two vital Sign Prediction

4.4 Conclusion

The development of the final Website and the wearable device are among the project's final results, which were described in this chapter. Our efforts to build and execute a thorough health surveillance system have culminated in these accomplishments. The finished website offers medical professionals a user-friendly interface for gaining access to real-time patient data, making appointments, keeping patient records, and fostering communication and teamwork. In addition, the hardware gadget guarantees precise and trustworthy measures of vital signs, enabling continuous monitoring and early detection of any anomalies or changes in the patient's health.

CHAPTER 05

CONCLUSION AND PERSPECTIVE

Chapter 5

Conclusion and perspective

This thesis introduces a novel health surveillance method that offers an innovative approach to predict vital signs. By utilizing advanced techniques and methodologies, this research makes a valuable contribution to the field by providing a unique and effective method for monitoring and predicting crucial physiological indicators. The thesis consists of several chapters that collectively advance knowledge and the design of a health surveillance system. The introductory chapter establishes the context by highlighting the significance of a health surveillance system in improving patient outcomes and healthcare delivery. The background chapter conducts a thorough review of relevant literature, laying the foundation for understanding and hypotheses. The design and implementation chapter outlines the systematic procedure employed to develop the health surveillance system, including decision-making processes, integration of IoT technologies, and rationale for design choices. The experimentation and results chapter presents empirical findings, evaluating the system's performance, predictive model effectiveness, and data analysis accuracy. Finally, the conclusion and perspective chapter summarizes the major findings and contributions, highlighting the implications of the novel health surveillance method for healthcare surveillance and suggesting avenues for future research. This thesis significantly contributes to the healthcare domain by introducing a novel approach that has the potential to transform healthcare practices through accurate vital sign prediction and proactive interventions.

In future work, a primary priority is the creation of an extensive health surveillance system. This entails extending the system to cover a wider range of vital signs and health indicators, improving its capacity to offer a comprehensive picture of a person's health. The accuracy and robustness of vital sign forecasts can also be increased by investigating the integration of other predictive models, such as deep learning or ensemble approaches.an other important area to explore our system is the use of alternative communication technologies, such as GSM (Global System for Mobile Communications), in the health surveillance device. Integrating GSM or other wireless communication methods can provide flexibility and expand the system's connectivity options beyond relying solely on Wi-Fi. Additionally, creating a mobile app with a user-friendly interface can improve accessibility and usefulness for both patients and healthcare professionals. By putting these innovations into practice, a more useful and user-friendly health surveillance system will be developed, ultimately enhancing healthcare outcomes.

Bibliography

[1] Python. https://www.python.org/. Accessed: May 2, 2023.

[2] What is a sensor? https://www.electronicshub.org/different-types-sensors/. Accessed on 2 april 2023.

[3] The anycubic i3 mega 3d printer. https://www.lesimprimantes3d.fr/comparateur/imprimante3d/anycubic/i3-mega/, 2023.

[4] Arduino. https://arduino.fr.softonic.com/, May 2023.

[5] Artificial intelligence. https://ospreydata.com/ai-ml-models-101-what-is-a-model/, May 2023.

[6] Body temperature dataser. https://data.world/laurie/skin-temperature, 2023.

[7] Chatgpt. https://chat.openai.com/?model=text-davinci-002-render-sha, 2023.

[8] colab. https://colab.research.google.com/, 2023.

[9] fritzing. https://fritzing.org/, 2023.

[10] Healthcare. https://marketbusinessnews.com/financial-glossary/health-care, May 2023.

[11] Heart beat dataser. https://ecg.mit.edu/time-series/, 2023.

[12] Machine learning types. https://www.analyticsvidhya.com/blog/2021/03/everything-you-need-to-know-about-machine-learning/, 2023.

[13] Machine learning vs deep learning. https://www.analyticsvidhya.com/blog/2021/03/everything-you-need-to-know-about-machine-learning/, May 2023.

[14] Reinforcement learning. https://robotecture.com/tech/5-most-used-types-of-machine-learning/, 2023.

[15] Reinforcement learning figure. https://http://www.datasciencelovers.com/machine-learning/machine-learning-types/, 2023.

[16] thingspeak. https://thingspeak.com/, 2023.

[17] tinkercad. https://www.tinkercad.com/, 2023.

[18] Transfer learning figure. https://madhuramiah.medium.com/deep-learning-using-resnets-for-transfer-learning-d7f4799fa863, 2023.

[19] Types of healthcare. https://en.wikipedia.org/wiki/Health_care, May 2023.

[20] What is the internet of things (iot) ? https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT, 2023.

[21] Ajay Agrawal, Joshua S. Gans, and Avi Goldfarb. Artificial intelligence: The ambiguous labor market impact of automating prediction. *The Journal of Economic Perspectives*, 33(2):31–50, 2019.

[22] Atra Akandeh and Fathi M. Salem. Slim lstm networks: Lstm_6 and lstm_c6. In *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 630–633, 2019.

