A Project Report on

Enhanced Depression Detection using Machine Learning with Real time ECG signals

submitted by

AVINASH MAURYA 2004331021

SNEHA SHUKLA 2004331049

UTKARSH TRIPATHI 2004331054

under the guidance of

DR. ATUL KUMAR DWIVEDI

in partial fulfilment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BUNDELKHAND INSTITUTE OF ENGINEERING AND TECHNOLOGY, JHANSI, INDIA

2023-2024

DECLARATION

We hereby proclaim that the work **Enhanced Depression Detection using Machine Learning with Real time ECG signals** has presented in this report. Information derived from the other sources to complete this work has been accredited in the text and a list of references has been given.

Avinash Maurya	**Sneha Shukla**	**Utkarsh Tripathi**
Roll no. 2004331021	Roll no. 2004331049	Roll no. 2004331054

CERTIFICATE

This is to verify that the project entitled **Enhanced Depression Detection using Machine Learning with Real time ECG signals** submitted by Avinash Maurya, Sneha Shukla, Utkarsh Tripathi for the award of the degree of B.Tech. is carried out by them under my guidance.

Dr. Atul Kumar Dwivedi
Department of ECE, BIET Jhansi

VERIFICATION

This is to verify that the project entitled **"Enhanced Depression Detection using Machine Learning with Real time ECG signals "** was submitted to the department of Electronics and communication engineering department of Bundelkhand Institute of Engineering and Technology, Jhansi.

Dr. Atul Kumar Dwivedi	**Prof. Deepak Nagaria**
Officer in-charge Projects(UG)	Head of Department, ECE, BIET

DEDICATION

To my beloved Parents and almighty

PROJECT OUTCOMES(Ps)

S. N.	**Project outcomes** After completing this project students will be able to	**Bloom's knowledge level**
P1	Apply the knowledge of Machine Learning,at various places depending upon the need.	KL3
P2	Design the Model for detection of depression by testing and training the model.	KL6
P3	Evaluate the models of Depression using accuracy and confusion matrix	KL5

KL: Bloom's knowledge level, KL1: remember, KL2: Understand, KL3: Apply, KL4: Analyse, KL5: Evaluate, KL6: Create/Design

Mapping of project outcomes with Program Outcomes (POs)

S. N.	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO 1**	**PSO 2**
P1	3	3	2	3	2	3	3	3	-	3	3	2	3	3
P2	3	3	3	3	2	3	3	2	-	3	3	3	3	3
P3	3	3	2	3	2	3	3	3	-	3	3	2	3	3

Mapping rules (Rubrics)

1: poor   2: medium   3: best

Mapping of Program Outcomes(POs) with Project

S. N.	**Program outcomes** After completing this project students will be able to	Map
PO1	**Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	3
PO2	**Problem Analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences..	2
PO3	**Design/development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	3
PO4	**Conduct Investigations of Complex Problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	2
PO5	**Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern Engineering and IT tools including prediction and modeling to complex Engineering activities with an understanding of the limitations	3
PO6	**The Engineer and Society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.	2
PO7	**Environment and Sustainability:** Understand the impact of the professional Engineering solutions in societal and Environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	2

PO8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the Engineering practice.	3
PO9	**Individual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	3
PO10	**Communication:** Communicate effectively on complex Engineering activities with the Engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	2
PO11	**Project Management and Finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	-
PO12	**Life-long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change	2
PSO 1	Apply the fundamental concepts of Electronics and Communication Engineering to design a variety of components and systems for applications like signal processing, Image processing, Communication, Networking, Microcontrollers, embedded system, VLSI, Control system.	3
PSO 2	Apply the concepts learned in Electronics, Signal Processing, Image processing, Communication, Networking, Microcontrollers, embedded system, VLSI, Control system to arrive at solutions to real-world problems.	3

Mapping rules (Rubrics) : 1- poor, 2- medium, 3- best

ABSTRACT

KEYWORDS: Machine Learning, Models, Algorithms, Naive Bayes, NLP, Ubidots, CNN, ECG

The prevalence of depression as a global mental health disorder underscores the urgency for early detection and intervention to prevent its devastating consequences, including suicide. This project addresses this critical need through a novel approach leveraging multimodel sensing and machine learning techniques, notably the Naive Bayes algorithm and Natural Language Processing (NLP). By integrating speech-to-text technology and ECG analysis, users are prompted to respond to questions assessing their mental state. The collected data is analyzed using the Naive Bayes algorithm to predict the likelihood of depression. Additionally, a hardware component employing an ESP8266 and AD8232 ECG sensor monitors users heart rate, transmitting data to the cloud for remote assessment by medical professionals. Concurrently, the project tackles the challenge of accurately detecting emotions in speech through an Automated Speech Emotion Recognition system utilizing Convolutional Neural Network (CNN) algorithms. By combining these methodologies, the project aims to provide a comprehensive solution for depression detection and intervention, ultimately improving patient outcomes and alleviating the burden on healthcare systems. Additionally, the incorporation of sentiment analysis into the speech emotion recognition framework represents a significant advancement, offering valuable insights into individuals mental health states and enhancing depression detection methods.

ACKNOWLEDGEMENT

We would like to express our deep sense of gratitude to our supervisor and guide Dr. Atul Kumar Dwivedi, Department of Electronics and Communication Engineering, BIET, Jhansi, for his constant guidance and motivation during the project work from last one year. We really appreciate and value their guidance and constant motivation from the beginning of the project. Without his guidance and persistent help this project would not have been possible. We are also indebted to Professor Deepak Nagaria, Head Department of Electronics and Communications Engineering for his encouraging co-operation and continued interest in shaping this project. We express our sincere gratitude to Prof. Deependra Singh, Director, BIET, Jhansi, who has been instrumental in providing an ideal environment for wholesome individual development and appreciating talents in both academic and extracurricular activities. We extend our sincere thanks to all faculties of the Department of Electronics and communication engineering, BIET, Jhansi, for their focused guidance and encouragement. We would like to thank our parents and friends and all those people who have supported me directly or indirectly to complete this project work.

Avinash Maurya	**Sneha Shukla**	**Utkarsh Tripathi**
Roll no. 2004331021	Roll no. 2004331049	Roll no. 2004331054

CONTRIBUTIONS

The main objective of this project is to predict the depression level of a person using speech and ECG signals. The main contributions of the work are

1 Machine Learning has been used for depression level prediction.

2 The Prediction is done using various model training and testing in machine learning.

3 Prediction has been done by converting speech to text and ML techniques Naive Bayes and Natural Language processing.

4 Detection of Depression through Speech Sentiment Recognition with the help of Convolution Neural Network.

5 ESP8266 and AD8232 ECG integrated on hardware component to monitor and transmit data for assessment by professions.

MEMBERS CONTRIBUTIONS

Although every member of the group equally contributed for the work, the division in their contributions can as below

Member 1 Conceptualization, Investigation, Methodology

Member 2 Data curation, Validation, Formal analysis ,Visualization

Member 3 Resources, Writing – original draft, Writing – review and editing.

Supervisor Conceptualization, Project Administration, supervision, review of final report draft.

ABBREVIATIONS AND SYMBOLS

ML	Machine Learning
HRV	Heart Rate Variability
ECG	Electrocardiography
NLP	Natural Language Processing
CNN	Convolutional Neural Network
DICOM	Digital Imaging and Communication in Medicine
API	Application Programming Interface
ANN	Artificial Neural Network
SVM	Support Vector Machine
RNN	Recurrent neural networks
MLP	Multilayer perceptrons
α	Current gain

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

Table of Contents (continued) **Page**

CHAPTER 1

INTRODUCTION

1.1 Introduction

Depression, or major depressive disorder (MDD), is a mental disorder characterized by feelings of sadness, emptiness, or hopelessness [?]. It's not just a passing mood or feeling down for a day or two; it's a prolonged state that significantly impacts an individual's thoughts, emotions, and behaviors.
Depression, a pervasive mental health condition, manifests through a multitude of symptoms across emotional, physical, cognitive, and behavioral domains. Emotionally, individuals may experience Persistent depression, hopelessness and lack of interest in previously enjoyed activities alongside irritability and worthlessness. Physically, depression may induce changes in appetite and sleep patterns, resulting in fatigue and bodily discomfort. Cognitive symptoms encompass negative thinking patterns, difficulty concentrating, and memory problems, while behavioral changes include social withdrawal, decreased productivity, and neglect of personal responsibilities. These symptoms collectively impact various aspects of functioning, leading to occupational, interpersonal, and academic challenges. Fortunately, depression is treatable through a combination of psychotherapy, medication, and lifestyle modifications. Psychotherapy techniques like cognitive-behavioral therapy help address negative thought patterns, while antidepressant medications regulate brain chemistry. Lifestyle changes, such as regular exercise and social support, contribute to overall well-being, offering hope for individuals grappling with depression to regain control over their lives.

It is important for people with symptoms of depression to seek help from a mental health professional. With the right treatment and support, many people with depression can benefit from their symptoms and regain control of their lives.

1.2 Methods of Depression Detection

Depression detection refers to the process of identifying symptoms and signs of depression in individuals. This can involve various methods, including self-assessment questionnaires, clinical interviews conducted by healthcare professionals, observation of behavioral patterns, and the use of technology such as machine learning algorithms [?] analyzing data from social media or smartphone usage. Conventionally, depression detection was done through extensive clinical interviews, where psychologists studied the subject's responses to determine their mental state[?].

In more recent approaches, multi-modal data fusion has been used. For example, a model fuses word context, audio, and video modalities to predict mental health outcomes [? ?]. However, video analysis is computationally very intensive and slow. A voice based depression level analysis based on response analysis for the questions posed can be quite computationally efficient and reliable approach. Voice analysis can carried out in two approaches. One is to analyze the language spoken [?] and another is to analyze the sentiment of the voice [?]. These approaches have been reported individually but the fusion of these two approaches is not reported. A number of works reported detection of depression using non-verbal signal Electroencephalogram (EEG) sensors [? ?].

Electrocardiography (ECG) has emerged as a potential tool for detecting depression by capturing subtle physiological changes associated with depressive symptoms. Analysis of Cardiac variability, which indicates the state of the autonomic nervous system function, has shown promise in identifying individuals with depression, who often exhibit reduced HRV. Machine learning algorithms trained on ECG data can further enhance depression detection by identifying patterns indicative of depressive states. Additionally, advancements in wearable ECG monitoring enable continuous ambulatory monitoring, facilitating real-time assessment of cardiac activity in relation to depressive episodes.

There are numerous methods for detection of depression such as Questionnaire based models, Machine learning model, Natural language processing model, Neuroimaging-based model, and Mobile wearable technology model. Hybrid models combine multiple data resources and analytical technique to improve depression detection accuracy. For example, integrating questionnaire responses with smartphone usage data or combining its features with NLP[**?**]-derived linguistic features can enhance predictive performance. Each type of model has things it's good at and things it's not so good at. Choosing the right one depends on factors like how much data you have, how accurate you need it to be, and what will work best in real-life situations, like in a clinic or during research.

Deep Learning: Deep learning can be likened to the human nervous system, as it involves the training of neural networks over collections of audio or image data to address various problems. These models, reminiscent the Information from the human brain designed to see and understand information in a manner similar to humans. Each node within a deep learning network functions akin to a neuron in the nervous system, collectively forming an artificial neural network [**?**]. As information passes through layers these networks, algorithms learn progressively deeper insights about the input, mirroring the way human cognition evolves. Initial layers focus on detecting low-level characteristics, such as edges in images, while subsequent layers build upon this foundation to extract more complex and abstract features, ultimately leading to a richer and more nuanced representation of the data. This hierarchical learning process allows deep learning models to develop increasingly sophisticated understandings of the input data, akin to the way the human brain processes information.

Deep learning[**?**], A subset of cognitive skills are good at recognizing patterns in digital Information such as images, sounds and text. In the process of prediction, data is pre-trained and divided into training and testing sets, with known results for evaluation. Deep learning aims to optimize the nodes within neural networks at different levels to make predictions, striving for the most satisfactory outcomes. The process is repeated prediction and optimization to support growth of true machine intelligence. One prominent architecture within deep learning is the Convolution Neural Network (CNN), taking cues from the intricate design and operational principles of the visual cortex found in animals. In a CNN, individual neurons respond to stimuli within specific regions known as receptive fields, with overlapping receptive fields enabling coverage of the entire visual area. The response of a neuron to stimuli within its receptive field is mathematically computed through convolution operations. CNN are designed to minimize the need for preprocessing and have found widespread applications in tasks such as image and video recognition, recommendation systems, and natural language processing (NLP). Their adeptness at emulating natural processes while adeptly handling intricate data renders them a formidable asset across diverse realms of Machine intelligence[**?**].

Purpose of a project using machine learning with real-time ECG signals is to leverage technological advancements to improve the detection, monitoring, and management of depression and related mental health conditions, ultimately enhancing the well-being and quality of life of affected individuals. If not given treatment at the right time or proper medication then it might compel people to do suicide.The primary purpose is often to develop a system capable of accurately detecting signs of depression or related mental health conditions based on real-time ECG signals. By detecting depressive symptoms early, the project seeks to enable timely intervention and treatment, potentially preventing the escalation of depression into more severe conditions. The project may aim to develop a system that can analyze ECG data to provide personalized recommendations or interventions tailored to the individual's specific needs and symptoms. Real-time ECG monitoring[**?**] allows for continuous tracking of an individual's physiological state, providing valuable insights into mood fluctuations and potential triggers for depressive episodes. Integrating the system into wearable devices enables remote monitoring, empowering individuals to actively manage their mental health and providing healthcare professionals with real-time data for decision-making. Beyond immediate clinical applications, the project contributes to ongoing research in understanding the relationship between physiological markers and mental health disorders, leading to advancements in diagnosis, treatment, and prevention. The observations include:

1. Prevalence and Impact of Depression: It affects a significant number of individuals worldwide, highlighting the urgency for effective detection and intervention strategies.

2. Objective: It specifically focuses on leveraging machine learning techniques, notably the Naive Bayes algorithm[**?**], and Natural Language Processing (NLP) to achieve this goal.

3. Utilization of Speech-to-Text Technology: These responses are then analyzed using the Naive Bayes algorithm to ascertain the likelihood of depression.

4. Implementation of Hardware Component: A hardware setup is integrated into the project, comprising an ESP8266 and an AD8232 ECG sensor. This hardware monitors users' heart rates, with the collected data transmitted to the cloud for remote assessment by medical professionals [**?**].

5. Comprehensive Approach: By combining machine learning algorithms with real-time physiological data, the project seeks to offer a holistic solution for depression detection and intervention. The ultimate goal is to enhance patient outcomes and alleviate the strain on healthcare systems through timely and effective interventions.

An innovative approach to audio emotion recognition, leveraging Convolution Neural Networks[**?**] (CNN) in tandem with Mel-frequency cepstral coefficients (MFCC) as the primary features for analysis. The overarching goal of this system is to accurately classify the emotional content conveyed within audio recordings across five distinct categories: anger, calmness, fear, happiness, and sadness. Emotion recognition from audio data holds significant implications across various domains, including but not limited to human-computer interaction, Emotional assessment, psychological care, consumer analysis. Understanding the emotional nuances within speech can enhance the quality of human-computer interfaces by enabling systems to respond appropriately to user emotions. Moreover, in applications related to mental health monitoring, analyzing audio recordings for emotional cues can aid in early detection and intervention for individuals experiencing distress or mood disorders. The choice of CNN and MFCC as key components of the system architecture is both intuitive and effective. CNN, Known for its ability to extract hierarchical features complex data, are well-suited for analyzing spectro-temporal patterns present in audio signals. By leveraging the hierarchical structure of CNN, the model can learn features at different levels of abstraction, thereby enhancing its ability to capture subtle variations in emotional expression. Meanwhile, MFCCs serve as a Compact and data representation of the spectral envelope of the audio signal, particularly suitable for speech and audio processing tasks. By computing MFCCs from audio recordings, the system transforms raw waveform data into a feature space that is conducive to pattern recognition and classification. This transformation enables the model to focus on relevant acoustic characteristics while mitigating the effects of irrelevant noise or variability in recording conditions.

