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 Number: GLSD01/M2/2021

Thesis

Presented to obtain the diploma of academic Master in

Computer Science

Option: **Software Engineering and Distributed Systems**

Monitoring System Development To Non-Invasively Forecast Future Body Temperature

By:
MAKHLOUF Khadidja

Defended the *July $\_^{th}$ 2021*, in front of the jury composed of:

/		President
HMIDI Zohra	MAA	Supervisor
/		Examiner

University Year: 2020/2021

Acknowledgements

All our thanks and gratitude for ALLAH the Almighty who gave us all the courage and the will to go till the end of this project.

I would like to express my special thanks of gratitude to my supervisor ***MRs. HMIDI Zohra****, to whom I am deeply grateful for her guidance, patience, availability and support, my deep respect and thanks for her commitment to giving me the opportunity to carry out this project.*

I am indebted to ***Pr. KAHLOUL Laid*** *for his consistent support and continuously provided encouragement to go throughout this experience.*

Warm thanks to LINFI Laboratory PhD students for their guidance and unconditional help they have given me.

Furthermore, I would like to thank all the Software Engineering and Distributed Systems teachers for their overall insights in this field and sharing their expertise.

Dedication

This work is dedicated to the bright memory of my late father ***MAKHLOUF M^{ED} Khider*** *, who would be the happiest if he could wait to see what I have become.*

My deepest gratitude to my mother for her continuous and unparalleled love, help and support.
I am grateful to my sister for her precious advices and belief on what I can do.
I really appreciate , value and respect my brother's trust and support in my life and study career.
Extended gratitude to all my family members and beloved ones who have always wanted to see me successful.

Abstract

Artificial Intelligence has lately begun to be used into medicine to improve patient care by speeding up processes and increasing accuracy, paving the way for improved healthcare in general. On the other hand, temperature is an important health factor that has to be regularly monitored, and even early detected in some situations.

Therefore, this project aims to invest the advances of AI to develop a monitoring system that early detects body temperature. The used technique relies on building a wearable device using a temperature sensor and a microcontroller with WiFi card integrated. Thus, the internet of things technology is mandatory to beneficiate from cloud storage and display the forecasted results on a reactive web application using the forecasting technique to get early body temperature values.

Key words: *Body temperature forecasting, Artificial Intelligence, Machine Learning, Internet of Things, Forecasting methods, Microcontroller, Temperature sensor.*

Résumé

L'intelligence artificielle a récemment commencé à être utilisée en médecine pour améliorer les soins aux patients en accélérant les processus et en augmentant la précision, ouvrant ainsi la voie à une amélioration des soins de santé en général. D'un autre côté, la température est un facteur de santé important qui doit être régulièrement surveillée, et même détectée de façon précoce dans certaines situations.

Par conséquent, ce projet vise à investir les progrès de l'IA pour développer un système de surveillance qui détecte précocement la température corporelle. La technique utilisée repose sur la construction d'un dispositif portable utilisant un capteur de température et un micro-contrôleur avec une carte WiFi intégrée. Ainsi, la technologie de l'internet des objets est requise pour bénéficier du stockage en nuage et afficher les résultats prévus sur une application web réactive utilisant la technique de prévision pour obtenir des valeurs de température corporelle précocement.

Mots clés: *Prévision de la température corporelle, intelligence artificielle, apprentissage automatique, Internet des objets, méthodes de prévision, micro-contrôleur, capteur de température.*

Contents

List of Tables

List of Figures

General Introduction

General Introduction

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, where approaching to patients is very perilous and harmful, taking as best example, COVID-19 crisis where permanent observation is crucial.
Moreover, body temperature is an early warning sign of infection, and monitoring it, even when healthy, can help detecting diseases early. Another motivational reason to track temperature is that the typical (normal) body temperature for people is just a myth, body temperature slightly varies from a person to another and it depends on many factors (gender, age, activity level, time of the day, measurement method, etc.). Thus, monitoring body temperature allows the obtention of a baseline temperature to enable comparison to be made with future recordings for further objectives such as the main objective of our project that we will discuss later, as it allows to monitor the effect of treatment for some therapies.
Monitoring body temperature may seem simple, but in fact, several issues can affect the accuracy of the reading including:

- site of measurement;
- reliability of the instruments;
- user's technique.

Amongst the potential uses of wearable devices designed for healthcare, we have thought about using them to ***forecast future body temperature*** of patients who are in critical situations, taking as best example infants having meningitis that causes real life-threatening problems. To do so, a huge ***time-series dataset*** containing varied body temperature values of people is indispensable, thereby, we have intended to build our own device and collect the time-series dataset due to the unavailability of this latter.

The device we are intending to build will be composed of a ***temperature sensor*** and a ***microcontroller*** that serves to recuperate the measured temperature values, then send these values to a website through IoT technology and store them as a time-series dataset on the same website that is only accessible by the treating doctor to ensure patients' privacy.
The main objective of our work is to ***accurately forecast human body temperature*** by continuous monitoring of individuals through ***wireless sensing technology*** in order to ***prevent*** patients enduring critical situations from temperature peaks during next hours and help doctors taking the right decision of treatment to avoid disastrous side-effects and save lives of hospitalized patients.

To achieve this goal, we first have to:

- Know some basic concepts of artificial intelligence and machine learning.
- Have a look on existing methods of measuring body temperature and their disadvantages to see what we can do to bring something new and reliable.

- Set a solution to achieve our main goal.

Therefore, we are writing this thesis that will be partitioned into two parts as follows:

1. Part one will be consecrated to the state of the art of our project, and we will divide it into two chapters, in the first one we will discuss the general concepts of artificial intelligence and machine learning and see how they can be used to improve health care; where in the second chapter we will discuss our case of study which is body temperature monitoring and talk about the internet of things that will be a major component of our monitoring system.

2. The second part will be the core of our project, where in the first chapter we will try to make a conception to bring a solution and achieve our goal, while the second chapter will describe how we will realise and implement our monitoring system.

To sum up this work, we will see a conclusion in which we will summarise what we have done during this project and talk about future works.

Part I

State of the art

Chapter 1

Artificial intelligence

Chapter 1

Artificial intelligence

1.1 Introduction

John McCarthy created the term Artificial Intelligence (AI) in 1956 during a conference on the subject. However, Alan Turing, who established the Turing test to distinguish humans from machines, highlighted the potential of machines being able to replicate human behaviour and really think. Since then, computer power has increased to the point where it can perform instant computations and evaluate fresh data in real time based on previously assessed data. In this chapter, we are going to see a background about artificial intelligence, giving its basic definition, present an understand the principles of machine learning and see the most commonly used algorithms and methods. Then, we are going to see how AI can contribute to improve healthcare with two of its methods that are *forecasting* and *prediction* in subsection 1.2.4. We are also going to give definition of both concepts and compare between them in order to see what best suits and can really help by coming up with a new aid and support in our hospitals.

1.2 Artificial intelligence and machine learning

1.2.1 Artificial Intelligence

Artificial intelligence (AI) is a wide-ranging branch of computer science concerned with building smart machines capable of performing tasks that typically require human intelligence [47].
Artificial Intelligence (AI) is a general term that implies the use of a computer to model intelligent behaviour with minimal human intervention [15].
AI applications that utilize machine learning are on the rise in clinical research and provide highly promising applications in specific use-cases.

1.2.2 Artificial intelligence in medicine

The application of AI in medicine has two main branches:

- ***The Virtual Branch:*** it is represented by Machine Learning and Deep Learning that are represented by mathematical algorithms that improve learning through experience [15].
- ***The Physical Branch:*** including physical objects, medical devices and increasingly sophisticated robots taking part in the delivery of care (carebots) [15].

1.2.3 Machine Learning

Machine learning is a collection of methods that enable computers to automate data-driven model building and programming through a systematic discovery of statistically significant patterns in the available data [3]. While machine learning methods are gaining popularity, the first attempt to develop a machine that mimics the behaviour of a living creature was conducted by Thomas Ross in 1930s [41].

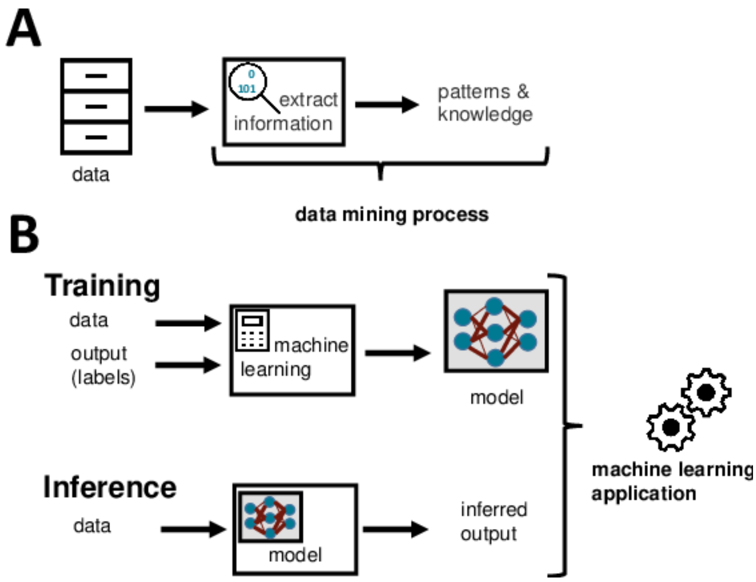

Figure 1.1: Training process in machine learning.

The above Figure 1.1 illustrates the machine learning process. The latter passes through two phases, phase A and phase B. In phase A, we have a large dataset that must be cleaned in order to select to most important data (data mining process). Whereas, in phase B, after picking a suitable algorithm for our problem, we have another 2 steps to pass by: **training**, that is to feed the chosen algorithm with the optimized dataset, in this step data fed to the algorithm must be accompanied with their outputs (labels) to get a trained model, and then we pass to the second step that is **inference** where we feed the obtained model with new data.

1.2.3.1 Approach for a machine learning algorithm development process

The flowchart in Figure 1.2 presents a typical approach for a machine learning algorithm development process. As depicted in the figure, the first step for any machine learning process is the **problem definition**, where the problem has to be well defined as it has a direct impact on the following steps and is the key solution to define the appropriate machine learning method to be used, to do so, we have to set the input and output variables to be considered and decide whether all the variables are of the same importance. After that, comes the **data collection** step in which we have to develop the list of feasible input/output variables then understand how much data is sufficient(i.e: size of the dataset). However, as presented in Figure 1.2, it is possible to develop a feedback algorithm that evaluates and trains the learning algorithm after the real world implementation.
The **evaluation** process of any ML model is one of the necessary steps. To fairly evaluate the model, we must not use a dataset containing data that have been utilised during the training process. Therefore, we have to initially divide our main dataset into two separate and completely independent ones (generally 80% data to train the algorithm and 20% to test the model's accuracy).

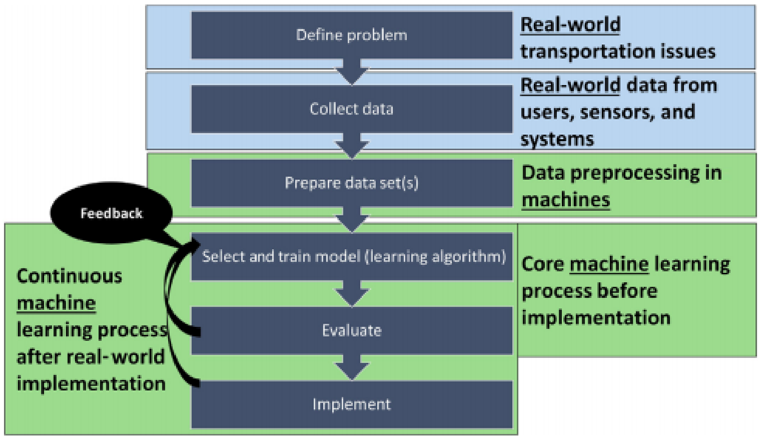

Figure 1.2: Machine learning algorithm development approach..
[3].

1.2.3.2 Machine learning methods

Machine learning methods can be characterized based on the type of "learning." There exist several basic types of learning methods, such as: *supervised learning*, where previously labelled data is used to guide the learning process; *unsupervised learning*, where only unlabelled data is used and *reinforcement learning*, where the learning process is guided by a series of feedback/reward cycles.

A) **Supervised learning:** it is an important form of ML. It is named as supervised, because the learning process is done under the seen label of observation variables. Given a database of training examples with a specific target label (property) in the form (x_1,y_1), (x_2 , y_2),..., (x_n , y_n) where x_i denotes the feature vector of the i^{th} example and y_i is its label, the goal is to construct a model $g : X \rightarrow Y$ that can accurately predict the label Y for future instances of data X. When this target property is a continuous real value, the task is referred to as regression. Otherwise, when the target property is a finite set of discrete values, the task is referred to as classification [12]. In supervised learning, *datasets*[1] are trained with the training sets to build ML, and then will be used to label new observations from the testing set Figure 1.3.

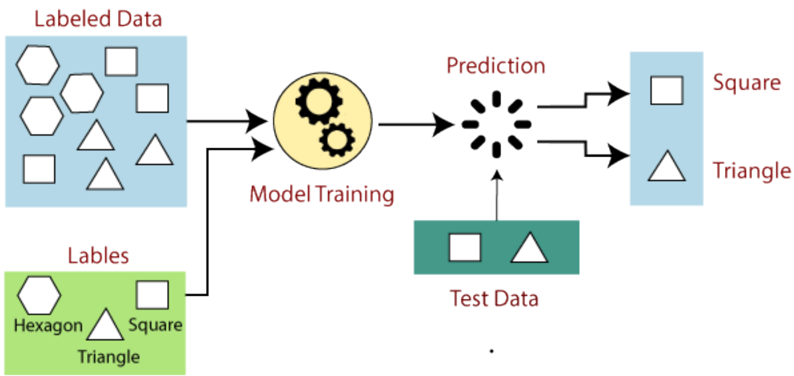

Figure 1.3: Supervised learning.

Supervised algorithms are best used for:

a. estimating life expectancy;

[1]a dataset is a set of data that is collected for a specific purpose

b. forecasting;
c. customer retention;
d. diagnostics;
e. identity fraud detection;
f. images classification;
g. advertising popularity prediction.

According to different types of output variables, supervised learning tasks can be divided into two kinds: classification task and regression task. [53]

a) *Classification :* for a classification problem, the goal of the machine learning algorithm is to categorize or classify given inputs based on the training data set. The training data set in a classification problem includes set of input/output pairs categorized in classes Figure 1.3. Many classification problems are binary, i.e., only two classes such as True and False are involved [3].
Classification models can be evaluated by calculating either the *accuracy, Log Loss*, *precision recall* or *ROC-AUC.*
b) *Regression :* for a regression problem, the goal of the machine learning algorithm is to develop a relationship between outputs and inputs using a continuous function to help machines understand how outputs are changing for given inputs. The regression problems can also be envisioned as prediction problems [3]. To evaluate our regression model, the most used metrics are *Root Mean Square Error(RMSE)* and *R Squared(R^2)*, we can also evaluate this kind of models with *Adjusted R Squared, MSAE, MSPE.*

Supervised learning algorithms: below are some of the most popular supervised learning algorithms:

- Decision tree;
- Linear regression;
- Logistic regression;
- Navie Bayes;
- Support vector machine (SVM);
- K-Nearest neighbour(KNN)...

B) **Unsupervised learning :** it is a type of machine learning in which models are trained using unlabelled dataset and are allowed to act on that data without any supervision Figure 1.4, unsupervised learning tasks can be divided into two kinds:

a) *Clustering:* clustering methods focus on grouping data in multiple clusters based on similarity between data points. Usually, clustering methods rely on mathematical models to identify similarities between unlabelled data points. The similarities between data points are identified by various methods such as Euclidean distance [3].
b) *Association:* association method focuses on identifying a particular trend (or trends) in the given data set that represents major data patterns or, the so-called significant association rules that connect data patterns with each other [1].