[23] Fatima Alshehri and Ghulam Muhammad. A comprehensive survey of the internet of things (iot) and ai-based smart healthcare. *IEEE Access*, 9:3660–3678, 2020.

[24] Jafar Alzubi, Anand Nayyar, and Akshi Kumar. Machine learning from theory to algorithms: an overview. In *Journal of physics: conference series*, volume 1142, page 012012. IOP Publishing, 2018.

[25] Matthew A. Andersson. Chronic disease at midlife: Do parent-child bonds modify the effect of childhood ses? *Journal of Health and Social Behavior*, 57(3):373–389, 2016.

[26] Xiangtian Bai, Ming Zhong, Yonghong Liu, Yafan Hu, and Jun Ma. Quantitative evaluation of smart bed comfort in different typical conditions during resting. 2023.

[27] Prarthana Bhattacharyya, Chengjie Huang, and Krzysztof Czarnecki. Ssl-lanes: Self-supervised learning for motion forecasting in autonomous driving. In *Conference on Robot Learning*, pages 1793–1805. PMLR, 2023.

[28] Manit Bhusal, Juan Calderon, and Hyun Jung Cho. Opinion mining of newspaper articles using natural language processing: Pilot test using texts on indian river lagoon. *Florida Scientist*, 84(2/3):pp. 226–231, 2021.

[29] BingLin Bie, WeiLin Xu, and YongGang Lv. Liquid metal-based textiles for smart clothes. *Science China Technological Sciences*, pages 1–19, 2023.

[30] Grant Blashki, Greg Armstrong, Helen Louise Berry, Haylee J. Weaver, Elizabeth G. Hanna, Peng Bi, David Harley, and Jeffery Thomas Spickett. Preparing health services for climate change in australia. *Asia Pacific Journal of Public Health*, 23(2):133S–143S, 2011.

[31] V Bonny, A Maillard, C Mousseaux, L Plaçais, and Q Richier. Covid-19: physiopathologie d'une maladie à plusieurs visages. *La Revue de médecine interne*, 41(6):375–389, 2020.

[32] Ron Brookmeyer and Jiangang Liao. Statistical modelling of the aids epidemic for forecasting health care needs. *Biometrics*, 46(4):1151–1163, 1990.

[33] Emery N Brown, Yong Choe, Harry Luithardt, and Charles A Czeisler. A statistical model of the human core-temperature circadian rhythm. *American Journal of Physiology-Endocrinology and Metabolism*, 279(3):E669–E683, 2000.

[34] W. A. Brownell. Theoretical aspects of learning and transfer of training. *Review of Educational Research*, 6(3):281–290, 1936.

[35] Jason Brownlee. *Introduction to time series forecasting with python: how to prepare data and develop models to predict the future*. Machine Learning Mastery, 2017.

[36] Benjamin P Chapman, Feng Lin, Shumita Roy, Ralph HB Benedict, and Jeffrey M Lyness. Health risk prediction models incorporating personality data: Motivation, challenges, and illustration. *Personality Disorders: Theory, Research, and Treatment*, 10(1):46, 2019.

[37] Lingkun Chen, Wenxin Chen, Lu Wang, Chencheng Zhai, Xiaolun Hu, Linlin Sun, Yuan Tian, Xiaoming Huang, and Lizhong Jiang. Convolutional neural networks (cnns)-based multi-category damage detection and recognition of high-speed rail (hsr) reinforced concrete (rc) bridges using test images. *Engineering Structures*, 276:115306, 2023.

[38] Taoli Cheng, Jean-François Arguin, Julien Leissner-Martin, Jacinthe Pilette, and Tobias Golling. Variational autoencoders for anomalous jet tagging. *Physical Review D*, 107(1):016002, 2023.

[39] Darius Chira, Ilian Haralampiev, Ole Winther, Andrea Dittadi, and Valentin Liévin. Image super-resolution with deep variational autoencoders. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 395–411. Springer, 2023.

[40] James J. Choi, David Laibson, Brigitte C. Madrian, and Andrew Metrick. Reinforcement learning and savings behavior. *The Journal of Finance*, 64(6):2515–2534, 2009.

[41] Marco Ciotti, Massimo Ciccozzi, Alessandro Terrinoni, Wen-Can Jiang, Cheng-Bin Wang, and Sergio Bernardini. The covid-19 pandemic. *Critical reviews in clinical laboratory sciences*, 57(6):365–388, 2020.

[42] Mark Coeckelbergh. Health care, capabilities, and ai assistive technologies. *Ethical Theory and Moral Practice*, 13(2):181–190, 2010.

[43] Tim F Cootes, Mircea C Ionita, Claudia Lindner, and Patrick Sauer. Robust and accurate shape model fitting using random forest regression voting. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12*, pages 278–291. Springer, 2012.

[44] Jianguo Cui, Desheng Song, Shiliang Dong, Mingzhuo Wang, Xiaopeng Liang, and Xinhe Xu. Research on airplane health forecast method based on the improved mgm(1, n). In *2009 Chinese Control and Decision Conference*, pages 5262–5265, 2009.