The objective of this project is to utilize the developed Automated Speech Emotion Recognition[?] system to differentiate between individuals who exhibit signs of depression and those who do not based on their speech patterns. By leveraging Convolution Neural Network (CNN) algorithms and employing various emotion recognition modules, the system aims to accurately classify speech samples into categories corresponding to different emotional states, including happiness, surprise, anger, neutral state, and sadness. The project utilizes a dataset of speech samples, with acoustic features extracted using the LIBROSA package, to train and test the system. Through the analysis of these extracted characteristics, the system aims to achieve high classification performance, enabling the identification of emotional states associated with depression in speech signals.

1.3 Machine Learning

Machine learning is the study of computer algorithms that improve with experience.Throughout history, humans have crafted various tools to simplify tasks, showcasing the remarkable creativity of the human mind. These inventions, from machines for travel to those powering industries and computing, have significantly enhanced daily life. One such innovation, Machine Learning (ML), as defined by Arthur Samuel, grants a computer's ability to learn without explicit instructions. Samuel, renowned for his checkers-playing program, laid the foundation for machines to handle data more efficiently. Sometimes, deciphering insights from vast datasets proves challenging for humans alone. This is where machine learning steps in, offering a solution by teaching machines to understand and process data autonomously. With the exponential growth of available datasets, the demand for machine learning continues to soar. Industries across the board are harnessing ML to extract invaluable insights from their data.At the heart of machine learning lies the goal of empowering machines to glean knowledge directly from the data they receive, devoid of explicit human guidance. Mathematicians and programmers are perpetually delving into a myriad of strategies to equip machines with the autonomy to

absorb insights autonomously from vast pools of information.
When delving into machine learning, a diverse range of algorithms come into play. Data scientists emphasize that there is no one-size-fits-all algorithm. Instead, the choice depends on the specific problem at hand, the difference between matching, and the most appropriate model.Some commonly used algorithms include Decision Trees, which act like a flowchart to make decisions based on input features; Support Vector Machines [**?**], One aspect aiding in the classification of data points into distinct categories involves the utilization of algorithms. Additionally, neural networks, drawing inspiration from the intricacies of the human brain, excel in handling intricate pattern recognition tasks. Each algorithm fulfills a specific role, finely tuned to address the intricacies of the problem domain, showcasing the agile and flexible essence of machine learning within our contemporary data-centric environment.
For the past ten years, traditional statistical models have not been performing well in estimating data. They struggle with handling grouped data, dealing with missing information, managing data that is spread out unevenly, and most importantly, they lack the ability to reason through complex data. Due to these shortcomings, more researchers are now turning to machine learning methods, which are more effective. Machine learning uses ideas from many different fields, such as statistics for data analysis, artificial intelligence for creating learning systems, philosophy for understanding knowledge, information theory for managing information, biology for natural learning processes, cognitive science for studying human thinking, computational complexity for understanding computational limits, and control theory for designing decision-making systems. This diverse foundation helps machine learning handle data better and make more accurate predictions.

1.4 Deep Learning

Depression persists as a profound and widespread health issue on a global scale, affecting millions of individuals and presenting formidable obstacles when left unrecognized or unaddressed. At the core of this challenge lies the critical need for promptly identifying those who may be at risk of developing depression or are already grappling with its effects. Presently, many existing methodologies hinge upon self-reported symptoms, which, while commonly utilized, come with inherent limitations. Self-reporting can be unreliable, as individuals may understate or fail to recognize their own symptoms, leading to delayed intervention or inaccurate assessments. Furthermore, relying solely on self-reporting may not capture subtle changes indicative of early-stage depression, thereby impeding timely interventions. As a result, there's a pressing necessity for more effective and nuanced approaches to detect depression, ones that transcend the reliance on self-reported symptoms and leverage a combination of objective indicators and advanced screening techniques. By adopting such comprehensive strategies, healthcare professionals can better identify individuals at risk and provide timely interventions, ultimately mitigating the impact of depression on individuals and societies.

The initiative represents a groundbreaking endeavor poised to transform the landscape of depression detection, offering healthcare professionals an unparalleled tool for comprehensive and precise assessment. It acknowledges the paramount significance of early intervention in addressing mental health conditions, particularly depression, which can exert profound and long-lasting effects on individuals if left untreated. By prioritizing early detection, the project aims to alleviate the strain on healthcare systems, which often face significant challenges in managing the consequences of undiagnosed or untreated depression. Through timely identification and intervention, the initiative aspires to optimize patient care, ensuring that individuals receive the support and treatment they need to navigate their mental health journey effectively. Ultimately, by facilitating early intervention and improving the accuracy of depression detection, the project endeavors to enhance the quality of life for millions affected by depression, offering hope for a brighter and healthier future.

The task of Automated Speech Emotion Recognition presents significant challenges due to the inherent gap between acoustic characteristics and human emotions. This gap is compounded by the variability in emotional expression among individuals, as well as the diverse ways in which emotions manifest in speech, including variations in energy and pitch. Consequently, detecting emotions in speech signals, particularly depression, poses a demanding task in computational vision. The objective of this project is to address this challenge by developing a system based on Convolution Neural Network (CNN) algorithms for automated speech emotion recognition. The system aims to differentiate between depressed and non-depressed individuals by classifying emotions such as happiness, surprise, anger, neutral state, and sadness. Leveraging a dataset of speech samples and acoustic feature extraction using the LIBROSA package, the project seeks to achieve accurate classification performance based on extracted characteristics, ultimately enabling the determination of the emotional state conveyed within speech signals for the identification of depression.

Deep learning is a part of machine learning that works like the human brain to understand and learn from data. It uses artificial neural networks, which are computer programs designed to act like brain cells.Deep learning exhibits exceptional prowess in managing extensive volumes of data encompassing images, text, and audio, leveraging this information to formulate predictions or decisions through learning processes. At the core of deep learning are artificial neural model networks [**?**] (ANN), Deep learning employs artificial neural model comprising interconnected nodes arranged in layers, which consist of an input layer, one or more hidden layers, and an output layer. Within this network, each node, analogous to a neuron, receives input signals, processes them, and transmits the outcomes to the subsequent layer. The connections between these nodes possess weights that dictate their strength. Through a process known as backpropagation, during training, these weights are iteratively adjusted to minimize the disparity between the predicted output and the ground truth, thereby enhancing the model's accuracy.

Deep learning models can be classified into two main types: feed forward neural networks and recurrent neural networks[**?**] (RNNs). Feed forward neural networks, also known as Multilayer perceptrons (MLPs), process data in a sequential manner, passing information from the input layer through the hidden layers to the output layer. RNN, on the other hand, are designed to handle sequential data, such as time-series data or natural language text. They have connections that loop back on themselves, allowing them to capture temporal dependencies in the data. A fundamental characteristic of hierarchical neural networks lies in its capacity to autonomously acquire hierarchical representations of data. Within a deep neural network comprising multiple hidden layers, each stratum progressively discerns abstract features or representations from the input data. Consider an image recognition scenario: the initial layer might discern rudimentary features such as edges and corners, whereas subsequent layers are adept at recognizing intricate patterns like shapes or objects, thereby reflecting the network's hierarchical learning process.

Deep learning has achieved remarkable success in a wide range of applications, including computer vision, natural language processing, speech recognition, and healthcare. It has revolutionized fields such as image classification, object detection, and language translation, often surpassing human-level performance on certain tasks. This is partly due to its ability to automatically learn intricate patterns and representations from vast amounts of data, without the need for explicit feature engineering. However, deep learning also has its Restraints. Training deep neural networks requires large amounts of labeled data and significant computational resources. Deep learning models can also be prone to more train it where they perform well on the training data but generalize poorly to unseen data. Additionally, interpreting and explaining the decisions made by deep learning models can be challenging, leading to concerns about their transparency and accountability. Overall, deep learning represents a powerful and versatile approach to machine learning, with the potential to drive advancements in artificial intelligence and transform various industries and domains. As research in this field continues to progress, deep learning is expected to play an increasingly significant role in solving complex real-world problems and unlocking new opportunities for innovation and discovery.

1.5 Natural Language Processing

Machine learning models represent sophisticated computational systems designed to extract intricate patterns and relationships inherent within datasets. Their primary function is to analyze data and derive insights that enable them to make informed predictions or decisions when confronted with new, previously unseen data. In the realm of Linguistic processing, the models are finely crafted to decipher and interpret human language, engaging in a wide spectrum of tasks including emotional analysis, translation, text classification, and more. By undergoing rigorous training on extensive collections of textual data, NLP models grasp the intricacies of language structure, semantics, and context. This empowers them to effectively grasp and analyze human-generated text with precision and accuracy.

Similarly, in the domain of image recognition, machine learning models are trained to identify and categorize objects depicted within images. This process involves exposing the model to a large dataset comprising pairs of input images and corresponding labels or categories. By analyzing these input-output pairs, the model learns to recognize distinctive features and patterns associated with various objects, thereby developing the ability to accurately classify objects within images. Through iterative exposure to diverse examples during the training phase, the model refines its understanding of visual concepts and generalizes its knowledge to effectively classify objects in new, unseen images.

Fundamental to the training The machine learning process is concept of learning from examples. By systematically exposing the model to a diverse range of input data and associated outcomes, it refines its internal parameters and algorithms to better capture underlying patterns and relationships in literature. This creates the pattern to generalize its knowledge and effectively apply it to make accurate predictions or classifications when presented with novel data instances. In essence, machine learning models represent powerful tools for extracting insights from complex datasets and facilitating automated decision-making across various domains, ranging from natural language understanding to image recognition.

The Naive Bayes[**?**] method stands out as a widely used algorithm in training machine learning models, belonging to the family of probabilistic classifiers rooted in Bayes theorem[**?**]. At its core, Bayes' theorem provides a framework for calculating conditional probabilities, enabling the assessment of the likelihood of a hypothesis given observed evidence. What distinguishes the Naive Bayes approach is its assumption of feature independence given the class variable, which simplifies the computational process but may not always reflect real-world dependencies accurately. This simplification earns the algorithm its "naive" moniker. Particularly in the domain of natural language processing (NLP), the Multinomial Naive Bayes algorithm holds prominence due to its effectiveness in handling tasks such as text classification and sentiment analysis.

In practice, the Multinomial Naive Bayes algorithm employs probabilistic calculations to address these NLP tasks by determining the most probable class or tag for a given input data instance. Despite its inherent simplifications, the Naive Bayes algorithm is widely favored for its simplicity, computational efficiency, and effectiveness across various classification tasks. Its straightforward implementation and relatively low computational requirements make it particularly appealing for applications where real-time or resource-constrained processing is essential. Consequently, the Naive Bayes algorithm has emerged as a cornerstone in the realm of machine learning, playing a pivotal role, especially in NLP applications where it continues to demonstrate its utility and versatility in handling diverse text-based tasks with admirable accuracy and efficiency.

In practical terms, the Naive Bayes algorithm functions by evaluating the likelihood of a specific class or label given the observed features present in the data. This evaluation is carried out by calculating the probability of each class or label occurring, given the values of the features, under the assumption of feature independence. This assumption simplifies the computation of these probabilities, making it feasible to estimate them even with large datasets. In the context of natural language processing (NLP) tasks, such as text classification, the Naive Bayes algorithm computes the probability of a particular class or category given the words or features present in the input text. Despite its simplistic assumption of feature independence, Naive Bayes often delivers surprisingly accurate results in practice, particularly in scenarios where the independence assumption holds reasonably well or when working with relatively small datasets.

One of the key advantages of Naive Bayes lies in its ease of implementation and computational efficiency. The algorithm's straightforward approach makes it relatively simple to understand and apply, even for those new to machine learning. Additionally, its computational efficiency makes it well-suited for handling large volumes of data, making it a practical choice for real-world applications. Moreover, Naive Bayes tends to perform admirably well in situations where the data exhibits a clear separation between classes or when the features are largely independent of each other. These favorable characteristics contribute to its widespread adoption across various domains, particularly within NLP, where it has proven to be effective in tasks such as text classification, spam detection, and sentiment analysis.

Despite its inherent simplicity and the oversimplified assumptions it operates under, Naive Bayes remains a valuable tool in the machine learning toolkit. Its robust performance in many practical scenarios, coupled with its ease of implementation and computational efficiency, make it a popular choice for a wide range of classification tasks. As such, Naive Bayes continues to be extensively utilized in machine learning workflows, contributing to the advancement of various applications, particularly within the realm of natural language processing.

The provided content outlines the foundational concepts and practical applications of machine learning, particularly focusing on natural language processing (NLP) and the Naive Bayes algorithm. Machine learning models are depicted as advanced computational systems adept at discerning intricate patterns and relationships within datasets, thereby facilitating informed predictions or decisions when presented with new data. In NLP, these models are tailored to comprehend and interpret human language, engaging in tasks such as sentiment analysis and language translation. Similarly, in image recognition, machine learning models are trained to identify and categorize objects within images, leveraging extensive datasets to refine their understanding of visual concepts.

The Naive Bayes algorithm is highlighted as a prevalent and effective approach within machine learning, particularly in NLP tasks. It operates by computing the probability of a given class or label based on observed features, assuming independence between features to simplify computations. Despite its simplistic assumptions, the Naive Bayes algorithm is favored for its simplicity, efficiency, and effectiveness across various classification tasks, especially when dealing with text-based data. Its ease of implementation and computational efficiency make it a practical choice for real-world applications, contributing to its widespread adoption in domains such as NLP, where it excels in tasks such as classifying text and evaluating opinions.

Overall, the content underscores the significance of machine learning in extracting insights from complex datasets and automating decision-making processes across diverse domains.

CHAPTER 2

LITERATURE REVIEW

Recent attention in research has been drawn towards mining health-related information from social media platforms like Twitter, Facebook, and Reddit. There has been a concerted effort to streamline methods for identifying health data within social media, leading to significant advancements in various subdomains such as pharmacology, disease surveillance, mental health, and substance abuse monitoring [?]. Additionally, A number of researchers concentrated on discerning the primary factors influencing suicide rates in specific regions of India [?]. In this work depression is analysed using audio, text and ECG signals. The related literature can be summarized as below:.

