ALGERIAN DEMOCRATIC AND POPULAR REPUBLIC

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

YAHIA FARES UNIVERSITY OF MEDEA



Faculty of Technology

Department of Electrical Engineering

End studies Project

sector: Automation

speciality: Automation and Systems

development of an intelligent IoT-based system for medical diagnostic assistance.

Presented by :

Bouteldja Othmane
Hosni Aniss Toufik

Proposed and Supervised by :

Dr.kara reda
Mr. Abderrahmane Saad

Class 2024

Acknowledgments

Before we begin, let us thank and praise Almighty God for everything He has given us, for granting us the will, health, patience, and courage to achieve our goal.

We are pleased to express our sincere gratitude to our supervisor, Mr. Kara, who despite his heavy responsibilities, generously agreed to guide us and meticulously review our work. We are particularly grateful for his continuous support.

We also thank Mr. Saad, who accompanied us at every step, big or small, of this work and who has been a helpful hand to us whenever needed. He deserves our full appreciation for the patience he has shown and for the advice, technical assistance, materials, and encouragement he provided us throughout the year.

ملخص:

تم تطوير نظام ذكي متصل يستخدم أحدث التقنيات في مجال الاتصالات والذكاء الاصطناعي لمراقبة حالة المرضى الذين يعانون من اضطرابات نظم القلب. يقوم النظام بتحليل بيانات القلب والكشف عن الأنماط غير الطبيعية في نبضاته، مما يمكن من تشخيص ومعالجة الحالات القلبية بدقة. النظام قادر على تحديد الارتجاج البطيني وغيره من الاضطرابات القلبية بدقة، مما يحسن جودة الرعاية الصحية ويوفر مراقبة مستمرة للمرضى. تم بناء نموذج CNN لعرض النتائج التنبؤية على منصة Node-Red، وتم التحقق من فعالية النظام وجدواه.

الكلمات المفتاحية: إشارات تخطيط القلب ECG، القلب، أألمراض القلبية، IOT، CNN، API، Node-Red.

Asbact

An intelligent system using advanced technologies in communications and artificial intelligence was developed to monitor patients with heart rhythm disorders. This system analyzes cardiac data comprehensively and detects abnormalities in heartbeats, enabling precise diagnosis and treatment of cardiac conditions. Its accuracy in identifying ventricular fibrillation and other heart disorders can improve healthcare quality and provide continuous patient monitoring. A CNN model was built to predict outcomes, displayed on the Node-Red platform, and the system's effectiveness was verified.

Keywords : ECG signals, heart, cardiovascular disease, CNN, IoT, API, Node-Red.

.

Résumé

Un système intelligent utilisant les dernières technologies en communications et en intelligence artificielle a été développé pour surveiller l'état des patients souffrant de troubles du rythme cardiaque. Ce système analyse les données cardiaques et détecte les anomalies des battements, permettant un diagnostic et un traitement précis des affections cardiaques. Grâce à sa capacité à identifier précisément la fibrillation ventriculaire et d'autres troubles cardiaques, ce système peut améliorer la qualité des soins de santé et fournir une surveillance continue des patients. Un modèle CNN a été construit pour prédire les résultats, affichés sur la plateforme Node-Red, et l'efficacité du système a été vérifié

.

Mots-clés : Signaux ECG, cœur, maladie cardiovasculaire, CNN, IoT, API, Node-Red

Table of Contents

LISTE OF FIGURES.....I

LIST OF TABLES.....III

LIST OF ABBREVIATIONS.....IV

GENERAL INTRODUCTION.....1

CHAPTER I: OVERVIEW OF THE HEART.....3

III.1 INTRODUCTION 4
III.2 THE HEART 4
III.2.1 Anatomy of the Heart *4*
III.3 STUDY OF A CARDIAC CYCLE 6
III.3.1 The mechanical activity of the heart *6*
III.3.2 The electrical activity of the heart *6*
III.4 ACTION POTENTIAL 7
III.5 ELECTROCARDIOGRAM (ECG) 8
III.5.1 Electrical Characteristic of Cardiac Tracing *8*
on each heartbeat, we can identify: *9*
III.5.2 Normal ECG *10*
III.6 STANDARD ECG LEADS 10
III.6.1 The bipolar peripheral derivations *11*
III.6.2 Derivations *11*
III.6.3 Precordial (thoracic) leads *12*
III.7 TYPES OF ECG SIGNAL NOISE 13
III.7.1 Technical Noise *13*
III.7.2 Physical Interferences *15*
III.8 CARDIAC DISEASES 16
III.8.1 Ventricular extrasystole (VES) *16*
III.8.2 Ventricular tachycardia *17*
III.8.3 Atrial fibrillation *17*
III.8.4 Ventricular Tachycardia *18*
III.9 CONCLUSION: 19

CHAPTER II: AI & IOT.....20

IV.1 INTRODUCTION 22
IV.2 OVERVIEW OF ARTIFICIAL INTELLIGENCE 22
IV.2.1 Intelligence Artificielle *22*
IV.3 MACHINE LEARNING 23
IV.3.1 Types of Machine Learning *23*
IV.4 DEEP LEARNING 24
IV.4.1 Supervised learning *24*
IV.4.2 Unsupervised learning *25*
IV.5 NEURAL NETWORKS 25
IV.5.1 Biological Neuron *25*
IV.5.2 The artificial neuron *26*
IV.5.3 The Activation Function *27*
IV.5.4 Type of neural networks: *31*
IV.6 INTERNET OF THINGS 36

IV.6.1 Definition *36*
IV.6.2 IOT Architecture *37*
IV.6.3 Components of IoT *38*
IV.7 Conclusion 40

CHAPTER III: MATERIALS AND SOFTWARE USED 41

V.1 Introduction 42
V.2 Application Programming Interfaces 42
V.2.1 How do Application Programming Interfaces (APIs) work? *42*
V.3 Fast API 43
V.3.1 Fast API's Features *43*
V.4 MQTT (Message Queue Telemetry Transport) 44
V.4.1 Operating Mode *44*
V.4.2 Components of MQTT *45*
V.4.3 Why MQTT is so important in IoT? *45*
V.5 Database 45
V.5.1 Definition of a database *45*
V.5.2 Role of a Database *46*
V.6 MYSQL 46
V.6.1 Definition *46*
V.6.2 Historical *47*
V.6.3 Benefits *47*
V.7 Python 47
V.8 Node-RED 48
V.9 Arduino IDE 49
V.10 AD8232 ECG Sensor 49
V.11 Libraries used 50
V.11.1 Tensorflow *50*
V.11.2 Keras *50*
V.11.3 Sklearn *51*
V.11.4 Matplotlib *51*
V.12 Conclusion 52

CHAPTER IV: IMPLEMENTATION AND EXPERIMENTAL TESTING 53

VI.1 Introduction 54
VI.2 Database (1000 Fragments ECG) 54
VI.2.1 Database Description *54*
VI.3 ECG Signal Preprocessing 55
VI.4 Model used : 56
VI.4.1 Multi layer perceptron *56*
VI.4.2 Cconvolutional neural network *58*
VI.4.3 Combining CNN with Long Short-Term Memory (CNN-LSTM) *60*
VI.4.4 Comparison of models *62*
VI.5 Experimental test 63
VI.5.1 Implementation of the acquisition node *63*
VI.5.2 Implementation *64*
VI.6 node red test 66
VI.7 Conclusion 68

GENERAL CONCLUSION 69

BIBLIOGRAPHIES 71

Liste of Figures

Figure I.1: The Thoracic Cage 4
Figure I.2: The Human Heart 5
Figure I.3: The heart envelope 6
Figure I.4:Electrical Operation 7
Figure I.5:The five phases of action potential and ion transfers at the cellular level 8
Figure I.6:ECG Signal Form 9
Figure I.7:ECG the isoelectric line 9
Figure I.8:PR Segment 10
Figure I.9:ST Segment 10
Figure I.10:Einthoven setup for recording bipolar limb leads 11
Figure I.11:Unipolar Derivation 12
Figure I.12:Positions of precordial electrodes 13
Figure I.13:50 Hz Line Interference 14
Figure I.14: Electrode Movements noise 14
Figure I.15: Drift of the Baseline 15
Figure I.16: Myoelectric Noise or Somatic Tremor 16
Figure I.17: Ventricular extrasystole (VES) 17
Figure I.18: Ventricular tachycardia 17
Figure I.19: Atrial Fibrillation 18
Figure II.1:Artificial Intelligence 22
Figure II.2:Machine Learning 24
Figure II.3:Deep Learning 24
Figure II.4: A neuron with its dendritic arborization 26
Figure II.5: Mapping biological neuron / artificial neuron 27
Figure II.6:threshold Functions 28
Figure II.7:Linear Functions 28
Figure II.8:SoftMax Functions 28
Figure II.9: Sigmoid Functions 29
Figure II.10:Tanh Functions 29
Figure II.11:ReLu Function 30
Figure II.12: Leaky ReLu Function 30
Figure II.13:ELU Function 31
Figure II.14:Gaussian Function 31
Figure II.15:Structure and behavior of perceptron 32
Figure II.16:Multilayer Perceptron 32
Figure II.17:CNN Architecture 33
Figure II.18:Convolution layer slides the filter over a given input. Output is the summation of an element by the element matrix multiplication of the filter and receptive field 34
Figure II.19:pooling layers 34
Figure II.20:Fully Connected layers 35
Figure II.21:Remeber Gate 35
Figure II.22:LSTM Architecture 36
Figure II.23: internet of thing 37
Figure II.24:IOT Architecture 37
Figure II.25:IOT Components 38
Figure II.26:Device 39
Figure II.27:connectivity 39
Figure III.1:: API Functionality 43
Figure III.2: MQTT Architecture 45
Figure III.3:My SQL logo 46
Figure III.4:python logo 47
Figure III.5:Node-red logo 48

Figure III.6: Arduino IDE logo 49
Figure III.7:AD8232 ECG Sensor 50
Figure IV.1:Data Processing Flowchart 56
Figure IV.2:result MLP 57
Figure IV.3:accuracy graph MLP 58
Figure IV.4:MLP confusion matrix ..... .58
Figure IV.5:CNN result 59
Figure IV.6:accuracy graph of CNN 60
Figure IV.7:CNN Confusion Matrix 60
Figure IV.8: (CNN _LSTM) result 61
Figure IV.9:(CNN-LSTM) accuracy graph 62
Figure IV.10: cnn lstm acquisition matrix 62
Figure IV.11:experment test map 64
Figure IV.12:the acquisition node 64
Figure IV.13:The electrodes placement 65
Figure IV.14:ECG sign 64
Figure IV.15:sensore sending data to database 66
Figure IV.16:model waiting for data 66
Figure IV.17:API get the data and update the state after the prediction 66
Figure IV.18:loaded model predict state successfully 66
Figure IV.19:node red flow 68
Figure IV.20:node red interface 68

List of Tables

Tableau IV.1:Description of the collected data..... 54

List of Abbreviations

3G: stands for third generation

AEs :auto-encoders

AI : Artificial Intelligence.

ANN : Artificial Neural Network.

API : Application Programming Interface.

AV : Auriculo Ventriculaire.

aVR, aVL, aVF : Dérivations unipolaires.

C++ : calcium

CNN : Convolutional Neural Network.

Cuda : Compute Unified Device Architecture.

DI, DII, DIII : Dérivations Standard (d'EINTHOVEN) de l'ECG.

DL: Deep Learning.

DNNs:deep neural networks

ECG : Electrocardiogramme.

ELU : Exponential Linear Units

ETSI: European Telecommunications Standards Institute

GRUs:gated recurrent units

GANs:Generative Adversarial Networks

Gnd :Ground

Tanh : Hyperbolic Tangent Function

Ibm : International Business Machines

IOT : Internet of things.

JSON :JavaScript Object Notation

Jwt :JSON web token

K+ : Potassium

LAN: stands for Local Area Network

Lstm : long short-term memory

ML : Machine Learning.

MLP :multilayer perceptron

MQTT: Message Queue Telemetry Transport

Na+ : Sodium

NFC :Near Field Communication

Relu :Rectified linear unit

RBMs:restricted Boltzmann machines (RBMs)

RFID : Radio-Frequency Identification

RNN: Recurrent Neural Network

SQL : Structured Query Language

Tcp : Transmission Control Protocol

Tls :Transport Layer Security

Ves : Ventricular extrasystole

Vt: Ventricular tachycardia

Wan : wide-area networks

Wi-Fi :refers to wireless networ

RFID : Radio-Frequency Identification

RNN: Recurrent Neural Network

SQL : Structured Query Language

Tcp : Transmission Control Protocol

Tls :Transport Layer Security

Ves : Ventricular extrasystole

Vt: Ventricular tachycardia

Wan : wide-area networks

Wi-Fi :refers to wireless networ

General Introduction

General Introduction

Cardiovascular diseases (CVDs) pose a significant global health challenge, with the World Health Organization (WHO) identifying them as the leading cause of death, claiming over 17.7 million lives in 2017 [1]. Accurate and prompt diagnosis of these conditions remains crucial within cardiology departments in hospitals. The Electrocardiogram (ECG) stands out as a fundamental tool for diagnosing and analyzing cardiac arrhythmias.

Traditional ECG diagnosis involves a non-invasive procedure where doctors assess heart function by placing electrodes on the skin surface to capture electrical signals reflecting the heart's activity over time [2]. However, challenges can arise when general practitioners without specialized training interpret ECG results. To address this, a cutting-edge smart system has been developed to diagnose cardiovascular diseases through advanced analysis of ECG signals.

The initial chapter delves into the intricate details of human heart anatomy, cardiac cycles, electrical activities, ECG signal components, and methodologies for acquiring and processing ECG data using various leads. Additionally, it explores different cardiovascular diseases and their underlying causes.

Moving forward, the second chapter delves into artificial intelligence (AI) and its subsets such as machine learning (ML) and deep learning (DL). The focus is on artificial neural networks (ANN) and convolutional neural networks (CNN), highlighting their architectural complexities and parameters in CNN 1D models. The chapter also touches on the Internet of Things (IoT) and its diverse applications.

Chapter three shifts the focus to the essential technologies and tools required for software development, IoT applications, data analysis, and machine learning. It discusses APIs, Fast API, MQTT, MySQL databases, Python programming language, Node-RED platform, Arduino IDE, and AD8232 ECG sensors, emphasizing their significance in developing intelligent systems.

The final chapter revolves around the pre-processing of ECG signals, database creation, and the development of a CNN model capable of automatically detecting and classifying cardiovascular diseases with high precision. The effectiveness of the model is tested using data collected from AD8232 ECG sensors. Real-time testing and validation are conducted by integrating the intelligent model into the Node-RED platform, ensuring seamless and accurate disease diagnosis in a timely manner.

Chapter I: Overview of the Heart

III.1 Introduction

Chapter I provides an in-depth investigation of the heart, a crucial organ essential for blood circulation in the body. The chapter examines the heart's structure, mechanical and electrical operations, and the complex mechanisms of the cardiac cycle. Additionally, it highlights the importance of electrocardiograms (ECGs) in tracking and identifying different cardiac conditions, offering understanding into the heart's electrical functions and the analysis of ECG readings.

III.2 The heart

The heart, a pump-like hollow muscular organ performs the crucial purpose of pumping blood through veins and arteries across the body. Its structure resembles an inverted cone with its apex pointing down to the left hand side while the base points upwards to the right. Located in the mediastinum—an area at the center of the chest that lies between lungs, sternum and spine—the heart is positioned slightly to the left of thoracic midline. It has a mean weight of about 300g among men and 250g in women, which is roughly equal to one and half times of clenched fist size. During rest it forces out approximately 5-6 liters per minute amounting to total volume of almost 220 million liters throughout life expectancy period of an average person (about 70 years).[3]

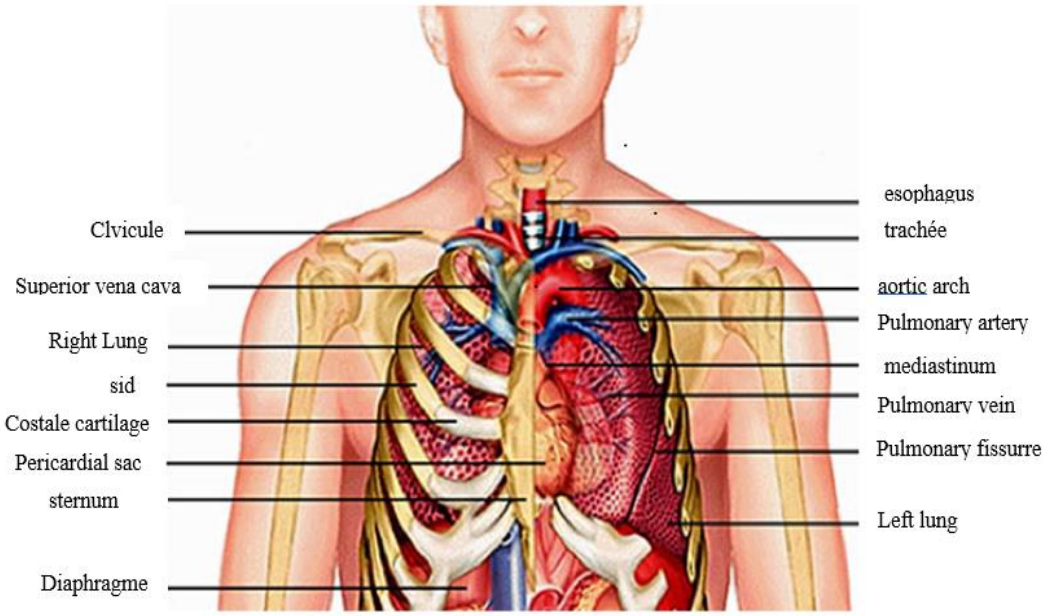


Figure I.1: The Thoracic Cage

III.2.1 Anatomy of the Heart

The contractions of its muscles make sure that blood is constantly flowing through the heart. It is a hollow, striated muscle separated into right and left halves by a thick septum. The

right side carries deoxygenated blood for pulmonary circulation, while the left side conveys oxygenated blood throughout the entire body. Each half consists of an atrium and a ventricle linked by inlet valves which control the flow of blood. Outlet valves allow communication between the ventricles and the pulmonary artery (pulmonary valve) as well as the aorta (aortic valve). These valves are placed at entry points into the pulmonary artery and aorta. For an illustration on how vessels in relation to heart architecture looks like see Figure I.2. The myocardium or cardiac muscle walls consist of cardiomyocytes..[4]

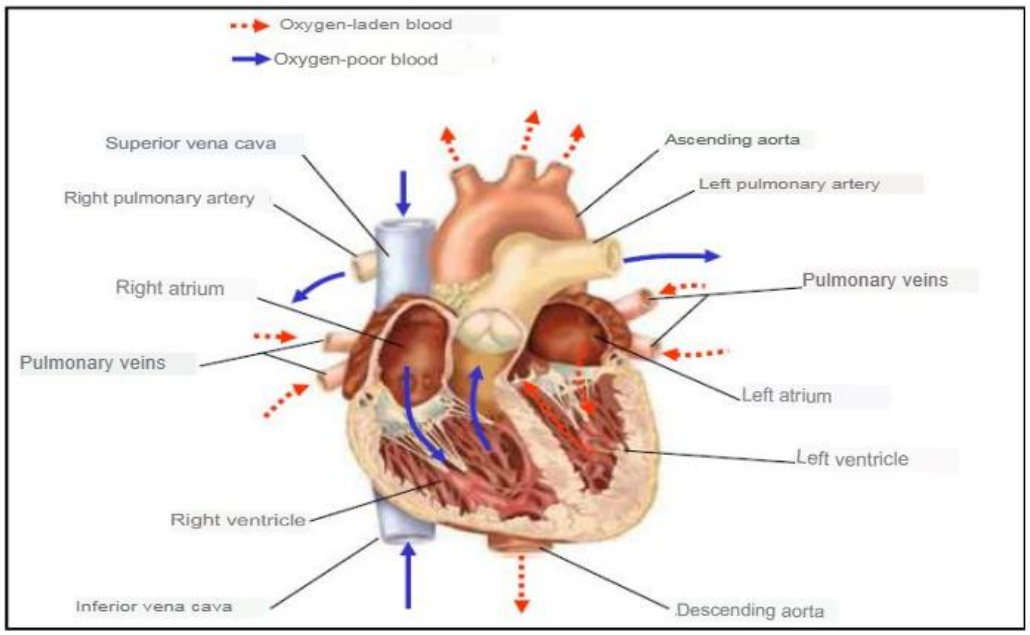


Figure I.2 The Human Heart

It is, therefore, important to note that the three layers of the heart wall are made up of a very strong muscle tissue.

- **Epicardium (epicardium):** Composed with mesothelium, a type of simple squamous epithelium that is lined by thin submesothelial layer which is separated from myocardium by connective tissue.
- **Myocardium (myocardium)**: Interstices holding dense connective tissues packed in elongated cardiac muscle fibers and where arterial, venous, and lymphatic capillaries extend between the myocardial fibers.
- **Endocardium:** Located internally lining this section is a slim subendothelial layer consisting of thin endothelium. It comprises elastic-connective layer as well as subendocardial loose connective tissue with nodal tissue elements.[5]

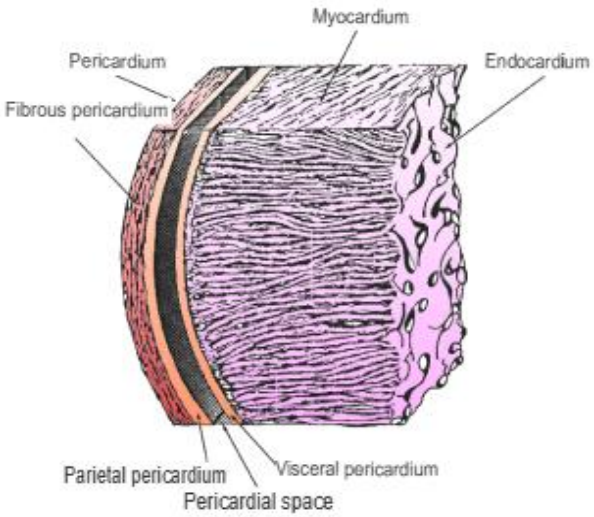


Figure I.3: The heart envelope

III.3 Study of a Cardiac Cycle

The cardiac cycle refers to the sequence of mechanical and electrical events that follow every heartbeat.