[45] Renan Gonçalves Leonel da Silva, Roger Chammas, and Hillegonda Maria Dutilh Novaes. Rethinking approaches of science, technology, and innovation in healthcare during the covid-19 pandemic: the challenge of translating knowledge infrastructures to public needs. *Health Research Policy and Systems*, 19(1):1–9, 2021.

[46] Patrick Davies and I Maconochie. The relationship between body temperature, heart rate and respiratory rate in children. *Emergency Medicine Journal*, 26(9):641–643, 2009.

[47] Bonnie Devet. The writing center and transfer of learning: A primer for directors. *The Writing Center Journal*, 35(1):119–151, 2015.

[48] Tashvik Dhamija, Anunay Gupta, Shreyansh Gupta, Anjum, Rahul Katarya, and Ghanshyam Singh. Semantic segmentation in medical images through transfused convolution and transformer networks. *Applied Intelligence*, 53(1):1132–1148, 2023.

[49] Nicole A. Drepaul. Sustainable cities and the internet of things (iot) technology: Iot technology improves the development of smart cities' infrastructures and reduces over-population stresses. *Consilience*, (22):39–47, 2020.

[50] Jakub J Dylag. Machine learning based prediction of glucose levels in type 1 diabetes patients with the use of continuous glucose monitoring data. *arXiv preprint arXiv:2302.12856*, 2023.

[51] Ikram Eddahmani, Chi-Hieu Pham, Thibault Napoléon, Isabelle Badoc, Jean-Rassaire Fouefack, and Marwa El-Bouz. Unsupervised learning of disentangled representation via auto-encoding: A survey. *Sensors*, 23(4):2362, 2023.

[52] Issam El Naqa and Martin J Murphy. *What is machine learning?* Springer, 2015.

[53] Reem H Elbekai and Ayman OS El-Kadi. Cytochrome p450 enzymes: central players in cardiovascular health and disease. *Pharmacology & therapeutics*, 112(2):564–587, 2006.

[54] Monica Eriksson and Bengt Lindström. Antonovsky's sense of coherence scale and the relation with health: a systematic review. *Journal of Epidemiology and Community Health (1979-)*, 60(5):376–381, 2006.

[55] GD Eslick and NJ Talley. Non-cardiac chest pain: predictors of health care seeking, the types of health care professional consulted, work absenteeism and interruption of daily activities. *Alimentary pharmacology & therapeutics*, 20(8):909–915, 2004.

[56] GuanWen Fang, Shuo Ba, Yizhou Gu, Zesen Lin, Yuejie Hou, Chenxin Qin, Chichun Zhou, Jun Xu, Yao Dai, Jie Song, et al. Automatic classification of galaxy morphology: A rotationally-invariant supervised machine-learning method based on the unsupervised machine-learning data set. *The Astronomical Journal*, 165(2):35, 2023.

[57] Xuanqi Feng and Masanori Yamada. An analytical approach for detecting and explaining the learning path patterns of an informal learning game. *Educational Technology Society*, 24(1):176–190, 2021.

[58] Ramouna Fouladi, Kyrylo Bessonov, François Van Lishout, and Kristel Van Steen. Model-based multifactor dimensionality reduction for rare variant association analysis. *Human Heredity*, 79(3/4):157–167, 2015.

[59] The Django Software Foundation. Django. https://www.djangoproject.com/, 2005.

[60] Martin Gaynor, Kate Ho, and Robert J. Town. The industrial organization of health-care markets†. *Journal of Economic Literature*, 53(2):235–284, 2015.

[61] Mehdi Gheisari, Fereshteh Ebrahimzadeh, Mohamadtaghi Rahimi, Mahdieh Moazzamigodarzi, Yang Liu, Pijush Kanti Dutta Pramanik, Mohammad Ali Heravi, Abolfazl Mehbodniya, Mustafa Ghaderzadeh, Mohammad Reza Feylizadeh, et al. Deep learning: Applications, architectures, models, tools, and frameworks: A comprehensive survey. *CAAI Transactions on Intelligence Technology*, 2023.

[62] Michel Girard. Standards for cybersecure iot devices: A way forward. Technical report, Centre for International Governance Innovation, 2020.

[63] José González-Alonso, Christina Teller, Signe L Andersen, Frank B Jensen, Tino Hyldig, and Bodil Nielsen. Influence of body temperature on the development of fatigue during prolonged exercise in the heat. *Journal of applied physiology*, 86(3):1032–1039, 1999.

[64] Jesús González-Rubio, J. Ramón Navarro-Cerdán, and Francisco Casacuberta. Dimensionality reduction methods for machine translation quality estimation. *Machine Translation*, 27(3/4):281–301, 2013.

[65] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[66] Isobel Claire Gormley, Thomas Brendan Murphy, and Adrian E Raftery. Model-based clustering. *Annual Review of Statistics and Its Application*, 10, 2023.

[67] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11516–11524, 2021.