2.1 Review of Speech-based Depression Detection

The literature review for our project report Give an overview of the methodology and findings in the domain of mental health prediction and analysis through text and social media data. Researchers like Elvis Saravia, Peter Burmap, and others have leveraged Methods such as Term Frequency-Inverse Document Frequency (TF-IDF) and Pattern of Life Features (PLF) to identify repetitive terms used by patients, aiding in emotion and action prediction. Learning-based analyses by J Ang L et al. explore sentence structures and human temperaments, employing Support Vector Machines (SVM) for understanding connections. Min Hang Aung et al. and S. S. K. Jandhyala et al. utilize supervised learning techniques and multiple execution metrics to assess human actions and classifier efficiency, respectively.
Yoshihiko et al. introduce the "Utsureko" application, utilizing Deep Learning to anticipate depression levels with high precision based on user data. Eric Gilbert et al. predict tie intensity in social networks using text analysis and web augmentation techniques, while S. C. Guntuku et al. investigate the impact of nighttime social media usage on

sleep quality and depression levels, particularly among youths. Moreover, Quan Hu et al. utilize Chinese text analysis software for depression prediction on Sina Weibo data, demonstrating the practicality of social media for mental health assessment. Munmun De Choudhury et al. and Keumhee Kang et al. employ crowdsourcing and crawling methods to retrieve Twitter data for SVM classifier development and analysis. Additionally, Maryam Mohammed Aldarwish and Hafiz Farooq Ahmed propose a web application for categorizing online media users into different levels of depression using data from Facebook and Twitter. These studies collectively highlight the significance of text and social media analysis in mental health prediction and emphasize the potential of machine learning techniques in improving public well-being, especially among students and working professionals. The literature underscores the importance of understanding user behaviors on social media platforms for early detection and intervention of mental health disorders.

2.2 Review of Audio based Depression Detection

Detecting Subtle Signs of Depression with Automated Speech Analysis in a Non-Clinical Sample [?]. Researchers analyzed speech features in healthy young adults. Even in non-clinical samples, changes in speech related to higher depression scores were observed. Investigating whether these speech features can serve as early markers for subsequent depression in individuals at risk is recommended. Deep Learning for Depression Detection from Textual Data [?], A productive model Using short-term memory (LSTM) and recurrent neural network (RNN) was proposed. The model achieved 99.0% accuracy in predicting depression from text, outperforming frequency-based models. Early diagnosis using this approach could reduce the number of affected individuals. A Review has been reported on Speech Recognition-Based Prediction for Mental Health and Depression [?]. Speech processing can predict mental health-related problems. The literature survey conducted for this project encompasses a diverse range of approaches and methodologies in the field of speech emotion recognition. Peng Song et al. introduce the Transfer Linear Subspace Learning (TLSL) framework, which aims to enhance cross-corpus recognition of speech emotions by extracting robust characteristic representations across different datasets. Their work demonstrates that TLSL outperforms baseline techniques and significantly excels in comparison to early transfer learning methods, providing promising results for speech emotion recognition [?]. Collectively, these studies contribute valuable insights and methodologies to the field of speech emotion recognition, addressing challenges such as cross-corpus recognition, unsupervised learning, automatic assessment of language impairments, spectral regression modeling, and depression detection using heterogeneous token-based systems. Their findings provide a comprehensive foundation for further research and development in the area of speech emotion recognition [?]. The literature review presents two significant contributions in the field of speech emotion recognition and automated assessment of Cantonese-speaking individuals with aphasia. Jun Deng et al. focus on unsupervised learning using automatic encoders to enhance speech emotion recognition, particularly in settings with limited labeled data. Their approach combines generative and unfair training, leveraging partially supervised learning algorithms to incorporate prior knowledge from non-labeled data. Through sequential evaluation on multiple databases, they demonstrate improved recognition performance, showcasing the model's ability to utilize both labeled and non-labeled data effectively. On the other hand, Ying Qin et al. introduce a completely automated assessment system for Cantonese-speaking individuals with aphasia, utilizing text characteristics to detect language impairments. Their methodology, driven by text characteristics and Siamese network analysis, correlates significantly with aphasia severity scores, emphasizing the importance of robust ASR output and the need for larger databases of pathological speech for improved classification. Both studies underscore Potential for advanced machine learning techniques to improve speech assessment for emotion recognition and automated assessment of language impairments, pointing towards future directions in research for improving clinical diagnostics and intervention strategies.

2.3 Review of ECG based Depression Detection

The increasing popularity of remote health monitoring systems in recent years, with a particular focus on heart monitoring. Numerous researchers and authors have contributed to the field, presenting various modifications and advancements in remote heart monitoring systems. These works emphasize real-time observation of patients' heart conditions and other physiological parameters, utilizing a range of sensors and communication technologies such as ECG, pulse, temperature, infrared sensors, Bluetooth, GSM, WiFi, and GPS modules. Each system is designed to cater to specific conditions and requirements, reflecting the diverse needs of healthcare applications. Considering the collective contributions from these studies, a remote heart monitoring system employing both pulse and ECG sensors simultaneously emerges, aiming to detect heart diseases based on ECG signals and heart rate measurements. Real-time monitoring stands out as a critical aspect of these systems, underscoring the importance of timely and accurate data acquisition for effective healthcare interventions. ECG-based depression detection reveals a growing interest in leveraging electrocardiogram (ECG) signals as potential biomarkers for detecting depression. Several studies have explored the relationship between cardiac activity and mental health, highlighting the potential of ECG-based approaches in diagnosing depression. For instance, researchers have investigated heart rate variability[**?**] (HRV) patterns and ECG signal characteristics as indicators of mood disorders, including depression. Studies such as those by [insert author names and references] have demonstrated promising results in using ECG features to differentiate between individuals with depression and healthy controls. Additionally, advancements in machine learning techniques have enabled the development of algorithms capable of analyzing ECG signals to predict depressive symptoms with high accuracy. These findings underscore the potential of ECG-based depression detection as a non-invasive and objective tool for early diagnosis and monitoring of depressive disorders, offering new insights into the physiological correlates of mental health conditions. Further research in this area holds promise for improving the understanding and management of depression through innovative ECG-based diagnostic approaches.

CHAPTER 3

METHODOLOGY

3.1 Project Flow

Project flow involves a series of systematic steps from data collection to making predictions. The first stage is Data Set Collection, where relevant data is gathered from a variety of sources to ensure they meet project objectives.Next comes Data Preprocessed, where the collected data is cleaned, transformed, and prepared for analysis. This step involves handling missing values, dealing with outliers, and converting Convert data into a format suitable for machine learning algorithms.

Following data preprocessing[**?**],Data is generally divided into training and testing .The training method is used to train the machine learning model, while the testing method is used to evaluate the performance of the model. The next crucial step is selecting and applying the appropriate Machine Learning Algorithm to the training data. This algorithm could be regression, classification, clustering, or any other method depending on the nature of the problem.

Once the training of the model concludes, the project transitions into the phase of result evaluation. During this stage, the model's efficacy is assessed through the examination of data. Performance metrics including accuracy, exactness, and retrieval are employed to gauge the model's effectiveness when presented with unseen data. Through analysis, a deeper understanding of the model's capabilities, limitations, and potential areas for enhancement is attained.

Finally, the model is ready for Predictions. It is deployed to make predictions on new, unseen data. This could involve predicting future trends, classifying new instances, or clustering new data points. Throughout this project flow, each step—Data Set Collection, Data Preprocessing, Data Partitioning, Machine Learning Algorithm selection, Result Analysis, and Predictions—logically leads to the next, requiring effective communication, meticulous planning, resource management, and flexibility to adapt to any changes that may arise during the project's life.

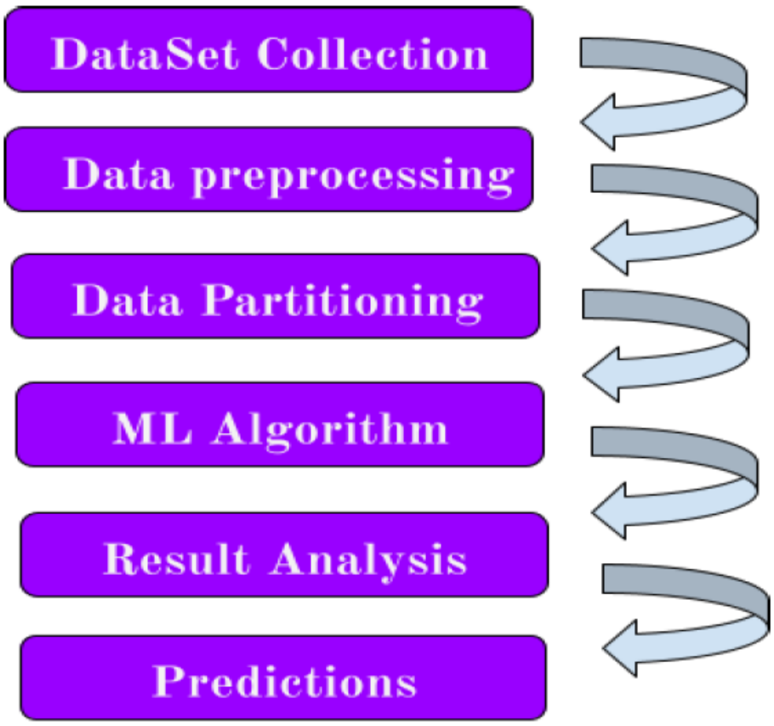


Fig. 3.1: Project Flow

3.2 Dataset Description

The Sentiment 140 dataset contains 1,600,000 tweets collected using the Twitter API. Every tweet in the dataset is annotated its polarity, indicating whether it is classified as depressed (0) or non-depressed (1), making it suitable for sentiment analysis tasks. The dataset consists of six fields:

1. target: This field represents the polarity of the tweet, where 0 indicates a depressed sentiment and 1 indicates a non-depressed sentiment.

2. ids: Each tweet is assigned a unique identifier (ID), facilitating easy reference and retrieval.

3. date: This field records the date and time when the tweet was posted, providing temporal context to the data.

4. flag: In instances where a specific query was used to extract the tweet, this field indicates the query term. If no query was used, the value is marked as NO QUERY.

5. user: The user handle of the individual who posted the tweet is recorded in this field, offering insight into the source of the tweet.

6. text: The text of the tweet is captured in this field, providing the actual content that was posted.

The sentiment140 dataset[**?**]serves it serves as a particularly useful resource for training and reviewing conceptual models. detecting depressed sentiments on social media platforms like Twitter. The dataset's annotations enable researchers to explore patterns and trends in online sentiment, contributing to a deeper understanding of mental health discourse and emotional expression in digital environments. For further details on the dataset's generation and methodology, refer to the official resources provided by the creators, including the dataset link and associated paper.

Table 3.1: Depression Detection Dataset

Target	ids	date	flag	user	text
0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO Query	TheSpecialOne	@switchfoothttp://twitpic.com 2y1zl-Awww,t..
1	1467810672	Mon Apr 06 22:19:49 PDT 2009	No Query	scotthamilton	is upset that he cant update his Facebook by..
2	1467810917	Mon Apr 06 22:19:53 PDT 2009	No Query	mattycus	@Kenichan I divided many times for the ball.Man...
3	1467811184	Mon Apr 06 22:19:57 PDT 2009	No Query	ElleCTF	my whole body feels itchy and like its on fire
4	1467811193	Mon Apr 06 22:19:57 PDT 2009	No Query	Karoli	@nationwideclass no its not behaving at all...

3.3 Project Procedure

The following process is used to implement the ML model of Depression Detection.

Step-1: Import Dependency and Dataset

In the preliminary phase of a data-driven project, the first step entails importing essential libraries and dependencies required for subsequent analyses and model development. These libraries often include popular Python packages such as NumPy is used for numerical calculations, Pandas for data management and analysis, Matplotlib and Seaborn for data visualization, and Scikit-learn for machine learning. Additionally, depending on the project's specific requirements, other specialized libraries might be imported for tasks like deep learning (e.g., TensorFlow or PyTorch) or natural language processing (e.g., NLTK or SpaCy). Furthermore, this step involves loading the dataset that serves as the foundation for all subsequent analyses and modeling endeavors. The dataset typically comprises structured or semi-structured data stored in various formats such as CSV files, Excel sheets, databases, or APIs. It's essential to ensure the dataset aligns closely with the project's objectives, containing relevant features and sufficient observations to facilitate meaningful analyses and model training[**?**]. Moreover, data integrity checks may be performed at this stage to identify any anomalies or inconsistencies in the dataset that might need to be addressed during preprocessing. Overall, this initial step lays the groundwork for the entire project, setting the stage for subsequent data exploration, preprocessing, model building, and evaluation phases.

```
In [ ]: import numpy as np
        import pandas as pd
        from matplotlib import pyplot as py
        from sklearn.model_selection import train_test_split
        from sklearn.naive_bayes import GaussianNB
        from sklearn.metrics import accuracy_score
        from sklearn.feature_extraction.text import TfidfVectorizer
        from sklearn.naive_bayes import MultinomialNB
        from sklearn.pipeline import make_pipeline
        from sklearn.metrics import confusion_matrix
        import seaborn as sns
        from sklearn.metrics import classification_report
        import string
        import nltk
        import re

In [2]: dataset_columns = ["target", "ids", "date", "flag", "user", "text"]
        dataset_encode = "ISO-8859-1"
        df=pd.read_csv("training.1600000.processed.noemoticon.csv", encoding = dataset_e
```

Step-2: Data Preprocessing

Data preprocessing is a crucial phase in any data analysis or machine learning project, where the collected dataset undergoes various transformations and enhancements to ensure its suitability for subsequent analyses and model building. This step As well as many cleaning products refining the data, addressing potential issues and inconsistencies that could hinder accurate analysis or model performance.

One of the main tasks of data preprocessing is to deal with missing values. Real-world data is often incomplete or incomplete due to various reasons, such as incorrect data entry, sensor failure, or some missing data. Strategies for dealing with missing values include imputation (replacing missing values with estimates or values derived from existing data), deleting rows or columns with missing values, or using advanced techniques (such as predictive modeling) to impute missing values. values. Relationships based on data values.

Another important aspect of data preprocessing involves removing duplicates, where identical or near-identical records within the dataset are identified and eliminated to avoid redundancy and ensure data integrity. Duplicate entries can skew analysis results and adversely affect model training, making their removal essential for accurate and reliable results.

Additionally, categorical variables—variables that represent categories often encountered in datasets. These variables need to be encoded into numerical representations suitable for machine learning algorithms, as most algorithms operate on numerical data. Common encoding techniques include one-hot encoding, where each category is represented by a binary (0/1) indicator variable, or label encoding, where categories are replaced with numerical labels.

Moreover, data preprocessing may involve scaling or normalization of numerical features to bring them within a similar range, preventing features with larger magnitudes from dominating the analysis or model training process. Standardization techniques such as z-score normalization or min-max scaling are commonly employed for this purpose.

Overall, data preprocessing plays a critical role in preparing the dataset for subsequent analysis and modeling tasks, ensuring that the data is clean, consistent, and appropriately formatted for effective utilization in machine learning algorithms. By addressing issues such as missing values, duplicates, and categorical variables, data preprocessing lays the foundation for accurate insights and robust model performance, ultimately leading to more informed decision-making and actionable outcomes.