III.3.1 The mechanical activity of the heart

An electrical impulse originating from the left atria's sinoatrial node initiates its mechanical function leading it to spread across right atria after which it reaches ventricles. At this point both atria contract together followed by simultaneous contraction of both ventricles which becomes an important part of this process. During diastole relaxation occurs while systole represents myocardiym contraction phase of this cycle. However, it should be noted that there are no simultaneous activities between the two chambers [6]

III.3.2 The electrical activity of the heart

Responsible for coordinating each heartbeat, the heart has an in-built electrical conduction system. The electrical signal that directs the contraction of cardiac muscles comes from the sinoatrial node or Keith and Flack node (located close to the superior vena cava in the right atrium) as can be observed in Figure I.3. In other words, this is a group of self-excitable cells that depolarizes approximately 60-100 per minute and it acts as a "pacemaker" for the heart beat.

This impulse propagated through both atria causes their systole while it also reaches Achoff-Tawara node or atrioventricular node at interventricular septum (see Figure I.4). This takes form of a transient lull during electric activity before going on to bundle of His and finally Purkinje network that starts ventricular systole. It is important because delayed

stimulation relative to atria allows ventricles enough time to fill up properly for atrial contraction.

This electrical system ensures rhythmical regularity of the heart and also coordinates contractions between atria and ventricles.. [7]

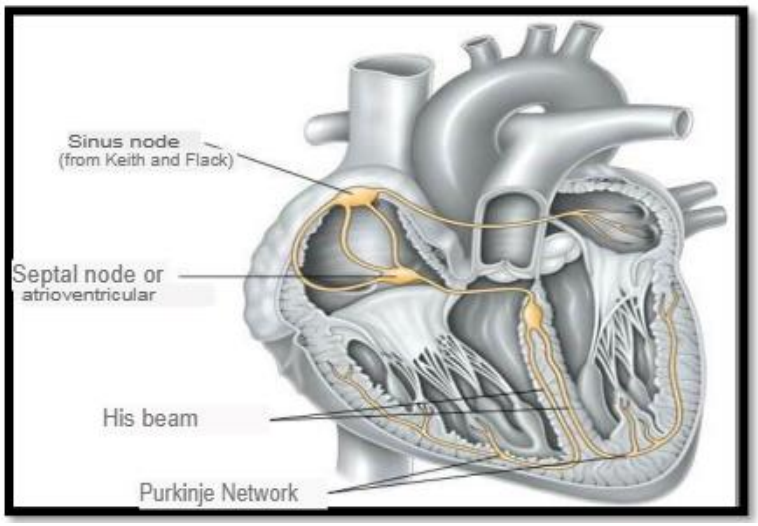


Figure I.4: Electrical Operation

III.4 Action potential

Cardiac cells contract through the exchange of ions which results in the formation of an action potential (Figure 5) of having five distinct phases. These are as follows:

- Phase 0 (rapid depolarization): This phase occurs when an electrical stimulus exceeding cell activation threshold forces Na+ ions to enter into the cell rapidly leading to sudden reversal of the polarity.
- Phase 1 (early repolarization): Deactivation of Na+ channels during this stage allows K+ ions to exit the cell, resulting in a brief and rapid repolarization.
- Phase 2 (plateau): In this part, Ca++ ions slowly diffuse into the cell such that they gradually balance outflowing K+ ions thereby delaying repolarization.
- Phase 3 (repolarization):Ion channels open thus restoring original resting potentials. During this time K+ions continue flowing out but it returns back to its resting state.
- Phase 4 (resting potential): It is a period where cells are at their baseline level and hence more excitable..[8]

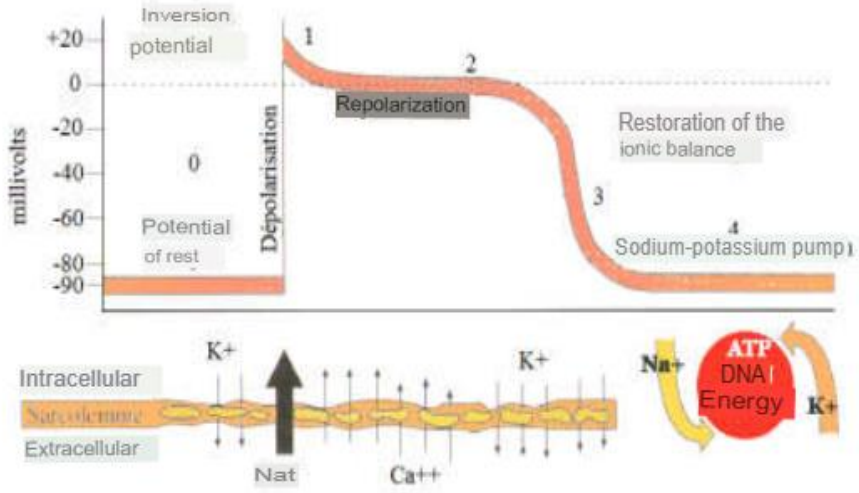


Figure I.5:The five phases of action potential and ion transfers at the cellular level

III.5 Electrocardiogram (ECG)

The electrocardiogram is a diagnostic test which records the electrical activity of the heart with each beat. The heart like any other muscle, contracts when there are electrical impulses (depolarizations) followed by reprorization, which can be observed and recorded. Electrocardiograph is used to describe the real process of the test and not the graphical representation of the obtained electrical activity as an electrocardiogram (ECG). This examination entails attaching electrodes to patients' skin surface in supine position. These electrodes collect electrical signals expressed as deflections across at least twelve leads; six in frontal plane (limb electrodes) and another six in horizontal plane (precordial electrodes). [9]

III.5.1 Electrical Characteristic of Cardiac Tracing

Distinct waves, that is, P, QRS and T ones completely correspond to certain functional features. For temporal positioning of these waves indicates that intervals are different from the normal electrocardiogram signal to those with cardiac problems. R-R interval, Q-T interval as well as P-Q interval along with influential components such as S-T segment and QRS complex exhibit variant patterns in various cardiovascular diseases thus making them useful in diagnosis. Also there are various pathological conditions which have diverse frequency characterizations of these waves. [10]

The electrocardiogram (ECG) signal can be presented as follows:

on

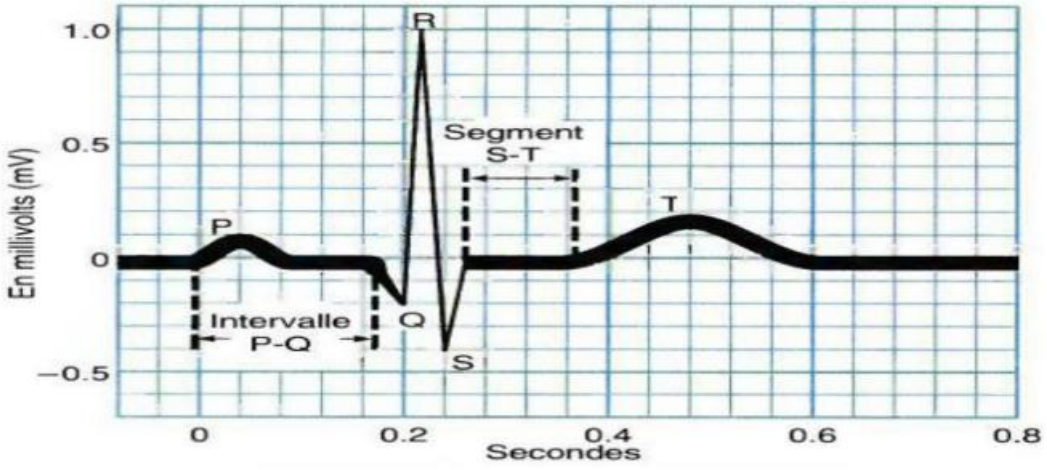


Figure I.6 :ECG Signal Form

each heartbeat, we can identify:

- **The isoelectric line** (Figure 1.7): The initial point of reference on an electrocardiogram trace is the isoelectric line, which signifies the baseline indicating no electrical activity. Signals above this line are termed positive waves, while those below are labeled as negative waves. A wave is considered biphasic if it contains segments both above and below the isoelectric line. The measurement of all waves begins from the start of their initial phase and extends to the isoelectric line..[11]



Figure I.7 :The isoelectric line

- **The P wave**: It signifies the depolarization of the atria, lasting about 90 milliseconds. The P wave is a positive wave with an amplitude typically less than or equal to 0.2 millivolts. [12].
- **The QRS complex:** It corresponds to the depolarization of the ventricles before the onset of mechanical contraction, typically lasting between 85 and 95 milliseconds. [12].
- **The T wave:** It represents the repolarization of the ventricles and has an asymmetrical shape, with a slightly oblique ascending branch and a steeper descending branch. The amplitude of the T wave is typically less than 2 millimeters, with its normal amplitude being lower than that of the QRS complex. [12].

- **The PR segment**(FigureI.8)**:** Extending from the conclusion of the P wave to the commencement of the QRS complex, the PR segment signifies the duration required for the impulse to traverse through the atria, the atrioventricular node, and the His-Purkinje system. Its duration typically ranges from 0.12 to 0.20 seconds.. [12]

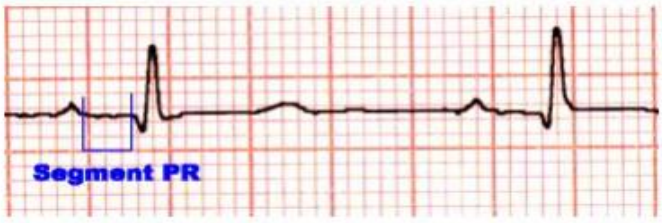


Figure I.8:PR Segment

- **The ST segment** (Figure I.9): It represents the interval of consistent ventricular excitation until the ventricular recovery phase. This segment is measured from the conclusion of the S or R wave to the onset of the T wave. Typically, it appears flat or slightly slanted +/− the isoelectric line. Any deviation of more than 1 millimeter from the isoelectric line, whether elevated or depressed, is considered abnormal. [12]



Figure I.9:ST Segment

III.5.2 Normal ECG

The normal values for the duration of waves and intervals in the ECG are as follows:

- **P wave**: less than or equal to 0.11 s;
- **Q wave**: less than 0.04 s;
- **QRS complex:** less than 0.1 s; usually between 0.06 and 0.08 s;
- **PR interval:** between 0.12 and 0.2 s; its duration varies inversely with heart rate;
- **QT interval**: varies in the same way as heart rate;
- **T wave:** average duration is 0.2 s.[13]

III.6 Standard ECG Leads

ECG waves reflecting the heart`s electric activity are recorded by means of leads put on certain parts of the body. Conventionally, ECG consists of recordings from twelve leads. Every lead has a negative and a positive electrode which determine its polarity and hence

permit measurement of potential difference within it. The line connecting these electrodes is referred to as the lead axis with orientation based on electrode placement. Half of them refer to cardiac electrical potentials in frontal plane (bipolar limb leads, unipolar limb leads) while the rest cover horizontal plane (precordial leads). To acquire a 12-lead ECG, ten skin electrodes have to be applied to the patient: six placed on the chest and four at its extremities. [14].

III.6.1 The bipolar peripheral derivations

Einthoven originated bipolar derivations that are still used today; D1, D2 and D3 were created in early 1900s according to this method. These derivations use three electrodes positioned on the subject's body : one on right arm, one left arm and one left leg forming Einthoven's triangle.

As their name suggests, these triangles are called bipolar since they measure potential difference between two electrodes; each side indicates one derivation involving a different pair of electrodes for each derivation..

The three derivations are:

- DI (derivation I) calculated as DI = VL - VR
- DII (derivation II) expressed as DII = VF - VR
- DIII (derivation III) defined by DIII = VF - VL

Here, VL signifies the potential on the left arm, VR the potential on the right arm, and VF the potential on the left leg. [15]

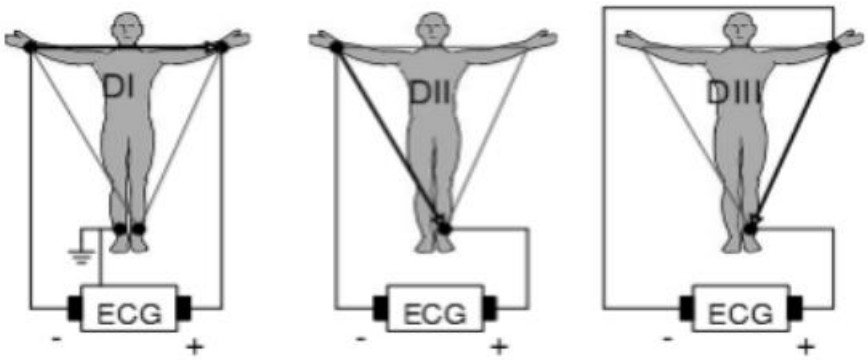


Figure I.10:Einthoven setup for recording bipolar limb leads

III.6.2 Derivations

Unlike Einthoven, Wilson was the pioneer of unipolar derivations. In his concept, these derivations are found between a test electrode positioned on the tip of Einthoven's triangle and a central terminal (neutral or indifferent electrode). This potential maintenance

brings out the unipolar leads VL, VR and VF which have a potential difference between one point and all others in ECG –in this case three points that make up the triangle. In future years goldberg developed wilson's system to create three augmented unipolar leads called aVL, aVR, and aVF.. (illustrated in Figure I.11).

The electrode designations by Goldberg are as follows [16]:

aVR = right arm.

aVL = left arm.

aVF = left leg.

The "a" prefix is an indication that voltage is amplified to produce waveforms with amplitudes equal to those of DI,DII,and DIII. [14],is shown in Figure I.11.

.

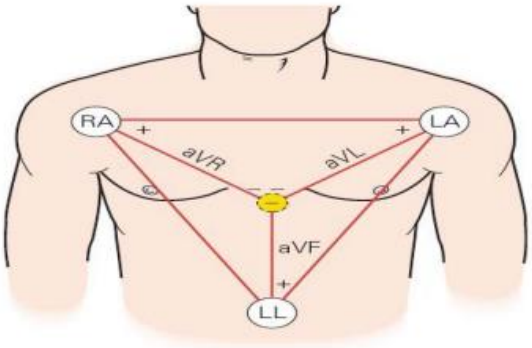


Figure I.11:Unipolar Derivation

III.6.3 Precordial (thoracic) leads

Wilson invented lateral or horizontal precordial (thoracic) leads V1, V2,V3,V4,V5,and V6 in 1944 which were situated around areas of close cardiac proximity in heart. These six leads are located on the left side of the chest and employ an exploring electrode (positive pole) placed upon the thorax and a reference electrode (negative pole) connected to Wilson's central terminal. These are referred to as near wall leads since one end is placed very close to each ventricle wall while at right angles there-to. [17].

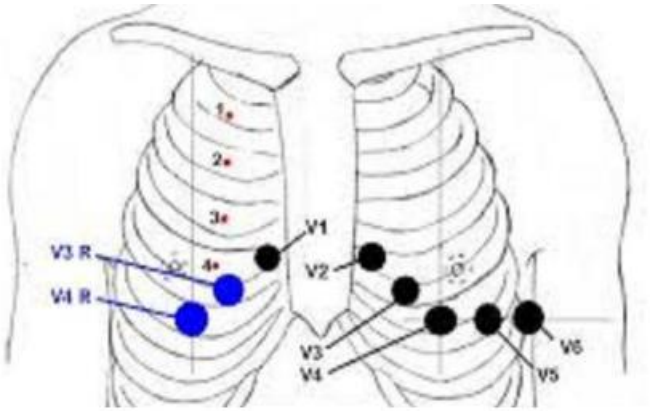


Figure I.1:Positions of precordial electrodes

These leads are positioned as follows[17]:

- V1: 4th intercostal space, right border of the sternum (parasternal line)
- V2: 4th intercostal space, left border of the sternum (parasternal line)
- V3: halfway between V2 and V4
- V4: 5th intercostal space, left midclavicular line
- V5: halfway between V4 and V6, on the anterior axillary line
- V6: same horizontal level as V4 and V5, midaxillary line.

III.7 Types of ECG Signal Noise

The trace might be masked by these occurrences and may lead to diagnosing inaccuracies. Trained eyes of the physician usually detect such disturbances before making a diagnosis. Several factors including technical, physical or pathological can cause these iatrogenesis. Our main focus will be on exploring the technical and physical attributes of noises and artifacts found in electrocardiogram tracings especially in Holter monitoring. In our attempt to view common electrocardiographic noises as additive, we shall determine the amplitude, periodicity and spectral range of these noises as our attention goes towards them. Artifacts are significant since they affect the functionality of algorithms employed for automated ECG analysis.[18]

III.7.1 Technical Noise

It is important to handle the recording equipment with care, as it has the potential to introduce unwanted noise during the recording process. The following are the most common types:

III.7.1.1 Line Noise

Electrocardiographic signal can sometimes be disrupted with a 50Hz harmonic

waveform by an electrical distribution network (as shown in Figure 1.13). This type of noise is pervasive throughout the recording and can be quite pronounced, but it can be effectively removed using a selective filter due to its narrow band high frequency nature.[15]

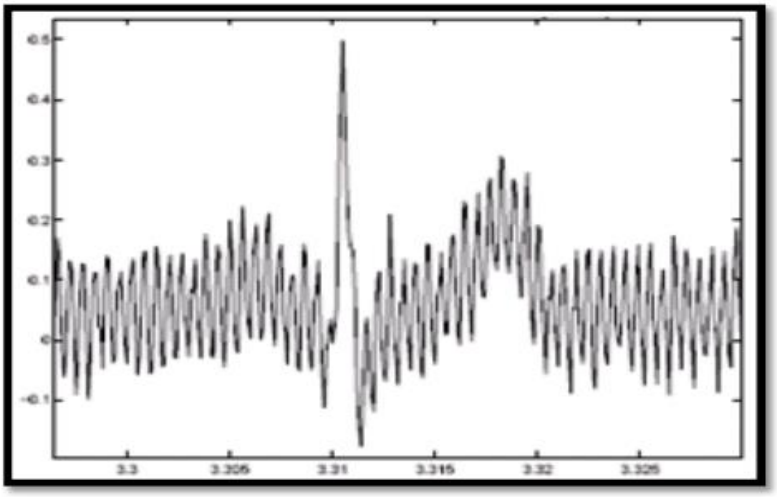


Figure I.13:50 Hz Line Interference

III.7.1.2 Electrode-Related Noise

When electrodes are not properly connected the baseline of the waveform can shift suddenly. This may lead to decreased amplitude or spikes appearing upon intermittent contact of the electrodes with skin. Such spikes, as presented in Figure 1.13, might be erroneously considered as regular waveform elements. Getting rid of this kind of intermittent noise which has a broad spectral band is difficult since its energy overlaps with that of QRS complex frequency range [19]

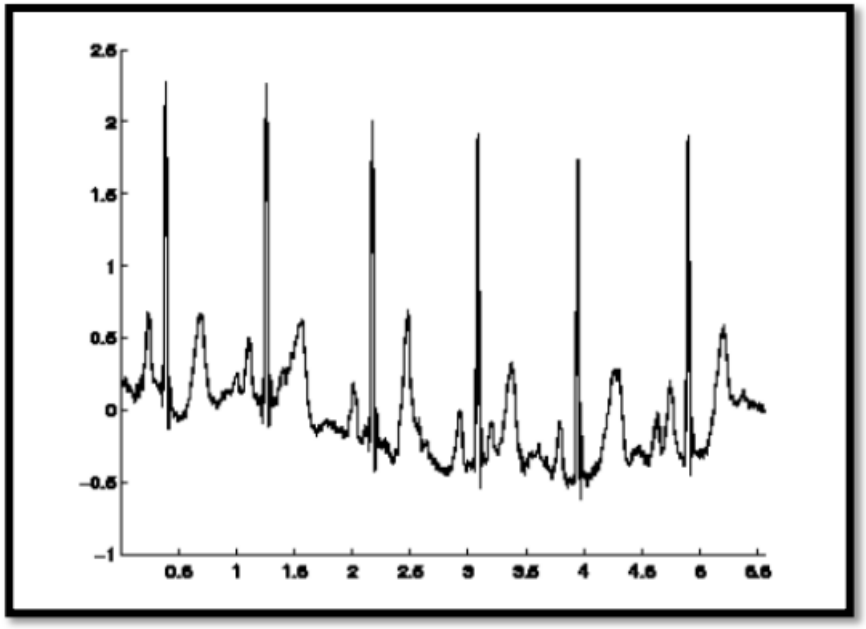


Figure I.14: Electrode Movements noise

III.7.1.3 Additional Common Sources of Interference

Among them are artifacts due to cable movement, instrument saturation, poor wiring and artifacts caused by synthetic clothing..[19]

III.7.2 Physical Interferences

Physical interferences arise from the body's electrical activities, such as muscle contractions or breathing.[20]

III.7.2.1 Fluctuations in Baseline

During electrocardiography recording respiration results in rhythmic fluctuations in ECG baseline as shown in Figure (1.15). The other type is when there is no any contact between skin and electrodes which may affect the normal line momentarily. These disturbances are usually unimportant for ECG diagnosis and can be mostly removed since they have low frequency magnitude that minimally intersects with normal ECG signals.[19]

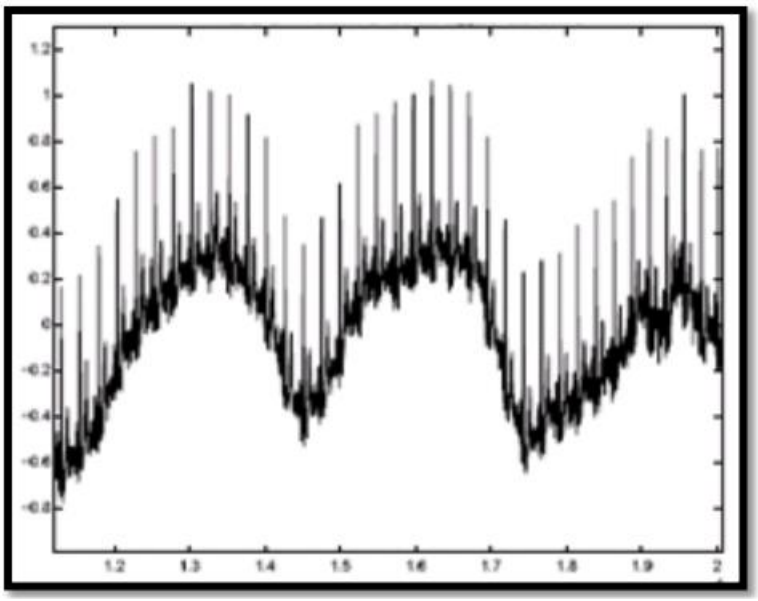


Figure I.15: Drift of the Baseline

III.7.2.2 Myoelectric Interference or Somatic Tremor

The depolarization of muscles cells initiates muscle contractions and ECG is more sensitive to heart frequencies though it can also sense skeletal muscle contractions as shown in figure 1.16. The commonest present symptom is a high frequency oscillation associated with increased muscle tone in a non-relaxing patient. If the patient moves considerably or shivers, these disruptions can be quite troublesome perhaps concealing the P and T waves and

hampering precise diagnosis; this interference depends on the state of the person under examination, for example highly stressed subjects or those suffering from Parkinson's disease may experience low quality recording due to all ECG leads.[19]

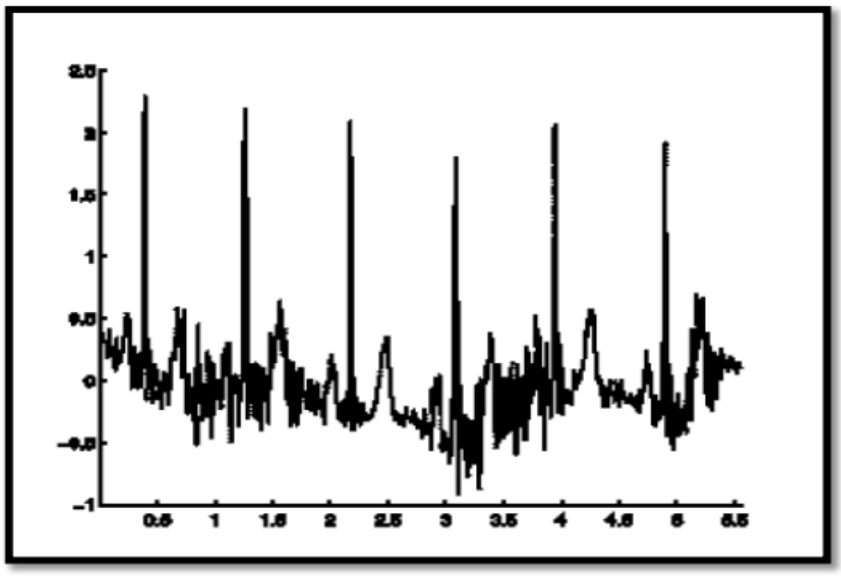


Figure I.16: Myoelectric Noise or Somatic Tremor

III.8 Cardiac diseases

III.8.1 Ventricular extrasystole (VES)

Almost all recordings will show ventricular extrasystole (VES), which is a common abnormal heartbeat, especially in the recovery period after exercise. Their presence may not necessarily represent any particular pathology, but if their frequency exceeds 6 per minute on a consistent basis VES can be a precursor to ventricular tachycardia which is a serious pathology. Unlike normal heartbeats that originate from the depolarization of sinus cells, VES comes about from the spontaneous depolarization of small group of ventricular cells called ventricular ectopic focus. The electrical impulse generated does not follow the normal conduction pathway (His bundle) thus it propagates more slowly through the ventricles. This prolonged ventricular contraction becomes less efficient. The waveform of VES beat has two characteristics: no P wave precedes R wave as there was no previous atrial activity and duration of complex is longer than that for a normal QRS complex. (Figure I.17) [21].