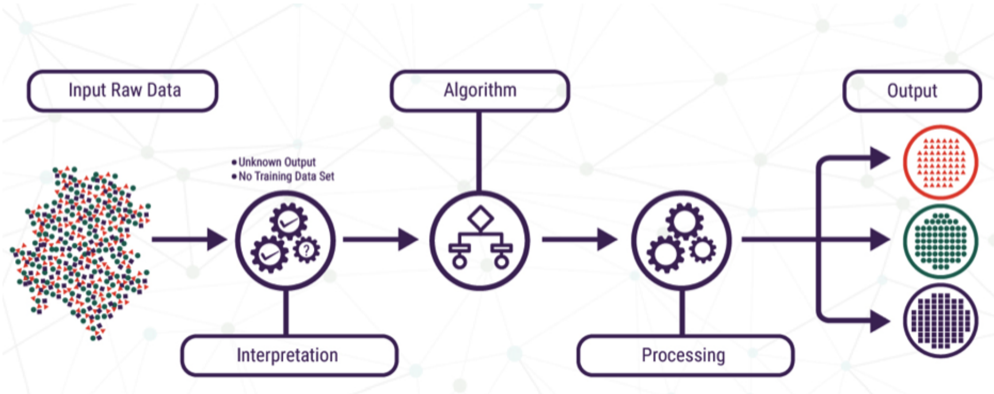

Figure 1.4: Unsupervised learning.

Unsupervised learning algorithms are mostly common in:

a. recommender systems;
b. targetted marketing;
c. customer segmentation;
d. meaningful compression;
e. big data visualisation;
f. structure discovery;
g. feature elicitation.

Unsupervised learning algorithms: below are some of the most popular unsupervised learning algorithms:

- K-means clustering;
- KNN (k-nearest neighbours);
- Hierarchal clustering;
- Anomaly detection;
- Neural Networks;
- Singular value decomposition...

Understanding data
It is important to note that in both supervised and unsupervised learning, the quality, type, and size of the data are significant factors that affect accuracy, efficiency, and robustness of the machine learning algorithm. While the goal of any machine learning application is to capture reality and model uncertainty, the learned model does not usually represent a real world but the reality presented by the data set [3].

C) **Reinforcement learning :** reinforcement learning is a subfield of machine learning that teaches an agent how to choose an action from its action space, within a particular environment, in order to maximize rewards over time Figure 1.5.

Reinforcement Learning has four essential elements:

- ***Agent:*** The program you train, with the aim of doing a job you specify.
- ***Environment:*** The world, real or virtual, in which the agent performs actions.

- ***Action:*** A move made by the agent, which causes a status change in the environment.
- ***Rewards:*** The evaluation of an action, which can be positive or negative.

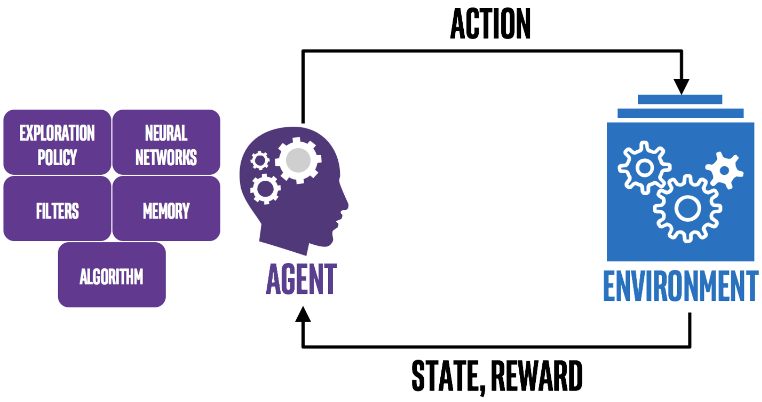

Figure 1.5: Reinforcement learning.

Reinforcement learning algorithms are used for:

a. learning tasks;

b. robot navigation;

c. real time decisions;

d. skill acquisition.

Below in Figure 1.6, a recapitulating picture of machine learning algorithms and their usages.

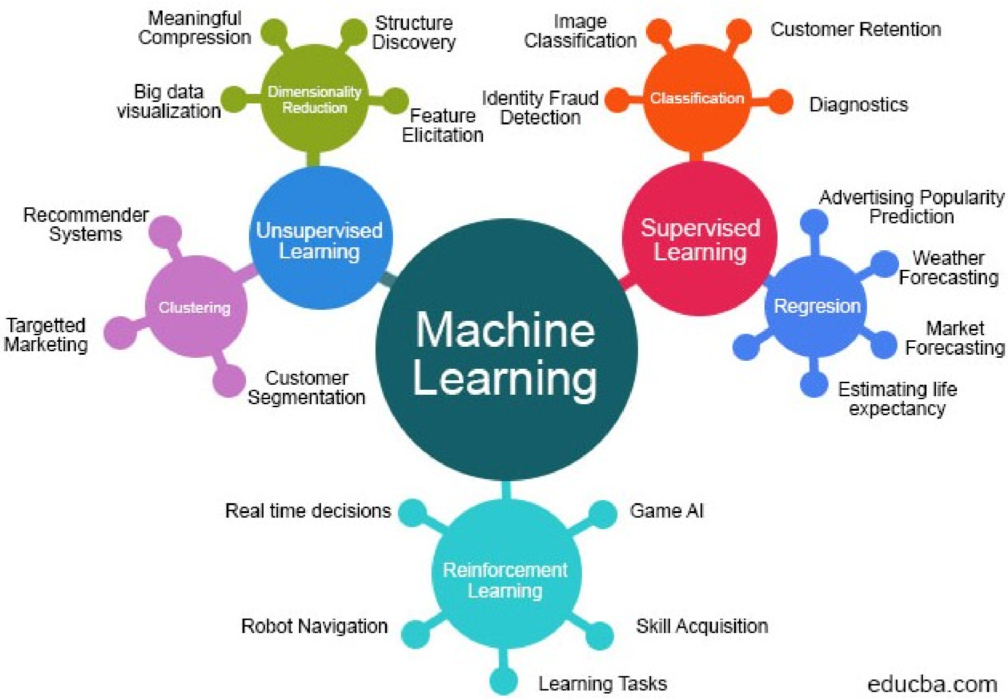

Figure 1.6: ML algorithms and usages.

1.2.3.3 Machine learning algorithms for data analytics

The type of machine learning algorithms may vary from linear regression and classification to complex neuro-fuzzy systems. Below, we are presenting selected popular machine learning algorithms that can be found implemented in a variety of open-source and commercial products.

Regression methods: Given a target variable, which up to measurement errors, depends on one or several input variables, regression describes the nature of dependence between the target and input variables and quantifies the error variance by finding a fitting function that maps the input variables to the target (i.e., output).
Some of the most popular regression algorithms:

- Linear regression;
- K-Neighbours regressor;
- Ridge regression;
- Random forest regressor;
- Lasso regression;
- Gradient boosting regressor;
- Decision tree regressor;
- Adaboost regressor;

These different types of regression analysis techniques can be used to build the model depending upon the kind of data available or the one that gives the maximum accuracy.

Regression analysis is used when you want to predict a continuous dependent variable from a number of independent variables. If the dependent variable is dichotomous, then logistic regression should be used. Note that no single algorithm works for all problems, there are many factors (dataset size and structure, result accuracy training time, etc.) that must be taken in consideration before choosing the appropriate algorithm as we have previously explained in the subsubsection 1.2.3.1.

1.2.4 Artificial intelligence and early body temperature detection

In many critical illness cases, the treatment prognosis is predicated on how early the diseases are detected. Ideally, symptoms show up early enough for us to know that something is wrong and give us ample time to seek professional help. However, there are some diseases that do not have the early warning signals and too often, we hear of cases where such signals come too late.

Furthermore, seeing medical specialists may not be something that many people do regularly. For some, the waiting time to see a medical professional may be a constraint too. This is where AI algorithms can help to do the first level screenings to pick up the subtle details that may point to some underlying issues and then refer them to the specialists.

To early detect body temperature, artificial intelligence has contributed with two of its technological advances that are ***prediction*** and ***forecasting***.

1.2.5 Prediction

Definition 1: a prediction is a statement which tries to explain a possible outcome or future event. It comes from the Latin term Pre which refers to before and dicer which means say in English. Companies and governments use predictions determined by experts to guide through uncertain projects despite their uncertainty .

Definition 2: a prediction is a technique performed on a database either to predict the response variable value based on a predictor variable or to study the relationship between the response variable and the predictor variables.

1.2.5.1 Health predicting

Personalized predictive medicine necessitates the modelling of patient illness and care processes, which inherently have long-term temporal dependencies. Healthcare observations, stored in electronic medical records are episodic and irregular in time [37].

There are several advantages to quantitative prediction tools that accurately foretell the occurrence of a disease, its prognosis or course, or an individual's likelihood to respond to a certain treatment. Such tools: [8]

✓ enable patients and their families to make more informed decisions about treatment and prevention (for instance, balancing the side-effects of a prevention regimen against the individual's likelihood of experiencing that outcome); [8]

✓ help clinicians precisely tailor care by planning treatment and prevention; [8]

✓ aid health care systems in allocating resources to patients most at risk for an outcome [8].

Prediction models in clinical medicine are not new. For instance, one of the most widely used prediction models is the Framingham Risk Score (FRS). The FRS takes data on cardiovascular factors such as smoking or obesity, and based on a validated logistic regression model a cardiovascular outcome such as stroke or myocardial infarction, produces a probability of that outcome [8]. This probability then informs treatment. For instance, The American College of Cardiology and American Heart Association recommend that statin treatment be initiated if a risk score of 7.5% chance of stroke or myocardial infarction in the next ten years is achieved for 40–75 year patients free from cardiovascular disease [14].

1.2.5.2 Predicting body temperature

Studies approved so far about predicting body temperature have shown good and promising results. However, all these studies were multi-parameter approach based, where the result (predicted temperature) is depending on different physical and physiological parameters like skin temperature, heart rate and particularly skin heat flux.

The paper published by Reto Niedermann [33] on which all the related studies were based, have concluded that multiple physical and physiological parameters at different body sites have to be measured for reliable prediction of core body temperature [33]. Accordingly, predictive models for body temperature measurement, although they resulted accuracies of the proposed methods that are considered to be sufficient to be used as a standard for anomaly detection, are not clinical applications, but to monitor thermal status while working in hazardous conditions and to prevent athletes from peaks of temperature while doing excessive exercises.

1.2.6 Forecasting

Definition 1: Forecasting refers to the process of analysing and elucidating a future state concerning any operation being undertaken. This process takes the past and the current information into account in a bid to predict facts for the future events. In short, forecasting refers to a process of looking forward, and predetermining future trends and the impact on the organization.

Definition 2: A forecast refers to a calculation or an estimation which uses data from previous events, combined with recent trends to come up a future event outcome.

1.2.6.1 Health Forecasting

Health forecasting is a novel area of forecasting, and a valuable tool for predicting future health events or situations such as demands for health services and healthcare needs. It facilitates preventive medicine and health care intervention strategies, by pre-informing health service providers to take appropriate mitigating actions to minimize risks and manage demand. Health forecasting requires reliable data, information and appropriate analytical tools for the prediction of specific health conditions or situations. There is no single approach to health forecasting, and so various methods have often been adopted to forecast aggregate or specific health conditions. Meanwhile, there are no defined health forecasting horizons (time frames) to match the choices of health forecasting methods/approaches that are often applied. The key principles of health forecasting have not also been adequately described to guide the process [49].

1.2.6.2 Health forecasting elucidation

Health forecasting is predicting health situations or disease episodes and forewarning future events. It is also a form of preventive medicine or preventive care that engages public health planning and is aimed at facilitating health care service provision in populations [40] [51] [44]. Health forecasting has been commonly applied to emergency department visits, daily hospital attendance and admissions [4] [19] [7].

There are important terms in forecasting that are worth noting because of the way in which they are used across various fields. The term prediction is mainly used across several fields of study to mean an opinion-based speculation with no explicit causal assumptions . In the health forecasting literature, however, the terms prediction and prognosis could mean different things, even though they are sometimes used interchangeably and without clarity. The term prognosis refers to a forecasting of outcomes under no intervention, whilst prediction is used to mean forecasting health outcomes that are associated with some health-related intervention [21] [38]. Syndromic surveillance is another closely related concept that is well known in disease surveillance literature. This concept focuses on case detection and events that lead to/precede an outbreak, and involves detecting aberrations in the patterns of diseases and using this information to determine future outbreaks [18] [6].

Principles of health forecasting

There are four main principles of health forecasting: [49]

1. the measure of uncertainty and errors;
2. the focus;
3. the nature of data aggregation and how it affects accuracy;
4. the horizon of health forecasting.

These properties are not only hypothetically important, but also have applications that are exemplified in the literature, as discussed below.

1.2.6.3 Measure of error

According to the definition of health forecasting, determining future health events or situations involves a degree of uncertainty, as it is virtually impossible to have a perfect (i.e. 100 % error free) prediction. We therefore describe the measurement of uncertainty and error of health forecasting as a principle in forecasting, because it is a basic requirement, and is also desirable for validation and determining the real value of a forecast. The data used is a major source of uncertainty and error, but this basic problem can partly be addressed methodologically, to obtain health forecasts with the least possible error [52] [49].

1.2.6.4 Focus of a health forecast

The focus of a health forecast relates to the central targeted issue that is being forecast. This is with reference to the basic unit of the health outcome measure that is being forecast. One focus of health forecasting is to predict population health outcome in terms of the number of events occurring within a space of time; for example, the forecasting of life expectancy and health expectancies [27]. Another focus is to determine the course of an ailment for a particular individual, which is usually referred to as prognosis [21]. These two categories are related to how the data is aggregated in health forecasting [49].

1.2.6.5 Horizon of health forecasting

A health forecasting horizon refers to the range of the period the forecast is intended to cover. The demand for a health forecast determines the forecast horizon (range), and this could be in a short, medium or long term. There are no clearly defined boundaries to health forecast horizons in the literature. However, borrowing the common classifications from other disciplines such as finance, business or econometric forecasting, a short-range forecast horizon refers to a period of 1 day to a quarter of a year; a medium-range forecast horizon refers to a quarter of a year to a year; and long-range forecasts refer to a year to five or more years. These horizons are, however, not fixed for all situations, but rather may be defined in relation to the qualitative indicator being forecast (e.g. life expectancy), as well as its weighting over an extended time period. Major population health issues, such as life expectancy or future health expectancies [27], or the forecasting of some chronic disease prevalence (i.e. obesity and diabetes) in large populations [45] [56], are often forecast with a long range. Short-range and medium-range health forecasts are applicable to routine health service uptake (e.g. hospital visits), and some chronic disease exacerbations resulting from environmental exposures [30] [29]. The choice of a long-range, medium-range, or short-range forecast is critical in developing a forecast,as health forecasting horizons also have applications in the planning of health care service deliveries [49].

The discussions around short, medium, and long range health forecasting do not identify some of the fundamental differences in assumptions between the various forecasting horizons. Yet, these differences are important since forecasting future events is based on a strong assumption that the current drivers or predictors will also follow the trend over the future horizon. Hence, long-range forecasting models will be prone to having more "shocks" compared to short-term forecasts. The “shocks” herein refer to disruptions/disturbances of function of the distributions’ equilibrium, which is caused by a significant change in magnitude of the forecast model predictor(s). This may then lead to a shift in the trend. Shocks also have effects on forecast errors because their occurrence, which is between the time of the forecast and the realization of the outcome, determines the error of the forecast. However, research on the mechanisms by which health forecasting models are developed to accommodate shocks at various thresholds is not explicit [49].

1.2.7 Prediction vs Forecasting

Forecasting and prediction are both relate to more or less the same concept, that is future oriented. There however is a fine line that differentiates them in term of :

1. ***Definition***
   - Predicting is saying or telling something before the event while forecasting is done on the basis of analysis of the past.
   - Forecast is scientific and free from intuition and personal bias, whereas prediction is subjective and fatalistic in nature.
2. ***Accuracy***
   - A Forecast is more accurate compared to a prediction. This is because forecasts are derived by analysing a set of past data from the past and presents trends. The analysis helps in coming up with a model that is scientifically backed and the probability of it being wrong are minimal.
   - On the other hand, a prediction can be right or wrong. For example, if you predict the outcome of a football match, the result depends on how well the teams played no matter their recent performance or players.
3. ***Application***
   - Forecasts are only applicable in the economic and meteorology field where there is a lot of information about the subject matter. When it comes to weather forecasting, meteorologist uses collected data such as wind speeds, temperatures, humidity to forecast future weather pattern. The same case applies to economics where current trends and previous performances are used to develop models which generate forecasts.
   - On the contrary, prediction can be applied anywhere as long as there is an expected future outcome.
4. ***Bias***
   Forecasting uses mathematical formulas and as a result, they are free from personal as well as intuition bias. On the other hand, predictions are in most cases subjective and fatalistic in nature.

   For example, if you are predicting the result between two teams, and then you happen to be a supporter of one team, there will be some bias. But this is not the case for scientific methods since they have a way of eliminating bias and enhancing the accuracy of the forecast.
5. ***Quantification***
   - When using a model to do a forecast, it's possible to come up with the exact quantity. For example, the World Bank uses economic trends, and the previous GDP values and other inputs to come up with a percentage value for a country economic growth.