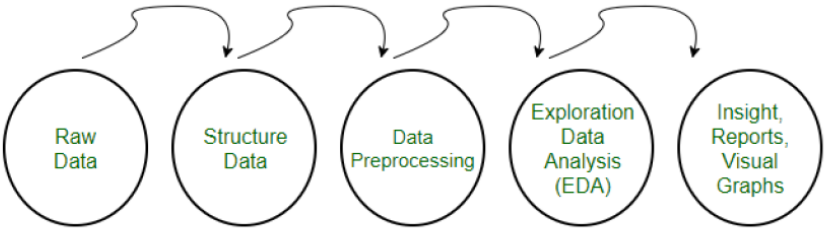


Fig. 3.2: Data Preprocessing

```
#remove punctuation
def remove_punctuation(text):
    no_punct=[words for words in text if words not in string.punctuation]
    words_wo_punct=''.join(no_punct)
    return words_wo_punct
df['clean_text']=df['text'].apply(lambda x: remove_punctuation(x))
df.head()
```

```
new_df = pd.DataFrame()
new_df['text'] = df['clean_text']
new_df['label'] = df['target']
new_df['label'] = new_df['label'].replace(4,1)
```

```
print(new_df.head())
print('Label: \n', new_df['label'].value_counts())
```

Need of Data Pre-processing: Data preprocessing plays a pivotal role in ensuring the effectiveness and accuracy of machine learning models in data-driven projects. One critical aspect is aligning the data format with the requirements of specific machine learning algorithms. For instance, algorithms like Naive Bayes have strict requirements regarding data format, often not supporting null values. Therefore, preprocessing steps such as handling missing values become essential to make the data compatible with such algorithms. Moreover, in complex projects where multiple machine learning and deep learning algorithms are employed, data preprocessing enables seamless execution and comparison across different models. By formatting the dataset appropriately and addressing issues such as outliers, scaling, and feature engineering, data preprocessing enhances the performance and speciality of machine learning models, ultimately facilitating better decision-making and insights extraction from the data.

Step-3: Dataset Partitioning

Dataset partitioning, often referred to as data segmentation, constitutes a pivotal phase in machine acquisition geared towards assessing the model's proficiency and effectiveness. It entails dividing the available data into distinct subsets: the training set and the testing set.

The training process makes the most of the dataset and serves as the basis for teaching the learning model to recognize patterns and relationships in the data. During training, the model learns from the features and labels (if any) included in the training process. By updating its internal parameters or coefficients, the model tries to minimize the difference between its predicted properties and the actual values present in the data.

Conversely, the testing set represents unseen data that the model has not encountered during the training phase. This set acts as an independent measure of the model's performance and generalization ability. through assessing the model's performance on unseen examples, we obtain understanding regarding its capacity to generate precise predictions for fresh, previously unobserved data instances. Testing methods represent real-world conditions and allow us to predict how the model will perform when used in production or when used to make predictions about future papers.

The partitioning of the dataset into training and testing sets is typically performed randomly or through more sophisticated techniques such as cross-validation. The random partitioning ensures that both sets are representative of the overall dataset, reducing the risk of bias in model evaluation. Common practices dictate that a significant portion of the dataset (e.g., 70-80 Percent) is allocated to the training set, with the remainder reserved for the testing set. However, the exact split may vary depending on factors such as dataset size, complexity, and the specific objectives of the project.

In summary, dataset partitioning plays a pivotal role in the machine learning workflow by enabling the assessment of model performance and generalization capability. By segregating the dataset into training and testing sets, we can train the model on one subset and evaluate its performance on another thereby gaining valuable insights into its effectiveness and suitability for real-world applications.

```
from sklearn.model_selection import train_test_split
X = new_df['text']
y = new_df['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.05, random

print(X_train.shape,X_test.shape,y_train.shape,y_test.shape)
```

```
(1520000,) (80000,) (1520000,) (80000,)
```

```
y_train.value_counts()
```

```
0    760001
1    759999
Name: label, dtype: int64
```

Step-4: ML Model

Automated learning, a subset of Cognitive computing, empowers machines to autonomously execute tasks by learning from data samples and examples. Through the utilization of algorithms and statistical analyses, machine learning uncovers intricate patterns and correlations within datasets, often imperceptible to human observers. This capability enables machine learning models to predict outcomes, classify data, or make decisions based on new information. For instance, in natural language processing, these models adeptly comprehend the meaning and context of sentences or phrases. Likewise, in image recognition tasks, machine learning models excel at identifying and categorizing objects depicted in images. During the training phase, a machine learning model is exposed to a vast dataset, enabling it to refine its internal parameters and rules for optimal performance. As the model iteratively learns from the data, its proficiency in accurate predictions or classifications steadily improves. Ultimately, this iterative learning process culminates in the creation of a machine learning model—a sophisticated computer program equipped with tailored rules and data structures, ready to tackle designated tasks with efficiency.

In supervised automated learning, models are trained using labeled data, allowing algorithms to learn patterns from input-output pairs for making predictions or classifications on unseen data. One prevalent use case of guided learning is in image recognition, where classification methods are utilized to assign images to predetermined categories or labels. Similarly, in predicting demographics such as population growth or health metrics, regression techniques are utilized to estimate continuous variables based on input features. One prominent supervised learning algorithm widely used for classification tasks is the Naive Bayes algorithm. Naive Bayes methods are grounded in Bayes' theorem, a fundamental concept in probability theory. The "naive" assumption in Naive Bayes refers to the simplifying assumption of conditional independence between every pair of features given the class variable. In other words, Naive Bayes models assume that the presence of a particular feature in a class is independent of the presence of any other feature, making the calculations simpler and more computationally efficient. Despite this simplification, Naive Bayes classifiers often perform remarkably well in practice, particularly in text classification and spam filtering tasks. By leveraging Bayes' theorem, Naive Bayes algorithms calculate the probability of a class given a set of input features, enabling them to make predictions or classifications with high accuracy and efficiency.

The Multinomial Naive Bayes algorithm is a powerful Bayesian learning technique widely utilized in Natural Language Processing (NLP) tasks. Specifically designed for handling text data, this algorithm excels in tasks such as text classification, sentiment analysis, and spam filtering. In NLP, the program leverages Bayes' theorem to infer the most likely tag or category for a given piece of text, be it an email, a news article, or any other textual content. It accomplishes this by calculating the likelihood of each possible tag given the observed features (words or tokens) in the text. The algorithm then chooses the tag with a higher probability than its prediction.. This approach is particularly valuable in contexts where understanding the context or sentiment of textual content is crucial, such as analyzing customer feedback, detecting spam emails, or categorizing news articles. With the vast amount of text data available across various platforms and sources, text data classification has become increasingly essential for extracting valuable insights and making informed decisions. By accurately classifying text data, Multinomial Naive Bayes enables organizations to gain deeper insights into user perceptions, sentiments, and preferences, thereby facilitating better product development, marketing strategies, and customer engagement initiatives.

```
In [16]: model = make_pipeline(TfidfVectorizer(), MultinomialNB())

In [17]: model.fit(X_train,y_train)

Out[17]: Pipeline(steps=[('tfidfvectorizer', TfidfVectorizer()),
                         ('multinomialnb', MultinomialNB())])
         In a Jupyter environment, please rerun this cell to show the HTML representation or trust
         the notebook.
         On GitHub, the HTML representation is unable to render, please try loading this page with
         nbviewer.org.
```

Step-5: Result Analysis

It's crucial to assess the performance of a machine learning model after training it on data by employing distinct datasets. This evaluation phase is essential for gauging how effectively the model aligns with the data and for pinpointing any potential issues or avenues for enhancement. Utilize a range of performance metrics to scrutinize the model's proficiency in prediction or classification tasks.

These metrics may include accuracy, which measures the proportion of correctly predicted instances out of the total instances; precision, which quantifies the proportion of correctly predicted positive instances out of all instances predicted as positive; recall, which measures the proportion of correctly predicted positive instances out of all actual positive instances; and F1 score, which represents the harmonic mean of precision and recall. Other metrics such as ROC curve and AUC (Area Under the Curve) are also commonly used for evaluating models, particularly in binary classification tasks.By scrutinizing these metrics, you gain insights into the capabilities and limitations of your model. This analysis aids in fine-tuning its performance and enhancing the accuracy of its predictions. Furthermore, this evaluative procedure offers invaluable insights for iteratively refining the model, thereby bolstering its reliability and effectiveness across real-world scenarios.

```
validation = model.predict(X_test)
```

```
validation1 = model.predict(X_train)
```

```
from sklearn.metrics import accuracy_score
accuracy_score(y_train, validation1)
```

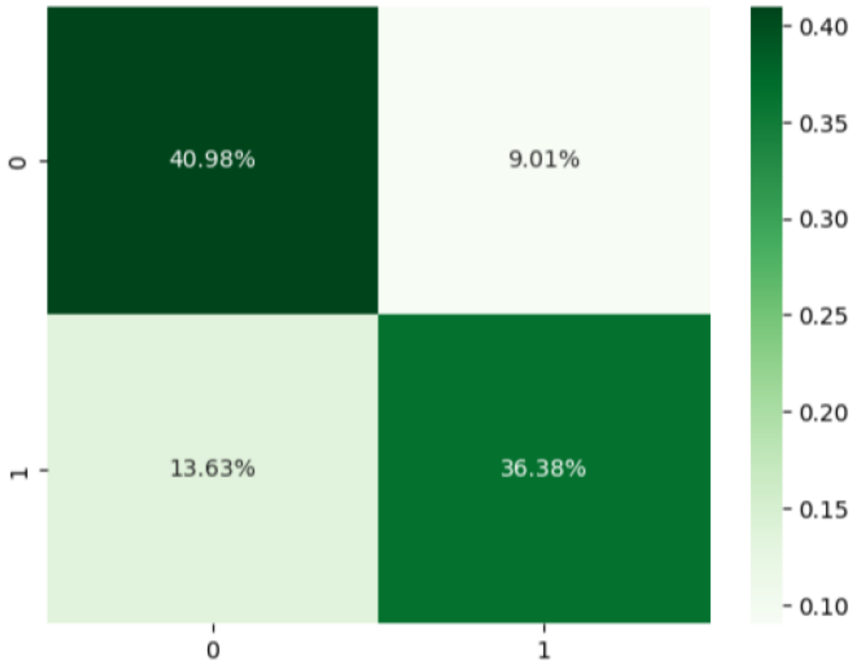
```
0.843316447368421
```

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test, validation)
```

```
0.7736
```

```
cf_matrix = confusion_matrix(y_test, validation)
sns.heatmap(cf_matrix/np.sum(cf_matrix), annot=True, fmt='.2%', cmap='Greens')
```

```
<AxesSubplot: >
```




```
print(classification_report(y_test, validation))
```

```
              precision    recall  f1-score   support

           0       0.75      0.82      0.78     39999
           1       0.80      0.73      0.76     40001

    accuracy                           0.77     80000
   macro avg       0.78      0.77      0.77     80000
weighted avg       0.78      0.77      0.77     80000
```

Step-6: Predictions

In the final step of the machine learning workflow, the trained model is deployed

to make predictions on new, unseen data. This phase represents the culmination of the model development process, where the insights and patterns learned during the training phase are applied to real-world scenarios. Depending on the nature of the problem and the task at hand, the model may be tasked with predicting outcomes, classifying new instances into predefined categories, or clustering new data points based on similarities. Consider a predictive maintenance scenario where the model predicts potential equipment failures by monitoring sensor data for operational anomalies. Alternatively, in a customer churn prediction task, the model distinguishes between customers likely to churn and those likely to remain loyal by analyzing their behavioral patterns and demographic characteristics.
Alternatively, in a customer segmentation analysis, the model has the capability to categorize incoming customers into distinct segments according to their purchasing behaviors and preferences. Regardless of the specific application, the deployment of the trained model enables organizations to leverage the power of machine learning to drive actionable insights and make informed decisions in real-time. By continuously monitoring and updating the model with new data, organizations can ensure that it remains accurate and relevant, thus maximizing its utility and impact in addressing real-world challenges.

```
In [24]: train = pd.DataFrame()
         train['label'] = y_train
         train['text'] = X_train

         def predict_category(s, train=X_train, model=model):
             pred = model.predict([s])
             return pred[0]