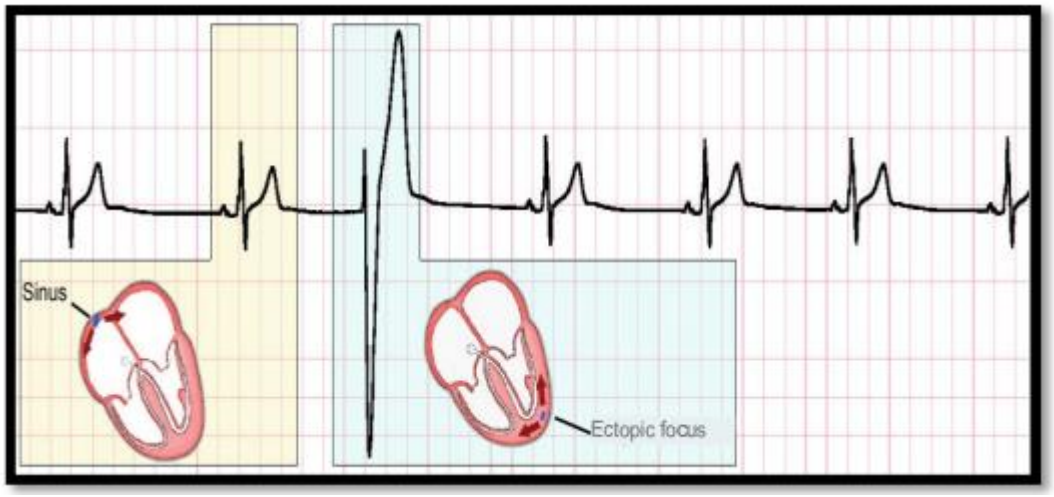


Figure I.17: Ventricular extrasystole (VES)

III.8.2 Ventricular tachycardia

Ventricular tachycardia is said to occur when there is tachycardia below His bundle bifurcation. In short, impulses move completely through ventricular myocardium after they have passed through His bundle down stream only. Ventricular tachycardia (VT) symbolizes a cardiac emergency par excellence. This criticality arises because it usually has poor hemodynamic tolerance often ending up in ventricular fibrillation and death without warning (Figure I .18). [22]

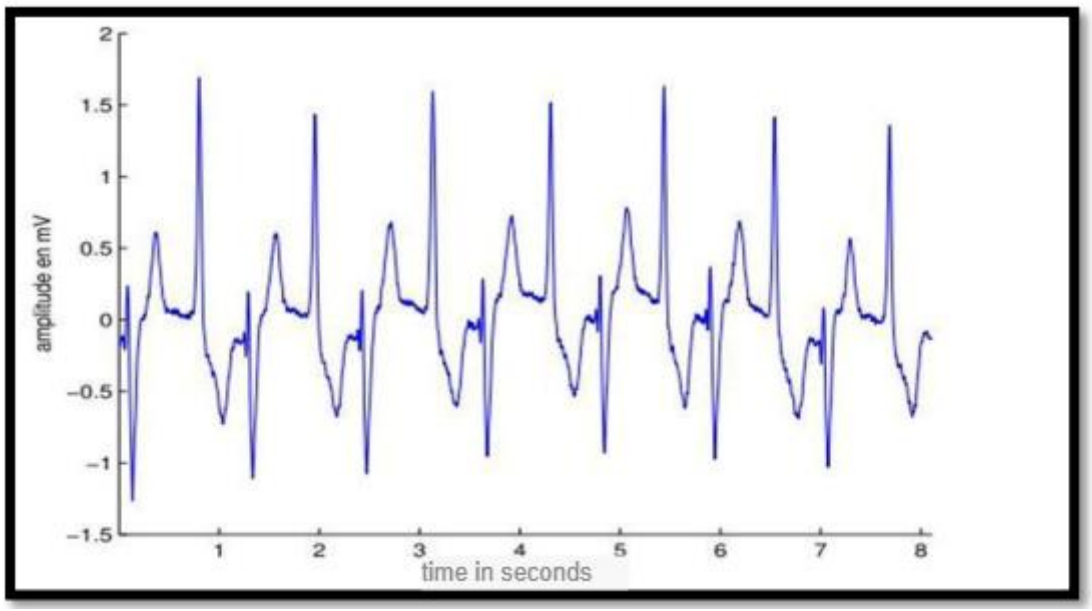


Figure I.18: Ventricular tachycardia

III.8.3 Atrial fibrillation

Atrial fibrillation is a supraventricular arrhythmia without organization.

Depolarization gets fragmented into multiple fronts of different directions and amplitudes,

leading to completely disordered electrical activity in the atria. This frequently sustained activity does not allow the atrial myocardium any "electrical rest". On an ECG, it is characterized by the disappearance of organized atrial waves, replaced by continuous activity resembling an irregular sine wave. The rate of depolarization varies within the atria but is typically very rapid. Multiple depolarization front leads to the atrioventricular node, which behaves as a “filter” only allowing few fronts to pass randomly causing totally irregular ventricular frequency that usually lies between 90 and 140/min depending upon its permeability. Absence of contractions in the atria per se is not dangerous but it significantly reduces cardiac efficiency through two ways: first, the heart cannot gain from atrial systole for ventricular filling; secondly, because on average heart beats above 100 bpm and may reach 200 bpm. The major risk factor associated with this condition is that a flutter might develop implying that emboli formed in the atria can be thrown into circulation (particularly harmful if it originates from the left atrium, since the left ventricle supplies blood to the systemic circulation, including the heart and brain). [15]

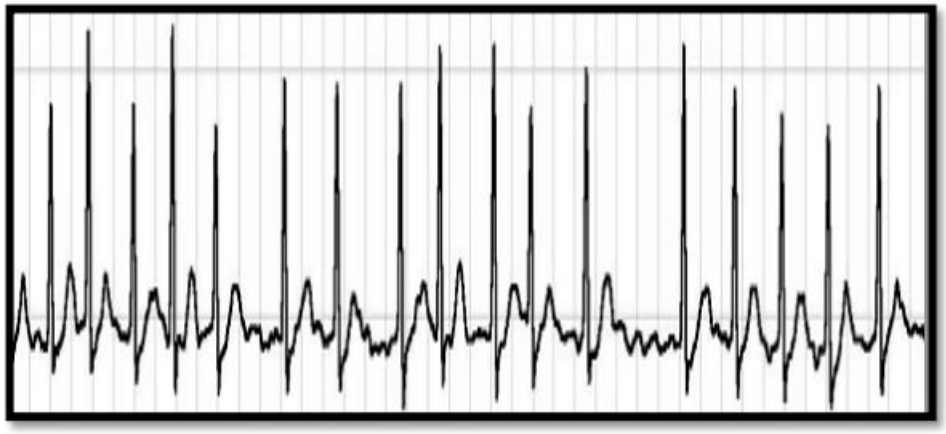


Figure I.19: Atrial Fibrillation

III.8.4 Ventricular Tachycardia

It is a rapid ventricular rhythm that may be regular or irregular (more than one hundred beats per minute). This form of arrhythmia represents only a small fraction of all arrhythmias and typically indicates poor ventricular function. A person with ventricular tachycardia will feel unwell, look pale, sweat heavily, and have palpitations. If the heart rate gets too fast, they can faint and even become dizzy..[4]

III.9 Conclusion:

In summary, this chapter offers a detailed examination of the heart, encompassing its structure, function, and the crucial importance of electrocardiograms (ECGs) in evaluating cardiac health. By delving into the intricate mechanisms of the heart, both mechanical and electrical, readers develop a deep comprehension of how the heart supports life and regulates circulatory efficiency. Emphasizing the critical role of ECGs in detecting cardiac issues, monitoring heart performance, and pinpointing irregularities in heart rhythm and conduction, the chapter acts as a foundational resource for unraveling the complexities of the heart and emphasizing its indispensable role in sustaining life and promoting cardiovascular well-being.

Chapter II: AI & IOT

IV.1 Introduction

The chapter titled "Artificial Intelligence and Internet of Things" delves into the realm of artificial intelligence and its various subfields, including Machine Learning and Deep Learning, and provides a comprehensive overview of the different types of Deep Learning. Additionally, it aims to elucidate the broader scope of neural networks, encompassing artificial neural networks and convolutional neural networks (CNN), while offering detailed insights into the parameters of each layer within CNN architectures. Lastly, the chapter aims to deliver a comprehensive description of the Internet of Things.

IV.2 Overview of artificial intelligence

IV.2.1 Intelligence Artificielle

John McCarthy, known as the father of AI, articulated the concept of artificial intelligence by defining it as "the science and engineering of making intelligent machines, especially intelligent computer programs." AI functions in a manner similar to human intelligence, encompassing learning, adapting, identifying, and correcting. McCarthy, the term's originator in 1956, described AI as "the science and engineering of creating intelligent machines." AI refers to a computerized system, such as a robotic system, designed to process information to produce results akin to those of an organization's workforce, utilizing its ability to learn, make decisions, and solve problems. Essentially, AI is the science of imitating human intelligence through computer technology. According to the Encyclopedia Britannica, artificial intelligence (AI) pertains to the capacity of a digital computer or computer-controlled robot to perform tasks typically associated with intelligent beings.[23]

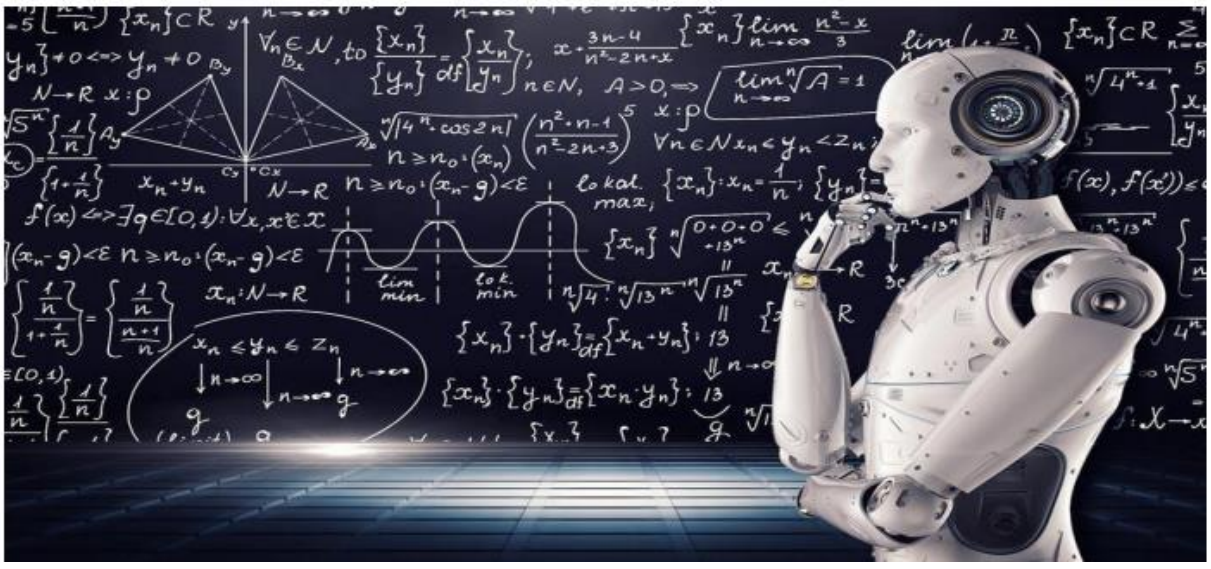


Figure IV.1:Artificial Intelligence

IV.3 Machine Learning

Machine learning is a subfield of artificial intelligence that enables a system to learn by itself from experience and improve without requiring explicit programming. It aims at creating computer programs which can access data and learn on their own. Basically, Machine Learning is the use of data and algorithms to generate artificial knowledge according to sets of predefined analytical rules for recognizing patterns in collected data. In other words, it is the use of quantitative and qualitative data in order to answer questions precisely and accurately. This will include; discovering, extracting and synthesizing relevant data using computers or computer-based systems; making independent predictions based on analyzed information; adapting automatically to certain changes; calculating probabilities for specified outcomes; as well as optimizing processes according to recognized patterns.[24]

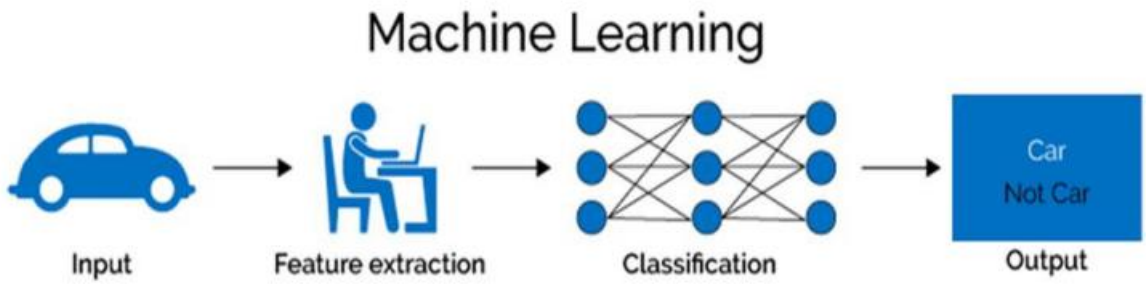


Figure IV.2:Machine Learning

IV.3.1 Types of Machine Learning

IV.3.1.1 Supervised Learning

The models in this type of machine learning are pre-defined, the systems learn from input-output pairs i.e. data. It is supervised learning if through labeled data examples and correct output the dots can be connected using adequate knowledge on data..[24]

IV.3.1.2 Unsupervised Learning

In simple words, AI learns without any supervision that means it has to find patterns in inputs that do not have targeted values. The unsupervised training is all about learning from mistakes by doing it over and over again. In contrast to supervised learning we don't have labeled data here; hence we don't show the answer to a machine. By looking at different algorithms that allow machines to join the dots our aim is for it to study and observe the data. As a result unsupervised training is likely to enable machines in subjecting inputs into classification or pattern recognition functions that humans could never have seen. [24]

IV.4 Deep Learning

Deep Learning is a branch of machine learning whose application involves one specific mode of learning it uses where each next level depends on what was gained previously. This hierarchical approach is based on how the brain processes information and learns as it responds to external factors. Each stage of education, according to theory, corresponds with a different part of the cerebral cortex.[25]

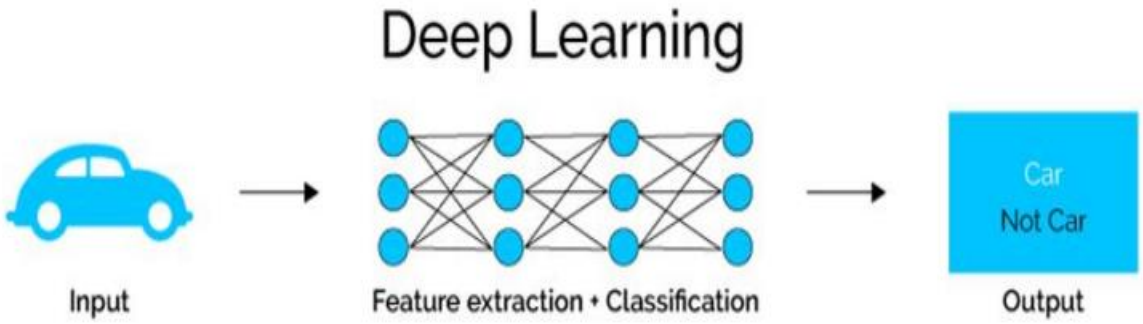


Figure IV.3:Deep Learning

And it is used in various fields:

- Image recognition.
- Automatic translation.
- Medical diagnosis.
- Personalized recommendations.
- Automatic social media moderation.
- Financial prediction and automated trading.
- Malware or fraud detection.
- Space exploration.
- Intelligent robots.

There are two main types of deep learning.

IV.4.1 Supervised learning

A machine learning model involves inferences of its algorithm from a labeled set of training data. This can be real-world examples such as hospital patient data or stock exchange values. In this case, we have a set of inputs and corresponding outputs (xt, yt). For example, an intelligent agent guesses at the input like xt and loses value. Then the agent enhances estimation for the desired outputs by iteratively updating network parameters. The successful

training gives an ability to provide correct answers to questions about the environment to an agent. Deep learning encompasses variety of supervised learning techniques including recurrent neural networks (RNNs), convolutional neural networks (CNNs), and deep neural networks (DNNs). Within the RNN class are gated recurrent units (GRUs) and long short-term memory (LSTM) approaches. The good thing about this technique is that it will either learn from existing data or create output on the basis of previous knowledge. However, it may be too rigid if decision boundary is derived with no samples that should belong to a specific class in the training set. To sum up, this method has relative simplicity but high performance in terms of learning among all others.[26]

IV.4.2 Unsupervised learning

Unsupervised learning, however, can proceed without the need for labeled data. Learning may proceed in this fashion and does not require labels. In this context, the agent acquires essential features or internal representations to unveil hidden structures or relationships within the input data. Various members of deep learning family have excelled in nonlinear dimensionality reduction and clustering tasks such as restricted Boltzmann machines (RBMs), auto-encoders (AEs), and more recently Generative Adversarial Networks (GANs). Moreover, RNNs with GRUs and LSTM approaches have found unsupervised learning applications in diverse fields. While there are advantages to using unsupervised learning it is limited in terms of providing accurate information regarding data categorization as well as involving computational complexity inherent to its algorithms Unsupervised learning has limitations when it comes to data categorization precision and underlying computational complexity involved. Finally, clustering remains one of most popular unsupervised learning methods.[26]

IV.5 Neural Networks

IV.5.1 Biological Neuron

The neuron is a cell composed of a soma and nucleus. Soma branches out to form dendrites (dendritic branches or dendritic arborization) that are sometimes so numerous they are called. The information carried from outside to the soma, the neuron’s cell body, is through its dendrites. Consequently, this processed information travels along the axon (singular) for transmission to other neurons. But all this happens only indirectly [27]

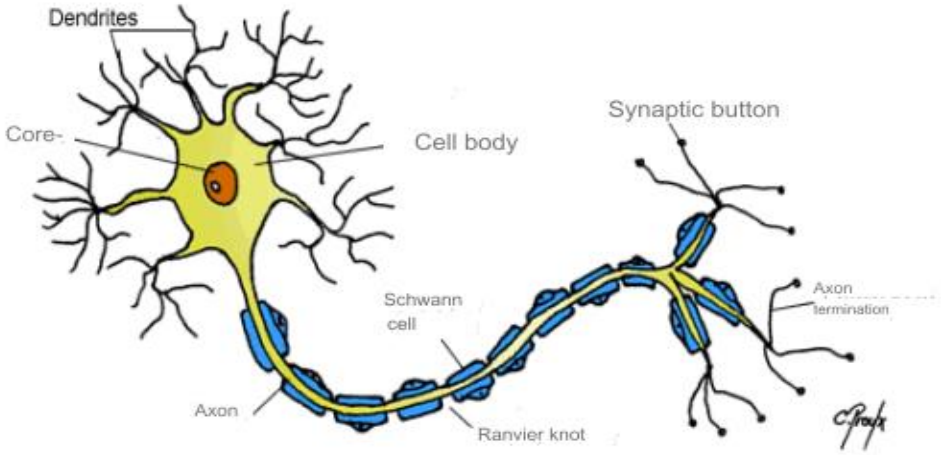


Figure IV.4: A neuron with its dendritic arborization

Indeed, there is an intercellular space of several tens angstroms (10e-9m) between an afferent neuron's axon and efferent neuron's dendrite..[27]

IV.5.2 The artificial neuron

ANNs are composed of interconnected elementary processors which work in parallel. All these units compute an output each from the info given to them.[28]

Every artificial neuron is a primitive processor that receives input from preceding neurons. These inputs have specific weights denoted by w, representing connection strength. A single output is available for each and every elementary processor which is then split to different numbers of downstream neurons. Each connection has a weight, and so on. So far all the connections are associated with some kind of weightage. In this regard, an artificial neural network's fundamental building block is called a formal neuron because it resembles a biological neuron's automaton model.[28]

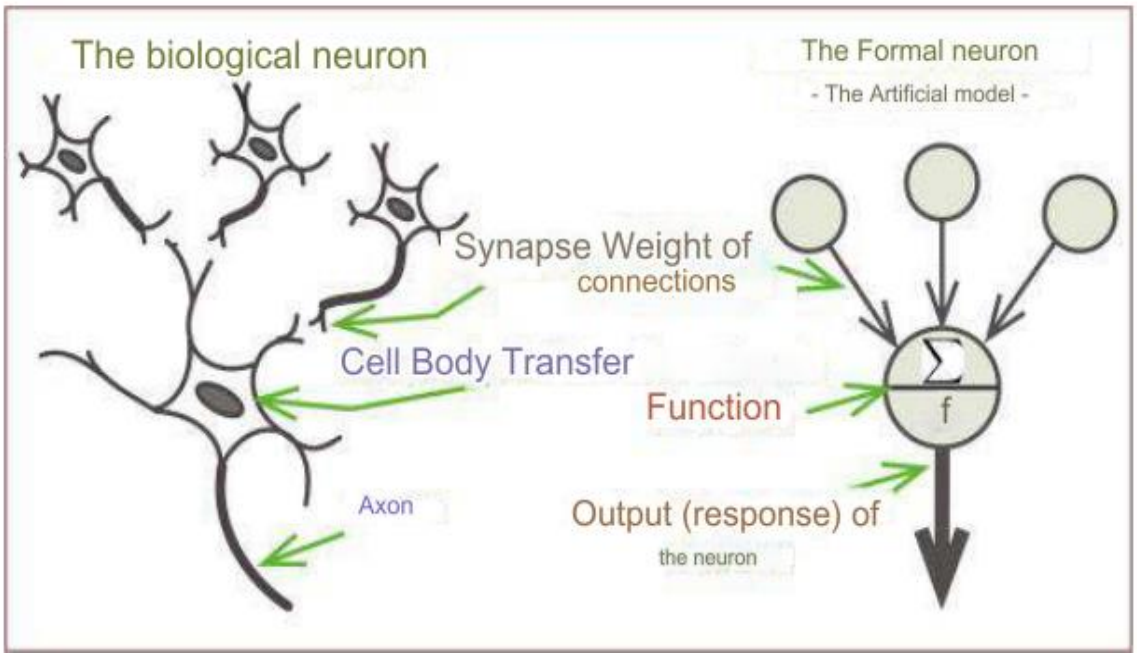


Figure IV.5: Mapping biological neuron / artificial neuron

It can be described by the following elements (for a neuron with index i):

- Its activation (also known as state) ai which may be either real or boolean generally corresponds to its output value;, Its input connections having associated weights w ij (where j represents the index of a connected neuron).
- The input function that makes preprocessing (usually weighted sum) on its inputs;
- Its activation function (or transfer function) Fi(x), which calculates the neuron's activation based on the result of the input function. The following figures provide some examples of the activation functions used.