- However, when doing prediction, since there is no data for processing, one can only say the economy of a given country will grow or not. As a result, a prediction value cannot be quantified and in most instances it's vague.

6. ***Basis***

- In most cases, predictions are based on arbitrary methods and experiences such as astrology, superstition, instincts etc.
- On the other hand, forecasts are done using scientific data that is analysed scientifically to generate a model. This implies that a forecast might change if the trends used to derive the models change.

7. ***Application level***

- Predictions are usually done at the instance or a customer level while forecasts are done at the aggregate level. This implies that when making a prediction needs to have a situation in hand which requires estimated future result.
- However, forecasts arise from analysis of data and they may take time to develop.

In short, all forecasts are predictions but not all predictions are forecasts.

FORECASTING VERSUS PLANNING

Basis of Comparison	Forecasting	Prediction
Meaning	Process of creating future predictions with relevant data	Process of creating future predictions with or without relevant data
Accuracy	More accurate	Lower probability of happening
Application	Mostly applied in the meteorology, economic and financial sectors	Can be applied almost anywhere
Bias	Forecasts are generated from calculation and data assessment	Is subject to bias
Quantification	Easily Quantified	Can't be quantified
Basis	Done using scientific methods	Arrived at by arbitrary methods e.g. instincts
Application level	Aggregate level	Customer level

Figure 1.7: Recapitulating table comparing between prediction and forecasting.

1.2.8 Patterns of health data and applications in forecasting

In health forecasting, the pattern of distribution of previous health data over a period of time (i.e. in the form of time series) is important for determining the choice of an appropriate forecasting method. Time series plays an important role in many forecasting approaches, and has been extensively used in subject areas such as climate science, finance and econometrics. The patterns of health data in time series, which are of importance to health forecasting are trend, seasonality, cyclicality, and randomness [29] [2].

Time Series and health forecasting Time series is defined by Shumway and Stoffer [46] as "*a collection of random variables indexed according to the order they are obtained in time*". In the broader literature, time series is similarly defined as a collection of data points that are typically measured at successive and uniformly spaced time intervals. In relation to health forecasting, the importance of this second definition is the emphasis it places on the "uniformly spaced time intervals", which is important in the use of health data for health forecasting. Thus, time series provides statistical setting for describing seemingly random fluctuating health data and projecting the data series into the future [46] [9].

Trend is the long-term variation in a time series that is not influenced by irregular effects or seasonally related components in the data. For instance, in health data, an overall record of a progressively increasing incidence over a specified period would show an increasing trend, irrespective of any random or systematic fluctuations.

When the pattern of health data (e.g. containing the incidence of health events/situations) is influenced by some periodic (long-term/short-term) fluctuations that are associated with other characteristics, it is described as cyclical.*Cyclicality* therefore refers to the extent to which disease incident data points are influenced by overall disease patterns. Seasonality is also a cyclic phenomenon, but is related to annual events, and is described as the predictable and repetitive positions of data points around the trend line within a year. A major difference between cyclical and seasonal patterns is that the former varies in length and magnitude, as compared to the latter. Chatfield describes how seasonality and cyclicality can be estimated either in an additive or multiplicative form [9]. *Additive seasonality* is estimated as a function of the sums of the de-seasonalized mean (m), the seasonal factor (S) and an error term (ε) (i.e. additive seasonality = m + S * ε). *Multiplicative seasonality* is defined by two functions, either the product of m, S and ε (multiplicative seasonality = m·S·ε), or the product of m and S and sum of ε (i.e. multiplicative seasonality = m·S + ε). In order to minimise the overall error, shorter cyclical effects that fall within the annual seasonal effect are best estimated with additive seasonality, whereas the effect of annual seasonality is best computed as "m·S·ε " [9] [49].

Randomness is also a common feature of all time series data, and refers to unexpected distortions of existing or anticipated trends [49].

Lag refers to the lapse of time before an effect is manifested. Lags have proven useful in forecasting events globally, and are a feature of time series data that is widely exploited in many forecasting techniques, e.g. in auto regressive integrated moving averages (ARIMA) [4]. In developing health forecast models for a particular condition/situation, the key questions are: how many days back should one go back in history to identify appropriate predictors, and how many lags should be included [49].

The properties of time series mentioned above require specific treatment prior to any analysis. However, the statistical forecasting models that involve time series analysis and are commonly used in health forecasting include moving average models, such as ARIMA, and smoothing techniques, e.g. the Holt-Winters methods. For instance, the Box–Jenkins ARIMA model, is commonly used in fitting forecasting models when dealing with a non-stationary time series, and this model has been used extensively in health forecasting [7] [4]. *Stationarity* is a feature of trend in a time series, and refers to the level of variation in the statistical properties (such as the mean, variance, auto-correlation, etc.) over time. Smoothing models have also been used in health forecasting studies conducted by Medina et al. [31] and Hyndman et al. [20] [32]. In the study conducted by Champion et al. [7], the authors identified trend, seasonal variations and randomness/"noise" in the data distribution, but used a time series statistical package to automatically identify optimal models to forecast monthly emergency department presentations. After, the authors proceeded to compare forecasts, based on a simple seasonal exponential smoothing model to an ARIMA model. Similarly, the study conducted by Medina et al [31]. also identified seasonal oscillations and trends in the time series data (of the diseases they analysed). The harmonics in the data distributions were handled as level, and trend components by the multiplicative Holt-Winters forecasting method, which is also a smoothing technique in forecasting [31].

1.2.9 Probabilistic health forecasting methods for peak events

Health forecasting techniques generally rely on modelling expectancy of the mean, but this is not useful for looking at extreme events. Nonetheless, extreme events represent the greatest test of a health system, because they expose the weaknesses of the system whenever they occur. A reliable method of modelling and predicting extreme events is therefore important. Quantile regression models (QRMs) and fractional polynomial models (FPMs) are potential probabilistic techniques that could be adopted for predicting extreme health situations/conditions.

Quantile regressions are extensions of the linear-regression models, and do not assume normality of the dependent variable. They model the conditional quantiles as functions of predictors, specifying changes in any conditional quantile [24] [16]. Unlike linear-regression models, QRMs have the ability to characterize the relationship between the dependent variable and the independent variable(s), particularly in the extremes of the distribution. They have common applications in medical reference charts, and could be used in preliminary medical diagnosis to identify unusual subjects by providing robust regressions for estimating extreme values [55]. QRMs also have the potential of predicting and forecasting extreme chronic respiratory illnesses like asthma. For instance, a QRM could be used to estimate extreme variations in daily asthma hospital admissions resulting from the changing patterns of selected meteorological and air quality indicators that are known to exacerbate asthma in a given location/area [48].

Williams [54] also showed how fractional polynomials could be used in modelling specific categories of dependant variables within a linear distribution of data, and thus target specific groups more precisely. In this study, the author used various categories of age groups as regressors to model a dichotomous health care demand. Logistic regression outputs of two arbitrary age-categorized models were then compared to a fractional polynomial model. The polynomial method of categorizing had clear advantages because it allowed a fuller representation of non-linear relationships between the predictor and outcome variables. This approach can be extended to a wide range of health situations or conditions.

Both approaches (QRM and FPM) can be adapted to suit extreme health forecasting.

1.2.10 Challenges in developing and using health forecasts

There are a number of challenging issues to be noted and addressed in developing and using a health forecast. These include limitations in the scope and reliability of health data, the robustness of health forecasting tools and techniques, and the poor demand for health forecasting [21] [28]. In recent times, technological advances have enabled health indicators to be more easily and cheaply measured, and yet the record capture of important population health indicators is not very efficient and not easily accessible or validated [21]. In the practice of personalised medicine, for instance, there are slight prognostic effects attributable to a wide range of complex factors (including some unknown factors), and these factors usually intermingle (randomly) to generate clinical outcomes. Data limitation on these complex factors can pose a challenge in developing a reliable health forecast. Aside from the data and methodological limitations in developing reliable health forecast, it is difficult to convincingly demonstrate the performance of a health forecasting model in realistic settings [5].

Health forecasting-related researches have sometimes focused on methods or procedures for forecasting aggregate health conditions, or on situations like crowding at emergency departments and total admissions [50] [43]. Even though these kinds of aggregate health forecasts are useful, health care providers would be better informed and prepared with condition-specific health forecasts. Therefore, health forecasts need to be more specific for particular health conditions. For example, the health forecast service provided by the United Kingdom Meteorological Office to some Primary Care Trusts (PCT) is very specific for conditions such as COPD [30] [49]. This kind of service is rare but useful.

Health forecasts are most valuable when they provide sufficient warning for timely, remedial action to be taken. Providers make critical decisions and resource allocations to meet the potential demand for health care services. Some of the complexities associated with these types of health care provider actions could range from providing basic social care for early symptoms, to using sophisticated staff and facilities and attending to extreme events [30] [17]. Meanwhile, being able to meet the demand for a health forecast that provides ample time for preparatory activities often requires the use of a good forecasting technique and ample reliable data. It also comes with an additional compromise as to the precision and accuracy of the forecast. Hence, finding a fine line between what is predictable vis-à-vis the demand for specific health forecast is a key challenge in health forecasting.

Another challenge in health forecasting relates to its practical use. A health forecast is usually developed to target the needs of susceptible individuals or institutions (health care providers). In any instance, there is a need for a technology with an intelligent early warning system that can communicate the forecast to the users. Automated telephone services, home visits/treatment, and direct health forecast (to individuals and service providers) are means through which some health forecast services have been delivered [10]. Although there have been some challenges and debates regarding the relevance of some of these existing health forecasting programmes, there are a couple of success stories which provide compelling evidence for their usage [30]. The case of the UK Meteorological Offices' COPD forecast, which was available to general practitioners in Bradford and Airendale, is an example. In 2009, Maheswaran et al. [26] evaluated this health forecasting alert service and failed to show that any change in admissions associated with the forecasting service was significant, and hence they challenged the effectiveness of the COPD forecast. Meanwhile, in cross-sectional study on the acceptability and utility of this same service in England, Scotland and Wales, Marno et al. [30] concluded that the service was both viable and useful. Further research to improve or develop new approaches or schemes in health forecasting is therefore important and will contribute to easing disease burden.

1.3 Conclusion

In this chapter, we have briefly explained some concepts of artificial intelligence and its subfields, presented the most commonly used algorithms and their usages as well as we have discussed both of prediction and forecasting and their utility in healthcare, defined the fine line difference between both and seen the challenges in health forecasting.

In the next chapter, we will introduce our case of study, that is body temperature, see methods of measurement, then talk about the internet of things technology to see how we can beneficiate of it in our project.

Chapter 2

Body temperature and Internet of Things technology

2.1 Introduction

Body temperature is an early warning sign of infection, and monitoring it, even when healthy, can help detect disease early, reason why health monitoring system has been an interesting topic recently among medical practitioners, engineers as well as IT professionals. However, the application of automatic temperature **remotely monitoring** system where physicians can monitor the temperature of their patients is practically new in Algeria.

In this chapter, we will give a background about the study case of our project that is body temperature, see current methods of taking measurement and the advantages and disadvantages of each. Then we are going to explain temperature sensors, define IoT, this new technology that offers the ability to objects to become intelligently connected, give its working principle and some of its benefits that incite us to do shift towards it. Finally, we will give a concise explanation about microcontrollers.

2.2 Temperature

Definition 1: Temperature is the measure of cold or heat. It is measured by a thermometer which has graduations corresponding to a temperature scale.

There are three main temperature scales:

- **Celsius (°C):** 0°C is the melting point of ice, and 100°C is the boiling point of water
- **Fahrenheit (°F):** t(°F) = 1,8 * t(°C) + 32.
- **Kelvin (Kelvin):** t(Kelvin) = t (°C) + 273,15

2.2.1 Methods of measuring body temperature

The reliability of the measurement differs according to the technique used: the most precise method remains the intra-rectal measurement, the others frequently lead to an underestimation of the body temperature at reference temperature, which is located in the centre of the body and is difficult to measure. We distinguish:

1. ***Non-invasive methods:*** Temperature measurement is controversial because the reference temperature is located in the centre of the body, where access is difficult. In these conditions, we are content to go second, taking the temperature in a place not too exposed to ambient air, and yet accessible:

- ***the rectum:*** provides reliable data. The thermometer must be cleaned and disinfected after use, it can be protected by a wrapper (single-use probe);
- ***the mouth:*** is placed under the tongue, mouth closed. The intake should be relatively distant from hot or cold absorption (single-use probe);
- ***the ear:*** infrared tympanic thermometer with an ear piece(single-use probe);
- ***armpit or axillary fold:*** the temperature is 0.5°C lower than the others taken with an electronic thermometer. It may be impossible for cachectic people. The skin should not be rubbed before taking;
- ***groin or inguinal fold:*** same as in the armpit;
- less used, and expensive, an ***infrared camera*** can reveal areas of inflammation (for example, revealing areas affected by arthritis, in veterinary medicine in particular).

2. ***Invasive methods:*** Reserved for hospitalized patients requiring intensive and continuous monitoring, the temperature measurement can be performed with a urinary catheter, an oesophageal catheter or with an arterial catheter equipped with a temperature probe (especially when measuring invasive blood pressure)

2.2.2 Different types of medical thermometers

2.2.2.1 Electronic thermometer

A thermometer that detects temperature changes using a thermoresistive device in which the electrical resistance changes in response to changes in temperature. Electronic thermometers are portable and can be used to measure oral, axillary, and rectal temperatures.

Drawback:

× Small children and people with breathing issues may not be able to keep their mouths closed long enough to acquire an accurate reading.

Figure 2.1: Electronic thermometer.

2.2.2.2 Forehead (infrared) thermometer

This kind of thermometers use infrared sensors to measure the temperature of the superficial temporal artery, which is a branch of the carotid artery. Some are known as non-contact infrared thermometers.

Forehead thermometers that require no physical contact have become very popular for use in venues such as airports, stores, and stadiums. Forehead temperature readings run around 1°F (0.6°C) cooler than oral temperature readings.

Figure 2.2: Forehead (Infrared) thermometer.

Drawbacks:

×Readings can be affected by external factors, including drafts, wind, indoor heating, and direct sunlight.

×Wearing certain clothing, such as hats or heavy coats, can skew the results.

2.2.2.3 Mercury thermometer(liquid in glass)

Invented by physicist Daniel Gabriel Fahrenheit in Amsterdam (1714), mercury thermometers were once the only option available before the technological evolution for taking temperature.

Figure 2.3: Gallium (ear) thermometer.

Drawback:

× Due to safety concerns, they're no longer widely available and may even be illegal where you live.

2.2.2.4 Pacifier thermometer

This may be an easy way to record an approximate temperature for infants who use pacifier.

Figure 2.4: Pacifier thermometer.

Drawbacks:

× Pacifier thermometers must remain in the mouth, without moving, for up to 6 minutes. Additionally, they provide an approximation of temperature rather than an exact reading.

2.2.2.5 Digital ear (tympanic) thermometer

Tympanic thermometers measure the temperature inside the ear canal through infrared ray technology.

Tympanic readings are 0.5°F (0.3°C) to 1°F (0.6°C) higher than oral temperature readings.

Figure 2.5: Tympanic (ear) thermometer.

Drawbacks:

× They may not fit properly in a small or curved ear canal.
× Obstructions like earwax may skew results.
× They must be positioned properly in order to get accurate results.

Assessment Patient health monitoring is a common thing done by doctors to monitor their patients' health. The most crucial reading monitored by doctors is the patient's read time body temperature. Unfortunately, current system used by doctors required them to see patients face to face, and the doctors will have to walk door to door to check the patient's temperature [25] .

As we have seen in subsection 2.2.2, temperature measurement devices used nowadays require direct contact between patient and health practitioner, which might be perilous in case of infectious diseases such as COVID-19

2.2.3 Another way of temperature monitoring

Recent advances in smartwatches have led to several applications in remote health monitoring and mobile health (mHealth) [23].The smartwatch is a new technology that combines features of smartphones with continuous data monitoring that promote health [13], such as temperature. They can provide feedback to users that allow them to monitor their health, perform just-in-time interventions such as medication use based on symptoms, and direct communication with caregivers and physicians [39].

Figure 2.6: Smartwatch.