In [25]: predict_category("i wanna shot myself")
Out[25]: 0

In [26]: predict_category("i Kill you")
Out[26]: 0

In [27]: predict_category("I'm cute")
Out[27]: 1
```

3.4 ML Model

Machine learning stands as a foundational pillar within the broader field of artificial intelligence, enabling systems to autonomously perform tasks by analyzing vast amounts of data and examples of desired behavior. This technology not only facilitates task execution but also illuminates intricate patterns within data that may be imperceptible to human observation alone. By systematically processing and data driven learning models that learn from data can reveal relationships and patterns, thereby enriching our comprehension of complex datasets and phenomena.At its essence, a data driven learning model embodies the culmination of this learning process, equipped with the ability to discern patterns and make informed decisions when confronted with new datasets. In the realm of natural language processing, these models showcase their proficiency by deciphering the underlying intent behind previously unseen sentences, showcasing their capacity to understand the subtleties and nuances of human language. Similarly, in image recognition tasks, machine learning models are trained to accurately identify and classify objects, demonstrating their versatility and applicability across a diverse range of domains and applications.The process of training a machine learning model involves feeding it with a substantial dataset, during which the algorithm refines itself to recognize specific patterns or produce desired outputs pertinent to the task at hand. This iterative optimization imbues the model with the ability to make accurate predictions or classifications. The resultant output of this training process manifests as a computer program equipped with defined rules and data structures, forming the essence of a machine learning model. Supervised machine learning, a prevalent paradigm in the field, entails furnishing the algorithm with an input dataset alongside corresponding desired outputs. Through this guidance, the algorithm learns to discern patterns and optimize its performance to align with the specified outputs. Classification, a technique within supervised learning, finds extensive utility in image recognition tasks, enabling models to categorize images into distinct classes. Moreover, supervised learning finds application in predicting various demographic factors such as population growth or health metrics, leveraging techniques like regression to extrapolate trends from data. This amalgamation of sophisticated algorithms and extensive datasets underscores the profound impact of supervised machine learning in diverse domains, elucidating its pivotal role in modern AI application.

3.5 ML Algorithm: Naive Bayes

Naive Bayes is a machine learning model widely used for classification tasks, particularly in scenarios with large volumes of data, including millions of records.Despite its simplicity, Naive Bayes generally produces good results, especially it is used for computational linguistics task such as emotion analysis. Recognized for its swiftness and user-friendly interface, this software offers simplicity and ease of use. Probability, in this context, denotes the likelihood of an event happening following the occurrence of another event. Bayes' theorem provides a mathematical framework for expressing this concept, enabling us to update our beliefs regarding event outcomes in light of fresh evidence.

Bayes' Theorem is represented as follows:

$$P(A|B) = P(B|A) \times P(A)/P(B) \tag{3.1}$$

where $P(A|B)$ is the conditional probability of event A given event B. $P(B|A)$ is the contextual probability of event B given event A. P(A) and P(B) are the probabilities of event A and event B occurring independently. Bayes' Theorem allows us to calculate the probability of an event A given the occurrence of event. B using prior knowledge about the probability of event A and the likelihood of event B given event A. This theorem is the foundation of the Naive Bayes algorithm, which applies Bayes' Theorem with the "naive" assumption of feature independence.In the context of Naive Bayes classification, Bayes' Theorem is used to calculate the probability of a class label given the features of an instance. By assuming that the features are conditionally independent given the class label, Naive Bayes simplifies the calculation of these probabilities, making it computationally efficient. Despite the simplifying assumption, Naive Bayes often performs well in practice, making it a popular choice for classification tasks, particularly in NLP applications like sentiment analysis. The Naive Bayes Classifier operates on the principles of Bayes' theorem, making predictions based on the probabilities of data points belonging to different classes. It calculates the likelihood of each class for a given set of data and selects the class with the highest probability as the most likely outcome. This approach, also known as Maximum A Posteriori (MAP), assigns membership probabilities to each class and identifies the class with the highest probability as the optimal choice. The MAP for a hypothesis is:

$$abc = abc \tag{3.2}$$

Evidence probability, utilized for result normalization, has no impact on the outcome when removed abc. In Naive Bayes classifiers, it is assumed that all variables or features are independent of each other. The presence or absence of one variable does not influence the presence or absence of any other variable.

Example: Even when a fruit exhibits characteristics typically associated with an apple, such as being red, round, and approximately 4 inches in diameter, a Naive Bayes classifier will still assess the independence of these features to determine the likelihood of the fruit being classified as an apple.

When applying our hypothesis to real datasets with numerous features, the computational complexity of our experiments increases.

Types of Naive Bayes Algorithms:

(a) **Gaussian Naive Bayes:**In Gaussian Naive Bayes, a variation of the Naive Bayes algorithm, the assumption is made that the characteristic values associated with each class follow a Gaussian or normal distribution when they are continuous in nature. This means that the distribution of these values for each class forms a bell-shaped curve, with most values clustered around the mean and fewer values spread out towards the tails of the distribution. By assuming this Gaussian distribution, the algorithm calculates the probability of observing a particular value given a class, based on the mean and standard deviation of the values for that class. This allows Gaussian Naive Bayes to effectively model the likelihood of observing certain continuous features given a class label, making it particularly suitable for datasets where the input features exhibit a continuous distribution.

(b) **Multinomial Naive Bayes:** Polynomial Naive Bayes is a specially designed variant of the Naive Bayes algorithm data that follows a multinomial distribution, making it particularly well-suited for text classification tasks in natural language processing (NLP). In text classification, each document is represented as a collection of words or tokens, and the occurrence of each word in the document is considered an event. Multinomial Naive Bayes models the probability of observing a particular word given the class label, utilizing counts of word occurrences to calculate these probabilities. Since text data often exhibits a multinomial distribution, where the frequency of occurrence of words follows a discrete distribution, Multinomial Naive Bayes is widely favored for tasks such as sentiment analysis, spam detection, and document categorization in NLP. By leveraging the frequency of word occurrences in documents, Multinomial Naive Bayes effectively learns the underlying patterns in text data and enables accurate classification of documents into predefined categories or labels.

(c) **Bernoulli Naive Bayes:** Bernoulli Naive Bayes is a variant of the Naive Bayes algorithm designed specifically for data that follows the multivariate Bernoulli distribution , where each feature is assumed to contain binary values. In other words, the features in the dataset are represented as binary variables, where each feature indicates the presence or absence of a particular attribute or characteristic. This variant of Naive Bayes is commonly used in scenarios where the presence or absence of certain features is crucial for classification tasks, such as document classification or spam detection. For example, in document classification, each feature may represent the presence or absence of a specific word in a document, with a value of 1 indicating its presence and 0 indicating its absence. By modeling the probability of observing each binary feature given the class label, Bernoulli Naive Bayes effectively learns the underlying patterns in the data and facilitates accurate classification based on the presence or absence of relevant features.

3.6 Deep Learning:Neural Network

Deep learning, a subset of machine learning, harnesses the power of artificial neuron network (ANN) to learn intricate patterns and relationships present within data. Unlike traditional programming where explicit instructions are provided for every task, deep learning systems learn autonomously from data, making them adept at handling complex tasks. This approach has gained immense traction in recent years, fueled by advancements in computational capabilities and the availability of vast datasets.Hierarchical neural networks revolves around neural networks at its foundation, specifically deep neural networks (DNN), which draw inspiration from the structure and functionality of biological neurons in the human brain. These networks consist of interconnected layers of artificial neurons that process and transform input data, gradually learning to extract meaningful features and make accurate predictions. By repetitively Repair connections between neurons feedback from the data, deep learning models continually improve their performance, enabling them to tackle a wide range of tasks across various domains with remarkable accuracy and efficiency.

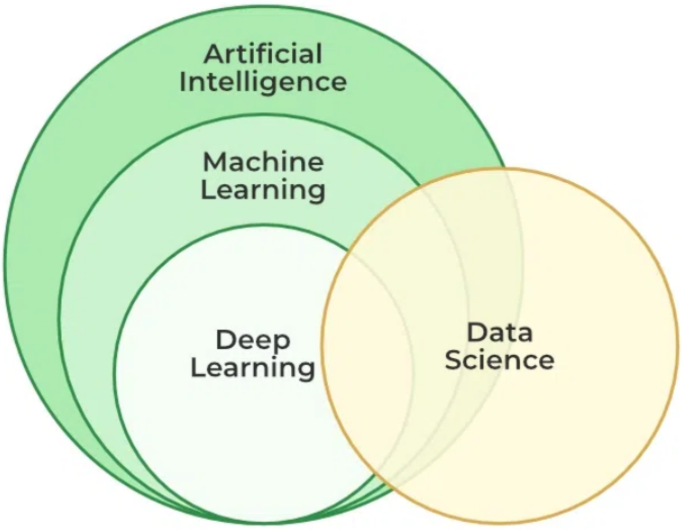


Fig. 3.3: Venn Diagram

At the heart of multi-layer learning lies the simulated network, a cornerstone concept reminiscent of the structure and functionality of human brain neural networks. These simulated networks comprise interconnected nodes known as neurons, organized into layers that typically encompass input, latent, and output processes. Within this framework, the input layer receives data, with each neuron representing a distinct feature. As data traverses through layers, intricate mathematical operations are conducted to uncover underlying patterns and correlations. The output layer generates the final outcome of the network's processing. Crucially, the connections between neurons are characterized by weights, dynamically adjusted during training via the process of Back propagation to minimize prediction errors. Activation functions introduce non-linearities, enhancing the network's capacity to comprehend complex data relationships. Through supervised learning, simulated networks iteratively refine their weights to minimize discrepancies between predicted and actual outputs, facilitating the ability to generalize and make accurate predictions on new data instances. This adaptability and adeptness at discerning subtle patterns render simulated networks indispensable for a myriad of tasks, including image recognition, natural language processing, and speech recognition, propelling advancements in artificial intelligence and machine learning.

3.6.1 Artificial Neural Network

The intricacy of a neural network is contingent upon the underlying data patterns and the myriad processes it undergoes, ranging from a few to potentially millions. Typically, a neural network comprises an input layer, one or more hidden layers, and an output layer. The ingress process sources information from various origins such as external devices, network analysis, or learning mechanisms. Subsequently, data traverses through layers, undergoing transformations to extract pertinent insights for subsequent processing. Each unit within the hidden layer receives input from preceding layer neurons, computes the weighted sum of these inputs, and applies a function to generate an output signal. The interconnections between units are governed by corresponding weights, dictating the influence of one unit's output on another. Throughout training, these weights are continually adjusted to optimize network performance. Ultimately, the egress process generates a response based on the processed input data. Drawing inspiration from human neuron structures and functions, this iterative process empowers artificial neural networks to glean knowledge from data and make accurate predictions or classifications.

In a neural network, Nodes, also called neurons, are arranged in layers typically consisting of three main layers: the input layer, the hidden layer(s), and the output layer.

1. Input Layer: The input layer is where the neural network receives its initial input data. Each node in this layer represents a feature or attribute of the input data. For example, in an image recognition task, each node might represent a pixel value in the image. The input layer serves to pass the input data to the subsequent layers for processing.

2. Hidden Layer(s): Situated between the input and output layers, the hidden layer(s) act as transitional components within the neural network structure. Their core role entails performing a series of mathematical computations to efficiently handle the input data. Each node in the hidden layer(s) receives input from every node in the previous layer and applies a transformation to generate its output. The number of hidden layers and the number of nodes in each hidden layer can vary depending on the complexity of the problem and the desired model architecture. Hidden layers enable the neural network to learn complex patterns and relationships within the data by extracting relevant features.

3. Output Layer: The output layer is the final layer of the neural network, where it produces the desired output or prediction. Each node in the output layer represents a class or category that the model aims to classify or predict. The output layer's size and structure depend on the nature of the task. For instance, in a binary classification problem, there may be two nodes representing the two possible classes, while in a regression problem, there might be a single node representing the continuous output value.

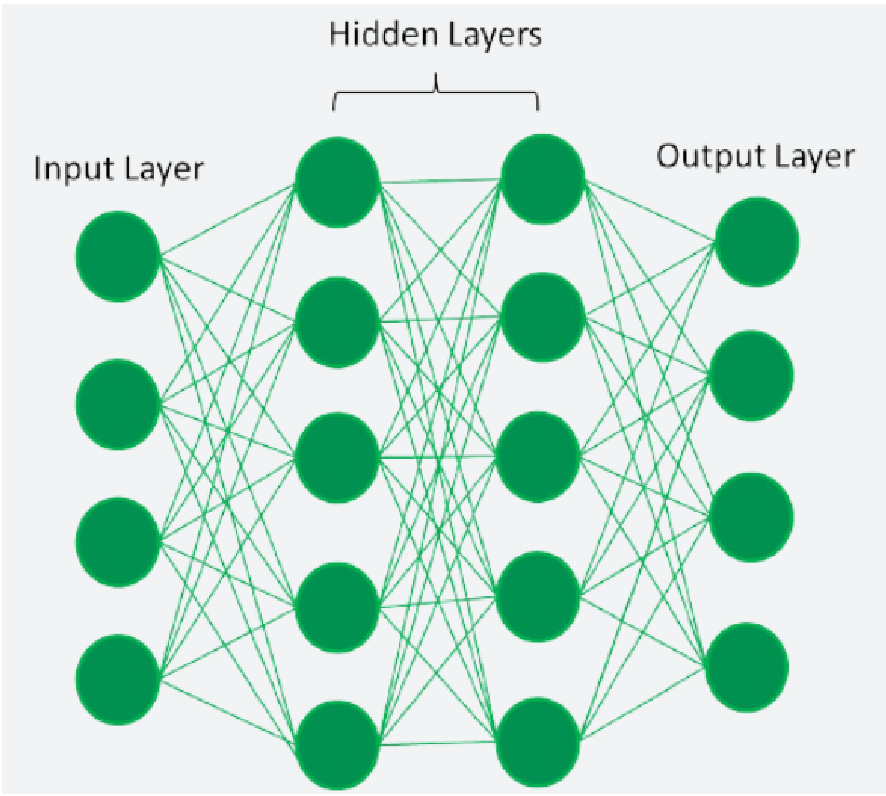


Fig. 3.4: Fully Connected Artificial Neural Network

Overall, the arrangement of nodes in these three layers forms the foundation of a neural network's architecture. Through training, which involves adjusting the parameters (weights and biases) of the connections between nodes, the neural network learns to map input data to the desired output, making it capable of performing various tasks such as classification, regression, and pattern recognition.

3.6.2 Artificial Neuron vs Biological Neurons

Artificial neural networks (ANN) are computational models inspired by the biological nervous system, particularly the structure and function of neurons in the brain. These networks consist of interconnected nodes, or artificial neurons, arranged in layers.

Structure: Biological neurons have a cell body that integrates incoming signals from dendrites, processes them, and generates output signals along the axon. In ANN, input nodes receive input signals from external sources, which are then transmitted through weighted connections to hidden layer nodes. The hidden layer nodes process these signals using activation functions and pass them to output layer nodes, which produce the final output of the network.

Table 3.2: Biological vs Artificial Neuron

Biological Neuron	**Artifical Neuron**
Dendrite	Input
Cell nucleus or soma	Nodes
Synapses	Weights
Axon	Output
Synaptic Plasticity	Backpropogation

Synapses: Synapses are the connections between neurons in the brain, where neurotransmitters are released to transmit signals from one neuron to another. In ANN, synapses are represented by the weighted connections between nodes in adjacent layers. These weights determine the strength of the connection between nodes and are adjusted during the training process to optimize the network's performance.

Learning: In biological neurons, learning occurs through synaptic plasticity, which involves strengthening or weakening of synaptic connections in response to neural activity. Similarly, in ANN, Learning is facilitated through training algorithms like backpropagation, which iteratively fine-tune the weights of connections between nodes by assessing the disparity between predicted and observed output values. This iterative process enables the network to glean insights from input-output pairs, progressively enhancing its performance over successive iterations.

Activation: Activation refers to the process by which neurons become active and transmit signals to other neurons. In biological neurons, activation occurs when the membrane potential reaches a certain threshold, triggering an action potential. In ANN, The activation function determines the output of the node based on the weight of the inputs. Common functions include sigmoid, tanh, ReLU (rectified linear unit), etc. and each of them has its own characteristics and uses. in different types of networks.