IV.5.3 The Activation Function

Recurrent Neural Networks (RNNs) use activation functions just like other neural networks to introduce non-linearity to their models. Here are some common activation functions used in RNNs:

IV.5.3.1 Threshold function

This is a very simple function with only one parameter: the threshold. When the input does not exceed the threshold, the function returns the value 0, and it returns 1 otherwise. This function can be used for a binary classifier .[37]

$$f(x)=\begin{cases}1 & \text{if } x>0\\ 0 & \text{if other}\end{cases} \qquad \text{(II.1)}$$

This is a step function:

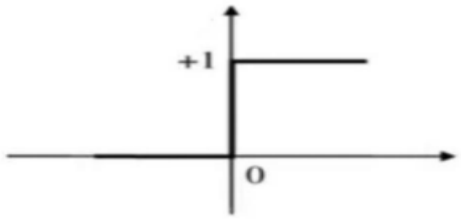


Figure IV.6:threshold Functions

$$\sum_{i=1}^{n} wi\ xi > 0 \rightarrow output = 1$$
$$\sum_{i=1}^{n} wi\ xi > 1 \rightarrow output = 0 \qquad \text{(II.2)}$$

IV.5.3.2 Linear function

It is a function that creates an output signal proportional to the input. It is more versatile than the threshold function because it can handle multiple outputs. For negative

values, it results in leakage. It allows for obtaining a non-zero gradient for the negative part.[37]

$f(x) = x$ (II.3)

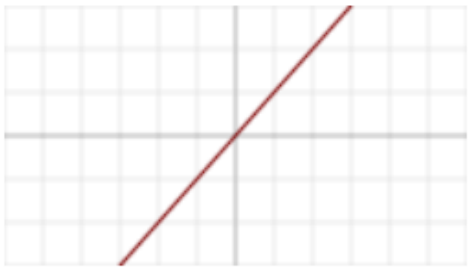


Figure IV.7:Linear Functions

IV.5.3.3 Softmax Function

The Softmax function calculates the probability distribution of an event over k different events and is very useful for output neurons. It gives the probability that the expected value is in a specific class. Softmax is generally used for networks requiring classification into multiple categories.

Typically, this function calculates the probabilities of each class relative to all possible classes. The sum of these probabilities must be equal to 1.[37]

$$f(a,x) = \begin{cases} -\dfrac{\ln(1-a(x+a))}{a} & \text{if } a < 0 \\ x & \text{if } a = 0 \\ \dfrac{e^{ax}-1}{a} + a & \text{if } a > 0 \end{cases} \quad \text{(II.4)}$$

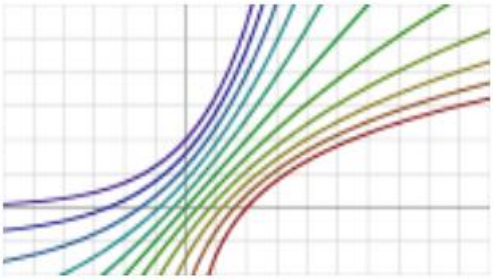


Figure IV.8:SoftMax Functions

IV.5.3.4 The Sigmoid Function

This function is the most common in the history of neural networks. It is the best choice for learning the simple step function as it is:

- Continues and differentiable everywhere.
- Symmetrical around the y-axis.

- Asymptotically approaches its saturation values.

$$f(x) = \frac{1}{1+e^{-x}} \quad \text{(II.5)}$$

$$x = \sum_{i=1}^{n} wi\ xi \quad \text{(II.6)}$$

This 'Sigmoid' function is used in the last layer (stage) of the Neuron network output for classification because it converts the input into a probability range (0,1). The 'Sigmoid' function is used for binary classification. [7]

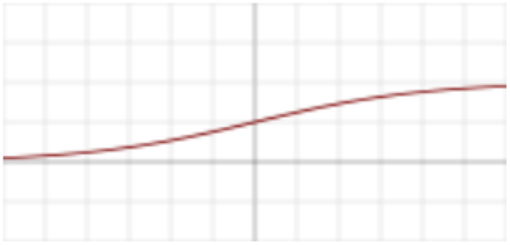


Figure IV.9: Sigmoid Functions

IV.5.3.5 The Hyperbolic Tangent Function 'Tanh'

This function has a similar shape to the 'Sigmoid' function, it has also been heavily used as an activation function until the late 1990s. The equation for this function is as follows:

$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad \text{(II.7)}$$

This function is centered on zero, but the gradients are always killed when the neurons become saturated.[37]

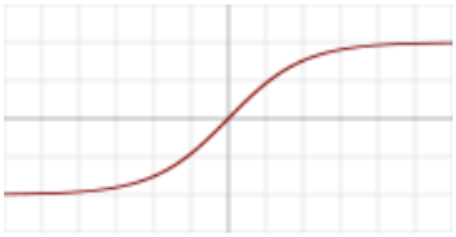


Figure IV.10:Tanh Functions

IV.5.3.6 The Relu function (Rectified linear unit)

This is a non-linear function (see Figure II-9) for general use commonly used. The equation of this function (Maas, Hannun et al. 2013). It gives x if x is greater than 0, 0 otherwise. In other words, it's the maximum between x and 0:

$$ReLu\ Function(x) = max\ (x, 0) \tag{II.8}$$

This function allows us to filter our data. It passes positive values (x > 0) to the next layers of the neural network. It is used almost everywhere but especially not in the final layer, it is used in the intermediate layers.[37]

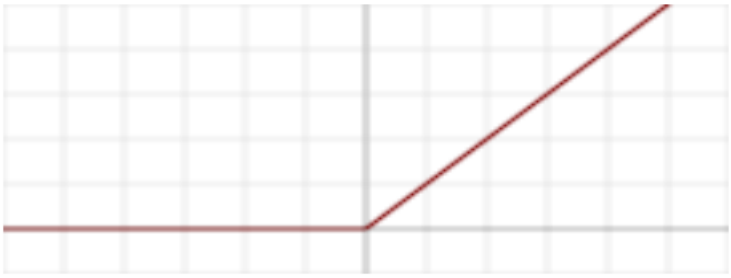


Figure IV.11:ReLu Function

IV.5.3.7 The Leaky ReLu Function

This is a variant function of ReLu, called Leaky ReLu allows a small non-zero gradient when the unit is not active.[37]

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ ax & \text{if other} \end{cases} \tag{II.9}$$

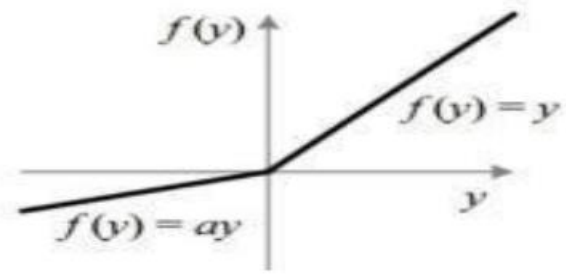


Figure IV.12: Leaky ReLu Function

IV.5.3.8 The ELU Function (Exponential Linear Units)

This function is defined according to the following equation:

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ a(e^x - 1) & \text{if other} \end{cases} \tag{II.10}$$

The value of α is constant and defined when the network architecture is instantiated.

A typical value for $\alpha = 1.0$

Thanks to the work of Clevert, the ELUs often obtain higher classification accuracy than the ReLus .[37]

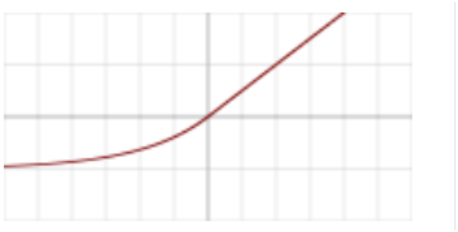


Figure IV.13:ELU Function

IV.5.3.9 The Argmax Function

It is used to obtain a discrete class from these continuous probability values, a decision function is applied. The simplest decision function for single-label multi-class classification consists of taking the largest value using the 'Argmax' function[37]

IV.5.3.10 The Gaussian Function

A Gaussian function is an exponential function of the negative square of the abscissa (a function in exp(-x2)).[37]

$$f(x) = e^{-x^2} \qquad \text{(II.11)}$$

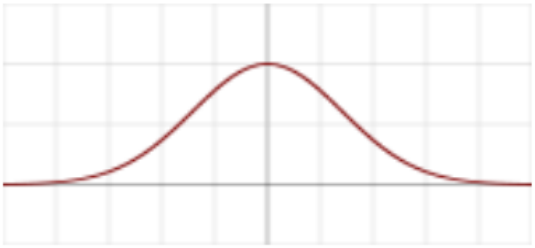


Figure IV.14:Gaussian Function

IV.5.4 Type of neural networks:

In this field, there are several neural network architectures used in experiments. The most commonly used architectures are as follows:

IV.5.4.1 Perceptron

The perceptron is a simple neural network that was proposed by Frank Rosenblatt in 1958. It is a linear, single-layer network with just two layers, one layer representing the system's inputs and the other for the outputs. The connections between these two layers are modifiable and bidirectional. The learning process is supervised, and the network is capable of solving simple logical operations.[30]

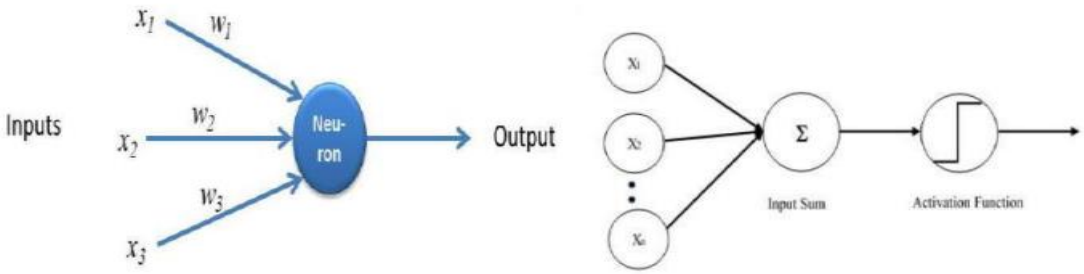


Figure IV.15:Structure and behavior of perceptron

The perceptron takes as input n values X1, X2, ...Xn, and calculates an output O, which depends on the sum of its components, weighted by real weights W. and the threshold θ The output O of this network will be in the following form:

$$o_k = \begin{cases} 1 & \text{if } w_n x_n \geq \theta_k \\ 0 & \text{if } w_n x_n < \theta_k \end{cases} \qquad \text{(II.12)}$$

The learning algorithm is Hebb's learning rule which affects the weight change by multiplying a neuron's input by its output and the network's learning rate.[30]

IV.5.4.2 The Multilayer Perceptron

A multilayer perceptron (MLP) is an improved version of the perceptron incorporating one or more hidden layers, which makes it a strong tool for complicated tasks. It is widely applied in facial recognition for decision making. In most cases, MLP networks are made up of fully connected networks. The first layer neurons receive the input vector and calculate their outputs which are sent to neurons in the second layer who calculate their own outputs , and so on from one layer to another until the output one .In contrast to MLP network, cells within the same layer do not have connections between them. Multilayer Perceptrons use supervised learning as well as back-propagation technique for error correction.[31]

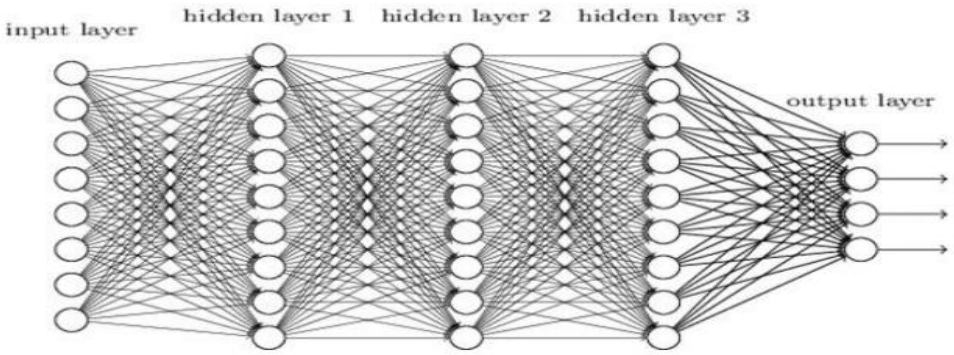


Figure IV.16:Multilayer Perceptron

IV.5.4.3 Convolutional Neural Network

In deep learning, convolutional neural network (CNN/ConvNet) is a type of modern technology which is often used to analyze pictures or images. ConvNets differ from ordinary neural nets based on matrix multiplications through Convolution method. Among other things, convolution in mathematics refers to the process of merging two functions into a new function that indicates how one function distorts another shape.[32]

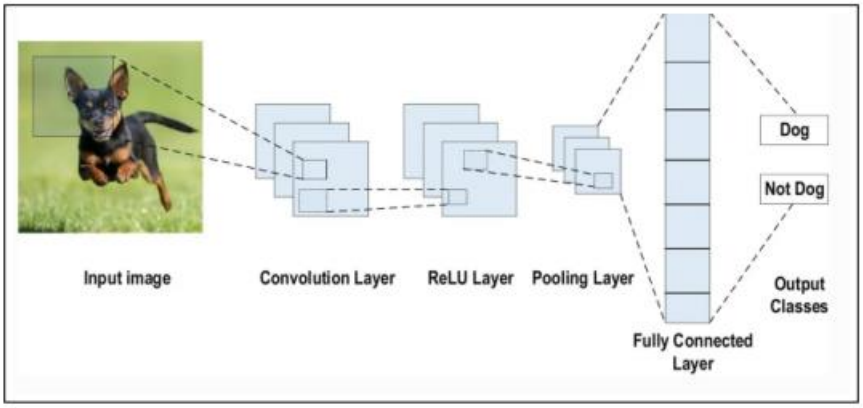


Figure IV.17:CNN Architecture

IV.5.4.3.1 The functioning of CNN

Before delving into the inner workings of the CNN model for image classification, let's start by defining what constitutes an image and how it is depicted. An RGB image, also known as a color image, is a matrix of pixel values with three levels. Conversely, a grayscale image shares the same characteristics as a color image but possesses a single level. Commonly, a CNN model comprises three core layers: a convolutional layer, a pooling layer, and a fully connected layer.[33]

1. Convolutional layer:

The input data is convolved and the results thereof are passed to the next layer. As shown in Figure 2.2, convolutional operation is an example. A particular neuron of the convolutional layer can only receive some portion of outputs obtained after convolution with a certain “kernel” from preceding layer. It should be noted that each neuron is associated with a set of output values, which it receives; as such this is referred to as “receptive field” for that specific neuron.[34]

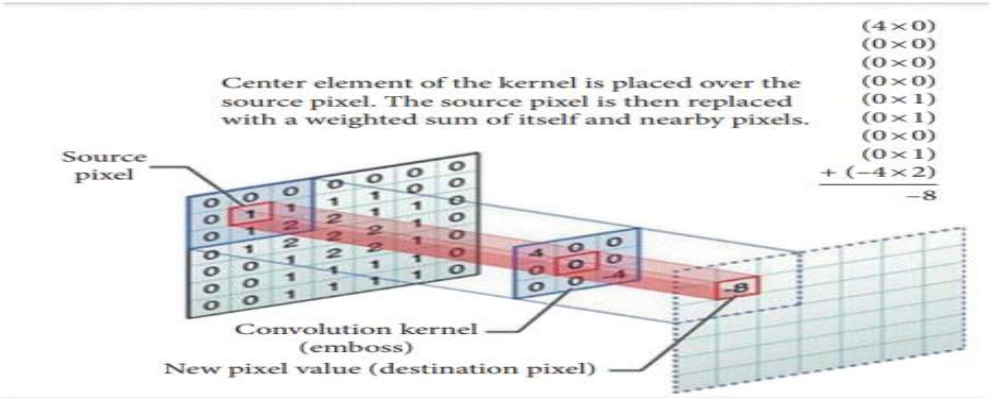


Figure IV.18:Convolution layer slides the filter over a given input. Output is the summation of an element by the element matrix multiplication of the filter and receptive field

2. Pooling layer:

This is another important component after the convolutional layers where each set of input outputs from previous layers gets consolidated into one cell. The common types of pooling include average pooling and max pooling illustrated in (Fig II. 17). Average pooling averages input values by taking their means while max pooling picks out the highest number.[34]

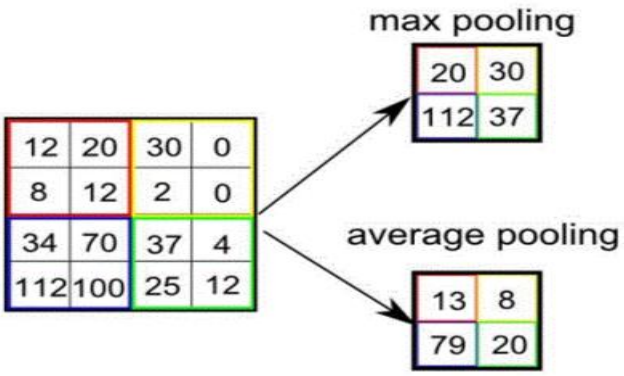


Figure IV.19:pooling layers

3. fully connected layer:

For instance, figure II. 18. shows that the fully connected layer acts as the last layer feeding into neural networks' network. Before being sent to neurons, matrices are usually flattened at this point. This becomes difficult at this point since there are many hidden layers with different weights attached to their outputs for each node meaning that tracking data becomes complicated when all computations are started here.[35]

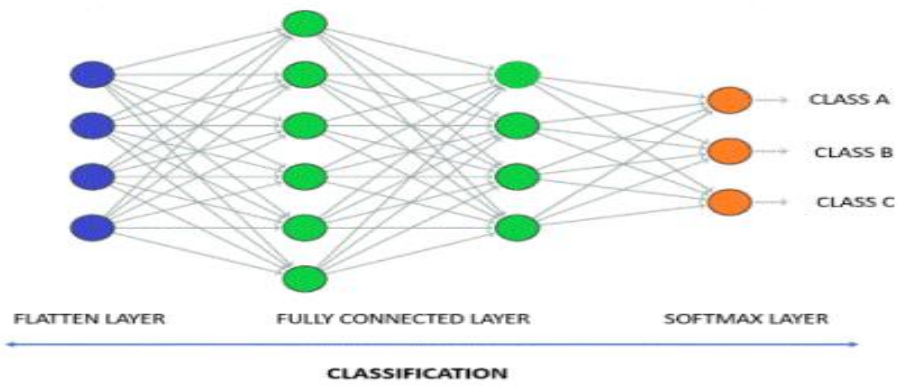


Figure IV.20:Fully Connected layers

IV.5.4.4 Long Short-Term Memory (LSTM)

Recurrence neural networks called Long Short-Term Memory (LSTM) is highly effective in picking out significant traits from data sequences. Time series, for example, have long-term dependencies that can be solved by LSTM. Another advantage of LSTM is its ability to handle multivariate time series data with multiple variables per time step. This enables it to perform e.g. multivariate forecasting Other than the above uses, LSTM is also applicable in multi-step forecasting because it has the capability to predict variable-length sequences. Similarly to Gated Recurrent Unit (GRU), LSTM makes use of gates for information flow control and as a result selective memory persistence and modification are realized with them. These gates play an important role in overcoming vanishing gradient problem associated with traditional RNNs due to which long term dependencies get lost. The features such as Forget gate, input gate, output gate and input modulation gates help LSTMs to efficiently capture both short-term and long term dependencies hence making them valuable tools for sequential data processing. [36]

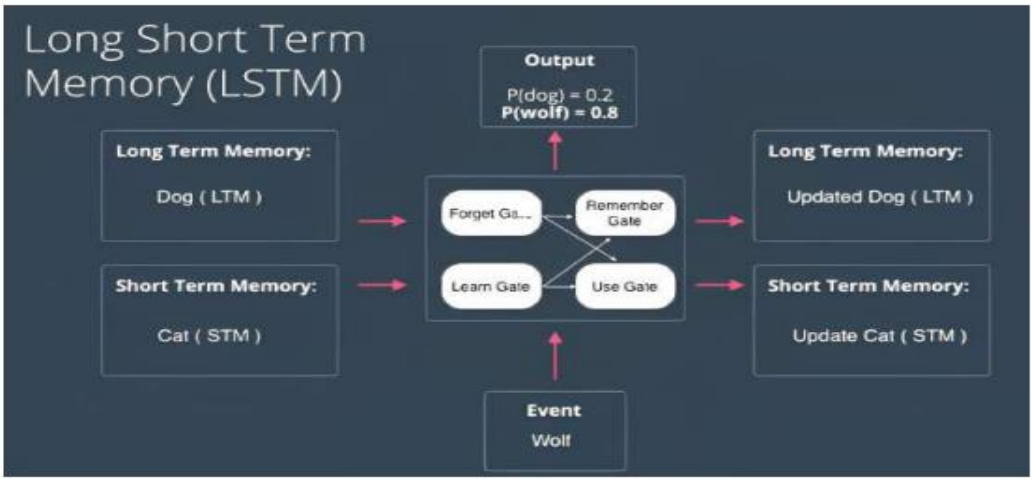


Figure IV.21:Remeber Gate

The above figure shows the simplified architecture of LSTMs.