[68] Jayadev Gyani, Ahsan Ahmed, and Mohd Anul Haq. Mcdm and various prioritization methods in ahp for css: A comprehensive review. *IEEE Access*, 2022.

[69] Halil Güvenç. Wireless ecg device with arduino. In *2020 Medical Technologies Congress (TIPTEKNO)*, pages 1–4, 2020.

[70] Abdul Mueed Hafiz, Mahmoud Hassaballah, and Adel Binbusayyis. Formula-driven supervised learning in computer vision: A literature survey. *Applied Sciences*, 13(2):723, 2023.

[71] Paul Hager and Phil Hodkinson. Moving beyond the metaphor of transfer of learning. *British Educational Research Journal*, 35(4):619–638, 2009.

[72] Pavel Hamet and Johanne Tremblay. Artificial intelligence in medicine. *Metabolism*, 69:S36–S40, 2017.

[73] Zainab Haroon, Azad Ali Azad, Mubashir Sharif, Ayesha Aslam, Komal Arshad, and Salma Rafiq. Covid-19 era: challenges and solutions in dental education. *J Coll Physicians Surg Pak*, 30(10):129–131, 2020.

[74] Pieter-Jan Hoedt, Sepp Hochreiter, and Günter Klambauer. Normalization is dead, long live normalization!

[75] Changqin Huang, Xuemei Wu, Xizhe Wang, Tao He, Fan Jiang, and Jianhui Yu. Exploring the relationships between achievement goals, community identification and online collaborative reflection: A deep learning and bayesian approach. *Educational Technology Society*, 24(3):210–223, 2021.

[76] Dong Huang, Chang-Dong Wang, Hongxing Peng, Jianhuang Lai, and Chee-Keong Kwoh. Enhanced ensemble clustering via fast propagation of cluster-wise similarities. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(1):508–520, 2021.

[77] Julian C Hughes, Claire Bamford, and Carl May. Types of centredness in health care: themes and concepts. *Medicine, Health Care and Philosophy*, 11:455–463, 2008.

[78] D Hutchinson, M Kunasekaran, A Quigley, A Moa, and CR MacIntyre. Could it be monkeypox? use of an ai-based epidemic early warning system to monitor rash and fever illness. *Public Health*, 220:142–147, 2023.

[79] J Ian. Goodfellow yoshua bengio and aaron courville. In *Deep Learning*. MIT Press, 2016.

[80] MohammadNoor Injadat, Abdallah Moubayed, Ali Bou Nassif, and Abdallah Shami. Multi-stage optimized machine learning framework for network intrusion detection. *IEEE Transactions on Network and Service Management*, 18(2):1803–1816, 2020.

[81] IoT Industriel. Architecture iot : l'essentiel à savoir. https://iotindustriel.com/iot-iiot/architecture-iot-lessentiel-a-savoir/, 2023.

[82] MARK ANDREW JAMES. An investigation of motivation to transfer second language learning. *The Modern Language Journal*, 96(1):51–69, 2012.

[83] Martin F Jarrold. Helices and sheets in vacuothe html version of this article has been enhanced with colour images. *Physical Chemistry Chemical Physics*, 9(14):1659–1671, 2007.

[84] JavaTpoint. Arduino ide. https://www.javatpoint.com/arduino-ide, n.d.

[85] Maria De Jesus and Chenyang Xiao. Predicting health care utilization among latinos: Health locus of control beliefs or access factors? *Health Education Behavior*, 41(4):423–430, 2014.

[86] Kan Jin. An efficient algorithm for association mining. In *2009 Second International Symposium on Knowledge Acquisition and Modeling*, volume 1, pages 291–295, 2009.

[87] Martin Kaloev and Georgi Krastev. Experiments focused on exploration in deep reinforcement learning. In *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 351–355, 2021.

[88] Rahul Katarya and Polipireddy Srinivas. Predicting heart disease at early stages using machine learning: a survey. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 302–305. IEEE, 2020.

[89] Aurangzeb Khan, Baharum Baharudin, Lam Hong Lee, and Khairullah Khan. A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology*, 1(1):4–20, 2010.

[90] Raheleh Khanduzi and Arun Kumar Sangaiah. An efficient recurrent neural network for defensive stackelberg game. *Journal of Computational Science*, 67:101970, 2023.

[91] T. Kinoue, Y. Watanabe, T. Watanabe, and I. Okazaki. Telemedicine session in tokai training course for future health leaders in collaboration with who-wpro, supported by jica. In *Proceedings. 6th International Workshop on Enterprise Networking and Computing in Healthcare Industry - Healthcom 2004 (IEEE Cat. No.04EX842)*, pages 133–135, 2004.

[92] Alice Y. Kolb and David A. Kolb. Learning styles and learning spaces: Enhancing experiential learning in higher education. *Academy of Management Learning Education*, 4(2):193–212, 2005.

[93] Dominik Kreuzberger, Niklas Kühl, and Sebastian Hirschl. Machine learning operations (mlops): Overview, definition, and architecture. *IEEE Access*, 2023.