Overall, ANNs emulate the structure and function of biological neurons to perform various computational tasks, including pattern recognition, classification, and regression. By learning from data and adjusting their internal parameters, neural networks can adapt to complex input-output relationships and Make predictions or decisions based on sample data.

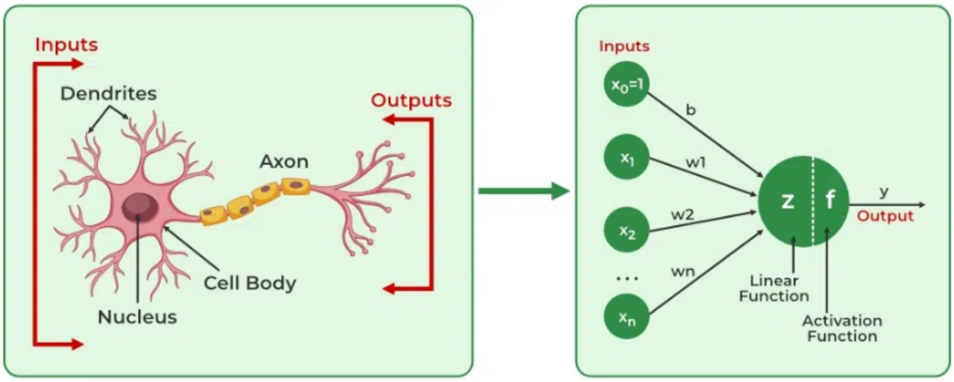


Fig. 3.5: Biological neurons to Artificial neurons

3.6.3 Learning of ANN

Artificial neural networks (ANN) are trained using a process called supervised learning, where they learn from labeled training data. In the example provided, let's consider training an ANN to recognize images of cats. The training process involves presenting the ANN with thousands of different images of cats, along with corresponding labels indicating that they are indeed images of cats.

During training, the ANN processes each image through its layers of interconnected nodes, with the input layer receiving pixel values from the image and passing them through the network. As the data propagates through the network, Adjust the weight of the connection between nodes based on the difference between predicted and actual results provided in the training data.
Once the ANN has been trained using a sufficient amount of cat images, it needs to be evaluated to determine its performance in correctly identifying cat images. This evaluation involves presenting the ANN with a new set of images, both cat images and images of other objects or scenes, and observing its classification decisions.
If the ANN fails to classify an image (i.e., it incorrectly identifies a non-cat image as a cat, or vice versa), back propagation is used to update the network's weights and biases. Back propagation involves Calculate the error, or loss, between the estimated output and the actual tag, and then propagate this error backwards through the network to adjust the weight and bias of the link to minimize errors.
This process of presenting data, calculating errors, and adjusting weights is repeated repetitively until the ANN achieves a satisfactory level of performance, typically measured by metrics such as accuracy or error rate. Through this iterative learning process, the ANN gradually learns to Identify specific patterns and features in cat images, allowing it to make accurate predictions on unseen data.
Overall, training an ANN involves exposing it to a large amount of labeled data, finetuning its parameters through back propagation, and repetitively refining its performance until it achieves the desired level of accuracy in classification tasks.

3.7 Algorithm:Convolution Neural Network

A Convolution Neural Network (CNN) is a type of deep learning algorithm primarily used for image recognition and classification tasks. CNN are inspired by the biologicalVisual cortex designed to automatically learn spatial hierarchies of features from input data [**?**].

Here's a simplified explanation of how CNN work: Sure, let's dive deeper into each component of a Convolution Neural Network (CNN) and how they work:

1. Convolution Layers: In convolution layers, Convolution Neural Networks (CNN) apply convolution Filter (also called kernel) data input. These filters are small matrices that slide over the input image, providing equal content and ensuring equal processing of each function. Each filter is designed to detect specific patterns or features in objects, such as edges, textures, or shapes. For instance, one filter might be sensitive to vertical edges, while another is tuned to detect horizontal edges. As the filters convolve across the input image, they generate feature maps that highlight the presence of different features. Each feature map represents the response of the corresponding filter to the input image and serves as a higher-level representation of the input data, capturing important visual patterns and structures. Through the learning process, CNN adjust the weights of these filters to effectively extract meaningful features from the input data, enabling them to perform tasks such as image classification and object detection.

2. Activation Function (ReLU): After the convolution operation in a Convolution Neural Network (CNN), an activation function is applied element-wise to introduce non-linearity into the model. This step is essential because it allows the network to capture complex relationships and patterns within the data, which may not be effectively represented by linear transformations alone. One of the most commonly used activation functions in CNN is the Rectified Linear Unit (ReLU). ReLU applies a simple mathematical operation to each element of the feature map, replacing any negative pixel values with zero while leaving positive values unchanged. This non-linear transformation enables the network to model more intricate features and interactions, improving its ability to learn and generalize from the input data. By introducing non-linearity through activation functions like ReLU, CNN can effectively handle the complexities of Functions such as image recognition, target detection and semantic classification, leading to improved performance and accuracy.

3. Pooling Layers: Pooling layers in Convolution Neural Networks (CNNs) serve the crucial role of reducing the width of the map while keeping the information simple. This area reduction is achieved by a function such as maximum pooling, which is a maximum pooling technique. During maximum pooling, a small window (usually 2x2 or 3x3) is shifted across the feature map and the maximum value in each window is retained while other values are discarded. By preserving only the maximum activation within each window, max-pooling effectively compress the feature maps, reducing computational complexity and memory requirements. Additionally, it aids in controlling training by enforcing spatial steady and promoting the detection of robust features across different regions of the input data. Overall, pooling layers play a critical role in extracting salient features from the input data while simultaneously reducing its spatial resolution, thereby facilitating efficient and effective feature extraction in CNN.

4. Flattening: In the process of building a Convolution Neural Network (CNN), after multiple convolution and pooling layers, the resulting feature maps are flattened into a one-dimensional vector. This step, known as flattening, is essential for preparing the data to be input into a fully connected neural network (FCNN). By flattening the feature maps, the spatial information captured during the convolution and pooling operations is converted into a linear format. This transformation is necessary because fully connected layers require one-dimensional input vectors. Flattening essentially collapses the multi-dimensional feature maps into a single continuous vector, where each element represents a specific feature or activation. This linear representation enables the subsequent fully connected layers to learn complex relationships between the extracted features and the target classes during the classification process. Overall, the flattening stage serves as a critical bridge between the convolution layers responsible for feature extraction and the fully connected layers responsible for classification, facilitating the seamless flow of information through the CNN architecture.

5. Fully Connected Layers: In the Convolution Neural Network (CNN) architecture, the flattened feature vector, obtained after the convolution and pooling layers, is fed into One or more layers, also called thick layers. All these layers play an important role in the classification process of learning intricate relationships between the extracted features and the target classes. Each neuron in the fully connected layer is connected to every neuron in the previous layer, enabling the network to capture high-level representations of the input data. Through this dense interconnection, the fully connected layers can effectively Learn and model complex patterns and dependencies found in data. Adjust weights and biases iteratively training process, these layers refine their parameters to optimize the classification accuracy, ultimately enabling the CNN to make accurate predictions on unseen data. Overall, the fully connected layers serve as the decision-making component of the CNN, where the learned features are transformed into predictions or classifications based on the network's learned parameters.

6. Output Layer: In convolutional neural network (CNN) architecture, the output layer serves as the final step in building the prediction network. It usually consists of one or more neurons, and each neuron corresponds to a group in the classification function. The number of neurons in the output layer matches the number of groups the CNN is designed to distribute. Different processes are used in the output layer depending on the nature of the job. For various classification functions, softmax initial functions are often used, which convert the scores of the raw data of each category into probabilities to ensure that the probability of each category equals 1. In contrast, in binary classification, a sigmoid activation function is used to make each class independent. The output value produced by the output method represents the probability estimate for each class, allowing the CNN to make informed decisions about classifying the input data according to the highest level. The latter process plays an important role in determining the overall performance of the network and its ability to analyze incoming data into its categories.

7. Training: Training a convolutional neural network (CNN) involves an iterative process of replacing a network failure to minimize performance loss. This process is caused by a process called backpropagation; where the error between the output estimate and the ground truth map is propagated back through the network, allowing for the slope of the loss function with respect to the network parameters. Optimization algorithms such as gradient descent are then used to adjust the weight and bias of the convolution filters and all layers in the network, thus gradually reducing the loss and improving the tuning of the network performance. The goal is to make the predicted output as close as possible to the real text in the training data, thus effectively learning underlying patterns and relationships in objects. Thanks to this training process, CNN learns the hierarchical representation of visual data, allowing it to perform many tasks such as image classification, target detection and image segmentation accurately and effectively.

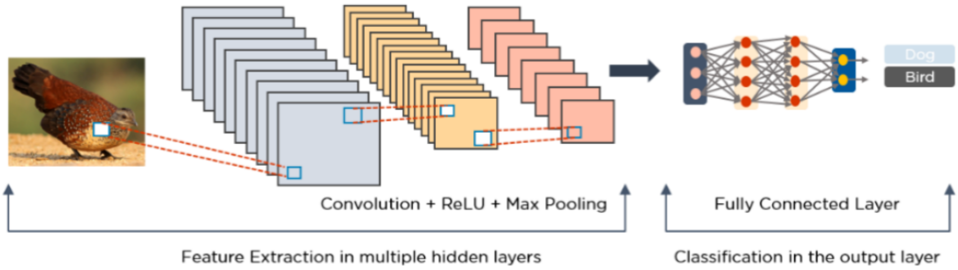


Fig. 3.6: Image Processed via CNN

Overall, CNN excel at learning spatial hierarchies of features from input data, making them well-suited for a wide range of computer vision tasks.

3.8 Data Set Description:Audio Sentiment Recognition

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) stands as a cornerstone resource for researchers delving into the intricate realm of emotional expression, particularly within the domains of speech and song. Boasting contributions from 24 seasoned actors, evenly divided between genders, RAVDESS offers a rich tapestry of vocal performances encompassing a diverse array of emotions. From the subtle tranquility of calm to the explosive intensity of anger, from the infectious joy of happiness to the haunting depths of sadness, the dataset encapsulates the full spectrum of human emotional experience.

One of the standout features of RAVDESS is its meticulous attention to emotional nuance. Each emotional expression is not only meticulously crafted but also presented at two distinct levels of intensity: normal and strong. This nuanced approach allows researchers to probe deeper into the intricate interplay between emotional states and vocal cues, shedding light on the subtle nuances that underpin human communication and perception.

Furthermore, RAVDESS offers researchers the flexibility to explore these emotional expressions across different modalities. Whether researchers seek to dissect the acoustic subtleties of speech, dissect the visual cues embedded in audio-video recordings, or solely focus on the visual aspects through video-only files, RAVDESS accommodates a wide range of experimental designs and analytical approaches.

The organizational structure of the dataset also merits praise for its clarity and efficiency. With files neatly organized into folders corresponding to individual actors, researchers can easily navigate through the wealth of data. Each file follows a standardized naming convention, providing detailed information about the modality, vocal channel, specific emotion, intensity level, statement, repetition, and actor involved. This meticulous labeling system streamlines data management and retrieval, ensuring researchers can swiftly locate and utilize the relevant data for their experiments.

Despite its many strengths, it's essential to acknowledge the limitations of this dataset. Actor number 18 notably lacks song version data, and certain emotions such as disgust, neutral, and surprised are absent from the song version recordings. Researchers must take these limitations into account when designing experiments and interpreting results to ensure the integrity and validity of their findings.

The dataset represents a treasure trove of invaluable resources for researchers venturing into the multifaceted realm of emotional expression. Its comprehensive coverage, nuanced emotional portrayals, versatile modalities, and meticulous organization make it an indispensable tool for advancing our understanding of human communication, perception, and cognition across a myriad of academic and commercial endeavors.

3.9 Project Flow of Audio based Depression Detection

The project flow for speech emotion recognition using Convolution Neural Network (CNN) entails a systematic process of analyzing audio data to discern emotional states. It begins with the provision of voice data in .wav format, which is then meticulously read and processed to extract relevant features. These features are visualized through waveform or spectrogram plots, aiding in understanding the underlying patterns within the audio signals. Subsequently, the dataset undergoes partitioning into training and testing subsets, facilitating the evaluation of the CNN model's performance. The CNN model is meticulously crafted and trained on the training data, enabling it to autonomously learn and extract salient features crucial for identifying emotions from speech. Post-training, the model is deployed to predict emotions or sentiments on the test data, which are then compared against ground truth labels using array sequences, thus quantitatively assessing the model's efficacy. The culmination of this process yields insights into the emotional state of the speaker, discerning whether they are stressed or not. Leveraging the capabilities of deep learning, the CNN model adeptly interprets emotional cues embedded within speech signals, thereby offering invaluable implications for diverse applications such as mental health monitoring and enhanced human-computer interaction paradigms.

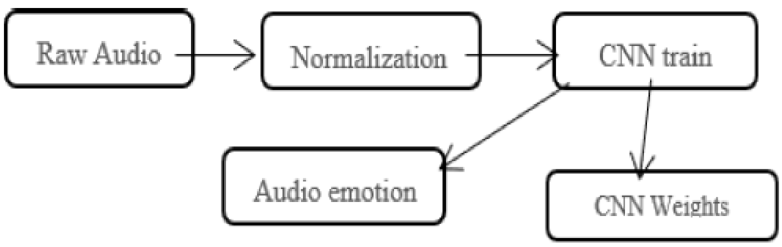


Fig. 3.7: Project Flow of CNN Model

The following process is used to implement the speech emotion recognition model of Depression Detection.

Step 1: Data Collection and Preprocessing

The data collection and pre processing steps involved in developing an emotion recognition system using audio recordings. The dataset used comprises recordings from various actors expressing different emotions. Prior to model training, the raw audio files undergo pre processing to extract Mel-frequency cepstral coefficients (MFCCs), which capture essential acoustic information while reducing extensity. Each audio sample is labeled with specific emotion categories such as anger, calmness, fear, happiness, or sadness, enabling supervised learning. This annotation schema facilitates the model's ability to learn patterns between acoustic features and emotional states. Overall, these preparatory steps lay the foundation for subsequent model training and evaluation, enhancing the system's accuracy in discerning emotions from real-world audio data.

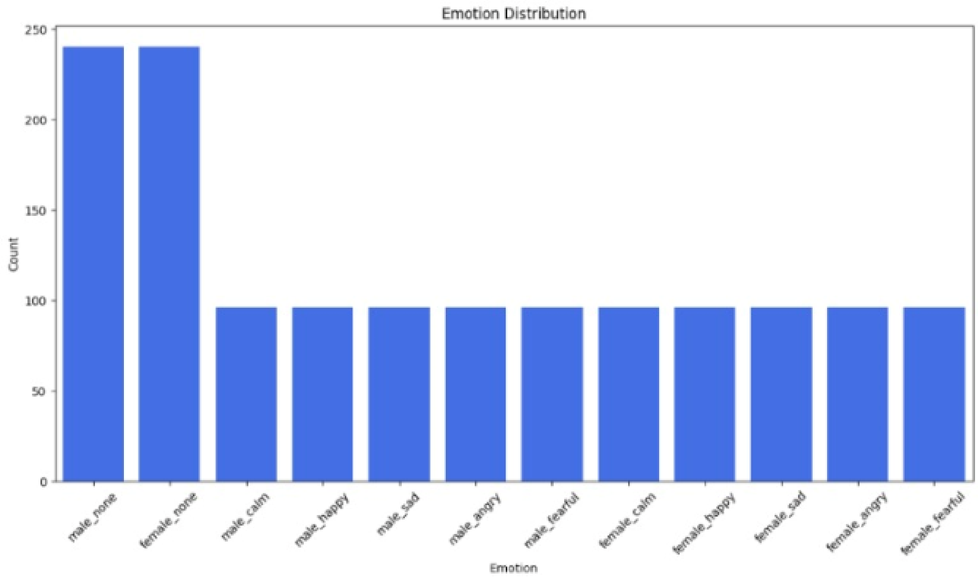


Fig. 3.8: Emotion Distribution

Step 2: Data Analysis and Visualization

In the realm of audio emotion recognition system development, data analysis and visualization play a pivotal role in understanding the dataset's characteristics and underlying patterns. By employing various techniques, analysts gain insights into the distribution of emotions within the dataset, using visualizations like bar charts to depict the prevalence of different emotional states. This analysis ensures the model is trained on a diverse and representative set of examples. Moreover, visualizing waveform and spectrogram provides a deeper understanding of the audio data temporal and spectral properties, aiding in the identification of patterns indicative of different emotional states.

Through thorough data analysis and visualization, researchers can uncover valuable insights, facilitating model development, evaluation, and the interpretation of predictions. Ultimately, this comprehensive understanding of the data enhances the robustness and accuracy of audio emotion recognition systems.

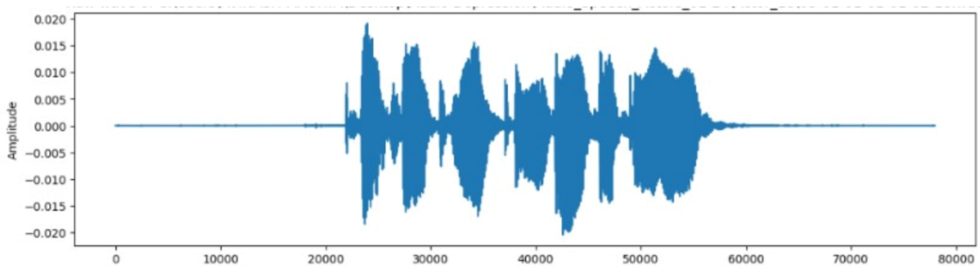


Fig. 3.9: Amplitude Plot of Raw wave

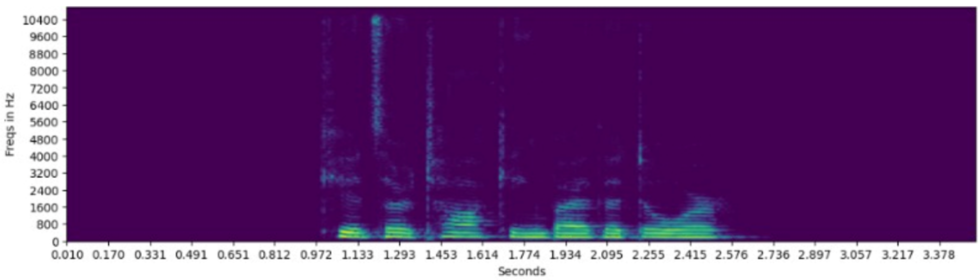


Fig. 3.10: Spectrogram Plot

Step 3: Data Augmentation