Besides the ability to wear smartwatches to collect continuous sensing data such as heart rate and activity, smartwatches have many other practical features that make them ideal platforms for healthcare applications [23] [11]. First, unlike smartphones, smartwatches are ubiquitous in that they are typically worn even when at home and during night-time. Also, similar to smartphones, smartwatches are able to combine sensor information such as accelerometers, gyroscopes, compasses, and heart rate, with global positioning satellite (GPS) data. This is particularly promising in applications that require continuous physical activity monitoring to identify unexpected changes in activity patterns and propose alarms and help based on the given localized area. Alarms and messages can also be more easily observed than those sent to smartphones, as individuals can receive vibrations, text, and sounds while wearing the watch. Finally, there is unlimited development potential regarding the use of smartwatches in healthcare applications, and the modularity of software applications (apps) allows for personalization to each individual's healthcare needs [23].

2.3 Temperature Sensors

Definition: A temperature sensor is an electronic device that detects and measures its environment hotness or coldness as input data and then converts these data into either an analogue or digital output to be recorded, monitored, or even signalled and reported as temperature changes. Temperature sensors are the most commonly used type of sensors.

Two kinds of temperature sensors can be distinguished in term of use method:

A. ***Contact temperature sensors:*** this type of temperature can be used to detect a wide range of temperatures requiring to be in physical contact with the sensed object.

B. ***Non-contact temperature sensors:*** is a kind of temperature sensors used to detected energy emitted from objects (generally liquids and gases) and transmitted in form of infrared radiation.

As we can classify temperature sensors into three main types: *Thermocouples, Thermistors and RTD (Resistive Temperature Devices).*

Thermocouple: is a sensor used to measure temperature. It consists of two metals of different kinds joined at one end. When the junction of the metals is heated or cooled, a variable voltage is produced, which can then be translated into temperatureFigure 2.7.

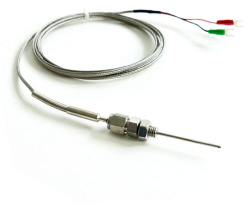

Figure 2.7: thermocouple.

Thermistor: is an electronic component whose electrical resistance varies with temperature and this variation is perfectly reversible. It is one of the main temperature sensors used in electronicsFigure 2.8.

Figure 2.8: Thermistor.

RTD (Resistance Temperature Detector:) is a sensor whose resistance changes as its temperature changesFigure 2.9.

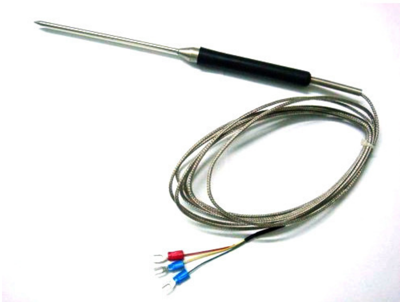

Figure 2.9: Resistance temperature detector (RTD).

The difference between these types is summarized in the table below Table 2.1:

Thermistor	Thermocouple	RTD
-80°C to 150°C	-270°C to 1800°C	-260°C to 850°C
Low	Low	Moderate
Moderate	High	Moderate
Moderate	Low	Best
Best	Low	Moderate
Poor	Moderate	Best
Best sensitivity	Highest temperatures	General purpose sensing

Table 2.1: Types of temperature sensors

2.4 Internet of Things

Till now, the Internet of Things (IoT) has no standard, unified and shared definition yet. Some definitions focus on the technical aspects of IoT and say that it is an extension of the Internet naming system and reflects a convergence of digital identifiers in the sense that it is possible to identify digital information elements and physical elements in a unified way. While others focus more on the uses and functionalities and define it as objects with virtual identities, operating in intelligent spaces and using intelligent interfaces to connect and communicate within various contexts of use.

According to Detlef Shoder [42], IoT is a world of interconnected things which are capable of sensing, actuating and communicating among themselves and with the environment (i.e, smart things or smart objects) while providing the ability to share information and act in parts autonomously to real/physical world events and by triggering processes and creating services with or without direct human intervention.

2.4.1 Working principle of IoT system

Each system is characterized by its working principle that can be unique. As its definition, IoT has no single unified architecture that is agreed on below, we are going to explain a principle of IoT technology that mainly relays on four components as illustrated in Figure 2.10:

1. **Sensor/Device:** the first step in the process is data collection by the sensor/device[1], for example temperature, humidity, etc., readings.

2. **Connectivity:** after collecting data, they have to be sent to the cloud[2]. To enable this step, sensor/device must be connected to the cloud, and this can be done through a variety of methods including: cellular, Wi-Fi, Bluetooth, low-power wide-area networks,etc.

[1] We use the composite term "*Sensor/Device*" rather than only sensor or device to refer that we can bundle several sensors together or sensors can be part of a device that does not only sense things

[2] An IoT cloud is a massive network that supports IoT devices and applications. This includes the underlying infrastructure, servers and storage, needed for real-time operations and processing. An IoT cloud also includes the services and standards necessary for connecting, managing, and securing different IoT devices and applications.

although the same task (connectivity) has to be accomplished, each of the options above has trade-offs (power consumption,range and bandwidth), and choosing the appropriate option comes to the specific IoT application.

3. **Data processing:** now comes the role of the software to process the data we got via the cloud to derive meaningful information. This step could be very simple as checking temperature readings values in a storage garage whether they are getting high. As it could be complex such as identifying a thief on a property using AI.

4. **User interface:** the user interface allows to proactively check in on the system. To complete the examples above, the user in the example above (storage garage manager), might have a mobile application that makes alerts in case of high temperature then he might remotely adjust the temperature inside the storage. As this step can be fully automatic as in case of the 2^{nd} example (detecting a thief through AI technology), where the system can directly notify the security rather than alerting the owner who might not pay attention to the alert. So, this final step depends on the IoT application.

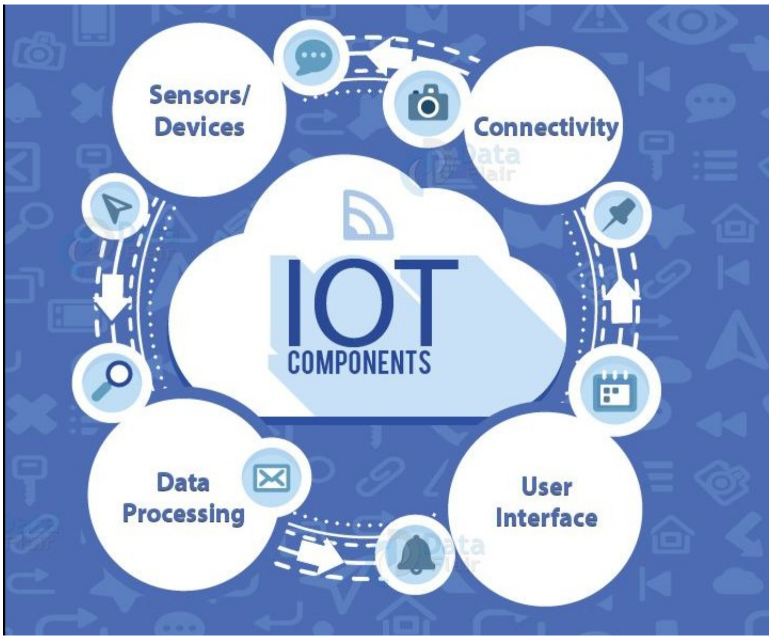

Figure 2.10: Main architecture components of the Internet of Things.

2.4.2 Benefits of IoT technology

- IoT security is the safety component tied to IoT, and it strives to protect IoT devices and networks against cybercrime.
- Helps improving machine learning process that requires a massive amount of data, and this data is being collected by billions of sensors.
- Necessary information are being easily accessed from anywhere, any device in real-time.
- Billions of devices are now capable of *immediately* sharing, receiving and analysing massive amounts of data.

All that we need to benefit from IoT technology goods is a good network.

2.4.3 Microcontrollers

Definition: a microcontroller (MCU) is a compact, easy to carry out (it can be embedded on any device), programmable integrated circuit designed to perform a specific operation in an integrated system. It consists of a processor, a memory and input/output ports on a single board or chip.

Microcontroller VS microprocessor:
Microcontrollers (MCUs) differ from microprocessors (MPUs) in that they are cheaper, easier to install and simpler to use MPUs. Figure 2.11 shows that an MPU consists of several chips that support various features such as memories, interfaces, etc., which is the key feature of it because it makes it possible design a system with a great flexibility.

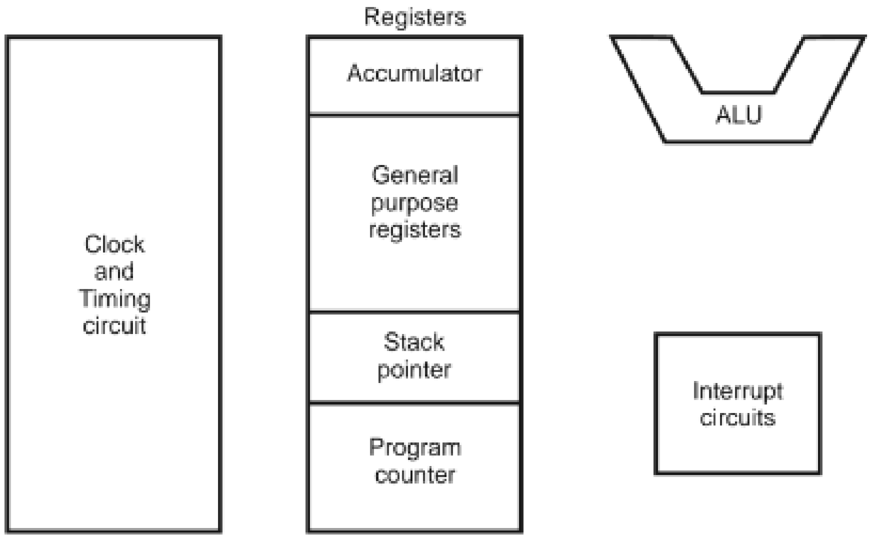

Figure 2.11: Microprocessor block diagram.

On the other hand, an MCU is considered as a single chip computer due to its built-in (on-chip) peripheral devices (ROM, RAM, parallel I/O, serial I/O, counters and clock circuit) that turn it a single-ship computer system as illustrated in Figure 2.12 that allow smaller access time and reduce the hardware size.

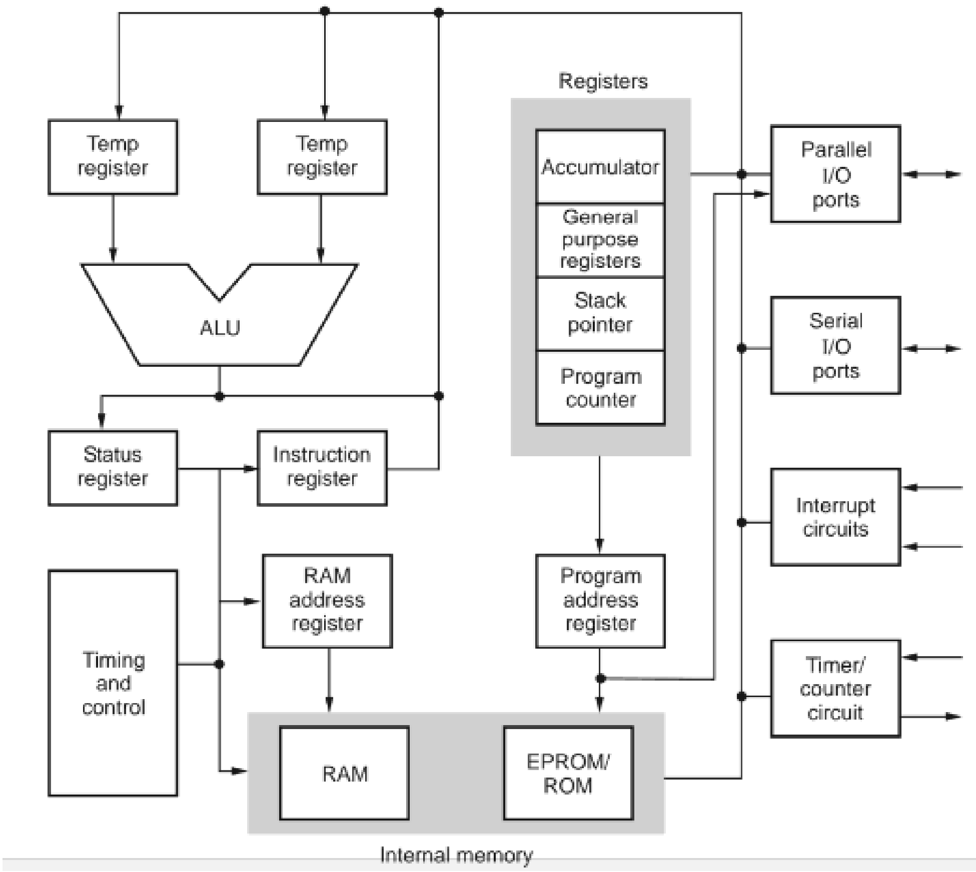

Figure 2.12: Microcontroller block diagram.

Criteria to choose a suitable microcontroller

To choose a microcontroller that is perfectly suited to the needs of the devices you want to build, it is important to take into consideration several factors.

a) ***Computational power:*** it is the first element to consider when choosing. It is depended on the the functionality of the end product, and whether a single core processor is needed or dual core processor, because as we know the higher number of cores, the faster is the microcontroller.

b) ***Energy efficiency:*** it is the trade-off between the computational performance and the power consumption of the microcontroller. The more powerful an MCU is, the more energy it will consume.

c) ***Security:*** with the high risk of hacking of connected objects, a well secured microcontroller is required. So, it is preferable to either choose devices that are certified according to the latest security standards or use microcontrollers with on-chip security.

d) ***Temperature:*** sometimes we may need models with better resistance to extreme temperatures , depending on the environment in which the microcontrollers are to be installed. The more tolerant MCUs are often more expensive.

e) ***Memory:*** the size of program memory (ROM) and random access memory (RAM) varies depending on the programs we want to run. More programs need more RAM.

f) ***Hardware interface:*** it depends on the nature of the task to be performed. If audio, video, camera, USB, Wi-Fi or Bluetooth functionalities are required, the microcontroller has to be chosen correspondingly.

g) ***Software:*** there are microcontrollers that run on several operating systems, and others that do not. It is preferable to use the same software architecture to increase interoperability.

2.5 Conclusion

In this chapter, we have presented body temperature and the ways to measure it, introduced temperature sensors by defining and classifying them. We have also briefly described IoT technology and summed up with presenting the microcontrollers that are necessary for IoT applications.

In the next chapter, we will see how we can beneficiate from the concepts explained in this chapter and the previous one to build a temperature monitoring system and use it to non-invasively forecast future body temperature.

Part II

Design and implementation

Chapter 3

Design

3.1 Introduction

By dint of the successful realization of IoT systems, many machine learning projects have emerged and succeeded thanks to the important amount of data collected. And as remote health monitoring is contemporary, we have decided to realise a wearable device that will play a significant role for the first time in Algeria to remotely monitor body temperature of hospitalized patients in critical cases and forecast their future body temperatures.

In this chapter, we will firstly introduce the idea of our project and present the components as well as the proposed architecture we need to build the wearable device. Then we will put forward the software that will be displayed in the user interface such that is a web interface in which we will plot the temperature values for better visualization.

3.2 Body temperature monitoring device

To realise a wearable device that takes temperature measurements and allows to remotely monitor it, we need a **hardware part** and a **software part**.