[94] Anupama Kulkarni, Mihai Patrascu, Yuri van de Vijver, Jaap van Wensveen, Robert Pijnenburg, and Stoyan Nihtianov. Investigation of long-term drift of ntc temperature sensors with less than 1 mk uncertainty. In *2015 IEEE 24th International Symposium on Industrial Electronics (ISIE)*, pages 150–155, June 2015.

[95] Sachin Kumar, Prayag Tiwari, and Mikhail Zymbler. Internet of things is a revolutionary approach for future technology enhancement: a review. *Journal of Big data*, 6(1):1–21, 2019.

[96] John G Kuschewski, Stefen Hui, and Stanislaw H Zak. Application of feedforward neural networks to dynamical system identification and control. *IEEE transactions on control systems technology*, 1(1):37–49, 1993.

[97] A. Lakshmanarao, A. Srisaila, and T. Srinivasa Ravi Kiran. Machine learning and deep learning framework with feature selection for intrusion detection. In *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, pages 1–5, 2022.

[98] GORDON-TAYLOR MEMORIAL LECTURE and SIGNIFICANC L TURNER. Annals of the royal college of surgeons of england. *Annals of the Royal College of Surgeons of England*, 29:204, 1961.

[99] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[100] Gwanhoo Lee, Rebekah Lewis, VAUGHAN TUREKIAN, TAEHEE JEONG, GIGI KWIK GRONVALL, ELIZABETH PRESCOTT, GWANHOO LEE, REBEKAH LEWIS, and BEAU WOODS. Building a smart partnership for the fourth industrial revolution. Technical report, Atlantic Council, 2018.

[101] JOHN LEE. The internet of things: China's rise and australia's choices. Technical report, Lowy Institute for International Policy, 2021.

[102] Rui Liang, Chao Chen, Akash Kumar, Junyu Tao, Yan Kang, Dong Han, Xianjia Jiang, Pei Tang, Beibei Yan, and Guanyi Chen. State-of-the-art applications of machine learning in the life cycle of solid waste management. *Frontiers of Environmental Science & Engineering*, 17(4):44, 2023.

[103] Björn Lindström, Ida Selbing, Tanaz Molapour, and Andreas Olsson. Racial bias shapes social reinforcement learning. *Psychological Science*, 25(3):711–719, 2014.

[104] Francisco L. Loaiza, John D. Birdwell, George L. Kennedy, and Dale Visser. Utility of artificial intelligence and machine learning in cybersecurity. Technical report, Institute for Defense Analyses, 2019.

[105] Craig Lockwood, Tiffany Conroy-Hiller, and Tamara Page. Vital signs. *JBI reports*, 2(6):207–230, 2004.

[106] NICO LÜCK. Machine learning-powered artificial intelligence in arms control. Technical report, Peace Research Institute Frankfurt, 2019.

[107] James G MacKinnon. Bootstrap inference in econometrics. *Canadian Journal of Economics/Revue canadienne d'économique*, 35(4):615–645, 2002.

[108] James G MacKinnon, Morten Ørregaard Nielsen, and Matthew D Webb. Testing for the appropriate level of clustering in linear regression models. *arXiv preprint arXiv:2301.04522*, 2023.

[109] Govind Singh Mahara and Sharad Gangele. Fake news detection: A rnn-lstm, bilstm based deep learning approach. In *2022 IEEE 1st International Conference on Data, Decision and Systems (ICDDS)*, pages 01–06, 2022.

[110] Batta Mahesh. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9:381–386, 2020.

[111] Khadidja MAKHLOUF. Monitoring system development to non-invasively forecast future body temperature. Master's thesis, University of Biskra, 2021.

[112] Khadidja Makhlouf, Zohra Hmidi, Laid Kahloul, Saber Benhrazallah, and Tarek Ababsa. On the forecasting of body temperature using iot and machine learning techniques. In *2021 International Conference on Theoretical and Applicative Aspects of Computer Science (ICTAACS)*, pages 1–6. IEEE, 2021.

[113] Aneil Malhotra, Harshil Dhutia, Gherardo Finocchiaro, Sabiha Gati, Ian Beasley, Paul Clift, Charlotte Cowie, Antoinette Kenny, Jamil Mayet, David Oxborough, et al. Outcomes of cardiac screening in adolescent soccer players. *New England Journal of Medicine*, 379(6):524–534, 2018.

[114] Danilo Mandic and Jonathon Chambers. *Recurrent neural networks for prediction: learning algorithms, architectures and stability.* Wiley, 2001.

[115] SHANNON MATTERN and JOHN DURHAM PETERS. *The Pulse of Global Passage: Listening to Logistics*, pages 75–90. Duke University Press, 2021.

[116] Derroh Amany Maxime Konan and Warish Patel. i-nxgevita: Iomt based ubiquitous health monitoring system using deep neural networks. In *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pages 552–557, 2018.

[117] Gary C McDonald. Ridge regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(1):93–100, 2009.