The actual mathematical architecture of LSTM is represented using the following figure:

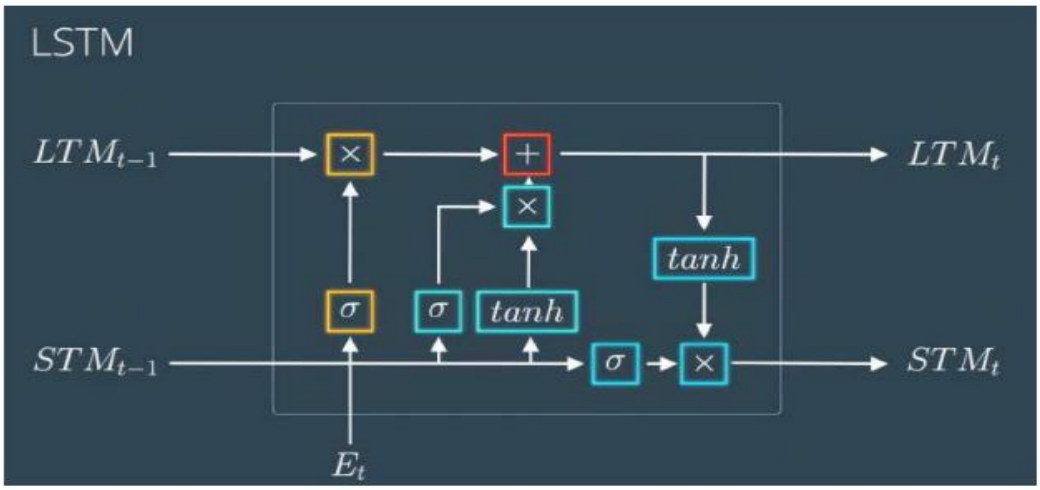


Figure IV.22:LSTM Architecture

IV.6 Internet of things

IV.6.1 Definition

The Internet of Things (IoT) is a worldwide open and flexible framework comprising a network of intelligent objects that interconnect and communicate through the Internet. These objects gather and exchange data using embedded software, cameras, and sensors capable of detecting light, sound, and motion. Furthermore, smart devices have the capacity to operate autonomously, as well as being remotely controlled and monitored. In the realm of IoT, an object can range from a person with a pacemaker to a farm animal equipped with a chip (transmitter), a car installed with sensors to alert the driver about low tire pressure, or any natural or artificial object that can support an IP address and transmit data across the network.[38]

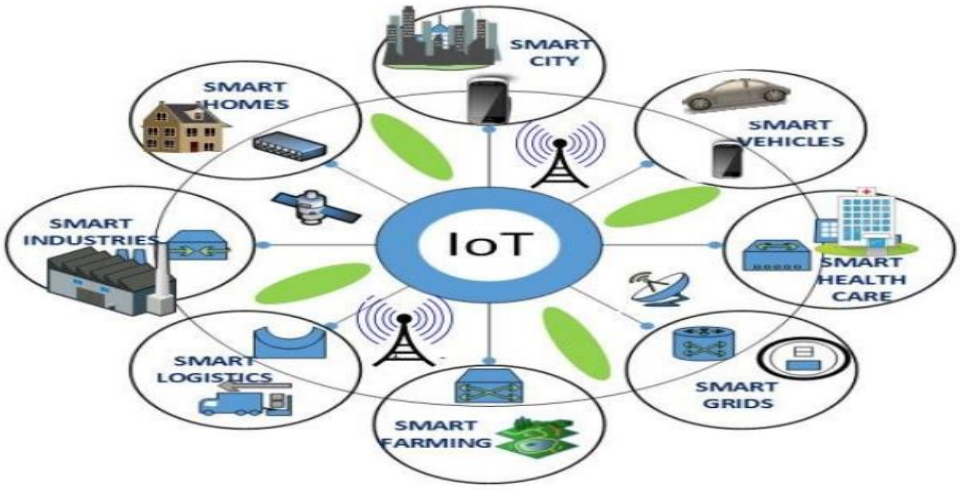


Figure IV.23: Internet Of Thing

IV.6.2 IOT Architecture

An architecture is a blueprint that details the physical components, technological organization, network configuration, operational protocols, and data requirements in use. The configuration of the Internet of Things changes according on operating timelines. To secure future success, IoT systems require strong design principles and defined processes. There are two potential IoT structures: three-layer and five-layer models. The European Telecommunications Standards Institute (ETSI) has proposed a three-layer design for IoT, consisting of the perception layer, the network layer, and the application layer. The five-layer concept goes beyond this by adding two more layers for processing and functionality, as seen in the accompanying diagram.[39]

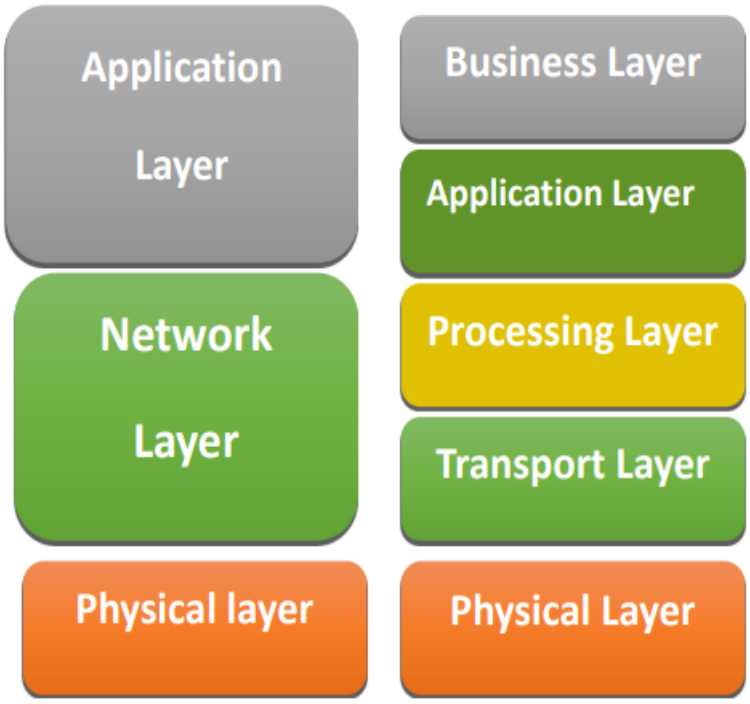


Figure IV.24:IOT Architecture

IV.6.2.1 Three-layer Architecture

It is arranged in the level :perception .network and application

1. **Perception Layer:** This layer includes a physical layer with sensors which sense and collect information about the environment, it detects particular physical parameters or recognizes other smart entities within its vicinity.
2. **Network Layer:** This part of the system is responsible for establishing connections among intelligent devices, network devices as well as servers and it goes beyond into transmitting sensor data and processing them.

3. **Application Layer:** The purpose of this layer is to provide user services that are connected to various applications. It lists different kinds of IoT applications like smart homes, smart cities, e-healthcare etc.[39]

IV.6.2.2 Five-layer Architecture

It includes all these three layers (perception, network and application) plus business and processing layers.

1. **Transport Layer:** This allows for the transfer of sensor data from perception layer to processing layer using different types of communication technologies including Wi-Fi, 3G, LAN, Bluetooth, RFID and NFC.

2. **Processing Layer (Middleware):** Data acquired from transport layer is stored, analyzed and processed by middleware that uses different technologies such as databases, cloud computing and data processing units.

3. **Business Layer:** It is responsible for managing the entire IoT ecosystem which includes applications, business strategies financial outcomes and user privacy protection.[39]

IV.6.3 Components of IoT

Here are four basic components of an IoT system that explain how it works:

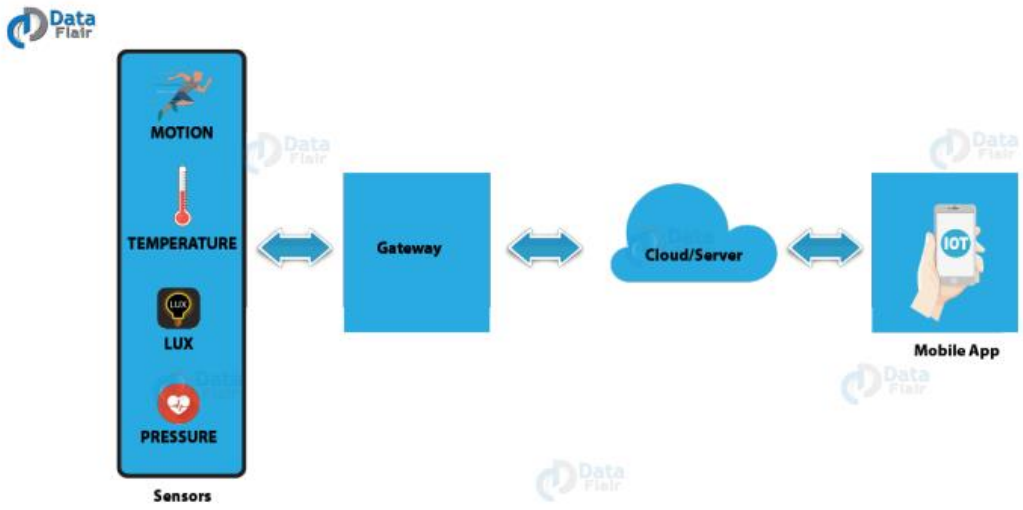


Figure IV.25:IOT Components

1. **Sensors/Devices:** The surrounding is a very important place for sensors to gather the most accurate data. It may be different levels of complexity of information collected ranging from simple temperature monitoring sensor to complex full video stream. When more than one sensor is combined into a single device, it can do more than just sensing and reacting. .[40]



Figure IV.26:Device

2. **Connectivity:** However, in order to be sent, this data must be transported after being accumulated and sent to the cloud platform from where it originated. For instance, sensors communicate with clouds over various communication and transport technologies such as cellular networks, satellite networks, wide-area networks (WAN), low-power wide-area networks, Wi-Fi as well as Bluetooth among other alternatives. .[40]



Figure IV.27:connectivity

3. Data Processing: Once data has been transmitted to the cloud after being collected, it is processed by software. This process may involve something like verifying that temperature readings on devices such as air conditioners or heaters are within normal ranges. [40]

4. User Interface :These IoT components underlie almost every IoT system around the globe today; thus they have been separated into multiple architectural layers which make the entire IoT network more efficient.[40]

IV.7 Conclusion

The chapter explores artificial intelligence, machine learning, deep learning and the internet of things shedding light on strong and weak AI, various types of machine learning, neural networks, applications of deep learning as well as IoT structure. It conducts a comprehensive investigation into these technologies with special regard to their effects on Intelligent systems and interconnected appliances.

Chapter III: Materials and software used

V.1 Introduction

In the document, chapter III addresses different areas concerning technology such as APIs, Fast API, MQTT, databases, MySQL, Python, Node-RED, Arduino IDEs and AD8232 ECG Sensors as well as TensorFlow, Keras Sklearn and Matplotlib libraries. In each of the sections here there are detailed explanations of how these technologies work in programming field; web development; IoT; data analysis; and machine learning. We have a glimpse into all these themes to realize their importance for the modern ways of doing things in the world.

V.2 Application Programming Interfaces

APIs stand for Application Programming Interfaces which are sets of regulations that enable software applications to communicate among themselves such as exchanging data and functionalities. They simplify app development and speed it up by allowing developers to use data and services from other apps instead of building them afresh. Another thing is that they make available a simple way for application holders to provide their application's information and functions through internal departments or business partners or even third parties. APIs improve system security by only sharing the required information, hiding the finer points of other internal systems. On the technical side, an API documentation acts as a guidebook for developers on how they should relate with it in order to improve user experience and optimize successful APIs.[41]

V.2.1 How do Application Programming Interfaces (APIs) work?

API functionality, however, can be best understood as an interactive process between a server and client. Where the client is usually an application that initiates requests while the server responds. The API acts as an intermediary in this connection.

For instance, consider third-party payment processing in e-commerce as a typical example of API operations. APIs are used to connect when a user selects "Pay with PayPal" for making payments.

- When I press on the pay button, it means that I am asking for data which is an API Call. This request is then sent over the web to the application's server via its own URI through which details like headers, verbs and sometimes even a body of request.
- Once a valid customer request has been received from the product page, the API will invoke some other program or even web server such as third party payment system.
- After this step is completed by providing requested information, the server returns response back to API.

- The data is then relayed back to the original requesting application such as product website by this Software Application Interface (API).

Regardless of how much these web services differ in terms of transferring data all interactions are conducted through APIs. That implies end users do not directly interact with APIs; they facilitate seamless transfer of information within systems or applications.[41]

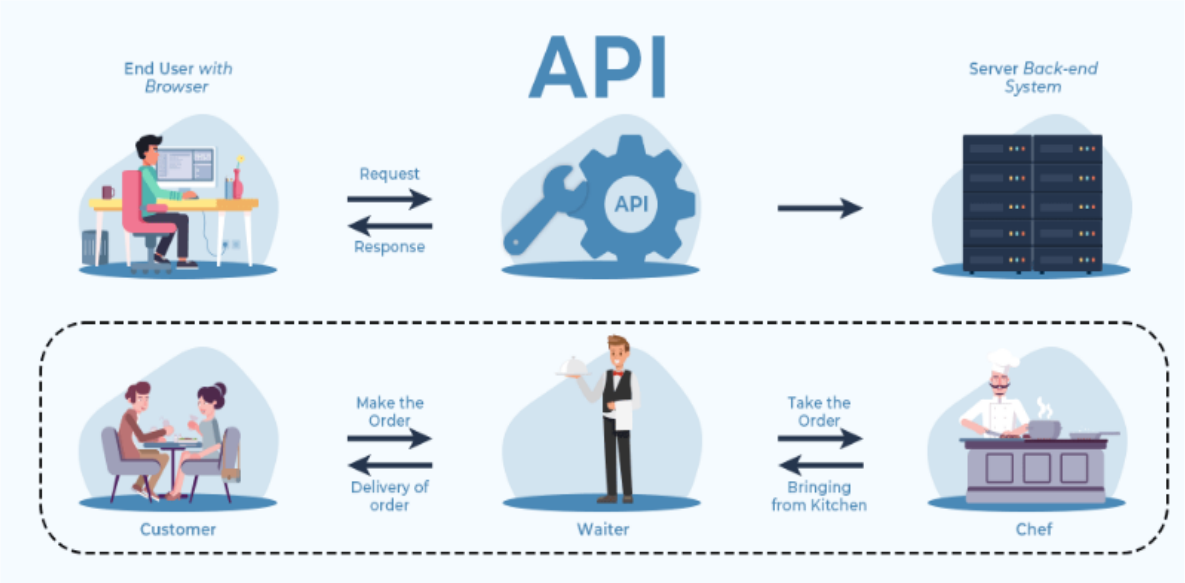


Figure V.1: API Functionality

V.3 Fast API

Fast API is a cutting-edge, high-speed web framework tailored for Python 3.7+ with standard Python type hints specifically meant for creating APIs. In general, this allows developers to build their own documentations with ease and understand the logic behind it. It can be used to make the process of testing web services easier by specifying required data and available operations. This makes it friendly to use as well as reducing coding errors in order to meet production needs very quickly; hence making its programming user-friendly. Moreover, Fast API is also compatible with Open API and JSON Schema widely supported standards for APIs.[42]

V.3.1 Fast API's Features

1. **Automatic Documentation:** Creates interactive API documentation using OpenAPI standard, which makes it easier to comprehend and test APIs.
2. **Python Type Hints:** Python-type hints are used for better code readability, data validation, and automatic documentation generation.
3. **Data Validation:** Automatic data validation, serialization, deserialization using Pydantic models

4. **Asynchronous Support:** It has full support for asynchronous operations with the help of Python's async-await for performance improvements.
5. **Dependency Injection:** It enables one to declare endpoints that depend on other modules hence promoting modularity, testability and maintainability.
6. **Security Features:** Some of its built-in security features include OAuth2, JWT as well as automatic validation that helps in mitigating some common security vulnerabilities.[42]

V.4 MQTT (Message Queue Telemetry Transport)

MQTT (Message Queuing Telemetry Transport) is a lightweight message protocol that works on the publish-subscribe mode. It's designed for constrained devices and bandwidth-limited, high latency or unreliable networks. MQTT is widely used in Internet of Things (IoT) scenarios to enable efficient communication among sensors, actuators and other devices.[43]

V.4.1 Operating Mode

In MQTT, messages are published through a broker: there is no direct client-to-client communication. Messages have two parts: content and topic. The last message on each topic is stored by the broker. To get such messages from the broker clients may connect to it. This method enables multiple clients' communication even if they are not connected at the same time. It means sorting messages by topics and distributing them according to subscription rules . It allows any application to send out its own messages using any topic it chooses; at the same time, applications can subscribe for all new postings on one chosen topic only.[44]

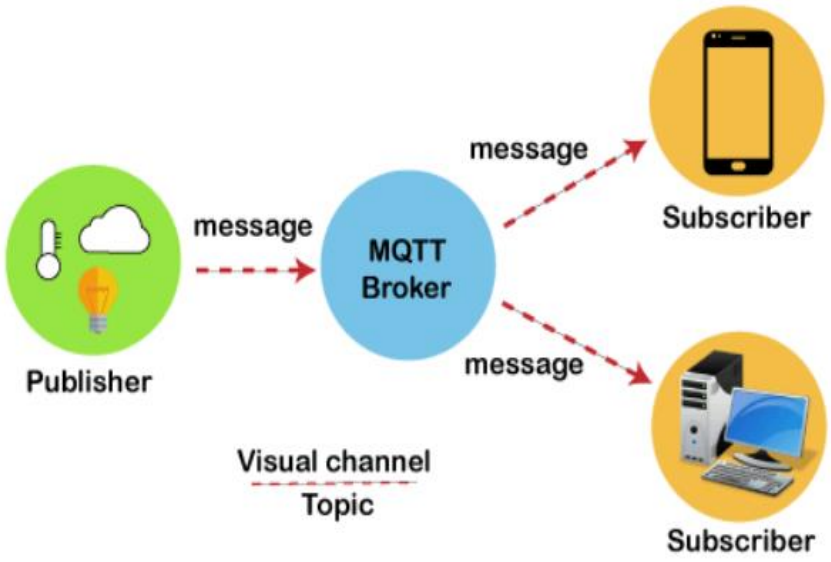


Figure V.2: MQTT Architecture

V.4.2 Components of MQTT

A. It is possible for a client to perform the duties of a publisher or a subscriber in MQTT.
B. The broker acts as an intermediary server that enables transmission of information between clients.
C. In MQTT, subject commonly known as topic is used by devices as connection point and central messaging hub to distribute messages in form of publication-subscription pattern.
D. Message or data is the purposeful exchange of data between devices this can be in form of commands or data.[45]

V.4.3 Why MQTT is so important in IoT?

As one of the key factors behind this, MQTT has become a critical protocol for IoT solutions.

Most notably, it differentiates itself as one of the lightest protocols used in IoT today. It is an open standard and can be integrated into any hardware or software. In addition, there are client libraries available in almost all major programming languages to support easy development of MQTT-based applications for IoT.

MQTT's publish/subscribe model enables unprecedented flexibility to accommodate various use cases and structures of IoT projects. Importantly, publishers and subscribers within the system operate independently from each other, while all connections are managed by broker.

This protocol enables development of highly scalable projects that may involve millions of connected devices under one system. By means of MQTT's bidirectional communication capability messages can be effectively broadcasted to large groupings of devices.

Lastly, various authentication methods and data security protocols such as TLS encryption have strong backing on MQTT platform.[46]

V.5 Database

V.5.1 Definition of a database

A data base is the container in the computer or the central part of computer equipment that is used to obtain, store, process and use information. The main ideas for designing a database during its design phase are based on entity, attribute and association concepts;

-Entity: An object, subject or concept within the field of activity for which the database is used and around which data is captured (e.g., persons, products, orders, reservations etc.).

-Attribute: Each entity has attributes stored in the database. For example, a person has such attributes as name and address. Attributes may also be referred to as fields or columns. In the schema entities are portrayed as a bundle of attributes associated with subject matter.

-Association: Associations represent relationships between different entities like seller-customer-store.[47]

V.5.2 Role of a Database

In contrast to the traditional approaches, the development of multi-user database reflects a shift in business management. It has the following roles:

- Centralization of data: Afterwards, information is no longer scattered all over and kept in different files.
- Integration (all activities are held by one department can be seen by all departments)
- Distribution of archived information (it's simple to spread if it's at least available in one point).