In the pursuit of fortifying the robustness and generalization capabilities of the audio emotion recognition system, data augmentation proves to be a crucial strategy. This process involves introducing variations and perturbations to the existing audio recordings through a suite of techniques such as noise addition, shifting, stretching, and pitch tuning. By augmenting the training dataset in this manner, the model becomes more adept at discerning emotional cues amidst varying levels of background interference, simulating real-world scenarios where environmental noise may obscure the underlying emotional expression. Moreover, temporal manipulations like shifting and stretching introduce variations in speech tempo and rhythm, enabling the model to generalize effectively across different speaking styles. Additionally, pitch tuning exposes the model to a spectrum of pitch variations, enhancing its understanding of the acoustic features associated with different emotional states. Overall, data augmentation enriches the training dataset with greater diversity and complexity, empowering the model to excel in diverse real-world environments characterized by noise, variability, and nuance. Through judicious application, data augmentation enhances the system's capacity for accurate emotion recognition, ensuring its resilience and adaptability across a wide range of scenarios.

Step 4: Model Architecture

At the core of the audio emotion recognition system lies a meticulously crafted model architecture built upon the foundational principles of Convolution Neural Networks (CNN). This architecture is designed to distill complex audio features into discernible emotional categories through a hierarchical structure. Beginning with convolution layers, strategically positioned to convolute over spectrogram representations of the audio data, the model extracts salient temporal and frequency patterns indicative of emotional nuances. Pooling layers then compress these features, capturing their spatial hierarchies while reducing computational complexity. Dense layers follow, consolidating hierarchical representations into compact feature vectors amenable to emotion classification through non-linear transformations. Dropout layers are strategically integrated to mitigate training by uncertainly deactivating neurons during training. Upon construction, the model undergoes compilation, utilizing categorical cross-entropy loss and the Adam optimizer to quantify prediction disparities and optimize parameters, respectively, thereby enhancing classification accuracy and convergence speed. This meticulously engineered architecture enables the system to discern subtle emotional distinctions encoded within audio data while ensuring robustness and generalization across diverse emotional expressions.

Step 5: Model Training

Model training constitutes the life-changing crucible where neural networks undergo iterative refinement, assimilating the intricacies of training data to become adept classifiers of emotional states within audio recordings. This process initiates with dataset partitioning into training and validation subsets, facilitating performance evaluation on unseen data. Amidst epochs of training, the model refines its parameters repetitively to minimize loss and maximize accuracy. Key interventions, orchestrated by callbacks, augment training efficacy and resilience. Model control unit ensures the preservation of optimal configurations, while learning rate reduction mechanisms dynamically navigate optimization landscapes. Metrics like loss and accuracy serve as performance barometers, guiding model architects towards iterative improvements in architecture and training protocols for superior classification outcomes. Through vigilant monitoring and adaptation, the model evolves to adeptly capture emotional dynamics within audio data, contributing to its efficacy in real-world applications.

Step 6: Model evaluation

During the model's evaluation on the test dataset, various performance metrics such as accuracy and F1-score are utilized to gauge its effectiveness in classifying emotions within audio recordings. These metrics provide a comprehensive assessment of the model's capabilities, while the confusion matrix offers a nuanced view of its classification prowess across different emotions. By analyzing the matrix, evaluators identify areas of strength and weakness, guiding future refinements and improvements. Ultimately, the evaluation serves as a catalyst for continued exploration and advancement in audio emotion recognition, empowering researchers to delve deeper into this complex domain armed with valuable insights gleaned from rigorous evaluation processes.

Step 7: Deployment and Testing

After the rigorous training process, the model is ready for deployment, marking the culmination of its development journey. Serialized for seamless integration into various environments, the model is tested with real-world audio files to evaluate its ability to recognize emotions accurately amidst everyday noise. Through iterative testing and refinement, the model adapts to diverse scenarios, enhancing its discernment and robustness. Deployment signifies not just the end of one phase but the beginning of another, as developers continually refine the model based on real-world feedback, ensuring its evolution into a dynamic tool for emotion recognition in various contexts.

3.10 Real Time ECG Monitoring

The article highlights the significance of mental health, particularly focusing on depression as a widespread condition impacting millions worldwide. It underscores the importance of early detection and intervention for effective treatment. Traditional approaches, such as self-reporting and clinical assessments, are acknowledged for their subjectivity and time-consuming nature. To address these challenges, the article introduces an innovative approach leveraging wearable technology and machine learning to transform depression detection. This novel method aims to provide a more objective and efficient means of identifying individuals at risk of depression, potentially revolutionizing the way mental health conditions are diagnosed and managed [**?**] .

The project introduces a hardware device crafted with an ESP8266 microcontroller [**?**] and an AD8232 sensor, designed to provide a non-invasive wearable solution. This wearable can be easily worn by users and operates by continuously monitoring their heart rate data. The collected physiological data is then transmitted without wire, potentially to a cloud-based platform, for in-depth analysis. This real-time monitoring offers a unique insight into the user's well-being, providing a continuous stream of physiological information. Such data could unveil subtle patterns and trends that traditional assessment methods might overlook, potentially offering valuable insights into the user's overall health status and aiding in early detection and intervention for conditions like depression.

The project extends beyond mere data collection by proposing the utilization of machine learning algorithms to analyze the heart rate information gathered by the wearable device. Machine learning, renowned for its ability to discern intricate patterns and correlations within complex datasets, is identified as a potent tool for this task. By applying these advanced algorithms to the collected heart rate data, the project aims to develop a sophisticated system capable of detecting potential indicators of depression. Through the analysis of subtle variations and trends in heart rate patterns, the system seeks to identify early signs of depressive symptoms. This early detection holds immense potential in facilitating timely intervention and treatment, enabling healthcare professionals to offer support and assistance to individuals at risk of depression before their condition exacerbates, ultimately improving patient outcomes and well-being.

The potential benefits of this project extend far beyond the realm of individual patients, impacting both healthcare outcomes and system efficiency. Early and precise detection of depression holds the promise of vastly improving patient outcomes by enabling timely intervention and treatment initiation. By identifying individuals at risk of depression at an early stage, healthcare professionals can intervene promptly, potentially preventing the condition from escalating and leading to more severe consequences. Moreover, this approach has the potential to alleviate the burden on healthcare systems by facilitating remote monitoring and early intervention. Through continuous monitoring of physiological data via wearable technology, healthcare providers can proactively identify and support individuals at risk, reducing the need for extensive clinical assessments and hospital visits. This not only improves patient care but also optimizes resource allocation within healthcare systems, ultimately enhancing overall system efficiency and effectiveness.

In conclusion, this project represents a major advance in depression research, leveraging the combined power of wearable technology and machine learning. Through continuous monitoring of heart rate data and the application of sophisticated algorithms, the project endeavors to achieve early and precise diagnosis of depression. The potential implications of this approach are profound, with the promise of improving patient outcomes by enabling timely intervention and treatment initiation. Moreover, the project has the potential to alleviate the burden on healthcare systems by enabling remote monitoring and early intervention, thereby optimizing resource allocation and enhancing overall system efficiency. The upcoming sections of the project will provide a detailed exploration of the technical aspects of the device, the machine learning methodology employed, and the broader implications for the field of mental health, shedding light on the life changing potential of this innovative approach.

3.11 Hardware and Methods

The project aimed at designing a heart monitoring system tailored to different weight categories of subjects, there are two crucial stages essential for achieving the proposed objective. Firstly, the sensing, processing, and display units must seamlessly collaborate to ensure the efficient functioning of the system. The heart electrical activities sensor, which outputs digital data, plays a pivotal role in sensing the physiological signals. This sensor's output can be seamlessly integrated with any of the digital pins of the Nodemcu microcontroller, facilitating communication between the sensor and the Nodemcu unit. This integration forms the foundational step in the data acquisition process, allowing the system to capture and process the heart's electrical signals effectively. Secondly, the communication aspect between the Nodemcu unit and the Ubidots [**?**] software is paramount for transmitting the collected data to the intended recipient, such as the patient's phone. Ubidots serves as the intermediary platform for this communication, enabling seamless data transfer and visualization for the patient's monitoring. The ESP8266 board, equipped with specific pins designated for transmission and reception of data, facilitates the communication interface between the Nodemcu unit and the Ubidots software. Through this synchronized communication setup, the system ensures real-time monitoring and tracking of the patient's heart activity, thereby contributing to enhanced healthcare management and patient well-being.

Stage 1 : Sensing

In the sensing stage of the heart monitoring system project, the primary objective is to capture the electrical activities of the heart using appropriate sensors. The chosen sensor for this project outputs a digital signal, which implies that it provides data in a digital format rather than analog. This digital output simplifies the interfacing process with the Nodemcu board, as digital signals are easier to handle and process within digital electronic systems.The digital output from the sensor can be seamlessly connected to any of the digital pins available on the Nodemcu board. These digital pins are versatile and can be configured to either read digital inputs or provide digital outputs, depending on the requirements of the system. By connecting the sensor's digital output to one of these pins, the Nodemcu board can effectively capture the heart's electrical activities in a format that is compatible with its digital processing capabilities.This integration of the sensor's digital output with the Nodemcu board forms the foundation of the sensing stage, enabling the system to accurately capture and process the heart's electrical signals. Through this setup, the system can gather essential physiological data necessary for monitoring the patient's heart health and facilitating further analysis or action as needed.

Stage 2: Processing and Display

2.1 Nodemcu Integration: In the heart monitoring system project, the Nodemcu board plays a pivotal role in integrating various components and facilitating communication between them. As a versatile microcontroller board based on the ESP8266 chip, the Nodemcu offers the necessary computational power and connectivity features required for this application. One of its primary functions in this project is to receive data from the heart sensor and relay it for further processing and transmission.The digital output from the heart sensor, which captures the electrical activities of the heart, is seamlessly connected to one of the digital pins on the Nodemcu board. These digital pins serve as the interface through which the Nodemcu communicates with external devices or sensors. By connecting the sensor's digital output to one of these pins, the Nodemcu establishes a direct pathway for receiving the heart signal data in digital format. Once the sensor's digital output is connected to a digital pin on the Nodemcu, the microcontroller can effectively receive and process the incoming data. This integration enables the Nodemcu to capture the heart's electrical activities in real-time, paving the way for further analysis or transmission as required by the project objectives. Through this seamless integration, the Nodemcu acts as a central hub for data acquisition, enabling the heart monitoring system to function efficiently and effectively.

2.2 Ubidots Software: In the heart monitoring system project, Ubidots serves as a pivotal component in facilitating seamless communication between the Nodemcu board, where the heart sensor data is received, and the user's smartphone. Ubidots is a cloud-based Internet of Things (IoT) platform designed to handle the collection, processing, and visualization of sensor data from various devices. Its versatile capabilities make it an ideal choice for integrating sensor data into cloud-based applications, enabling real-time monitoring and analysis. In this project, Ubidots acts as the intermediary platform between the Nodemcu board and the user's phone. Once the Nodemcu receives the digital output from the heart sensor and processes it, the data is transmitted to Ubidots for further handling. Ubidots then stores the received data securely in the cloud, making it easily accessible to the user through their smartphone. This allows the user to monitor their heart activity remotely and in real-time, providing valuable insights into their health status.

2.3 Data Transmission: The ESP8266 board features GPIO (General Purpose Input/Output) pins that can be configured for various purposes, including digital input/output and communication interfaces such as UART (Universal Asynchronous Receiver-Transmitter) and SPI (Serial Peripheral Interface). For communication with Ubidots, specific GPIO pins are chosen and configured to serve as the communication interface between the ESP8266 board and the Ubidots platform. Typically, the selected pins are configured to establish a serial communication link, such as UART or SPI, which allows for the transmission and reception of data between the ESP8266 board and the Ubidots servers. This serial communication protocol ensures reliable data transmission and synchronization between the devices, enabling seamless integration of the heart monitoring system with the Ubidots platform.By utilizing specific pins dedicated to data transmission and reception, the ESP8266[**?**] board can effectively communicate with the Ubidots platform, enabling the seamless transmission of sensor data from the Nodemcu to the cloud-based Ubidots servers. This integration facilitates real-time monitoring and analysis of heart activity data, empowering users to track their health metrics and receive timely insights into their heart health status.

Stage 3: Communication Flow

3.1 Sensor Data to Nodemcu: In the heart monitoring system, the heart sensor collects data pertaining to the subject's heart electrical activities, which is essential for monitoring their cardiac health. This data is captured by the sensor and converted into a digital format, ensuring compatibility with digital processing systems like the Nodemcu board. The digital output from the heart sensor is transmitted to the Nodemcu board through a digital pin connection. This connection establishes a pathway for the sensor data to be received and processed by the microcontroller unit. By connecting the sensor's output to a digital pin on the Nodemcu board, the system ensures a direct and efficient transfer of the digital data.Once the sensor data is received by the Nodemcu board, it can be further processed, analyzed, or transmitted to external devices or platforms for visualization and monitoring. This integration of the heart sensor with the Nodemcu board forms a critical component of the heart monitoring system, enabling the real-time capture and processing of essential physiological data for monitoring the subject's heart health.