[118] David Mendes, Manuel Lopes, Pedro Parreira, and César Fonseca. Deep learning and iot to assist multimorbidity home based healthcare. 2017.

[119] Massimo Merenda, Carlo Porcaro, and Demetrio Iero. Edge machine learning for ai-enabled iot devices: A review. *Sensors*, 20(9):2533, 2020.

[120] Madison Milne-Ives, Ching Lam, Caroline De Cock, Michelle Helena Van Velthoven, Edward Meinert, et al. Mobile apps for health behavior change in physical activity, diet, drug and alcohol use, and mental health: systematic review. *JMIR mHealth and uHealth*, 8(3):e17046, 2020.

[121] Lina Miltenienė. Bendradarbiavimo modelio konstravimas tenkinant specialiuosius ugdymosi poreikius. 2005.

[122] Christina Y Miyake, Erica J Lay, Cheyenne M Beach, Scott R Ceresnak, Caridad M Delauz, Taylor S Howard, Christopher M Janson, Kate Jardine, Prince J Kannankeril, Maina Kava, et al. Cardiac crises: Cardiac arrhythmias and cardiomyopathy during tango2 deficiency related metabolic crises. *Heart rhythm*, 19(10):1673–1681, 2022.

[123] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Dimensionality Reduction*, pages 281–292. The MIT Press, 2012.

[124] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Reinforcement Learning*, pages 313–338. The MIT Press, 2012.

[125] Jessica Morley, Caio CV Machado, Christopher Burr, Josh Cowls, Indra Joshi, Mariarosaria Taddeo, and Luciano Floridi. The ethics of ai in health care: a mapping review. *Social Science & Medicine*, 260:113172, 2020.

[126] Ilias Mylonidis. An ai forecasting method based on nonlinear dynamics and neural networks: The case of gas-emissions by cars. 2023.

[127] Belkacem NACERI. Internet of things/trust in iot. Master's thesis, University of Biskra, 2019.

[128] Resmiye Nasiboglu and Efendi Nasibov. Wabl method as a universal defuzzifier in the fuzzy gradient boosting regression model. *Expert Systems with Applications*, 212:118771, 2023.

[129] Tigor Hamonangan Nasution and Lukman Adlin Harahap. Predict the percentage error of lm35 temperature sensor readings using simple linear regression analysis. In *2020 4rd International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM)*, pages 242–245, 2020.

[130] Masooma Nazari, Akhtar Hussain, and Petr Musilek. Applications of clustering methods for different aspects of electric vehicles. *Electronics*, 12(4):790, 2023.

[131] Duduzile Ndwandwe and Charles S Wiysonge. Covid-19 vaccines. *Current opinion in immunology*, 71:111–116, 2021.

[132] ELINA NOOR and MARK BRYAN MANANTAN. Raising standards: Data and artificial intelligence in southeast asia. Technical report, Asia Society, 2022.

[133] Varun Kumar Ojha, Ajith Abraham, and Václav Snášel. Metaheuristic design of feedforward neural networks: A review of two decades of research. *Engineering Applications of Artificial Intelligence*, 60:97–116, 2017.

[134] Vincent K Omachonu and Norman G Einspruch. Innovation in healthcare delivery systems: a conceptual framework. *The Innovation Journal: The Public Sector Innovation Journal*, 15(1):1–20, 2010.

[135] Eva Ostertagová. Modelling using polynomial regression. *Procedia Engineering*, 48:500–506, 2012.

[136] R Lyman Ott and Micheal T Longnecker. *An introduction to statistical methods and data analysis*. Cengage Learning, 2015.

[137] Mike Owens and Grant Allen. *SQLite*. Springer, 2010.

[138] Ashwin Pajankar and Aditya Joshi. Preparing data for machine learning. In *Hands-on Machine Learning with Python: Implement Neural Network Solutions with Scikit-learn and PyTorch*, pages 79–97. Springer, 2022.

[139] Bharati Panigrahi, Krishna Chaitanya Rao Kathala, and M Sujatha. A machine learning-based comparative approach to predict the crop yield using supervised learning with regression models. *Procedia Computer Science*, 218:2684–2693, 2023.

[140] Si Hyun Park, Canada Vancouver, British Columbia, Student, and BFA Design. Covid-19.

[141] Cong Peng, Prashant Goswami, and Guohua Bai. A literature review of current technologies on health data integration for patient-centered health management. *Health informatics journal*, 26(3):1926–1951, 2020.

[142] Trang Pham, Truyen Tran, Dinh Phung, and Svetha Venkatesh. Predicting healthcare trajectories from medical records: A deep learning approach. *Journal of biomedical informatics*, 69:218–229, 2017.

[143] Thomas G Pickering, Nancy Houston Miller, Gbenga Ogedegbe, Lawrence R Krakoff, Nancy T Artinian, and David Goff. Call to action on use and reimbursement for home blood pressure monitoring: executive summary: a joint scientific statement from the american heart association, american society of hypertension, and preventive cardiovascular nurses association. *Hypertension*, 52(1):1–9, 2008.