This leads to benefits such as:

- Consistency of Information (one piece of similar information having just one value)
- Reduction in Redundancy (information that is not diverse is saved only once)
- Less inputting and updating laboriousnesses (i.e., if information needs to be stored only once, then it will be entered only once; also its update will be done only once).[48]

V.6 MYSQL

V.6.1 Definition

MySQL is a database management system with SQL for its operations. MySQL is open source, thus permitting free use and alteration of the software. The Internet is where you download it without pay, and individuals may study or tailor the source code as they please. Nonetheless, if you wish to include MySQL in commercial applications you must have a license from MySQL AB..[49]

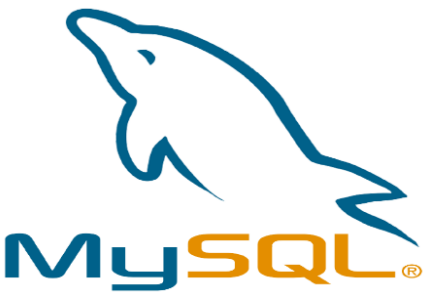


Figure V.3:My SQL logo

V.6.2 Historical

Widely known and trusted, MySQL is a Database Management System that has been around since 1995. This has made it undergo various versions and updates in the quest to enhance its performance as well as include more features. Most websites being large and some of them being applications make it the best choice among so many others.

MaxDB was produced by MySQL AB working together with SAP AG in 2003 which integrated with the SAP suit thereby providing more comprehensive database management solutions. Following Oracle's purchase of Sun Microsystems in 2009, questions arose regarding the future of MySQL under Oracle's management.

MySQL still remains a strong tool for boosting SQL use and providing effective ways of managing data. For businesses or professionals looking forward to coming up with highly efficient applications, this is an ideal option for them. As a result, through advancing its SQL capabilities and aggregating information sources into one place, it may be seen as a free resource helping companies develop future sales strategies and industry progression.[49]

V.6.3 Benefits

- Open Source.
- Reliable and relatively high-performance.
- Rich in functionality.
- Simple to use and administer.
- Inheritance of tables.[49]

V.7 Python

The ease of learning the programming language and its readability has made Python a very versatile and friendly tool. Python was created in the 1980s by Guido van Rossum, and it has undergone several versions since then, with Python 3.11 being the latest one released in 2022. It is becoming increasingly popular, particularly in areas like machine learning, data analysis, and web development.



Figure V.4:python logo

Python's simplicity, being object-oriented, open-source availability, extensive library support and robust community are its main selling points. Nevertheless, apart from the easy availability and increased productivity offered by Python there are certain limitations as well. Knowing both its strengths and weaknesses will give users vital information necessary for deciding on whether to use it for educational or developmental purposes or not.[50]

V.8 Node-RED

Node-RED is a graphical development tool that eases the connection of hardware devices, APIs, and online services. Initially developed for IBM's internal use but made open-source since 2016, it is now widely embraced in automation industry as a user friendly method for developing Industrial IoT applications that effectively collect, process and exchange data over the internet.[51]

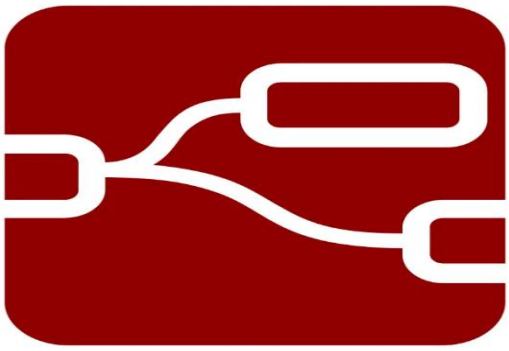


Figure V.5:Node-red logo

There are two main ways Node-RED simplifies the development of Industrial IoT applications:

For example, Node-RED has plenty of nodes that are pre-configured to perform complex tasks such as MQTT-based data transport, Modbus/TCP or emailing. Besides that, other nodes provide data publishing services into online providers including Dropbox and Google Drive. As at today, there are more than two thousand library nodes available for creating applications within Industrial IoT space with more constantly added by committed open source community.

Nodes are dragged onto an application's canvas then linked together using this graphical development environment offered by Node-RED. These may involve data collection, processing or transmission among other things. This graphic orientation improves how cost-effective management environments can be created for efficient running of Industrial IoT applications.[51]

V.9 Arduino IDE

The Arduino IDE is open-source software developed by Arduino.cc for writing, compiling and uploading codes to different Arduino modules. It is the official software of Arduino that allows code compilation to be simplified through a friendly interface and beginners can easily get into the learning process.[52]

Available on MAC, Windows and Linux operating systems, Arduino IDE runs on Java Platform. Its features include debugging, editing, and compiling codes. The most common types of these modules are: Arduino Uno, Arduino Mega, Arduino Leonardo, and Arduino Micro.

The user makes sketches in the IDE which then becomes Hex Files before being uploaded onto the microcontroller located on the board. There are an Editor for coding and a Compiler for compiling / uploading code to your chosen Arduino board in this environment.

This makes it ideal for C as well as C++ programming languages making it a versatile tool for developing codes. With its extensive functionality and flexibility, it caters for both beginners and experienced users.[52]

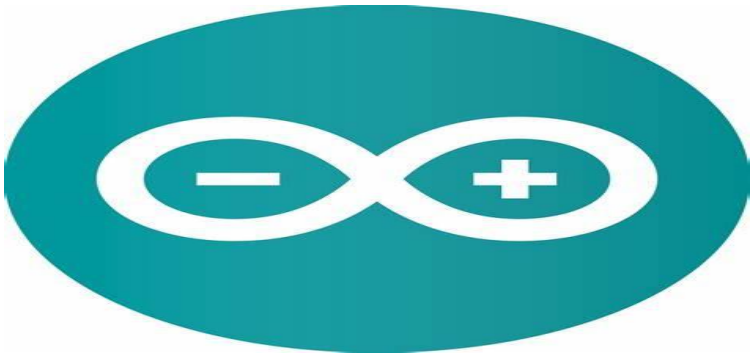


Figure V.6: Arduino IDE logo

V.10 AD8232 ECG Sensor

This is a cheap board used for sensing the electricity in the heart. The ECG represents this electrical activity and shown as an analog reading. While ECGs are often noisy, the AD8232 Single Lead Heart Rate Monitor acts as an operational amplifier (op-amp) to help in getting a clean signal from the PR and QT Intervals with simplicity. For instance, AD8232 is an integrated signal conditioning block specifically designed for ECG, and other biopotential measurement applications. It is built to recover amplify, and filter small bio potential signals embedded with noise caused by movements or placing remote electrodes.[53]

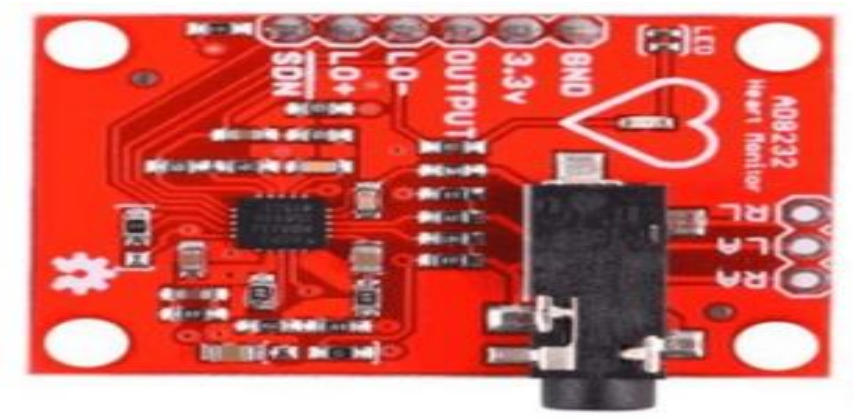


Figure V.7:AD8232 ECG Sensor

It has 9 possible breakout points from the IC that you can solder pins, wires or other connectors to. These pins like SDN, LO+, LO-, OUTPUT, 3.3V and GND are important in making it interface with an Arduino or any developmental board. Also, the board has separate RA (Right Arm), LA (Left Arm) and RL (Right Leg) pins made for attachment and use of user-customized sensors. Additionally, there is integrated LED indicator light pulsating as per the cardiac rhythm.[53]

V.11 Libraries used

V.11.1 Tensorflow

TensorFlow, a modern, open-source library, is used by software engineers and data scientists to create, train and deploy deep learning models that support data-driven analysis. Data flow graphs allow complex neural networks for image recognition, voice recognition or time series analysis in TensorFlow. The fact that several programming languages are supported like Python which has a rich ecosystem for data science speaks to its versatility. Its core library is written in C++ and CUDA to ensure fast and efficient development of models thus making it one of the most popular libraries across different industries such as healthcare and construction. For this reason, TensorFlow continues to be a leading-edge technology driving innovations in machine learning as well as artificial intelligence with its advanced features alongside real-world applications.[54]

V.11.2 Keras

Keras is that high-level API of TensorFlow, giving a friendly and productive interface to solve machine learning problems especially on the modern deep learning techniques. Keras covers all aspects of the ML workflow including data processing, hyperparameter tuning and deployment, with a strong emphasis on enabling fast experiments.

Users can make use of Keras for full leverage of TensorFlow's scalability and cross-platform compatibility, running their models on TPU Pods or GPU clusters and exporting them to be used in browsers or mobile devices. Moreover, Keras models can be served through web APIs.

The design of Keras tries to decrease cognitive load by:

- Presenting easy-to-understand but unified interfaces
- Reducing common activities for well-known scenarios
- Giving precise error messages
- Complying with the principle of gradual complexity revelation that starts from simple level moving towards advanced features
- Helping developers make shorter codes which are also understandable.[55]

V.11.3 Sklearn

Sklearn (Scikit-learn) is a python library that provides an easy way to work with different machine learning, data preprocessing, cross-validation and visualization techniques. This is a complete one-stop shop for many machine learning tasks such as Classification, Regression, Clustering and many more. The current release of Sklearn is 1.1 and it works well with Python 3.8 upwards.

To effectively set up Sklearn, ensure you have installed NumPy or SciPy first. Afterward, confirm that NumPy alongside SciPy are available in your system and then install Sklearn.[56]

V.11.4 Matplotlib

Matplotlib is a multipurpose Python plotting library that assists in the generation of static, moving and interactive visualizations for effective data analysis. Created initially by John D. Hunter in 2003, Matplotlib is currently supported by a dynamic community which has resulted in an array of powerful graphic capabilities for users.

Matplotlib allows different types of plots to be made, extensive customization options for aesthetic control, integration with NumPy for simple data array plotting, creating superior publications, and expanding its functionality using toolkits such as Seaborn and Basemap.[57]

V.12 Conclusion

To sum up, Chapter III explains several technological concepts and tools used in modern software development, IoT applications, data analysis, and machine learning. The chapter contains a wide variety of technology topics from the basics of APIs and Fast API to MQTT functionalities, databases, Python language programming, Node-RED platform usage, Arduino IDE tool and AD8232 ECG Sensor. Moreover, there is also a mention on TensorFlow along with other famous libraries such as Keras Sklearn Matplotlib among others which are critical to developers as they create deep learning models and visualizations for different areas. This chapter's content discusses readers' familiarity with these major technological constituents that drive the changing digital world.

Chapter IV: Implementation and experimental testing

VI.1 Introduction

The objective of this chapter is to analyze the ECG signal database by conducting pre-processing procedures in a structured manner and designing MLP CNN 1D LSTM networks for the purpose of recognizing and classifying cardiovascular diseases. Upon evaluating the results, the most efficient model will be selected for further examination in the study's concluding phase. In this procedure, ECG signals captured with an AD8232 ECG sensor will be utilized as the primary inputs for our CNN network in the testing stage. Subsequently, the findings will be demonstrated through implementations on the Node-Red platform.

VI.2 Database (1000 Fragments ECG)

To evaluate the classifier developed, we utilized electrocardiogram (ECG) signals from the MIT-BIH Arrhythmia database, which is available online at [58]. This database is referred to as the "1000 ECG Fragments."

VI.2.1 Database Description

This database is structured as follows :

- ECG signals from 45 patients, including:
  - 19 females aged between [23 - 89] years.
  - 26 males aged between [32 - 89] years.
- The ECG signals encompass 17 classes:
  - Normal Sinus Rhythm.
  - Pacemaker Rhythm.
  - 15 types of cardiac dysfunctions.
- All ECG signals were recorded at a sampling frequency of 360 Hz and a gain of 200 μV/mV.

For analysis, 1000 non-overlapping 10-second fragments (files) of the ECG signal (3600 samples (points)) were randomly selected. The description of the collected signals is detailed in Table

number	Class (Diseases)	Fragments	Patients
1	Normal Sinus Rhythm	283	23
2	Atrial Fibrillation	135	6
3	Premature Ventricular Contraction	133	14
4	Left Bundle Branch Block Beat	103	3

Tableau VI.1:Description of the collected data

An important consideration is maintaining a balanced dataset. It's crucial that the number of signal fragments representing physiological cardiac variations does not disproportionately exceed those of other ECG signal classes. Such an imbalance could artificially enhance the effectiveness of cardiac disease recognition. To ensure a balanced dataset across various diseases, our focus was on analyzing the four specific diseases listed in the table

VI.3 ECG Signal Preprocessing

In this section, we worked with ECG signals from different cardiac diseases in our database, specifically focusing on four selected classes. Each file spans 10 seconds at a sampling frequency of 360 Hz, totaling 3600 samples per file. These files underwent processing following the steps outlined in Figure VI.1's flowchart. Before segmentation, we managed a total of 654 signals.

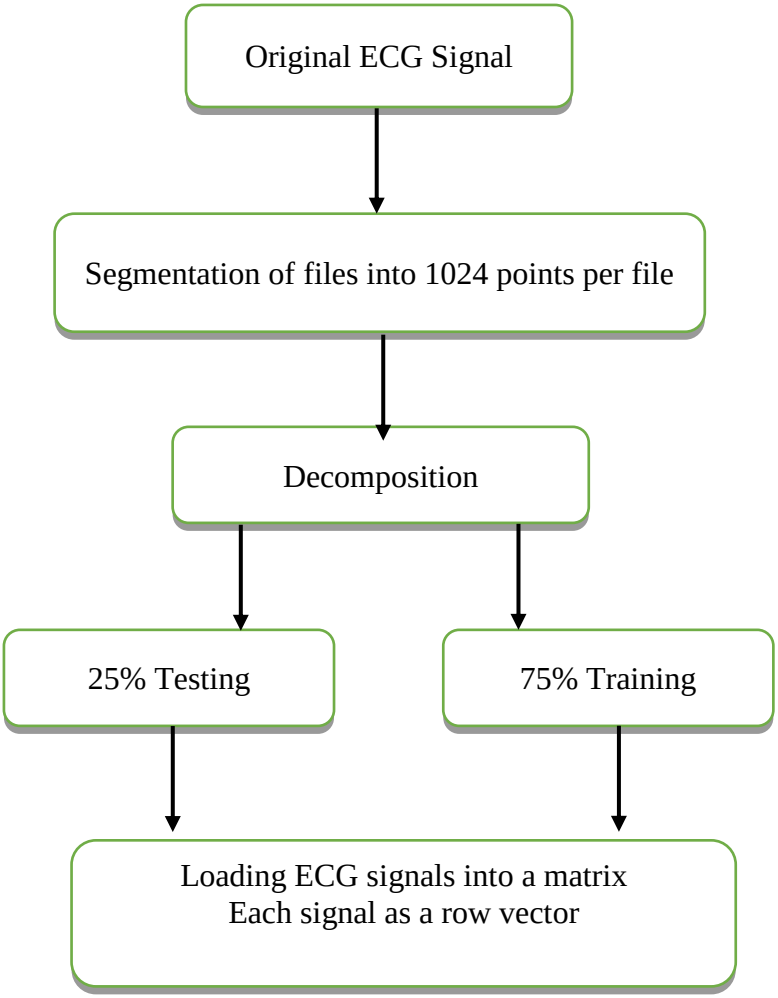


Figure VI.1:Data Processing Flowchart

VI.4 Model used :

In order to find the best neural network, we will test the three most common ones ,we selected Multi-Layer Perceptron (MLP), Convolutional Neural Networks (CNN), and a hybrid model combining CNN with Long Short-Term Memory (CNN-LSTM) because these architectures are well-established and widely used in deep learning. Each model offers unique strengths in processing and analyzing data. In this project, we will implement and evaluate these three network types to determine which one performs best in predicting cardiac diseases. By comparing their performance, we aim to identify the most effective approach for our embedded system We have evaluated approximately 30 models to identify the best performers within each neural network type

VI.4.1 Multi layer perceptron

We initially implemented a Multi-Layer Perceptron (MLP) model to predict cardiac diseases due to its straightforward architecture and effectiveness in handling structured data. The MLP architecture comprises an input layer with 512 neurons activated by ReLU, followed by a dropout layer to prevent overfitting. This is succeeded by a dense layer of 256 neurons, another dropout layer, and a final dense output layer using the softmax activation function for classification. The model was compiled using the Adam optimizer with a learning rate of 0.001 and trained with categorical cross-entropy loss. However, the MLP model did not yield satisfactory results, achieving a test loss of 0.96, a test accuracy of 0.59, an F1 score of 0.57, precision of 0.96, and recall of 0.49 as you see in the figureVI.2 . These results indicate that while the model was able to achieve high precision, its overall accuracy and recall were relatively low, suggesting it struggled to generalize well to the test data. This prompted further exploration of more complex models to improve predictive performance.

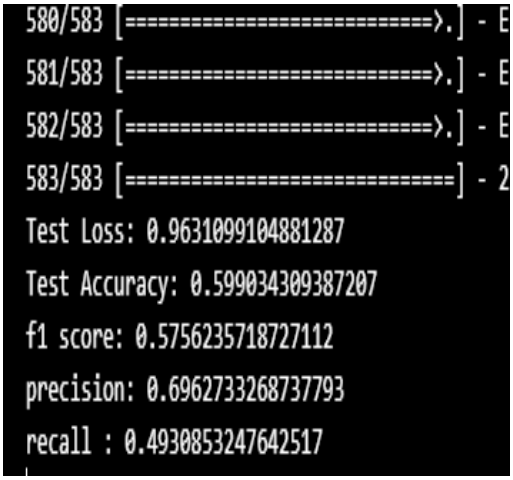


Figure VI.2:result MLP

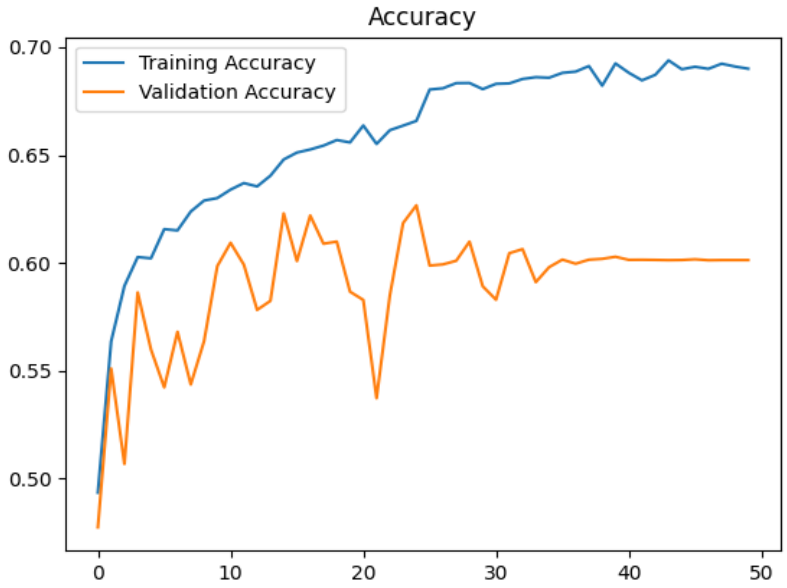


Figure VI.3:accuracy graph MLP

- There is a significant increase in accuracy in the early epochs between 0 and 10 epochs, followed by a more gradual increase throughout the rest of the training. This suggests the model learns the simpler patterns in the data quickly and then requires more epochs to refine its performance on more complex patterns.
- The final accuracy after 50 epochs is around 59% as you see in the figure

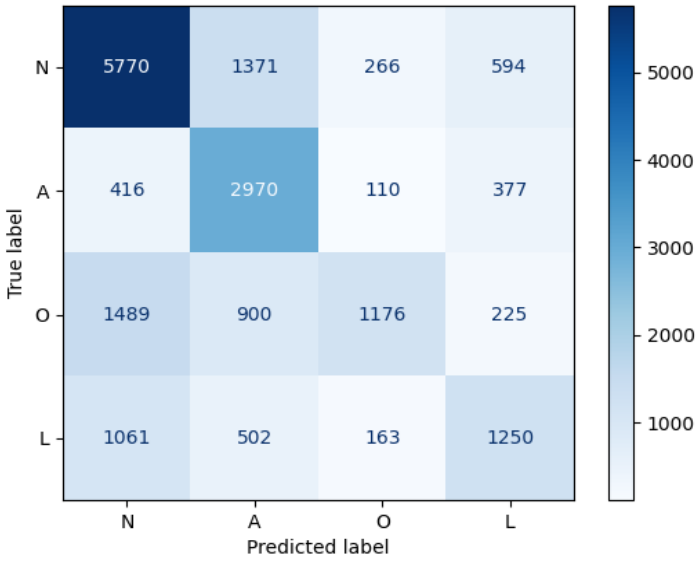


Figure VI.4:Figure MLP Confusion Matrix

Based on the confusion matrix, we can make the following observations:

- Class N is the most common class: There are more instances of class N than classes A and L.
- The model is most accurate for class N: The model has the highest number of correct predictions for class N (5770).
- The model has the most difficulty with class O: The model has the lowest number of correct predictions for class O(1176).
- The model frequently misclassifies class O as class L: There are 1489 instances of class O that were incorrectly classified as class N.
- Overall, the model seems to perform well for class N but struggles with classes O and L.