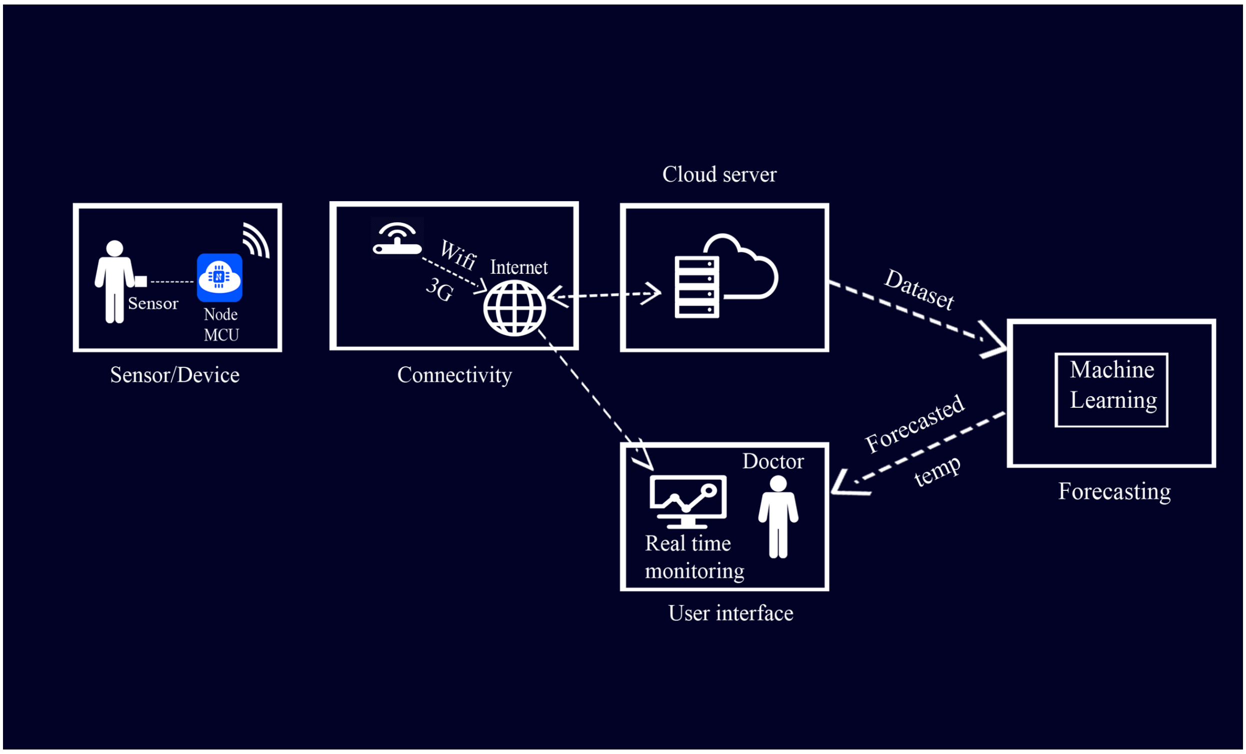

Figure 3.1: General architecture of our project.

Figure 3.1 illustrates the general architecture that will be adopted for this project. As previously explained in subsection 2.4.1, the whole IoT application will be divided into four components where, in our case, for *"sensor/device"* part, we will have the wearable device itself (temperature sensor, microcontroller, battery) that will be explained in subsection 3.2.1. For connectivity, we need a network connection (SSID + password) to enable the device store data in the cloud, and on the other hand, display these data in the user interface subsection 3.2.2

3.2.1 Hardware part

The components below are to be used for the hardware part:

- a sensor that measures temperature;
- a microcontroller to retrieve data values detected by the sensor;
- a Wi-Fi module that allows wireless communication between the device and the cloud;
- a cover to protect the above components.
- a battery.

For our device we have chosen the components described below:

Temperature sensor: as a temperature sensor, we are going to use the LM35 series Figure 3.2 that are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of $\frac{1}{4}$°C at room temperature and, $\frac{3}{4}$°C over a full 55°C to 150°C temperature range.

As the LM35 device draws only 60 µA from the supply, it has very low self-heating of less than 0.1°C in still air.

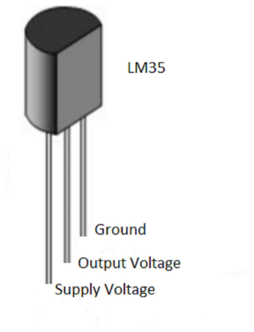

Figure 3.2: LM35 sensor.

Microcontroller and WiFi module : for this part of the device, we normally need two electronic cards (a microcontroller and a WiFi module), but as our device is intended to be wearable, we wanted it to be comfortable thus as small as possible, therefore, we are going to use a microcontroller with an integrated WiFi module.
The ESP8266 Figure 3.3 is a System on a Chip (SoC), manufactured by the Chinese company Espressif. It consists of a Tensilica L106 32-bit micro controller unit (MCU) and a Wi-Fi transceiver. It has 11 GPIO pins* (General Purpose Input/output pins), and an analogue input as well. This means that we can program it like any other microcontroller. And on top of that, you get Wi-Fi communication, so you can use it to connect to our Wi-Fi network, connect to the Internet, host a web server with real web pages, let smartphones connect to it, etc. The possibilities are endless! It is no wonder that this chip has become the most popular IoT device available. It contains a built-in 32-bit low-power CPU, ROM and RAM. It is a complete and self-contained Wi-Fi network solution that can carry software applications as a stand-alone device or connected with a microcontroller (MCU). The module has built-in AT Command firmware to be used with any MCU via COM port. The ESP8266 can be flashed and programmed using the Arduino IDE. Due to its large open source developer community, a large number of libraries for this popular microcontroller is available.

Figure 3.3: ESP8266 NodeMCU.

Battery: In order to power the previous components and make the final realisation portable, we need to add a power supply.

3.2.1.1 Hardware system design

The process of hardware system design includes circuit connections, schematic design, simulation, verification, and testing. This design process provides a detailed understanding of the project and the hardware components. It also helps in preliminary verification and testing of the project architecture. To design the hardware system we need a helpful tool, for our project, we used *Fritzing* Figure 4.10 to model our circuit with the following connections:

- The Vcc of the LM35 goes to 3v of the ESP8266.
- The Out pin of LM35 goes to the A0 of the ESP8266 which is the only analogue pin of the NodeMcu.
- The GND of LM35 goes to the GND of the ESP8266.

The conceptional result is down below Figure 3.4.

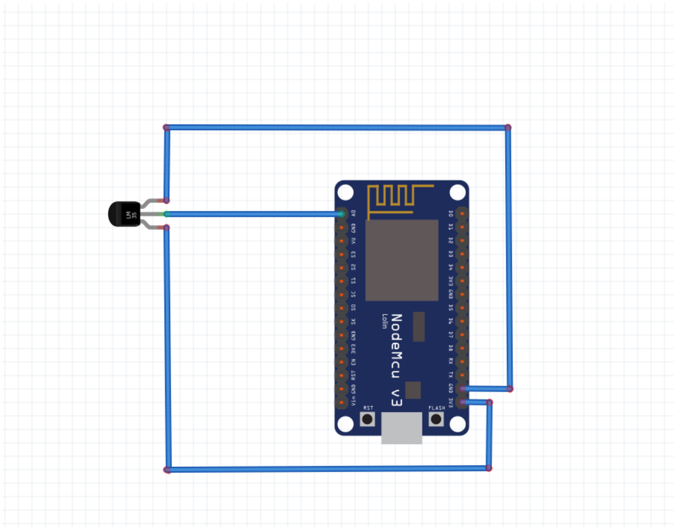

Figure 3.4: Hardware architecture design.

3.2.2 Software part

3.2.2.1 Dataset Collection and storage

As to achieve the flexibility of collecting and storing data (temperature readings), and real-time accessing them, we will create a channel in the open source IoT application "ThingSpeak" Figure 4.12 that is a data platform and application programming interface (API), where we can send and store our data sent from the sensor to the cloud through the WiFi module integrated in the NodeMCU.

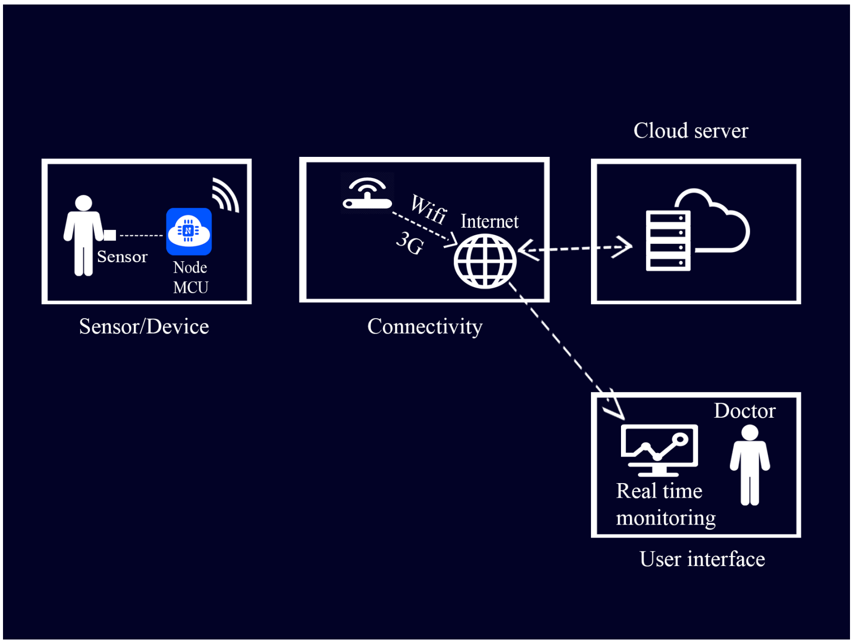

Figure 3.5: Architecture of data collection, storage and display in the UI .

The reason behind choosing cloud storage for our project is to find enough storage space to hold the data we are acquiring. Besides storing data in a remote database, in our project will allow the doctor who is monitoring his patient to access and visualize new data from anywhere and anytime and not only from his office.

Once we sign into ThingSpeak, we will have to create a channel where to store our data privately, visualize and retrieve them as a time-series dataset.

In order to forecast future body temperature based on historical values (time-series dataset), we have proposed a machine learning model that is described in Figure 3.1. Where, after collecting data-set using our body temperature monitoring device and storing it in the cloud, we retrieve it to feed the appropriate algorithm to train it, after that, we get the forecasted body temperature values for next hours which will be displayed in the User Interface Figure 3.6.

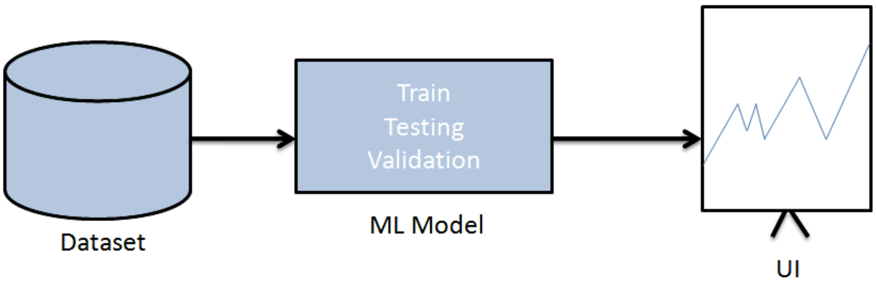

Figure 3.6: Forecasting process.

For our project, to forecast body temperature based on time-series data, we will use the following regression methods as they are used for predicting/forecasting target variables on a continuous scale:

1. **Linear regression:** that fits a linear model with coefficients $w = (w_1, \ldots, w_p)$ to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation [36].

2. **Ridge regression:** that is a method of estimating multiple regression model coefficients in scenarios where independent variables are highly correlated.

3. **Decision tree regressor:** used to fit a sine curve with addition noisy observation. As a result, it learns local linear regressions approximating the sine curve. If the maximum depth of the tree (controlled by the max_depth parameter) is set too high, the decision trees learn too fine details of the training data and learn from the noise, i.e. they overfit [36].

4. **Random forest regressor:** A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max_samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree [36].

5. **K-Neighbours regressor:** meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction. As such, subsequent regressors focus more on difficult cases [36].

6. **Lasso regressor:** Least Absolute Shrinkage and Selection Operator regression, is a shrinkage and variable selection method for regression models, is an attractive option as it addresses both problems,aims to identify the variables and corresponding regression coefficients that lead to a model that minimizes the prediction error [22].

7. **Gradient boosting regressor:** GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage a regression tree is fit on the negative gradient of the given loss function [36].

8. **Adaboost regressor:** it is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction [36].

Among the methods described above, three of them (Gradient Boosting Regressor, Adaboost Regressor and Random Forest Regressor) belong to **ensemble methods** set, where ensemble learning is a technique that combines different models of machine learning in order to increase the performance and accuracy.

It is not arbitrary to elect the appropriate model to be used, as we have previously mentioned in subsubsection 1.2.3.2, there are ways to evaluate models, and for regression models that we have used we can use the Root Mean Square Error (RMSE) and R^2 calculating techniques to evaluate the performance of each model where :

- RMSE is the standard deviation of the prediction/forecast errors, it is defined as follows:

$$\boldsymbol{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\Delta_i^2)} \quad [35]$$

  Where N represents the total number of data samples. Δ_I= Y'- Y is the error between the predicted Y' value and the true Y for the i_{th} test samples.

- R^2 is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model, where:

$$\boldsymbol{R}^2 = \boldsymbol{1}\text{-}\frac{\sum_i (y_i - \hat{y})^2}{\sum_i (y_i - \overline{y})^2} \text{ [34]}$$

Concerning the user interface, we will develop a temperature monitoring application to monitor temperature evolution of patients during 24 hours. The application will allow doctors, patients, and other possible third parties (eg: parents) to check the temperature of patients and display the forecast of its evolution in the next hours based on the data collected which was predicted in the previous step using machine learning techniques.

3.3 Conclusion

In this chapter, we have precisely explained the idea of our project, as we have presented tools and methods to realise it in both hardware and software part step by step.

In the upcoming chapter, we will see how to realise the whole project and implement the software part then forecast the future body temperature.

Chapter 4

Implementation

4.1 Introduction

As part of our attempt to realise a wearable device that collects a time-series dataset. This latter will be used to forecast future body temperature as we have introduced in the previous chapter, this chapter is made to achieve our objective.

In this chapter, we will present the whole process of realising our project, from making the electronic device that collects data, to the way of storing then doing our forecasts, and finally displaying the forecasted temperature values on a reactive web interface.

4.2 Hardware realisation

After modelling our circuit using Fritzing software Figure 4.10, we have connected the real components with the following connections:

- The Vcc of the LM35 goes to 3v of the ESP8266.
- The Out pin of LM35 goes to the A0 of the ESP8266 which is the only analogue pin of the NodeMcu.
- The GND of LM35 goes to the GND of the ESP8266.

The result of the hardware connection is down below in Figure 4.1

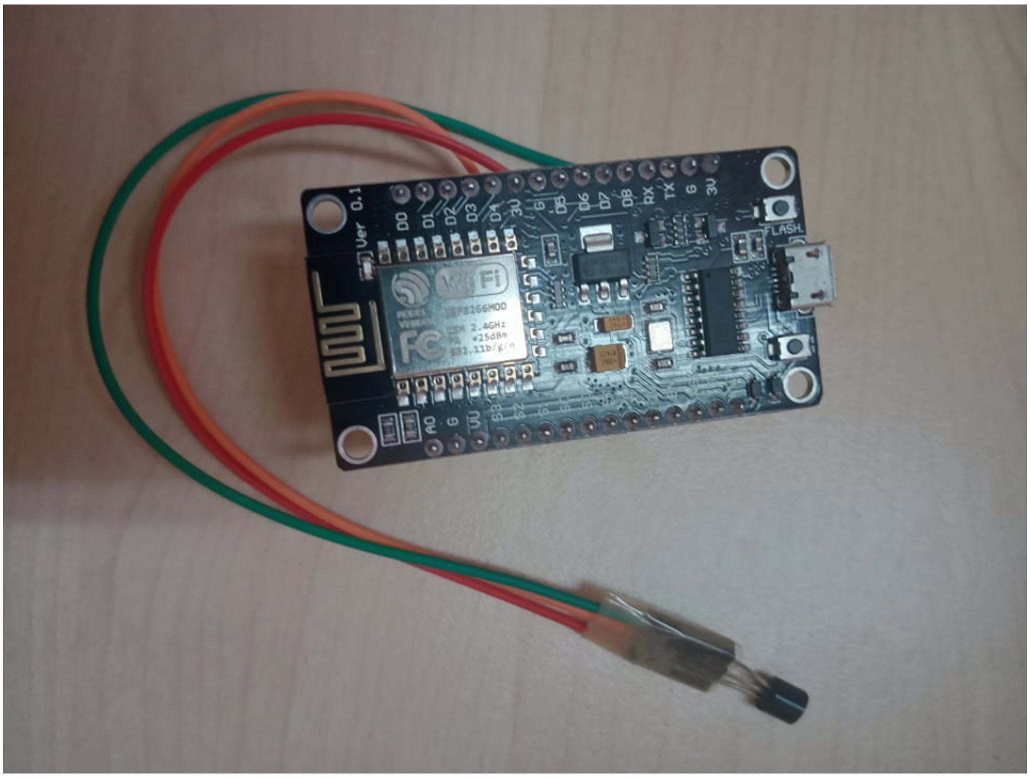

Figure 4.1: First connection of the hardware using the sensor and ESP8266.

- The Vcc pin is connected through the green wire.
- The Out pin is connected through the orange wire.
- The GND pin is connected through the red wire.