[144] K Poikolainen. A comment diagnosis as a means of health forecasting. *Social science & medicine*, 13(2):165–167, 1979.

[145] Mihail Popescu, George Chronis, Rohan Ohol, Marjorie Skubic, and Marilyn Rantz. An eldercare electronic health record system for predictive health assessment. In *2011 IEEE 13th International Conference on e-Health Networking, Applications and Services*, pages 193–196, 2011.

[146] Satya Prakash, Anand Singh Jalal, and Pooja Pathak. Forecasting covid-19 pandemic using prophet, lstm, hybrid gru-lstm, cnn-lstm, bi-lstm and stacked-lstm for india. In *2023 6th International Conference on Information Systems and Computer Networks (ISCON)*, pages 1–6, 2023.

[147] Sandy Suryo Prayogo, Yulisdin Mukhlis, and Bayu Kumoro Yakti. The use and performance of mqtt and coap as internet of things application protocol using nodemcu esp8266. In *2019 Fourth International Conference on Informatics and Computing (ICIC)*, pages 1–5, 2019.

[148] Jasper Prijs, Zhibin Liao, Minh-Son To, Johan Verjans, Paul C Jutte, Vincent Stirler, Jakub Olczak, Max Gordon, Daniel Guss, Christopher W DiGiovanni, et al. Development and external validation of automated detection, classification, and localization of ankle fractures: inside the black box of a convolutional neural network (cnn). *European Journal of Trauma and Emergency Surgery*, 49(2):1057–1069, 2023.

[149] Charles Protzman, George Mayzell, and Joyce Kerpchar. *Leveraging lean in healthcare: Transforming your enterprise into a high quality patient care delivery system.* CRC Press, 2010.

[150] Ganesan Raman. Identifying extra-large pore structures in zeolites with a machine learning approach and its deployment into production. *Microporous and Mesoporous Materials*, 348:112362, 2023.

[151] Veenu Rani, Syed Tufael Nabi, Munish Kumar, Ajay Mittal, and Krishan Kumar. Self-supervised learning: A succinct review. *Archives of Computational Methods in Engineering*, pages 1–15, 2023.

[152] N Sri Sai Venkata Subba Rao and S John Justin Thangaraj. Flight ticket prediction using xgbregression compared with kneighbour regression algorithm.

[153] Daniele Ravì, Charence Wong, Fani Deligianni, Melissa Berthelot, Javier Andreu-Perez, Benny Lo, and Guang-Zhong Yang. Deep learning for health informatics. *IEEE journal of biomedical and health informatics*, 21(1):4–21, 2016.

[154] Karen Rose, Scott Eldridge, and Lyman Chapin. The internet of things: An overview. *The internet society (ISOC)*, 80:1–50, 2015.

[155] Ricardo Adán Salas Rueda and José de Jesús Vázquez Estupiñán. Aplicación en la nube lucidchart:¿ herramienta necesaria para la innovación del proceso educativo en el siglo xxi? *Revista de Comunicación de la SEECI*, (44):115–126, 2018.

[156] Vemuri Richard Ranjan Samson, U Bharath Sai, P L S D Malleswara Rao, K Kedar Eswar, and S Pradeep Kumar. Automatic oxygen level control of patient using fuzzy logic and arduino. In *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*, pages 98–102, 2017.

[157] Gillian M Sandstrom, Neal Lathia, Cecilia Mascolo, and Peter J Rentfrow. Opportunities for smartphones in clinical care: the future of mobile mood monitoring. *The Journal of clinical psychiatry*, 77(2):13476, 2016.

[158] Iqbal H Sarker, Alan Colman, Muhammad Ashad Kabir, and Jun Han. Individualized time-series segmentation for mining mobile phone user behavior. *The Computer Journal*, 61(3):349–368, 2018.

[159] Christina Schweikert, Yanjun Li, David Dayya, David Yens, Martin Torrents, and D. Frank Hsu. Analysis of autism prevalence and neurotoxins using combinatorial fusion and association rule mining. In *2009 Ninth IEEE International Conference on Bioinformatics and BioEngineering*, pages 400–404, 2009.

[160] Xiaoyu Shi, Shengzhi Huang, Qiang Huang, Xuewen Lei, Jiangfeng Li, Pei Li, and Mingyang Yang. Deep-learning-based wind speed forecasting considering spatial–temporal correlations with adjacent wind turbines. *Journal of Coastal Research*, pages 623–632, 2019.

[161] Yitao Shi, Haiyang Tang, Zhuhui Wang, and Weijie Zhu. Automatic machine learning model construction and performance validation in face recognition. In *Proceedings of the World Conference on Intelligent and 3-D Technologies (WCI3DT 2022) Methods, Algorithms and Applications*, pages 23–41. Springer, 2023.

[162] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security*, pages 587–601, 2017.

[163] Malcolm Strens. A bayesian framework for reinforcement learning. In *ICML*, volume 2000, pages 943–950, 2000.