VI.4.2 Cconvolutional neural network

Next, we implemented a Convolutional Neural Network (CNN) to predict cardiac diseases, leveraging its strength in feature extraction. The CNN architecture begins with a 1D convolutional layer with 32 filters and a kernel size of 5, activated by ReLU, followed by a max-pooling layer to reduce dimensionality. This process is repeated with 64 and 128 filters in subsequent convolutional layers, each followed by max-pooling and batch normalization to enhance training stability and performance. After the convolutional layers, the model includes a flattening layer to prepare the data for the dense layers. The dense layers consist of 256 and 128 neurons respectively, with ReLU activation, interspersed with a dropout layer to prevent overfitting. Finally, the output layer uses the softmax activation function for classification into the predefined number of classes. The model was compiled with the Adam optimizer, set to a learning rate of 0.001, and trained using categorical cross-entropy loss. This architecture proved effective, achieving a test loss of 0.24, test accuracy of 0.92, F1 score of 0.92, precision of 0.92, and recall of 0.92 as you see in the figure IV.5 . These results highlight the CNN's ability to accurately extract and classify features from the input data, making it well-suited for our project’s goal of real-time cardiac disease prediction

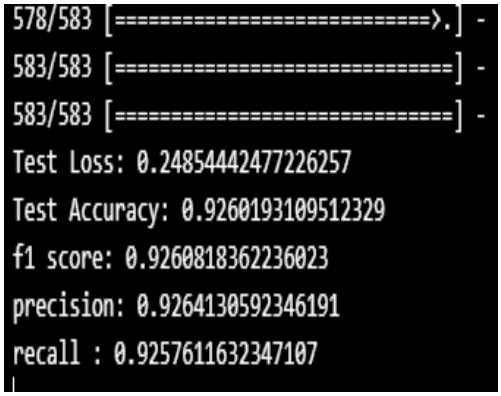


Figure VI.5:CNN result

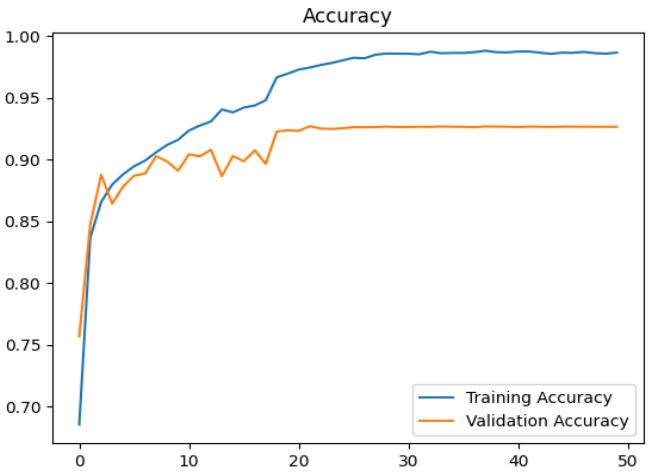


Figure VI.6:accuracy graph of CNN

- The graph exhibits a positive trend of increasing accuracy throughout the training epochs, indicating successful learning by the CNN model as you see in the figure IV.6. This suggests the model is effectively extracting relevant features from the training data and progressively improving its classification capability.
- The initial epochs demonstrate a steeper rise in accuracy, followed by a more gradual increase. This behavior is characteristic of CNNs, where they often learn simpler patterns quickly and then require more epochs to refine their performance on more intricate features within the data

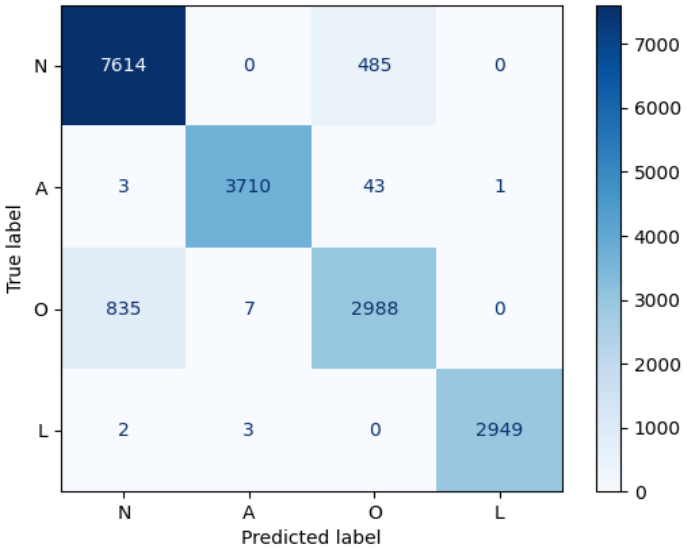


Figure VI.7:CNN Confusion Matrix

- The provided confusion matrix reveals a good class distribution. Class N exhibits a significantly higher number of true positives. While the overall accuracy cannot be

definitively determined from the matrix alone, the diagonal elements suggest the model performs well

- The model exhibits some difficulty in differentiating between classes O and N

VI.4.3 Combining CNN with Long Short-Term Memory (CNN-LSTM)

To further enhance the performance of our predictive system for cardiac diseases, we implemented a hybrid model that combines Convolutional Neural Networks (CNN) with Long Short-Term Memory (LSTM) layers. The architecture starts with a 1D convolutional layer with 32 filters and a kernel size of 5, activated by ReLU, followed by a max-pooling layer to reduce dimensionality. This is followed by additional convolutional layers with 64 and 128 filters, each accompanied by max-pooling and batch normalization layers to improve stability and performance. The output from the convolutional layers is then passed to an LSTM layer with 128 units, which is capable of capturing temporal dependencies and sequential patterns in the data. An Attention mechanism is incorporated to allow the model to focus on the most relevant parts of the input sequence. The model continues with a flattening layer, followed by dense layers with 256 and 128 neurons respectively, using ReLU activation and a dropout layer to prevent overfitting. The final layer is a softmax layer for classification into the predefined number of classes. Compiled with the Adam optimizer and a learning rate of 0.0001, and trained using categorical cross-entropy loss, this model achieved a test loss of 0.054, test accuracy of 0.98, F1 score of 0.98, precision of 0.98, and recall of 0.98 as you see in the figure VI.3. The inclusion of the LSTM layer significantly boosts the model's performance by effectively capturing the temporal relationships within the ECG data, resulting in high accuracy and robust predictive capabilities for cardiac disease detection.

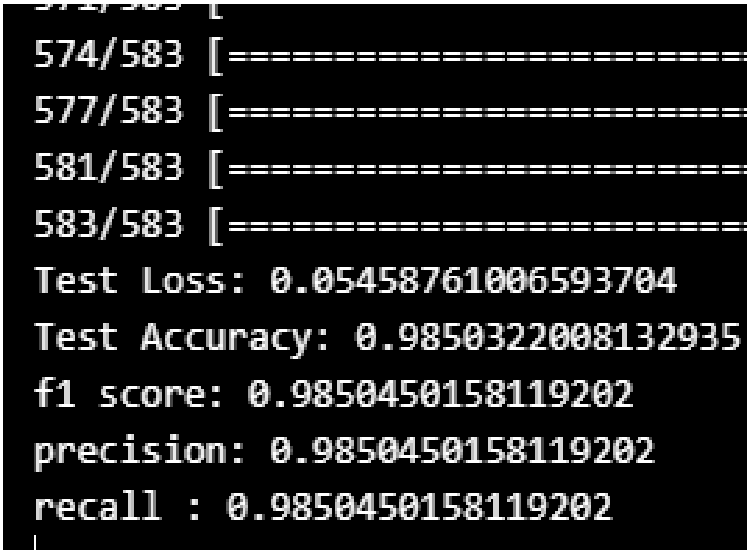


Figure VI.8: (CNN _LSTM) result

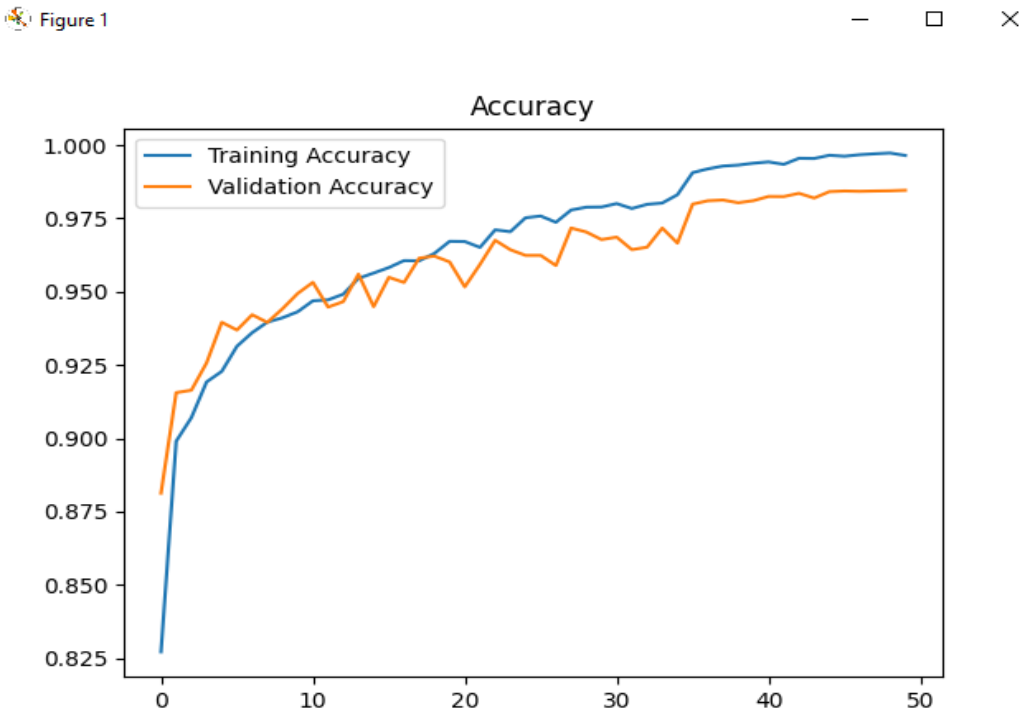


Figure VI.9:(CNN-LSTM) accuracy graph

- The accuracy increases throughout the training epochs, which signifies successful learning by the CNN-LSTM model. This indicates the model is effectively capturing both spatial and temporal features from the data and progressively improving its classification capability.
- The initial epochs demonstrate a steeper rise in accuracy, followed by a more gradual increase. The training and validation accuracy curves appear to level off around epoch 36, suggesting potential overfitting.. This indicates the model might be memorizing the training data rather than generalizing well to unseen examples.

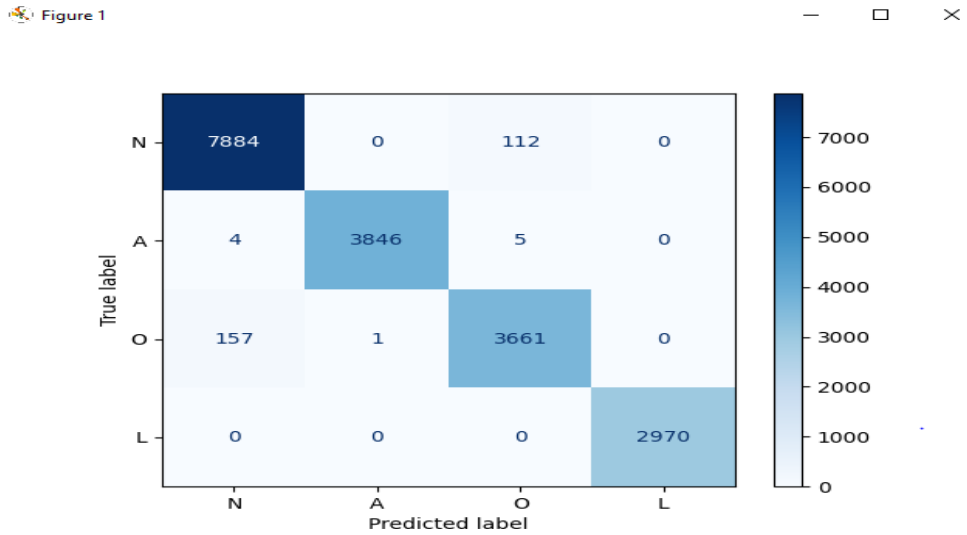


Figure VI.10: cnn lstm acquisition matrix

- The confusion matrix showcases a strong classification performance for classes N, A, and L, reflected by the high values along the diagonal. This suggests the CNN-LSTM model has effectively learned to distinguish these classes. While misclassifications are minimal, a closer look reveals a slight confusion between class O and class N

VI.4.4 Comparison of models

The MLP model consists of dense layers with 512 and 256 neurons, respectively, and incorporates dropout layers to prevent overfitting. Despite its simplicity and robustness for structured data, the MLP model yielded suboptimal results, with a test loss of 0.96, test accuracy of 0.59, F1 score of 0.57, precision of 0.96, and recall of 0.49. The relatively low accuracy and recall suggest that the MLP struggled to generalize well to the complex and sequential nature of ECG data, highlighting its limitations for this application.

The CNN model, designed for effective feature extraction, includes multiple convolutional layers with increasing filter sizes (32, 64, and 128), max-pooling layers to reduce dimensionality, and batch normalization layers to stabilize training. This architecture resulted in significantly better performance, achieving a test loss of 0.24, test accuracy of 0.92, F1 score of 0.92, precision of 0.92, and recall of 0.92. The CNN's ability to capture spatial features from the input data contributed to its improved predictive capabilities compared to the MLP.

To further enhance the model's ability to capture temporal dependencies in ECG data, we implemented a hybrid CNN-LSTM model. This model incorporates the same convolutional layers as the CNN, followed by an LSTM layer with 128 units and an Attention mechanism. The addition of LSTM allows the model to effectively learn and remember sequential patterns in the data. This architecture achieved the best performance, with a test loss of 0.17, test accuracy of 0.95, F1 score of 0.95, precision of 0.95, and recall of 0.95. The superior results can be attributed to the LSTM layer's capability to handle temporal sequences, which is crucial for accurate ECG analysis.

In summary The MLP model, while simple, was not well-suited for this task due to its inability to effectively capture the complex and sequential nature of the data. The CNN model demonstrated significant improvements by leveraging its feature extraction capabilities. However, the CNN-LSTM hybrid model provided the best performance, combining the strengths of CNN for spatial feature extraction and LSTM for temporal sequence learning, making it the most suitable choice for our project.

VI.5 Experimental test

Having thoroughly analyzed the results, we can now combine all the components we've developed to evaluate the system's predictive capabilities. The experiment will occur as shown in Figure VI.11

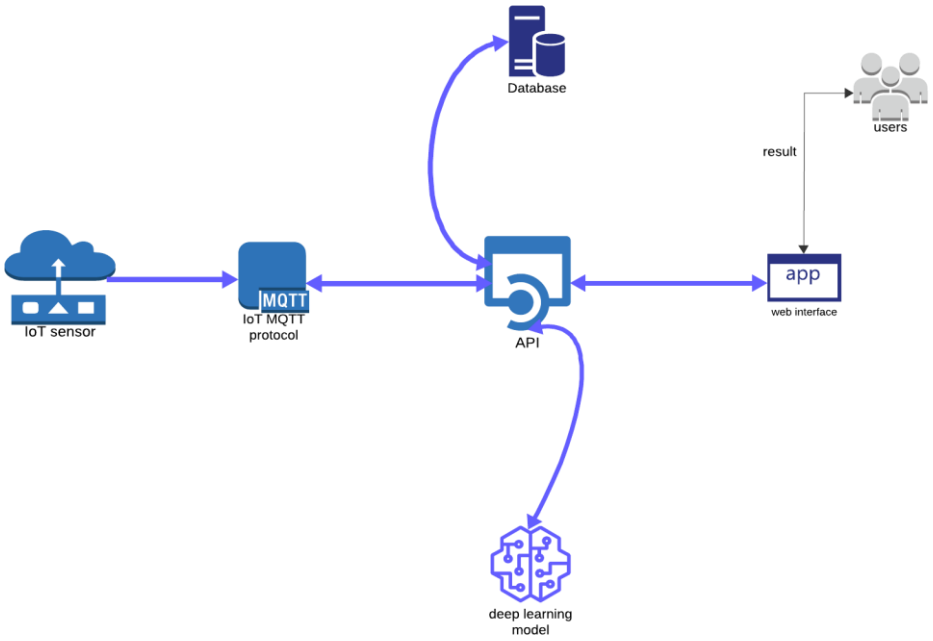


Figure VI.11:experment test map

VI.5.1 Implementation of the acquisition node

The initial step involved establishing the electrical connection between the ESP32 microcontroller and the AD8232 ECG sensor. This involved implementing the circuit configuration as depicted in the Figure VI.12

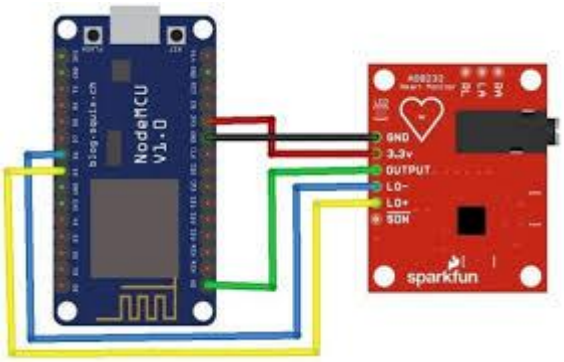


Figure VI.12:the acquisition node

The placement of electrodes on the human body is shown in the figure

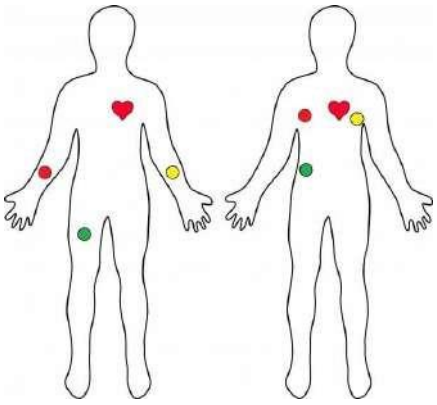


Figure VI.13:The electrodes placement

Upon executing the code and configuring the MQTT broker parameters, the signals can be observed on the serial plotter and the values displayed on the serial monitor. The signal measured by the AD8232 is presented in the figure

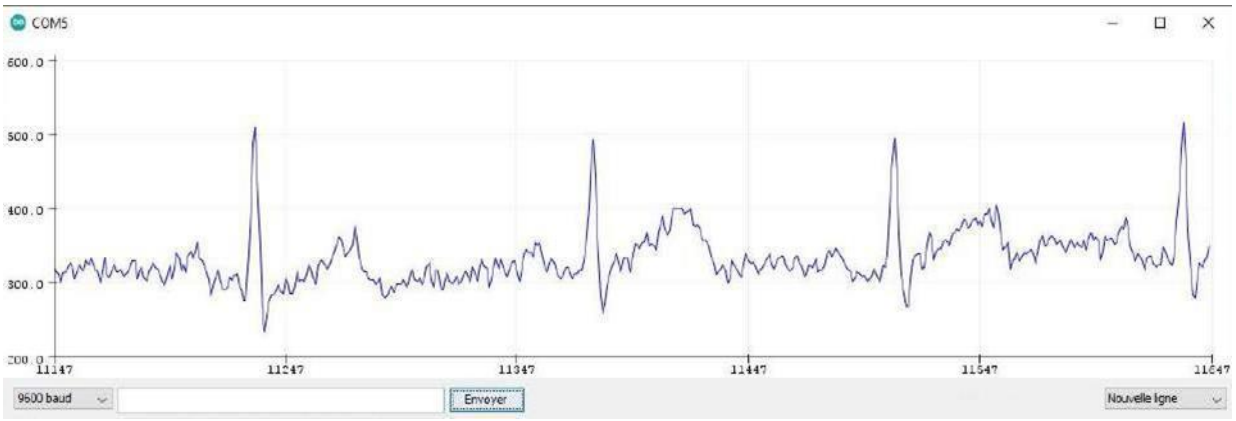


Figure VI.14:ECG signal

VI.5.2 Implementation

Utilizing a Mosquitto broker for the MQTT protocol, we have established a real-time data acquisition system. The system is configured to seamlessly integrate with a MySQL database. Upon sensor data transmission, a designated API receives the data in real-time and promptly stores it within the MySQL database for the prediction as you see in figure VI.15

```
Sending PUBLISH to auto-9172FEEE-BA48-AEB4-9709-620482061222 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Sending PUBLISH to nodered_87abce8aac5f94b9 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Received PUBLISH from esp32-client-C8:F0:9E:F4:92:50 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Sending PUBLISH to auto-9172FEEE-BA48-AEB4-9709-620482061222 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Sending PUBLISH to nodered_87abce8aac5f94b9 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Received PUBLISH from esp32-client-C8:F0:9E:F4:92:50 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Sending PUBLISH to auto-9172FEEE-BA48-AEB4-9709-620482061222 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Sending PUBLISH to nodered_87abce8aac5f94b9 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Received PUBLISH from esp32-client-C8:F0:9E:F4:92:50 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Sending PUBLISH to auto-9172FEEE-BA48-AEB4-9709-620482061222 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Sending PUBLISH to nodered_87abce8aac5f94b9 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Received PUBLISH from esp32-client-C8:F0:9E:F4:92:50 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Sending PUBLISH to auto-9172FEEE-BA48-AEB4-9709-620482061222 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Sending PUBLISH to nodered_87abce8aac5f94b9 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Received PUBLISH from esp32-client-C8:F0:9E:F4:92:50 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Sending PUBLISH to auto-9172FEEE-BA48-AEB4-9709-620482061222 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Sending PUBLISH to nodered_87abce8aac5f94b9 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Received PUBLISH from esp32-client-C8:F0:9E:F4:92:50 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
Sending PUBLISH to auto-9172FEEE-BA48-AEB4-9709-620482061222 (d0, q0, r0, m0, 'ecg/data', ... (32 bytes))
```

Figure VI.15:sensore sending data to database

The signals are sent to the API, which forwards them to the database in batches of 1024 data points. Each batch constitutes a single database insertion. Upon reaching 1024 data insertions in a column, the API triggers a code that loads our pre-trained CNN-LSTM model and performs predictions as you see in the figures *VI.16 and VI.17*