And we got the result illustrated below in Figure 4.2

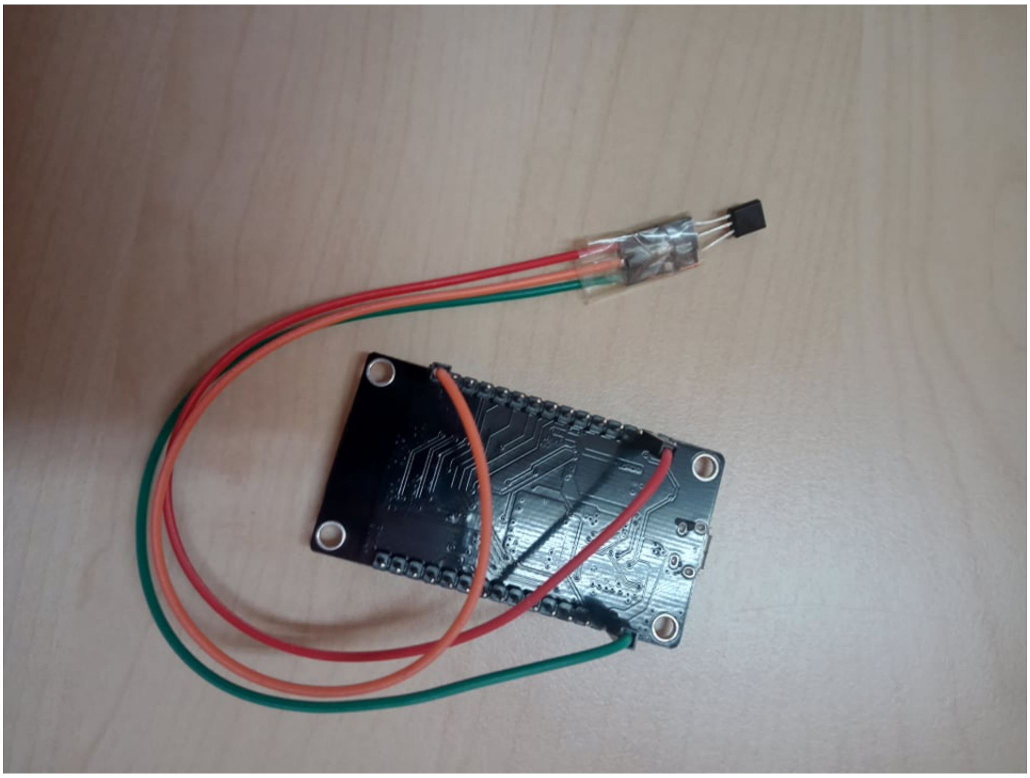

Figure 4.2: First connection of the hardware using the sensor and ESP8266 (other side).

In order to power our device we have used a 5600mAh external battery and a data cable (containing a data line to enable data sharing)Figure 4.3.

Figure 4.3: The external battery used in our project.

Figure 4.4: "Sensor/Device" powered Figure 4.4.

4.3 Software implementation

4.3.1 Development tools and languages

Python is an open source, interpreted, object-oriented, high-level programming language with dynamic semantics programming language that includes a lot of supporting libraries. Python is used for data analytics, machine learning, and even design.

Figure 4.5: Python Logo

Figure 4.6: Matplotlib Logo

Matplotlib is a plotting library for Python. It is used along with NumPy to provide an environment that is an effective open source alternative for MatLab. It can also be used with graphics toolkits like PyQt and wxPython.

Figure 4.7: Google Colab Logo

Colaboratory, often shortened to "Colab", is a product of Google Research. Colab allows anyone to write and run any Python code of their choice through the browser. It is an environment particularly suited to machine learning, data analysis and education.

Figure 4.8: sklearn Library Logo

Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy .

Figure 4.9: pandas Library Logo

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. Pandas makes it simple to do many of the time consuming, repetitive tasks associated with working with data (Data cleansing, data fill, data inspecting,data visualization, etc.).

Figure 4.10: Fritzing Software logo.

Fritzing is a free circuit modelling software that can be installed in pretty much any computer the good thing about it is that it does not use a lot of resources and can work in any computer, the default version of Fritzing comes without the NodeMcu v3 model, which means that we had to download it and manually installed it.

Figure 4.11: Arduino IDE logo.

The Arduino IDE is a cross-platform application developed in Java that can be used to develop, compile, and upload programs to the Arduino board, The IDE contains a text editor for coding, a menu bar to access the IDE components, a toolbar to easily access the most common functions, and a text console to check the compiler outputs. A status bar at the bottom shows the selected Arduino board and the port name that it is connected to, as shown in . An Arduino program that is developed using the IDE is called a sketch. Sketches are coded in Arduino language, which is based on a custom version of C/C++.

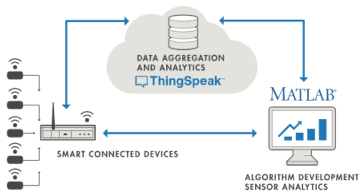

Figure 4.12: ThingSpeak Logo

ThingSpeak is an IoT analytics service that allows you to aggregate, visualize, and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices to ThingSpeak. With the ability to execute MATLAB code in ThingSpeak, you can perform on-line analysis and process data as it comes in. ThingSpeak is often used for prototyping and proof-of-concept IoT systems that require analytics.

Figure 4.13: dash Library Logo

Dash is an open-source Python framework used for building analytical web applications. It is a powerful library that simplifies the development of data-driven applications. Users can create amazing dashboards in their browser using dash because it ties modern UI elements like dropdowns, sliders and graphs directly to their analytical python code.

4.3.1.1 Arduino Sketch

We have used ArduinoIDE to program our device (body temperature monitor), and as we have used a microcontroller with a built-in Wi-Fi module. We first had to call the *ESP8266.h* library, that is a Wi-Fi library developed using the naming conventions and overall functionality of the *Arduino WiFi* library.

#include <ESP8266WiFi.h>

Then added the followed three lines of code:

- **String apiWritekey = "03T4AADIQ6L###";** this string is provided by the API of ThingSpeak service for each channel we create, in our case for each user of the device. Otherwise, if we had realised two copies of this device we would have had two apiWriteKey and then for each copy we have to modify this line of code with the specific write key before the implementation. *Note:* if we share this write key with another device, there would be an overlap and confusion in the channel, which means that we would have 2 values at the same.

- **const char* ssid = "khadijaMK";** this is the name of the network we were connected to (Gateway).

- **const char* password = "12#####"** ;and of course we have also to write the password of the gateway.

WiFiClient client; allows the instantiation of a client from WiFiClient class. #include <ESP8266WiFi.h>) **const char* server = "api.thingspeak.com";** indicate that we are hosting our web interface in thingspeak server.
While **resolution** variable is created to simplify the function of conversion the read value from voltage to a clear value, where 3.3 is the voltage given to LM35 sensor and 1023 is the analogue pin resolution starting from 0.

```
LM35ESP8266
#include <ESP8266WiFi.h>

String apiWritekey = "03T4AADIQ6L[redacted]";
const char* ssid = "khadijaMK"; //
const char* password = "12[redacted]" ;//

const char* server = "api.thingspeak.com";
float resolution=3.3/1023;//
WiFiClient client; //
```

To successfully compile the Arduino sketch, it is required to use both of *setup()* and *loop()* that are are mandatory functions.

Setup code explanation:

The void setup() is the first function to be executed in the sketch and it is executed only once. It usually contains statements that set the pin modes.

LM35ESP8266

```
void setup() {
  Serial.begin(115200);
  WiFi.disconnect();
  delay(10);
  WiFi.begin(ssid, password);

  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("NodeMcu connected to wifi...");
  Serial.println(ssid);
  Serial.println();
  Serial.println("IP address :");
  Serial.print(WiFi.localIP());
  Serial.println();
}
```

Serial.begin(115200) opens serial port, sets data rate to 115200 bps, what comes after that is connecting to a new session on the given Network SSID and password

Loop code explanation:

This is where the main code of Arduino sketch is executed. The program runs over and over since the board is empowered.

LM35ESP8266

```
void loop() {
  float temp = (analogRead(A0) * resolution) * 100;

  if (client.connect(server,80))
  {
    String tsData = apiWritekey;
           tsData +="&field1=";
           tsData += String(temp);
           tsData += "\r\n\r\n";

     client.print("POST /update HTTP/1.1\n");
     client.print("Host: api.thingspeak.com\n");
     client.print("Connection: close\n");
     client.print("X-THINGSPEAKAPIKEY: "+apiWritekey+"\n");
     client.print("Content-Type: application/x-www-form-urlencoded\n");
     client.print("Content-Length: ");
     client.print(tsData.length());
     client.print("\n\n");
     client.print(tsData);
```

The first line in the function represents the conversion of voltage reading to a normalized value (celsius in here).
After that, if the device is successfully connected to the server, we start uploading data to it.

```
     Serial.print("Temperature: ");
     Serial.print(temp);
     Serial.println("uploaded to Thingspeak server....");
  }
  client.stop();

  Serial.println("Waiting to upload next reading...");
  Serial.println();
  // thingspeak needs minimum 15 sec delay between updates
  delay(15000);
}
```

This fragment of code, is done to display all the above information to the serial monitor, and the result is down below:

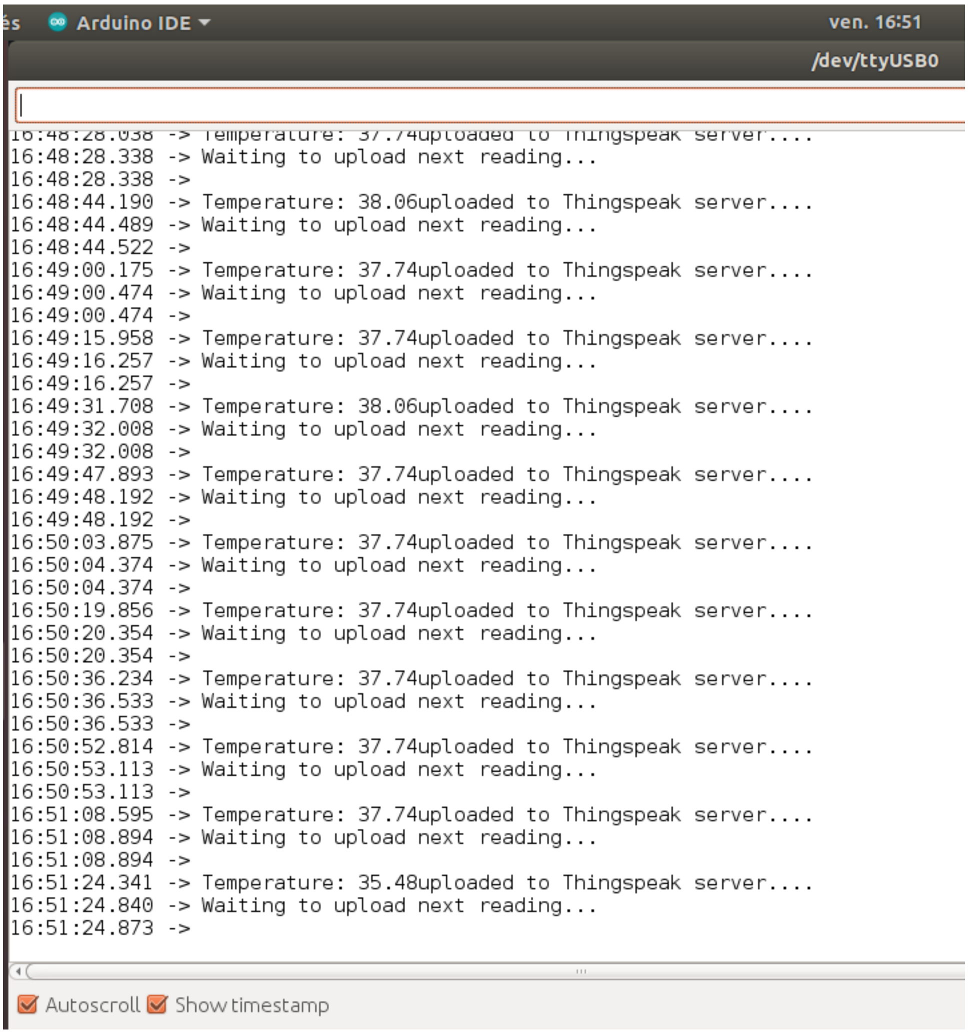

Figure 4.14: Collected temperature values displayed on ArduinoIDE Serial Monitor.

Above in Figure 4.14, we can see temperature readings every 15 seconds, the sensor is stable and reliable as it has returned stable values when put on the skin (37.74°C), returned above 38°C when quietly warmed it up and 35.48°C when cooled down.

After that we have stored data in ThingSpeak, we had exported them as a csv file to use it for forecasting.

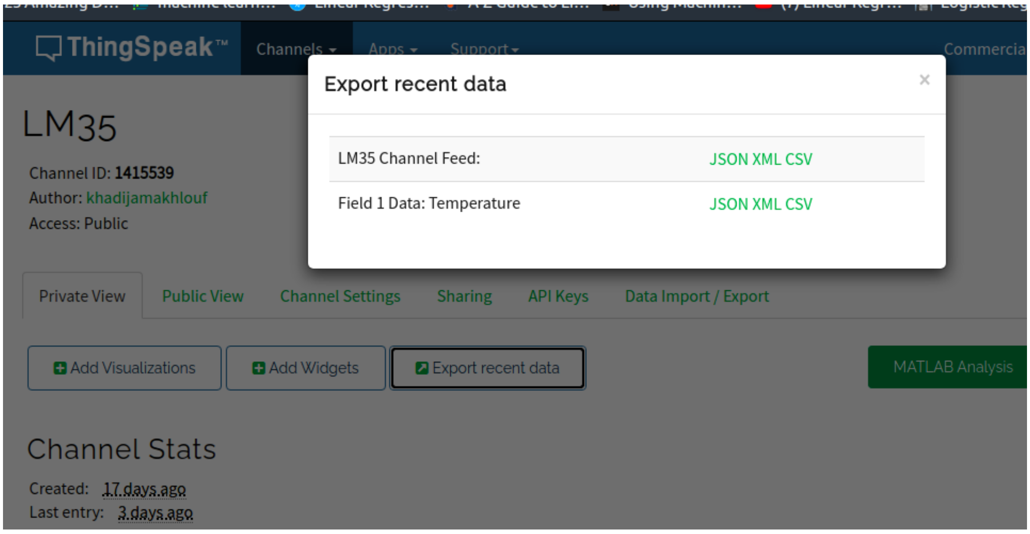

Figure 4.15: Exporting time-series dataset from the ThingSpeak.

4.3.1.2 Machine learning models implementation

Now comes the machine learning process to forecast body future temperature values. To do so, we have used **google colab** to implement the models. First, we imported the necessary packages depicted in Figure 4.16.

```python
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import (RandomForestRegressor, GradientBoostingRegressor,
                              AdaBoostRegressor)
from sklearn.metrics import mean_squared_error, r2_score
```

Figure 4.16: Necessary python packages.

We have downloaded a dataset from DataWorld, a cloud-native data catalogue, that contains wrist skin temperature of the same person, captured by a BASIS smart watch each minute during 24hours and imported it to python code to train our model.

```python
df = pd.read_excel
("/content/drive/MyDrive/basis_70days_skintemp_conditional_formatting_visual.xlsx",
 na_values=" None")
```

Figure 4.17: Dataset import.

The imported dataset is split into training and forecasting sets, 25% of the measurements are selected for the training of the forecasting models, while the remaining 75% are aimed for the forecasting, in this step, we used a function that splits arrays or matrices into random train and test subsets Figure 4.18.

```
x_train, val_X, y_train, val_Y = train_test_split(x, Y,
                    test_size=0.5, random_state = 1)#, random_state = 1
```

Figure 4.18: splitting dataset.

Training step

Various popular machine learning methods are performed for comparison purposes, including:"Linear Regression", "Ridge Regression", "Lasso Regression', "K Neighbors Regressor", "Decision Tree Regressor", "Random Forest Regressor", "Gradient Boosting Regressor", and "Adaboost Regressor", using default parameters.

```
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import (RandomForestRegressor, GradientBoostingRegressor,
                              AdaBoostRegressor)
names = ['Linear Regression', 'Ridge Regression', 'Lasso Regression',
         'K Neighbors Regressor', 'Decision Tree Regressor',
         'Random Forest Regressor', 'Gradient Boosting Regressor',
         'Adaboost Regressor']
models = [LinearRegression(), Ridge(), Lasso(),
          KNeighborsRegressor(), DecisionTreeRegressor(),
          RandomForestRegressor(), GradientBoostingRegressor(),
          AdaBoostRegressor()]
from sklearn.metrics import mean_squared_error, r2_score
Model = []
RMSE = []
R_sq = []

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)
  RMSE.append(mean_squared_error(val_Y.reshape(-1, 1), Y_prd.reshape(-1, 1)))
  R_sq.append(r2_score(val_Y.reshape(-1, 1), Y_prd.reshape(-1, 1)))

evaluation = pd.DataFrame({'Model': Model,
                           'RMSE': RMSE,
                           'R Squared': R_sq})
print("FOLLOWING ARE THE TESTING SCORES: ")
evaluation
```

Figure 4.19: Performed models.