[164] Sandya Subramanian, Bryan Tseng, Riccardo Barbieri, and Emery N Brown. Unsupervised machine learning methods for artifact removal in electrodermal activity. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, pages 399–402, 2021.

[165] Fengxin Sun, Yongsheng Zhu, Changjun Jia, Tianming Zhao, Liang Chu, and Yupeng Mao. Advances in self-powered sports monitoring sensors based on triboelectric nanogenerators. *Journal of Energy Chemistry*, 2023.

[166] Mark C. Taylor. *Internet of Bodies*, pages 109–142. Columbia University Press, 2021.

[167] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[168] Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. Semi-supervised model adaptation for statistical machine translation. *Machine Translation*, 21(2):77–94, 2007.

[169] Bradley University. Bradley university scout 2006 2007.

[170] Jeroen van Meijgaard, Jonathan E. Fielding, and Gerald F. Kominski. Assessing and forecasting population health: Integrating knowledge and beliefs in a comprehensive framework. *Public Health Reports (1974-)*, 124(6):778–789, 2009.

[171] Luis M Vaschetto. *CRISPR-/Cas9 Based Genome Editing for Treating Genetic Disorders and Diseases*. CRC Press, 2022.

[172] Deepankar Verma and Chhavi Rana. Comparative analysis of time series forecasting algorithms. In *Proceedings of Third International Conference on Communication, Computing and Electronics Systems: ICCCES 2021*, pages 239–255. Springer, 2022.

[173] Michael Jeffrey Volk, Ismini Lourentzou, Shekhar Mishra, Lam Tung Vo, Chengxiang Zhai, and Huimin Zhao. Biosystems design by machine learning. *ACS synthetic biology*, 9(7):1514–1533, 2020.

[174] Darrell M. West. *ARTIFICIAL INTELLIGENCE*, pages 19–42. Brookings Institution Press, 2018.

[175] Jin Xiao, Ling Xie, Dunhu Liu, Yi Xiao, and Yi Hu. A clustering and selection based transfer ensemble model for customer credit scoring. *Filomat*, 30(15):4015–4026, 2016.

[176] Tingting Xu, Yunting Zong, Heng Su, Aohua Tian, Jay Gao, Yurui Wang, and Ruiqi Su. Prediction of multi-scale socioeconomic parameters from long-term nighttime lights satellite data using decision tree regression: A case study of chongqing, china. *Land*, 12(1):249, 2023.

[177] Xin Yan, Yanqin Bai, Shu-Cherng Fang, and Jian Luo. A kernel-free quadratic surface support vector machine for semi-supervised learning. *The Journal of the Operational Research Society*, 67(7):1001–1011, 2016.

[178] Stephen J. H. Yang. Guest editorial: Precision education - a new challenge for ai in education. *Educational Technology Society*, 24(1):105–108, 2021.

[179] Amir Alipour Yengejeh. Genome-wide association study of the maize crop by the lasso regression analysis. 2023.

[180] Jun Yuan, Zhili Qin, Haikun Huang, Xingdong Gan, Shuguang Li, and Baihai Li. State of health estimation and remaining useful life prediction for a lithium-ion battery with a two-layer stacking regressor. *Energies*, 16(5):2313, 2023.

[181] Xingyu Zeng, Wanli Ouyang, and Xiaogang Wang. Multi-stage contextual deep learning for pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 121–128, 2013.

[182] Jiawei Zhang, Yi Wang, Mingyang Sun, Ning Zhang, and Chongqing Kang. Constructing probabilistic load forecast from multiple point forecasts: A bootstrap based approach. In *2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, pages 184–189, 2018.

[183] Jinghua Zhang, Chen Li, Yimin Yin, Jiawei Zhang, and Marcin Grzegorzek. Applications of artificial neural networks in microorganism image analysis: a comprehensive review from conventional multilayer perceptron to popular convolutional neural network and potential visual transformer. *Artificial Intelligence Review*, 56(2):1013–1070, 2023.

[184] Kun-Shan Zhang, Chiu-Mei Chen, and Wen-Yu Chang. Customers' willingness to import hotel robots under the background of artificial intelligence based on spss data analysis. In *2021 2nd International Conference on Artificial Intelligence and Education (ICAIE)*, pages 81–84, 2021.

[185] Zhongheng Zhang. Introduction to machine learning: k-nearest neighbors. *Annals of translational medicine*, 4(11), 2016.

[186] Jing Zhao, Li-an Chen, and Zhi-ying Chen. Application of a grey combined prediction model with weighted compensation. In *IET International Conference on Information Science and Control Engineering 2012 (ICISCE 2012)*, pages 1–5, 2012.

[187] Ruoqing Zhu, Donglin Zeng, and Michael R. Kosorok. Reinforcement learning trees. *Journal of the American Statistical Association*, 110(512):1770–1784, 2015.

[188] Wilson A Zuniga-Galindo. p-adic statistical field theory and deep belief networks. *Physica A: Statistical Mechanics and its Applications*, page 128492, 2023.