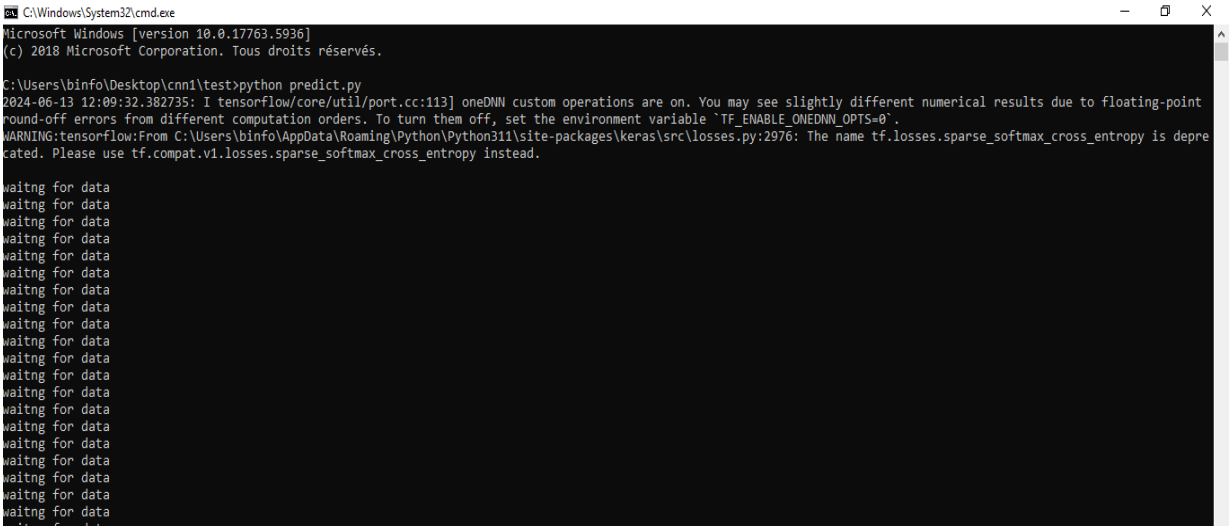


Figure VI.16:model waiting for data

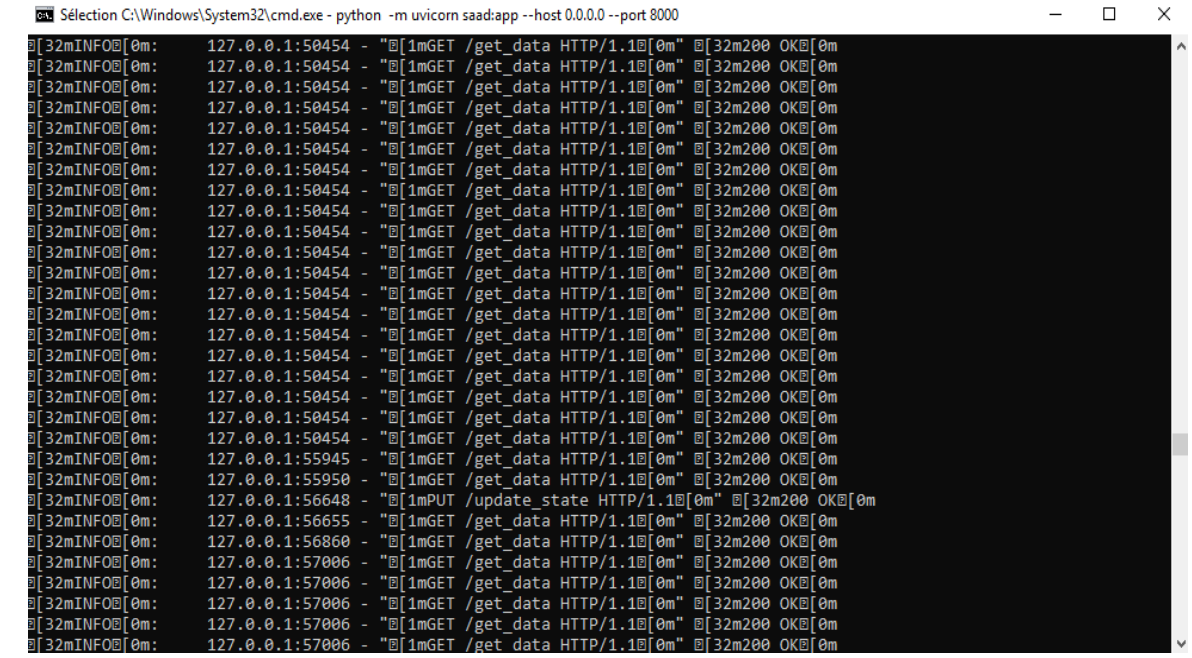


Figure VI.17:API get the data and update the state after the prediction



Figure VI.18:loaded model predict state successfully

VI.6 node red test

In this Node-RED configuration, we will establish an MQTT input to receive sensor data. An API request node will be implemented to periodically fetch new data for real-time ECG graph plotting. Following data acquisition, the prediction code will be triggered to classify the ECG signal. The results will then be utilized to dynamically generate the ECG graph and control the color of an LED based on the classification outcome as you see in Figure VI.19

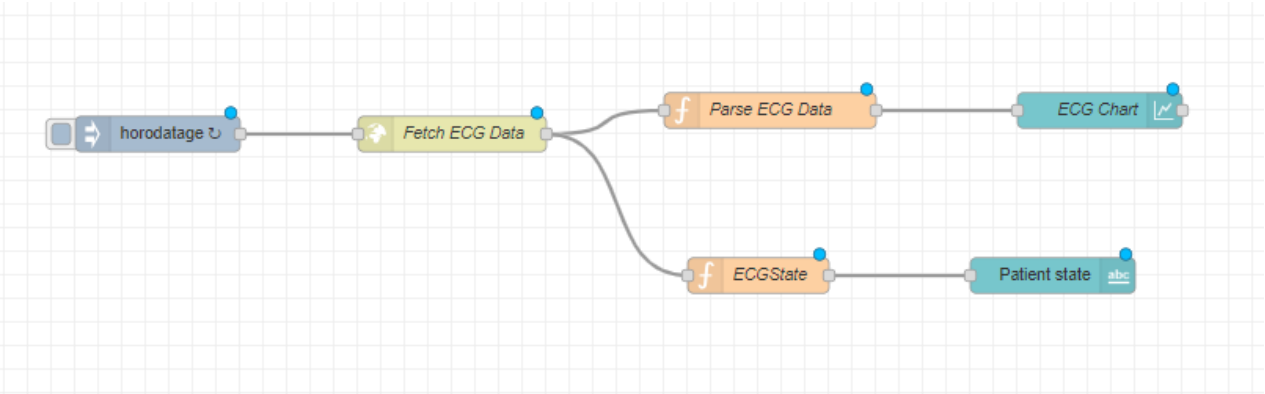


Figure VI.19:node red flow

After deploying the Node-RED flow, we see an interface with the ECG chart and the patient's state displayed above the chart, as shown in Figure VI.4

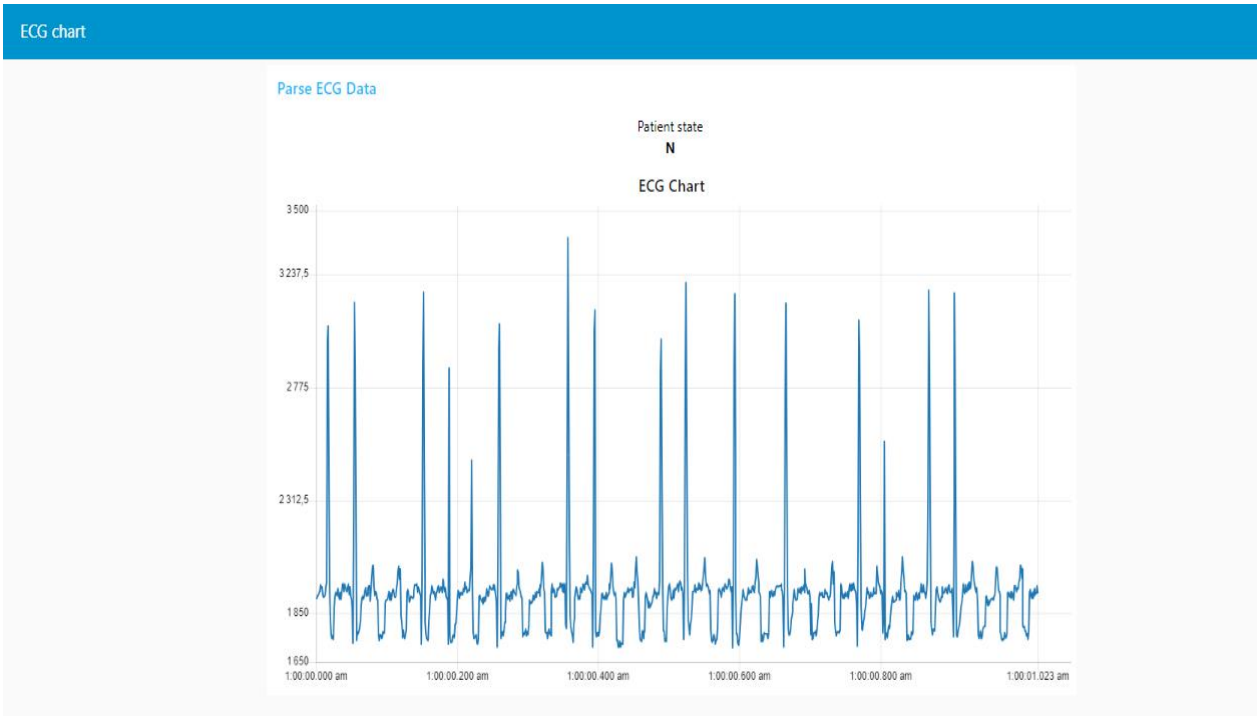


Figure VI.20:node red interface

VI.7 Conclusion

Chapter IV presents the practical development and testing of an intelligent IoT-based system for ECG signal analysis and medical diagnostic assistance. The project focused on evaluating three neural network models—MLP, CNN, and CNN-LSTM—to identify the most effective in diagnosing cardiovascular diseases.

Data collection involved using the MIT-BIH Arrhythmia database, followed by preprocessing to prepare the ECG signals for model training. Among the models tested, the CNN-LSTM hybrid outperformed others, effectively handling spatial and temporal features of ECG data.

The implementation included setting up an ESP32 microcontroller with an AD8232 ECG sensor, using MQTT for data transmission, and an API to manage predictions and storage. A Node-RED interface provided real-time visualization of ECG data and patient states.

The CNN-LSTM model achieved the highest accuracy and reliability, confirming its suitability for ECG classification. This chapter demonstrates the successful integration of advanced neural networks and IoT for enhanced cardiac monitoring, setting a solid foundation for future advancements in medical diagnostics.

General conclusion

General conclusion

The project "Development of an Intelligent IoT-Based System for Medical Diagnostic Assistance" showcases the successful integration of modern technologies to advance medical diagnostics, particularly in cardiac monitoring. By utilizing IoT, machine learning, and real-time data processing, this system provides a robust solution for patient monitoring and diagnostic support. The core of the system involves an ESP32 microcontroller paired with an AD8232 ECG sensor to capture patient ECG signals, which are then transmitted for analysis via the MQTT protocol to a central server housing a pre-trained model for real-time predictions and analyses.

Throughout the project, three neural network architectures—MLP, CNN, and a hybrid CNN-LSTM—were assessed, with the CNN-LSTM model proving to be the most accurate and efficient for classifying ECG signals and diagnosing cardiac conditions. The system's API manages incoming data, stores it in a MySQL database, and oversees the prediction process, ensuring efficient data processing and timely predictions based on set thresholds.

The implementation of a Node-RED interface allows for real-time visualization of ECG data and patient states, offering healthcare providers an intuitive tool for monitoring patient data and receiving alerts based on the model's classifications, thereby enhancing patient care. The project's objective was to develop an intelligent system capable of systematically detecting and classifying cardiovascular diseases using CNN networks, with a focus on the segmentation of ECG signals to enhance classification accuracy and database expansion.

This innovative system not only addresses potential inaccuracies in ECG interpretation by non-specialized practitioners but also serves as a prototype that can be enhanced to detect a wider range of diseases beyond cardiovascular conditions. By expanding its capabilities, the system aims to significantly contribute to saving lives through early and accurate disease detection, promising a transformative impact on healthcare for a better future.

Bibliographies

Bibliographies

1. Organisation mondiale de la santé. (2021). Cardiovascular diseases (CVDs). Disponible sur: https://www.who.int/fr/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds), consulté le 29 mai 2024.
2. Cabrol, C., R. Vialle, and H. Guérin-Surville, *Anatomie du cœur humain.* Niveau PCEM2-EIA cardiologie. Université Pierre et Marie Curie, 2002.
3. TALBI Mohamed Lamine, 2011: «Analyse et traitement du signal électrocardiographique (ECG)». Thèse de doctorat, université de Constantine Algérie
4. Nadia KRICHANE et Samira TAZEBOUDJT,2015: «Classification des signaux ECG par les réseaux de neurone probabiliste» .Mémoire de Fin d'Etudes De MASTER ACADEMIQUE. UNIVERSITE MOULOUD MAMMERIDETIZI-OUZOU.
5. Berkani Mohamed Rafik et Aymene Achir Zahra.2022: «Etude et implémentation d'un système connecté intelligent pour la surveillance de l'état de santé des patients souffrant d'arythmie cardiaque. » Mémoire de Fin d'Etudes De MASTER. UNIVERSITE YAHIA FARES DE MEDEA
6. NAIT-HAMOUD Mohamed Cherif. 2010 «Segmentation et Classification du Signal ECG Par les SVM Flous Multi-classes». Magistère. Tébessa.
7. LAMANI Meriem & HALITIM Rania.2020: «Extraction des caractéristiques et classification des signaux ECG». Mémoire de Fin d'Etudes De MASTER. Université Mohamed Boudiaf - M'sila.
8. AMRI MOHAMED CHAKIB, Développement et réalisation d'un électrocardiographe ECG, Mémoire master 2017, Université M'Hamed Bougera-Boumerdes
9. DAHOU NIHAD et KHADIR SAMIA . 2023: «Extraction automatique du complexe QRS à partir du signal ECG par les ondelettes en vue d'une identification directe des extrasystoles. » Mémoire de Fin d'Etudes De MASTER. UNIVERSITE M'HAMED BOUGARA-BOUMERDES.
10. BEREKSI REDUIG Fethi « conception et réalisation d'un dispositif d'exploration fonctionnelle cardiovasculaire » Université Abou Baker Belkaid Tlemcen, Année 2011-2013 projet national de recherche
11. Salim GRAJA «Segmentation et classification de l'onde P d'un électrocardiogramme : détection d'un risque de fibrillation auriculaire». Thèse de doctorat. l'école nationale supérieure des télécommunications de Bretagne. 03 Avril 2008.
12. BOUKHEDIMI Lylia et ZIANE Hanane.2020. «Classification des signaux biomédicaux par la méthode Neuro-floue d'ANFIS» Mémoire de Fin d'Etudes De MASTER. UNIVERSITE M'HAMED BOUGARA-BOUMERDES
13. HAMITOUCHE AREZKI.2012. «Application des réseaux de neurones probabilistes et la transformée en ondelettes à la détection de la fibrillation auriculaire» .Mémoire de Fin d'Etudes De MASTER. UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU

14. GUEDOUARI Mouhamed.Nadjib et BOUANEM Karim « Etude et réalisation d'un Electrocardiogramme » Université Mhamed Bougra Boumerdas, Année 2013, Master Imagerie et Appareillage Biomédical
15. BENALI Radhwane: Analyse du signal ECG par réseau adaptif d'ondelettes en vue de la reconnaissance de pathologies cardiaques.Thèse de doctorat. Université Abou Bekr Belkaid. Avril 2013.
16. Karima CHEBLI et Lilia LAOUFI: Conception et réalisation d'un ECG à deux électrodes à base d'une carte Arduino UNO . Mémoire de Fin d'Etudes De MASTER ACADEMIQUE.2016. UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU.
17. KHADDOUMI Balkine, 2005 : « analyse et modélisation d'électrocardiogrammes dans le cas de pathologies ventriculaires ».thèse doctorat, université de Nice-Sophia Antipolis.
18. http: //sofia.medicalistes.org/spip/IMG/pdf/La fibrillation ventriculaire nouveaux defibril lateurs.pdf
19. Chahada et Brahim. Analyse des données Electrocardiogramme par un systéme d'aide au diagnostic. Mémoire de Master. Université Abdelhamid Ibn Badis-Mostaganem.2016
20. Messioud . M .L . (2019) .Classification des signaux ECG en utilisant les réseaux de Neurones , Université 8 Mai 1945 – Guelma
21. Rémi DUBOIS. Application des nouvelles méthodes d'apprentissage à la détection précoce d'anomalies en électrocardiographie. L'UNIVERSITÉ PARIS 6
22. TRIQUI Bohra. Classification des arythmies cardiaques par la carte TSOM (Temporal Self Organizing Map), Doctorat, Oran, 2017-2018.
23. R. Geetha and D. Bhanu Sree Reddy, "Recruitment through artificial intelligence: A conceptual study," Int. J. Mech. Eng. Technol., vol. 9, no. 7, pp. 63–70, 2018.
24. Abubakar, A., et al. (2021). "Review of Artificial Intelligence-Based Failure Detection and Diagnosis Methods for Solar Photovoltaic Systems." Machines 9(12)
25. «Deep Learning with TensorFlow [Book]»https://www.oreilly.com/library/view/deeplearning-with/9781786469786/ (consulté le 15 mai 2024).
26. M. Saeed, Z. Al Aghbari, and M. Alsharidah. Big data clustering techniques based on spark.
27. P.K. Simpson. « Artificial Neural Systems », Pergmon Press Elmsford, New York, 1989.
28. Dreyfus G., Martinez J-M., Samuelides M., Gordon M.B., Badran F., Thiria SS., Hérault L, Réseaux de neurones : Méthodologie et Applications, Eyrolles, 2002.
29. alexandre, « Recurrent Neural Networks: The complete Guide », Data Science Courses | DataScientest, 19 janvier 2023. https://datascientest.com/en/recurrent-neural-network-whatis-it (consulté le 21 mai 2023).
30. N. BENAHMED, «optimisation de réseaux de neurones pour la reconnaissance de chiffres manuscrits isolés : sélection et pondération des primitives par algorithmes génétiques,» Université du Québec, Montréal, 2002.

31. M. NOURESSADAT, «Etude des performances des réseaux de neurones dynamiques à représenter des systèmes réels :une approche dans l'espace d'état,» These de magister, Université de Setif, 2009.
32. M. Mandal, "Introduction to Convolutional Neural Networks (CNN)," May 2021.
33. M. Mayank, "Convolutional Neural Networks, Explained | by Mayank Mishra | Towards Data Science," Aug. 2020.
34. Q. Sellat, S. K. Bisoy, and R. Priyadarshini, "Semantic segmentation for self-driving cars using deep learning: a survey," in Cognitive Big Data Intelligence with a Metaheuristic Approach, pp. 211–238, Elsevier, 2022.
35. A. Ajit, K. Acharya, and A. Samanta, "A review of convolutional neural networks," in 2020 international conference on emerging trends in information technology and engineering (ic-ETITE), pp. 1–5, IEEE, 2020
36. Chouiha Boudjemaa Abdelkrim. "Accurate Electricity Consumption Prediction Using Deep Learning". Master Academic . Université KASDI MERBAH OUARGLA. 2023
37. SOLTANI Khadidja. Etude d'un Système Intelligent Pour Suivre les Patients Atteints de Maladies Cardiaques. Mémoire de fin d'études de Master. UNIVERSITE ABDELHAMID IBN BADIS DE MOSTAGANEM.2023
38. https://www.researchgate.net/publication/335233372_STUDY_OF_INTERNET_OF_THINGS_AND_DEVELOPMENT_TOOLS_AND_TECHNOLOGY
39. Tati Amani, BenSaci Cheyma. Simulation and evaluation of an IOT communication protocol. Mémoire de fin d'études de Master. UNIVERSITE KASDI MERBAH OUARGLA .2021
40. Data Flair. How IoT Works. Disponible sur : https://data-flair.training/blogs/how-iot-works/ (consulté le 20 mai 2024).
41. IBM. What Is an API (Application Programming Interface)?. Récupéré sur https://www.ibm.com/topics/api (consulté le 20 mai 2024).
42. GeeksforGeeks. (s.d.). FastAPI - Introduction. Récupéré sur https://www.geeksforgeeks.org/fastapi-introduction/ (consulté le 22 mai 2024).
43. EMQ. (s.d.). MQTT Broker: How It Works, Popular Options, and Quickstart. Récupéré sur https://www.emqx.com/en/blog/the-ultimate-guide-to-mqtt-broker-comparison (consulté le 22 mai 2024).
44. ARADJ TAHA MOHAMED EL-AMINE. "Usage du protocole MQTT dans une application de suivi". Mémoire de fin d'études de master académique en Informatique. Université Mohamed Khider – BISKRA.2017
45. BENCHOULA Rayane et HANACHI Nour Elhouda." Etude comparative de protocoles de communication dans l'IOT". MEMOIRE DE FIN D'ETUDES. UNIVERSITE KASDI

MERBAH OUARGLA .2020

46. akenza. (s.d.). What is MQTT and why it is important for the Internet of Things. Récupéré sur https://akenza.io/blog/what-is-mqtt (consulté le 20 mai 2024).

47. Khadmallah wafa et Djoudi souhila." Création et analyse de la base de donnée «DESASTER » ,des accidents des plateformes de forage offshore". Mémoire de Fin d'étude MASTER PROFESSIONNEL. UNIVERSITE KASDI MERBAH OUARGLA.2013

48. BOUNEGAB Malika et GHEZAL Fadila." Mise En Œuvre d'Une Base de Données OLAP Pour le Décisionnel". Mémoire MASTER ACADEMIQUE. UNIVERSITE KASDI MERBAH OUARGLA.2016

49. DJEBALI Hadjer et KENOUZE Ilyas. "Un Outil d'administration du Système de Gestion de Base de données Oracle (SGBD Oracle)". Mémoire de fin d'études de Master. UNIVERSITE KASDI MERBAH OUARGLA.2014

50. GeeksforGeeks. (s.d.). What is Python? its Uses and Applications. Récupéré sur https://www.geeksforgeeks.org/what-is-python/(Consulté le 24 mai 2024.)

51. RealPars. (s.d.). What is Node-RED? | How Can I Use it to Create IoT Applications? Récupéré sur https://www.realpars.com/blog/node-red (Consulté le 24 mai 2024).

52. The Engineering Projects. (s.d.). Introduction to Arduino IDE. Consulté le 27 mai 2024. Récupéré sur https://www.theengineeringprojects.com/2018/10/introduction-to-arduino-ide.html.

53. How2electronics. (consulté le 27 mai 2024). AD8232 ECG Sensor & Arduino Interfacing w ECG Graph. Récupéré de https://how2electronics.com/ecg-monitoring-with-ad8232-ecg-sensor-arduino/

54. Built In. (consulté le 27 mai 2024). What Is TensorFlow? (Definition, Python Use, Difficulty). Récupéré de https://builtin.com/data-science/tensorflow

55. TensorFlow Core. (consulté le 28 mai 2024). Keras: The high-level API for TensorFlow. Récupéré de https://keras.io/?ref=phannhatchanh.com

56. GeeksforGeeks. (consulté le 28 mai 2024). Learning Model Building in Scikit-learn. Récupéré de https://www.geeksforgeeks.org/learning-model-building-scikit-learn-python-machine-learning-library/"

58. GeeksforGeeks. (consulté le 28 mai 2024). Introduction to Matplotlib. Récupéré de https://www.geeksforgeeks.org/python-introduction-matplotlib/

59. http://www.physionet.org. (consulté le 26 mai 2024)