Among all methods, Linear Regression, Ridge Regression, Lasso Regression and Adaboost Regressor performed worse (higher RMSE) than the remaining methods. On other hand, K-Neighbors Regressor , Decision Tree Regressor , Random Forest Regressor , Gradient Boosting Regressor, Adaboost Regressor achieved a lower RMSE than other comparing methods. We picked the best model based on RMSE metric, which is Random Forest Regressor Table 4.1.

Model	RMSE	R^2
Linear Regression	9.664388	0.136574
Ridge Regression	9.664388	0.136574
Lasso Regression	9.666494	0.136386
K-Neighbour Regressor	0.397253	0.964509
Decision Tree Regressor	0.295361	0.973612
Random Forest Regressor	**0.265717**	0.976261
Gradient Boosting Regressor	0.407368	0.963605
Adaboost Regressor	1.641101	0.853382

Table 4.1: Machine learning training scores

As Random forest regressor performed best, we have elected it to forecast future body temperature and it has been trained on the first 360 measurements, and predicted the body temperature in the upcoming 1081 measurements, as shown below in Figure 4.20

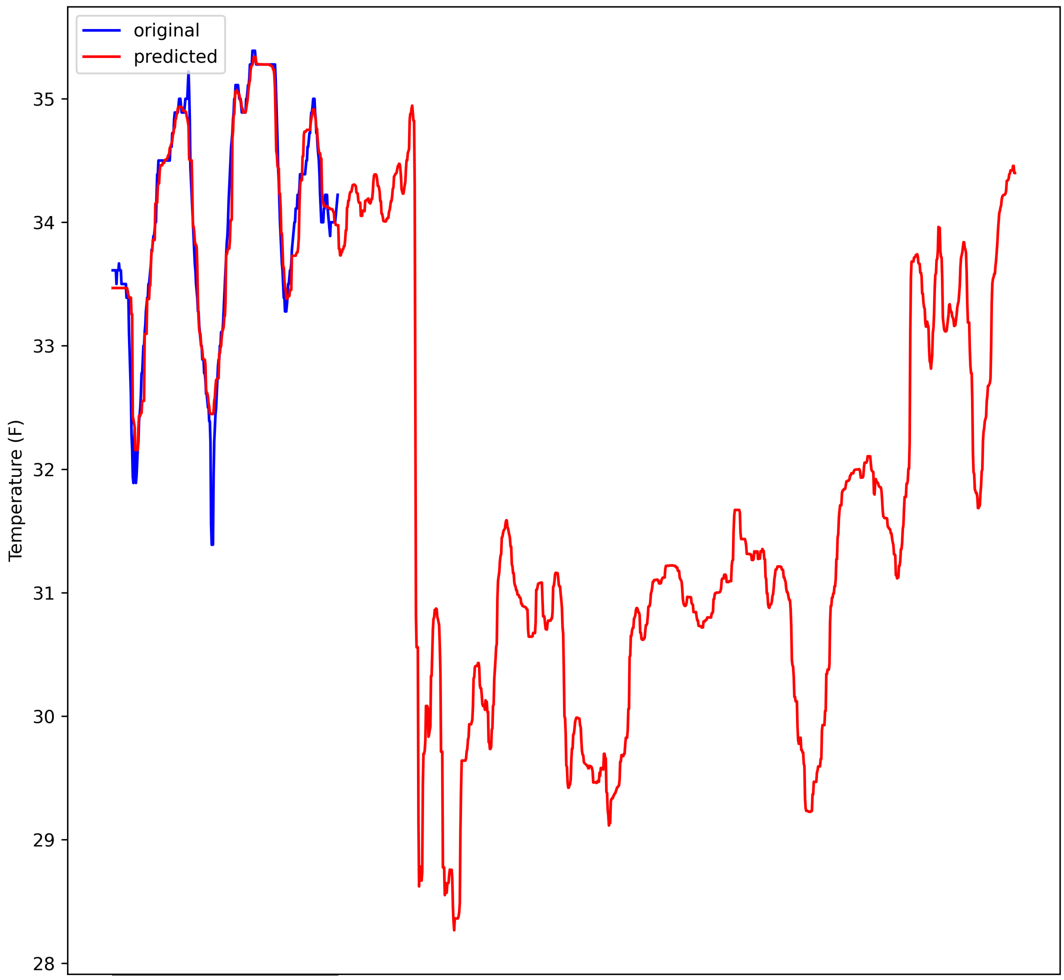

Figure 4.20: plot of the training step.

Testing step Predicted body temperature of Random Forest method are depicted in Figure 4.21. It is worth noting that the body temperature forecasting values are more reliable and closer to the true values, where the forecasted values have the same trend of change with true values.

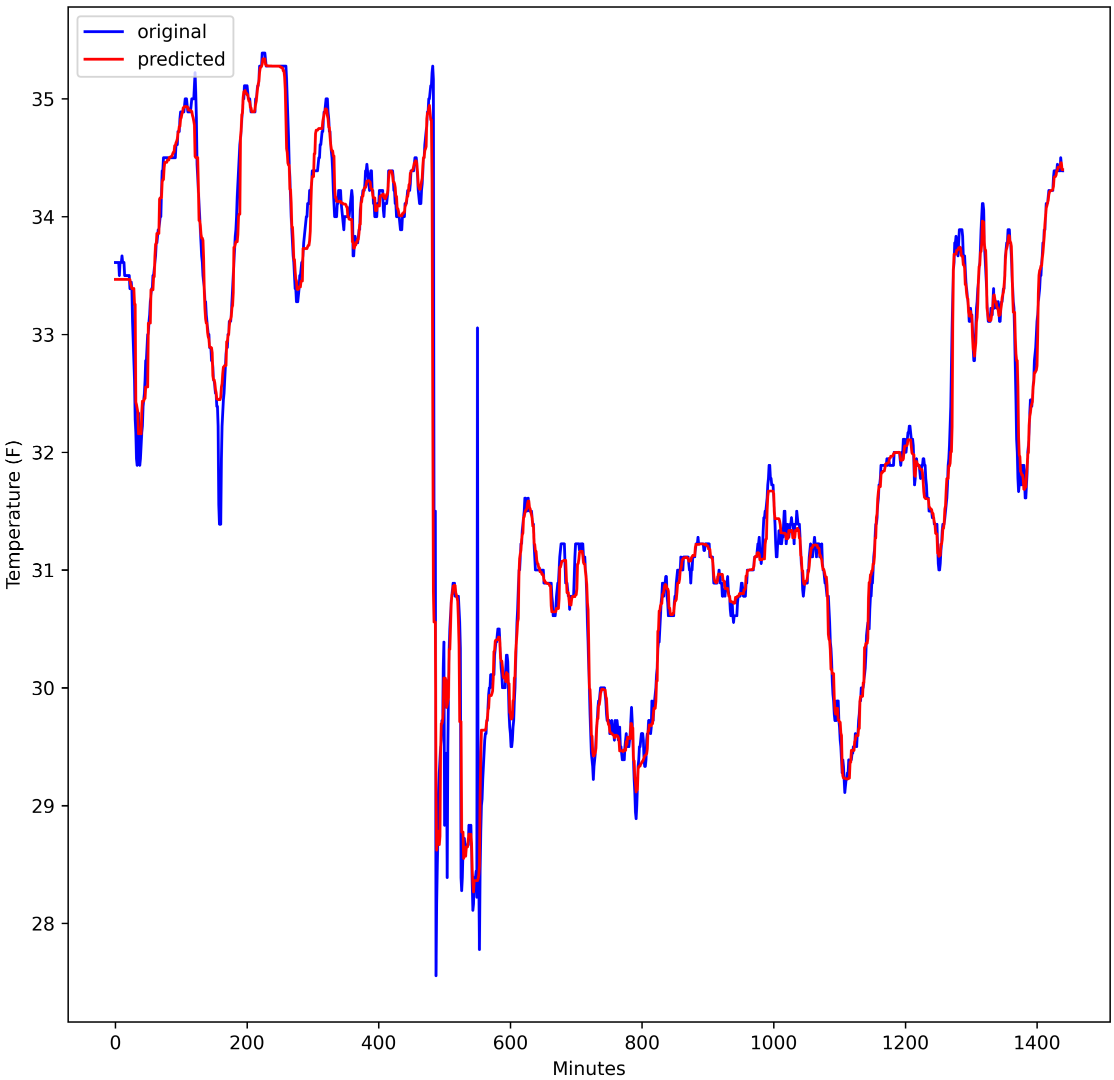

Figure 4.21: Plot of the testing step.

Training using our own dataset
We have imported our dataset that we have started collecting using the device we made and applied the same previous process on it Figure 4.22.

```
df = pd.read_csv("/content/drive/MyDrive/feed.csv")
```

Figure 4.22: Importing our own dataset.

But unfortunately, as we could not get a varied dataset (i.e: different body temperature measurement that we usually get from ill people), the result we have got were not satisfying and are depicted below in Figure 4.23 for training step and Figure 4.24 for testing step:

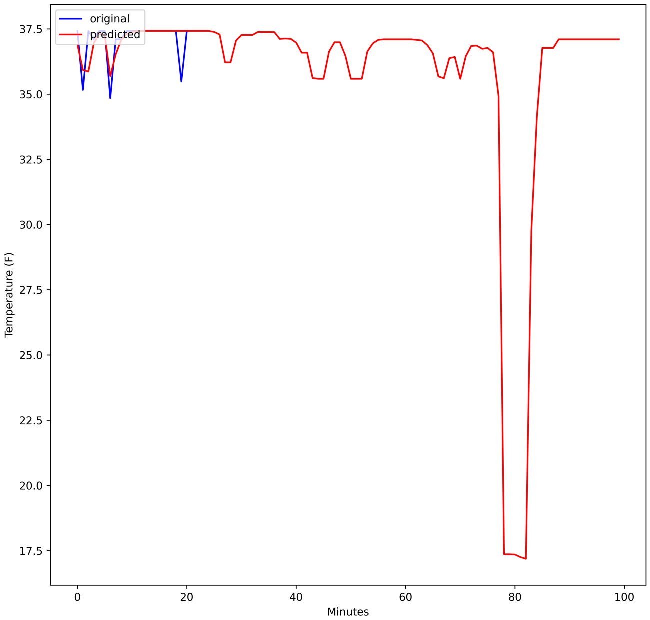

Figure 4.23: Training step.

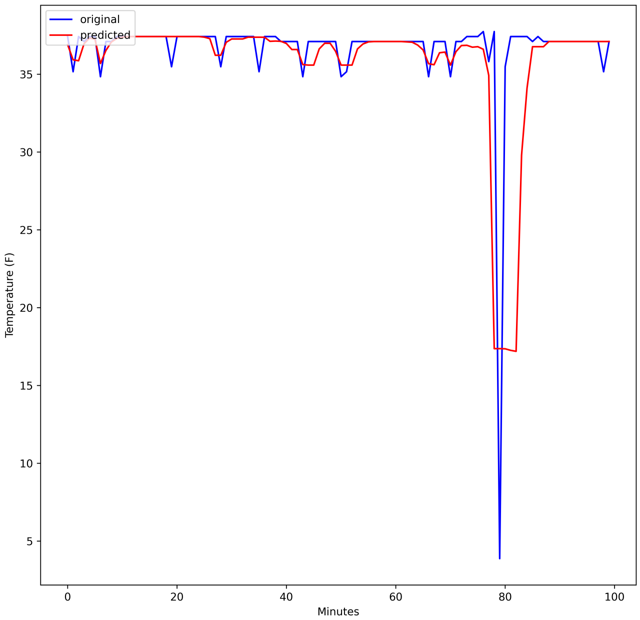

Figure 4.24: Testing step.

Web application for visualization

The Temperature Monitoring application was developed as an interactive Web Application that is accessible via internet to all authorized users in a secure manner. The development steps of this application are as follows:

1. **Install the required libraries to run the web application:** Required libraries are: pandas, plotly, dash(+ dash-html- components, dash-core-components).

```
[ ]  1  pip install -r /content/drive/MyDrive/Requirements.txt #install the required packages

Requirement already satisfied: pandas==0.24.2 in /usr/local/lib/python3.7/dist-packages (from -r /content/drive/MyDrive/Requirements.txt (line 1)) (0.24.2)
Requirement already satisfied: plotly==4.1.0 in /usr/local/lib/python3.7/dist-packages (from -r /content/drive/MyDrive/Requirements.txt (line 2)) (4.1.0)
Requirement already satisfied: dash==1.2.0 in /usr/local/lib/python3.7/dist-packages (from -r /content/drive/MyDrive/Requirements.txt (line 4)) (1.2.0)
Requirement already satisfied: dash-html-components==1.0.1 in /usr/local/lib/python3.7/dist-packages (from -r /content/drive/MyDrive/Requirements.txt (line 6)) (1.0.1)
Requirement already satisfied: dash-core-components==1.1.2 in /usr/local/lib/python3.7/dist-packages (from -r /content/drive/MyDrive/Requirements.txt (line 8)) (1.1.2)
Requirement already satisfied: python-dateutil>=2.5.0 in /usr/local/lib/python3.7/dist-packages (from pandas==0.24.2->-r /content/drive/MyDrive/Requirements.txt (line 1)) (2.8.1)
Requirement already satisfied: numpy>=1.12.0 in /usr/local/lib/python3.7/dist-packages (from pandas==0.24.2->-r /content/drive/MyDrive/Requirements.txt (line 1)) (1.19.5)
Requirement already satisfied: pytz>=2011k in /usr/local/lib/python3.7/dist-packages (from pandas==0.24.2->-r /content/drive/MyDrive/Requirements.txt (line 1)) (2018.9)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from plotly==4.1.0->-r /content/drive/MyDrive/Requirements.txt (line 2)) (1.15.0)
Requirement already satisfied: retrying>=1.3.3 in /usr/local/lib/python3.7/dist-packages (from plotly==4.1.0->-r /content/drive/MyDrive/Requirements.txt (line 2)) (1.3.3)
Requirement already satisfied: dash-table==4.2.0 in /usr/local/lib/python3.7/dist-packages (from dash==1.2.0->-r /content/drive/MyDrive/Requirements.txt (line 4)) (4.2.0)
Requirement already satisfied: flask-compress in /usr/local/lib/python3.7/dist-packages (from dash==1.2.0->-r /content/drive/MyDrive/Requirements.txt (line 4)) (1.10.1)
Requirement already satisfied: Flask>=1.0.2 in /usr/local/lib/python3.7/dist-packages (from dash==1.2.0->-r /content/drive/MyDrive/Requirements.txt (line 4)) (1.1.4)
Requirement already satisfied: future in /usr/local/lib/python3.7/dist-packages (from dash==1.2.0->-r /content/drive/MyDrive/Requirements.txt (line 4)) (0.16.0)
Requirement already satisfied: dash-renderer==1.0.1 in /usr/local/lib/python3.7/dist-packages (from dash==1.2.0->-r /content/drive/MyDrive/Requirements.txt (line 4)) (1.0.1)
Requirement already satisfied: brotli in /usr/local/lib/python3.7/dist-packages (from flask-compress->dash==1.2.0->-r /content/drive/MyDrive/Requirements.txt (line 4)) (1.0.9)
Requirement already satisfied: Jinja2<3.0,>=2.10.1 in /usr/local/lib/python3.7/dist-packages (from Flask>=1.0.2->dash==1.2.0->-r /content/drive/MyDrive/Requirements.txt (line 4)) (2.11.3)
Requirement already satisfied: itsdangerous<2.0,>=0.24 in /usr/local/lib/python3.7/dist-packages (from Flask>=1.0.2->dash==1.2.0->-r /content/drive/MyDrive/Requirements.txt (line 4)) (1.1.0)
Requirement already satisfied: Werkzeug<2.0,>=0.15 in /usr/local/lib/python3.7/dist-packages (from Flask>=1.0.2->dash==1.2.0->-r /content/drive/MyDrive/Requirements.txt (line 4)) (1.0.1)
Requirement already satisfied: click<8.0,>=5.1 in /usr/local/lib/python3.7/dist-packages (from Flask>=1.0.2->dash==1.2.0->-r /content/drive/MyDrive/Requirements.txt (line 4)) (7.1.2)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dist-packages (from Jinja2<3.0,>=2.10.1->Flask>=1.0.2->dash==1.2.0->-r /content/drive/MyDrive/Requirements.txt (line 4)) (2.0.1)
```

Figure 4.25: Required libraries installing.

2. **Install the ngrok package:** To create a machine learning Web Application, ngrok is installed. This tool generates a public URL that can be shared with anyone in the world (temporarily if the free version of ngrok is used, if its paid then the URL is permanent).

```
[5] # To run a Dash app in Google Colab
    ## It is required to install ngrok
    #which creates a secured URL accessible on any machine and by anyone.

    !wget https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.zip
    !unzip ngrok-stable-linux-amd64.zip

    --2021-07-01 21:03:55--  https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.zip
    Resolving bin.equinox.io (bin.equinox.io)... 52.73.79.40, 50.17.89.192, 34.195.88.198, ...
    Connecting to bin.equinox.io (bin.equinox.io)|52.73.79.40|:443... connected.
    HTTP request sent, awaiting response... 200 OK
    Length: 13832437 (13M) [application/octet-stream]
    Saving to: ‘ngrok-stable-linux-amd64.zip.1’

    ngrok-stable-linux- 100%[===================>]  13.19M  6.53MB/s    in 2.0s

    2021-07-01 21:03:58 (6.53 MB/s) - ‘ngrok-stable-linux-amd64.zip.1’ saved [13832437/13832437]

    Archive:  ngrok-stable-linux-amd64.zip
    replace ngrok? [y]es, [n]o, [A]ll, [N]one, [r]ename:
```

Figure 4.26: Ngrok installing.

3. **Run ngrok to generate an URL for our Web application:** ngrok takes a port that is available on your localhost and exposes it to the internet with a public URL in a secure manner. The generated URL in Figure 4.27 is *http://f4f8a3350311.ngrok.io*

```
[6]

### Run ngrok to tunnel Dash app port 8050 to the outside world.
### This command runs in the background.
get_ipython().system_raw('./ngrok http 8050 &')

### Get the public URL where you can access the Dash app. Copy this URL.
! curl -s http://localhost:4040/api/tunnels | python3 -c \
    "import sys, json; print(json.load(sys.stdin)['tunnels'][0]['public_url'])"
```

```
http://f4f8a3350311.ngrok.io
```

Figure 4.27: Generated URL.

4. **Create the web application:** in this step, we have loaded the collected temperatures from 0 to 360 minutes (0 to 4hours), in addition to the forecasted temperatures from 360 to 1440 minutes (4 to 24 hours). Then, we have created our web application which is designed using html. The application contains a title “html.H1” and a graph that plots the curve of temperature evolution in time range of 24 hours “dcc.graph”.

```
%%writefile my_app1.py
import dash
import dash_html_components as html
import pandas as pd
import dash_core_components as dcc

real = pd.read_csv('/content/real.csv') # read the file which contains the data collected by us.
real.drop(real.columns[0], axis=1, inplace=True)
real.to_csv('real.csv', index=True)
real = pd.read_csv('/content/real.csv')

forcasted = pd.read_csv('/content/Y_predict.csv', sep=',') #read the file enerated by our Machie learning model of the forecasted data.

indexes = [x for x in range(360, 1441)] #used to generate x-axis value for the forecasted values.

external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']

app = dash.Dash(__name__)  #create the app
                           #design the app and graph
app.layout = html.Div(children=[
    html.H1(children='Temperature in 24 hours (1440 minutes)'),  #head title of our web application

    dcc.Graph(
        id='body-temperature',
        figure={
            'data': [
                {'x': real.index, 'y': real.iloc[0:360, 1], 'type': 'line', 'name': u'Measured Temperature'},
                {'x': indexes, 'y': forcasted.iloc[359:1140, 0], 'type': 'line', 'name': u'Forecasted Temperature'},
                ],
        }),
])

if __name__ == '__main__': #launch the web application
    app.run_server(debug=True)
```

```
Overwriting my_app1.py
```

Figure 4.28: Main Python code.

5. **Run the web application**

```
1  !python my_app1.py

Running on http://127.0.0.1:8050/
Debugger PIN: 855-229-541
 * Serving Flask app "my_app1" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
Running on http://127.0.0.1:8050/
Debugger PIN: 437-682-223
```

Figure 4.29: Running the web application.

6. **Navigate to Web Application:** navigate to the web application using the URL (Figure 4.27). The application could be consulted with other users any time and anywhere.

Figure 4.30: Visualization of the forecasted temperature.

4.4 Conclusion

In this chapter we have seen how we realised our wearable device and collected data, stored them then retrieved as time-series data set. Then we have seen the machine learning implementation, performed models and comparison between them and how we have chosen the most appropriate model that has fit with the dataset and given the best result. Then we have done the same process with the dataset we have started collecting by our device and we have summed up by displaying the forecasted results on a reactive web application.

General Conclusion and perspectives

General Conclusion and perspectives

Early detection of body temperature measurement has recently became an important subject in health field. Therefore, many researchers in medical field are investing their experience and exploiting the technical advances of artificial intelligence and machine learning to get non-invasive and accurate early human body temperature measurements.

In this thesis, we have presented machine learning in chapter 1, and how it can be beneficial in health care. In chapter 2, we have briefly defined body temperature in general and the ways it is being measured as we have introduced temperature sensors. After that, we have precisely talked about IoT technology and its benefits to see why and how we can use it to achieve our goal, as we have also presented the microcontrollers as they are an essential part to be included in our monitoring system.

In the third chapter, we have planned for our project and precisely given both hardware and software designs that have been realised and implemented in chapter 4. Where we have seen the realisation of the device that collects data and stores it in the cloud from where we could retrieve them afterwards and fed the appropriate machine learning model to forecast future body temperature. We choose to use Random Forest Regressor because it performed best among all the methods that achieved lower RMSE.

In this work, we have realized a hard part which is the wearable device to monitor the temperature and collect dataset. Additionaly, we trained a machine learning model and tested it on an upload dataset the first time and our dataset the second time. We have developpend also a reactive web application to visualize the forecast temperature.

From our experiments, we came to the conclusion that when we use the data set that we collected using the the wearable device, we do not get satisfactory results. This deficiency is due to the lack of variation in the temperature readings as we could not collect them from a person with a fever.

Hence, this work opens a field to several future works starting by collecting a consistent and varied dataset to get better forecasts or evolving the user interface application.

Bibliography

[1] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, 1993.

[2] adrian g barnett and annette j dobson. *Analysing seasonal health data*, volume 30. Springer, 2010.

[3] Parth Bhavsar, Ilya Safro, Nidhal Bouaynaya, Robi Polikar, and Dimah Dera. Machine learning in transportation data analytics. In *Data analytics for intelligent transportation systems*, pages 283–307. Elsevier, 2017.

[4] Justin Boyle, Melanie Jessup, Julia Crilly, David Green, James Lind, Marianne Wallis, Peter Miller, and Gerard Fitzgerald. Predicting emergency department admissions. *Emergency Medicine Journal*, 29(5):358–365, 2012.

[5] David L. Buckeridge, Howard Burkom, Murray Campbell, William R. Hogan, Andrew W. Moore, et al. Algorithms for rapid outbreak detection: a research synthesis. *Journal of biomedical informatics*, 38(2):99–113, 2005.

[6] Howard S. Burkom, Sean Patrick Murphy, and Galit Shmueli. Automated time series forecasting for biosurveillance. *Statistics in medicine*, 26(22):4202–4218, 2007.

[7] Robert Champion, Leigh D. Kinsman, Geraldine A. Lee, Kevin A. Masman, Elizabeth A. May, Terence M. Mills, Michael D. Taylor, Paulett R. Thomas, and Ruth J. Williams. Forecasting emergency department presentations. *Australian Health Review*, 31(1):83–90, 2007.

[8] Benjamin P. Chapman, Feng Lin, Shumita Roy, Ralph H. B. 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.

[9] Chris Chatfield. *The analysis of time series: an introduction*. Chapman and Hall/CRC, 2003.

[10] Richard Cooper and Rachel O'Hara. Patients' and staffs' experiences of an automated telephone weather forecasting service. *Journal of health services research & policy*, 15(2_suppl):41–46, 2010.

[11] Frederic Ehrler and Christian Lovis. Supporting elderly homecare with smartwatches: advantages and drawbacks. *Studies in health technology and informatics*, 205:667–71, 2014.

[12] Bradley J Erickson, Panagiotis Korfiatis, Zeynettin Akkus, and Timothy L Kline. Machine learning for medical imaging. *Radiographics*, 37(2):505–515, 2017.

[13] E. M. Glowacki, Y. Zhu, and E. Hunt. *University of Texas, Austin*, 2016.

[14] David C. Goff, Donald M. Lloyd-Jones, Glen Bennett, Sean Coady, Ralph B. D'agostino, Raymond Gibbons, Philip Greenland, Daniel T. Lackland, Daniel Levy, Christopher J. O'donnell, et al. 2013 acc/aha guideline on the assessment of cardiovascular risk: a report of the american college of cardiology/american heart association task force on practice guidelines. *Journal of the American College of Cardiology*, 63(25 Part B):2935–2959, 2014.

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

[16] Lingxin Hao, Daniel Q. Naiman, and Daniel Q. Naiman. *Quantile regression*. Number 149. Sage, 2007.

[17] Deborah Hemming, A. Colman, P. James, N. Kaye, P. Marno, D. McNeall, R. McCarthy, T. Laing-Morton, E. Palin, P. Sachon, et al. framework for copd forecasting in the uk using weather and climate change predictions. In *IOP Conference Series: Earth and Environmental Science (EES)*, volume 6, 2009.

[18] Kelly J. Henning. What is syndromic surveillance? *Morbidity and mortality weekly report*, pages 7–11, 2004.

[19] Nathan R. Hoot, Larry J. LeBlanc, Ian Jones, Scott R. Levin, Chuan Zhou, Cynthia S. Gadd, and Dominik Aronsky. Forecasting emergency department crowding: a discrete event simulation. *Annals of emergency medicine*, 52(2):116–125, 2008.

[20] Rob J. Hyndman, Anne B. Koehler, Ralph D. Snyder, and Simone Grose. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of forecasting*, 18(3):439–454, 2002.

[21] John P. A. Ioannidis. Limits to forecasting in personalized medicine: an overview. *International Journal of Forecasting*, 25(4):773–783, 2009.

[22] R. Jonas and J. A. Cook. lasso regression. *British Journal of Surgery*, 105(10), 2018.

[23] Christine E. King and Majid Sarrafzadeh. A survey of smartwatches in remote health monitoring. *Journal of healthcare informatics research*, 2(1):1–24, 2018.

[24] Roger Koenker and Kevin F. Hallock. Quantile regression. *Journal of economic perspectives*, 15(4):143–156, 2001.

[25] Tonny Heng Yew Ling, Lim Jin Wong, Jocelyn Ee Hung Tan, and Kwong Yao Kiu. Non-intrusive human body temperature acquisition and monitoring system. In *2015 6th international conference on intelligent systems, modelling and simulation*, pages 16–20. IEEE, 2015.

[26] Ravi Maheswaran, Tim Pearson, Nagpal Hoysal, and Michael J. Campbell. evaluation of the impact of a health forecast alert service on admissions for chronic obstructive pulmonary disease in bradford and airedale. *Journal of public health*, 32(1):97–102, 2010.

[27] Istvan Majer. Modeling and forecasting health expectancy: Theoretical framework and application. 2011.

[28] Spyros Makridakis and Nassim Taleb. Decision making and planning under low levels of predictability, 2009.

[29] P. Marno, C. Bryden, W. Bird, and H. A. Watkin. how different measures of cold weather affect chronic obstructive pulmonary disease (copd) hospital admissions in london. *European Respiratory Review*, 15(101):185–186, 2006.

[30] Penny Marno, Melanie Chalder, Tish Laing-Morton, Mark Levy, Patrick Sachon, and David Halpin. can a health forecasting service offer copd patients a novel way to manage their condition? *Journal of health services research & policy*, 15(3):150–155, 2010.

[31] Daniel C. Medina, Sally E. Findley, Boubacar Guindo, and Seydou Doumbia. forecasting non-stationary diarrhea, acute respiratory infection, and malaria time-series in niono, mali. *PLoS One*, 2(11):e1181, 2007.

[32] Joshua A. Mott, David M. Mannino, Clinton J. Alverson, Andrew Kiyu, Jamilah Hashim, Tzesan Lee, Kenneth Falter, and Stephen C. Redd. cardiorespiratory hospitalizations associated with smoke exposure during the 1997 southeast asian forest fires. *International journal of hygiene and environmental health*, 208(1-2):75–85, 2005.

[33] Reto Niedermann, Eva Wyss, Simon Annaheim, Agnes Psikuta, Sarah Davey, and René Michel Rossi. Prediction of human core body temperature using non-invasive measurement methods. *International journal of biometeorology*, 58(1):7–15, 2014.

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

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

[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[37] 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.

[38] K. Poikolainen. A comment diagnosis as a means of health forecasting. *Soc Sci Med Part A Med Psychol Med Sociol*, 13:165–166, 1979.

[39] Blaine Reeder and Alexandria David. Health at hand: a systematic review of smart watch uses for health and wellness. *Journal of biomedical informatics*, 63:269–276, 2016.

[40] David P. Rogers, Melvyn A. Shapiro, Gilbert Brunet, Jean-Claude Cohen, Stephen J. Connor, Adama Alhassane Diallo, Wayne Elliott, Kan Haidong, Simon Hales, Debbie Hemming, et al. Health and climate–opportunities. *Procedia Environmental Sciences*, 1:37–54, 2010.

[41] Thomas Ross. The synthesis of intelligence—its implications. *Psychological Review*, 45(2):185, 1938.

[42] Detlef Schoder and Q. F. Hassan. introduction to the internet of things. In *Internet of Things A to Z: Technologies and Applications*, pages 1–50. John Wiley & Sons, Inc. Hoboken, NJ, USA, 2018.

[43] Lisa M. Schweigler, Jeffrey S. Desmond, Melissa L. McCarthy, Kyle J. Bukowski, Edward L. Ionides, and John G. Younger. Forecasting models of emergency department crowding. *Academic Emergency Medicine*, 16(4):301–308, 2009.

[44] Neelam Sekhri, Rob Chisholm, Andrea Longhi, Peter Evans, Mark Rilling, Edward Wilson, and Yvette Madrid. global health forecasting working group background paper. 2006.

[45] Lu Shi, Jeroen van Meijgaard, and Jonathan Fielding. peer reviewed: forecasting diabetes prevalence in california: a microsimulation. *Preventing chronic disease*, 8(4), 2011.

[46] Robert H. Shumway, David S. Stoffer, and David S. Stoffer. *Time series analysis and its applications*, volume 3. Springer, 2000.

[47] Zdenko Sonicki and Josip Kern. Is artificial intelligence really a new topic in medical education? *Croatian Medical Journal*, 62(2):200–200, 2021.

[48] I. N. Soyiri and D. D. Reidpath. predicting extreme asthma events in london using quantile regression models. *Using administrative databases to identify cases of chronic kidney disease: a systematic review*, page 39, 2011.

[49] Ireneous N. Soyiri and Daniel D. Reidpath. An overview of health forecasting. *Environmental health and preventive medicine*, 18(1):1–9, 2013.

[50] Yan Sun, Bee Hoon Heng, Yian Tay Seow, and Eillyne Seow. Forecasting daily attendances at an emergency department to aid resource planning. *BMC emergency medicine*, 9(1):1–9, 2009.

[51] J. van Meijgaard, J. Fielding, R. Shimkhada, E. Eslami, and B. Cole. Forecasting health outcomes related to air quality for placer county. *Issue brief. Health Forecasting, UCLA School of Public Health*, 2010.

[52] 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*, 124(6):778–789, 2009.

[53] Lizhi Wang, Zhaohui Zhang, Xiaobo Zhang, Xinxin Zhou, Pengwei Wang, and Yongjun Zheng. Chapter one-a deep-forest based approach for detecting fraudulent online transaction. *Adv. Comput.*, 120:1–38, 2021.

[54] Jennifer Stewart Williams. Assessing the suitability of fractional polynomial methods in health services research: a perspective on the categorization epidemic. *Journal of health services research & policy*, 16(3):147–152, 2011.

[55] Keming Yu, Zudi Lu, and Julian Stander. Quantile regression: applications and current research areas. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52(3):331–350, 2003.

[56] Paola Zaninotto, Heather Wardle, Emmanuel Stamatakis, Jennifer Mindell, and Jenny Head. *Forecasting obesity to 2010.* Department of Health London, 2006.