3.2 Nodemcu to Ubidots: In the heart monitoring system, the heart sensor collects data pertaining to the subject's heart electrical activities, which is essential for monitoring their cardiac health. This data is captured by the sensor and converted into a digital format, ensuring compatibility with digital processing systems like the Nodemcu board.The digital output from the heart sensor is transmitted to the Nodemcu board through a digital pin connection. This connection establishes a pathway for the sensor data to be received and processed by the microcontroller unit. By connecting the sensor's output to a digital pin on the Nodemcu board, the system ensures a direct and efficient transfer of the digital data. Once the sensor data is received by the Nodemcu board, it can be further processed, analyzed, or transmitted to external devices or platforms for visualization and monitoring. This integration of the heart sensor with the NODEMCU board forms a critical component of the heart monitoring system, enabling the real-time capture and processing of essential physiological data for monitoring the subject's heart health.

3.3 Ubidots to Phone: Upon reaching the Ubidots platform, the heart monitoring data undergoes storage and processing, becoming readily accessible to the patient or user through the Ubidots mobile application. Ubidots serves as a secure and reliable cloud-based Internet of Things platform, where data collected from various sensors, including the heart sensor in this case, is stored securely. The platform offers robust features for data management, visualization, and analysis, ensuring that users can easily access and interpret their heart activity data.Through the Ubidots mobile application, users can conveniently monitor their heart's electrical activities in real-time. The application provides a user-friendly interface where users can view their heart monitoring data in various formats, such as graphs, charts, or numerical values. This real-time access empowers users to track changes in their heart activity over time, allowing them to stay informed about their cardiac health status and make timely decisions regarding their well-being. Furthermore, the Ubidots mobile application may offer additional functionalities such as setting up alerts for abnormal heart activity, viewing historical data trends, and sharing data with healthcare providers or family members. These features enhance the user experience and enable proactive management of cardiac health. Overall, the seamless integration between Ubidots and the mobile application ensures that users have easy access to their heart monitoring data, promoting continuous monitoring and proactive management of their cardiovascular health.

3.11.1 ESP 8266 :

ESP8266 is a versatile and useful Wi-Fi module that combines a microcontroller and a Wi-Fi chip in a single package; making it ideal for IoT applications. With its low power consumption, extensive GPIO pins, and support for various programming languages like Arduino IDE and MicroPython, the ESP8266 enables developers to easily add Wi-Fi connectivity to their projects. Its built-in Flash memory provides ample space for program storage, while its compatibility with a wide range of sensors and peripherals makes it suitable for diverse applications. Popular for its affordability and ease of use, the ESP8266 has become a staple in the maker community, powering countless IoT devices and projects worldwide. The ESP8266 comes in various variants, ranging from the ESP8266-01 to ESP8266-13, each building upon its predecessor in terms of hardware capabilities. The modules differ in features such as the number of GPIO pins, presence of shield and antenna, package type (Through-hole or Surface mount), memory, and ability to handle external analog signals. For instance, the ESP8266-01, the most basic variant, offers 2 GPIO pins, UART communication, a low-powered 32-bit CPU, and a PCB antenna. In contrast, other modules like the ESP8266-12 boast additional features such as ADC input capabilities, SPI, I2C, and more GPIO pins, making them suitable for a wider range of applications.

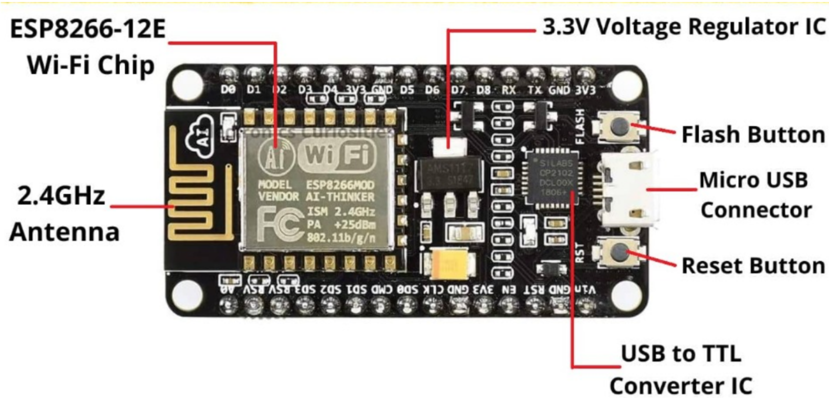


Fig. 3.11: ESP 8266

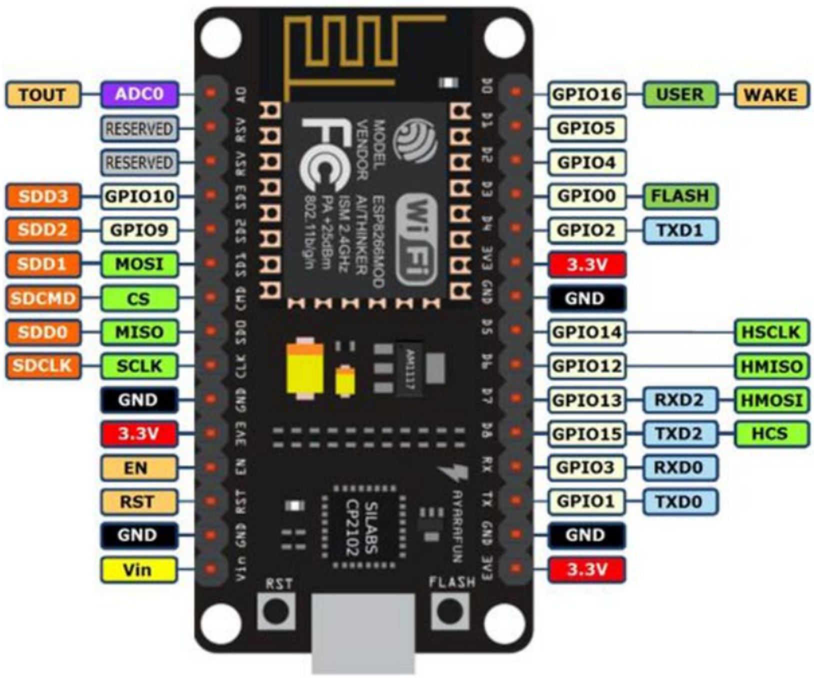


Fig. 3.12: ESP 8266 Pinout

3.11.2 AD 8232

The AD8232 is a single-lead, heart rate monitoring analog front-end (AFE) integrated circuit designed by Analog Devices for applications in portable ECG (electrocardiogram) devices and heart rate monitors. This highly integrated chip includes instrumentation amplifiers, right-leg drive amplifiers, lead-off detection, and pace detection circuitry to provide a complete solution for biopotential signal acquisition. The device offers high input impedance, low input bias current, and low input-referred noise, ensuring accurate and reliable signal acquisition from the body. Its flexible design accommodates various electrode configurations and provides adjustable gain to optimize signal quality. The AD8232 operates from a single power supply, making it suitable for battery-powered applications. With its small form factor, low power consumption, and robust performance, the AD8232 simplifies the design of portable ECG devices and enables accurate heart rate monitoring in wearable health and fitness applications. The healthcare industry has developed IoT real-time monitoring that allows doctors and specialists to diagnose patients quickly, smartly and effectively. Although there are many researches and studies that have developed methods for remote monitoring of ECG signals, the methods for monitoring and distributing these signals have not yet been agreed upon, so the process must be used to create complete health. Distribute the ECG signal. In this paper, we propose an ECG monitoring and classification. The application process is based on the communication of AD8232 sensor and ardunino node MCU, analog to digital converter and output ECG signal, which is used to convert the signal into a higher signal, and then send the output signal to the signal.Cloud-to-cloud use anywhere, the signal is pre-processed to remove the noise and QRS complex[**?**] is detected to determine the other characteristics of the signal such as heart rate, also to determine one cycle of ECG signal, later the signal is classified by using proposed convolution neural network model to detect the signal status. The extracted ECG signal is transmitted in real time to cloud (Ubidots cloud is used) through ESP8266 over to the cloud using WiFi based on MQTT publishing method. The experimental results are performed on different signals and the different stage of de-noising and QRS detection are applied and different pooling layers are used in the proposed CNN model[**?**]. The results show that the proposed classification model achieve accuracy up to 94%. he basic components of AD8232 are:

1. Instrumentation Amplifiers:

An instrumentation amplifier is a specialized type of operational amplifier[**?**] (op-amp) configuration designed to amplify small differential signals while rejecting common-mode noise. It typically consists of three op-amps configured in a differential amplifier topology, with additional resistors for precise gain control. The primary function of an instrumentation amplifier is to amplify the voltage difference between two input signals while attenuating any common-mode signals present at both inputs. This allows for accurate measurement of small differential signals, making instrumentation amplifiers widely used in applications such as sensor interfacing, data acquisition systems, and medical instrumentation, where precision and noise rejection are paramount.

2. Right Leg Drive:

Right leg drive (RLD) is a technique commonly employed in biomedical instrumentation, particularly in electrocardiography[**?**] (ECG) systems, to reduce common-mode interference and enhance the quality of acquired signals. In RLD, an electrode is placed on the patient's right leg and connected to the ground of the amplifier circuitry. By doing so, the common-mode voltage at the patient's body surface, including both the left and right arms, is shifted or driven towards the right leg. This effectively creates a reference point at the right leg, helping to cancel out common-mode noise picked up by the body. RLD is crucial in ECG systems as it improves the signal-to-noise ratio and minimizes artifacts, resulting in clearer and more accurate ECG recordings, particularly in environments prone to electrical interference.

3. Lead-Off Detection Circuitry:

Lead-off detection is a crucial feature in biomedical instrumentation, especially in electrocardiography (ECG) systems, aimed at ensuring the quality and reliability of the acquired signals. Lead-off detection identifies instances where the electrodes lose contact with the patient's skin or when the impedance between the electrode and the skin becomes too high, indicating a potential issue with signal acquisition. When lead-off is detected, it triggers an alert or warning, notifying the user or system operator of the problem. This feature is essential for maintaining the integrity of physiological measurements and preventing erroneous interpretations of data due to poor electrode contact or signal loss. Lead-off detection is often implemented using dedicated circuitry or algorithms that continuously monitor the electrical impedance at each electrode site, promptly identifying any abnormalities and allowing for timely corrective action to be taken.

4. Single Power Supply Operation:

The AD8232 is a single-lead electrocardiogram (ECG) analog front-end (AFE) integrated circuit (IC) manufactured by Analog Devices. One of its notable features is its ability to operate from a single power supply, which simplifies system design and reduces the overall component count required for building ECG monitoring systems. This characteristic is particularly advantageous for portable or wearable applications where space and power efficiency are critical considerations. By eliminating the need for dual power supplies, the AD8232 streamlines the design process, reduces system complexity, and lowers manufacturing costs. Additionally, operating from a single power supply enhances the versatility and ease of integration of the AD8232 into various ECG monitoring devices,becoming a popular choice among designers and engineers healthcare and wellness industries.

5. Low Input-Referred Noise:

"Low input-referred noise" refers to the minimal level of electrical noise introduced by an electronic device, such as an amplifier or sensor, at its input. In the context of biomedical instrumentation like electrocardiography (ECG), this term is particularly relevant as it pertains to the quality of the acquired physiological signals. For instance, in an ECG system, low input-referred noise means that the amplifier used to capture the heart's electrical signals[**?**] introduces minimal additional noise to the signal being measured. This is crucial because ECG signals are typically very small in amplitude, measured in millivolts, and any additional noise can distort the signal, leading to inaccurate readings or difficulty in interpreting the data. A low input-referred noise level ensures that the acquired signals maintain high fidelity and accuracy, allowing for reliable diagnosis and monitoring of cardiac activity. Achieving low input-referred noise often involves careful design considerations, including the use of high-quality components, proper shielding techniques, and optimized circuit layouts.

6. Flexible Input Configuration:

Firstly, the AD8232 can accommodate both single-lead and multi-lead ECG configurations. Single-lead configurations are commonly used in portable or wearable devices for basic heart rate monitoring, while multi-lead configurations, such as the standard 12-lead ECG, are utilized for more detailed cardiac assessments in clinical settings. The AD8232's adaptability to different lead configurations to ensure widespread use of monitoring scenarios, from basic health and wellness applications to medical diagnostics. Moreover, the AD8232 is compatible with various types of electrodes, including disposable electrodes with adhesive gel, reusable electrodes with snap connectors, and dry electrodes with conductive materials. This compatibility enables users to choose the most suitable electrode type for their specific application requirements, considering factors such as comfort, convenience, and signal quality. Additionally, the AD8232's adjustable gain settings and integrated right-leg drive feature further enhance its versatility, allowing users to optimize signal acquisition for different electrode types and placements.

7. Built-in Filters:

The AD8232 includes built-in filters designed to enhance the quality of electrocardiogram (ECG) signal acquisition by mitigating noise and interference. These filters are integral to the analog front-end (AFE) circuitry of the AD8232 and serve to remove unwanted signals while preserving the desired ECG waveform [**?**]. One of the primary filters incorporated into the AD8232 is a configurable low-pass filter. This filter attenuates high-frequency noise and artifacts that may be present in the ECG signal due to sources such as electromagnetic interference (EMI) or muscle activity. By selectively filtering out high-frequency components, the low-pass filter helps to improve the signal-to-noise ratio and enhance the clarity of the ECG waveform. Additionally, the AD8232 may include other filters such as a high-pass filter to remove baseline drift and DC offset, as well as notch filters to suppress specific frequencies of interference, such as powerline noise (50 Hz or 60 Hz). These filters work synergistically to ensure that the acquired ECG signal is clean and free from common sources of distortion or interference.

8. Small Form Factor:

The AD8232 is indeed engineered in a compact form factor, which significantly enhances its suitability for portable and wearable applications across various industries, particularly in healthcare and wellness. Its small size and lightweight design make it highly conducive for integration into a wide range of devices, including fitness trackers, wearable monitors, and portable medical devices. The compact form factor of the AD8232 enables manufacturers to design sleek and unobtrusive wearable devices that can be comfortably worn by users throughout the day. Whether incorporated into wrist-worn fitness trackers or chest strap heart rate monitors, the AD8232's compact size ensures that the wearable device remains discreet and inconspicuous, promoting user comfort and convenience. Additionally, the small footprint of the AD8232 facilitates the development of portable medical devices for ECG monitoring[**?**] applications. By integrating the AD8232 into handheld or handheld devices, healthcare professionals can conveniently perform on-the-go ECG monitoring, enabling timely diagnosis and intervention for patients in various settings such as clinics, ambulances, or remote locations. Moreover, the compact size of the AD8232 allows for efficient use of space within the device, leaving room for other essential components such as batteries, microcontrollers[**?**], and display screens. This optimization of space ensures that portable and wearable devices equipped with the AD8232 can offer comprehensive functionality while maintaining a sleek and ergonomic design.

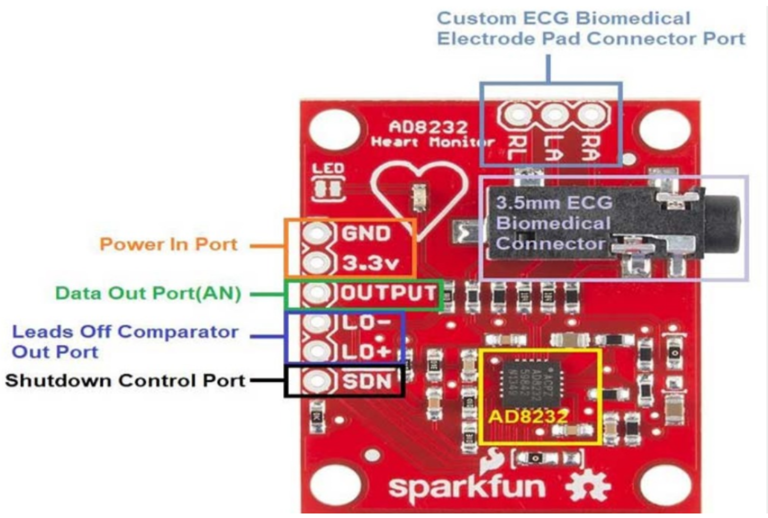


Fig. 3.13: AD 8232

3.11.3 ECG Electrode

Electrocardiogram (ECG) electrodes [**?**] are essential components used in biomedical instrumentation for capturing the electrical activity of the heart. These electrodes are typically placed on the patient's skin at specific locations to detect the electrical signals generated by the heart's contractions. ECG electrodes come in various forms, including disposable adhesive electrodes, reusable electrodes with snap connectors, and dry electrodes with conductive materials. They function by establishing electrical contact between the patient's skin and the monitoring device, allowing for the transmission of bioelectrical signals to the sensor or amplifier. ECG electrodes must adhere securely to the skin to ensure reliable signal acquisition and minimize motion artifacts, while also being gentle enough to prevent skin irritation or discomfort. Overall, ECG electrodes[**?**] play a crucial role in obtaining accurate and reliable ECG readings, making them indispensable components in healthcare settings for diagnostic, monitoring, and research purposes. Here are the details of each component of an ECG electrode:

1. Adhesive Patch/Pad: Adhesive patches or pads are specialized types of electrodes commonly used in electrocardiogram (ECG) monitoring and other biomedical applications. These electrodes are designed with an adhesive backing that allows them to securely adhere to the patient's skin, ensuring reliable contact and signal acquisition throughout the monitoring period. Adhesive patches or pads are particularly advantageous for long-term monitoring or ambulatory ECG recordings, as they provide a stable and comfortable attachment that minimizes the risk of electrode displacement or motion artifacts. These electrodes come in various shapes and sizes, including rectangular patches, circular pads, or custom-shaped designs, to accommodate different electrode placements and patient preferences. They are typically equipped with conductive gel or solid conductive materials to facilitate the transmission of electrical signals from the patient's skin to the monitoring device. Adhesive patches or pads are also available in disposable or reusable options, depending on the specific application requirements and cost considerations.

2. Conductive Gel: Conductive gel is a specialized substance used in conjunction with electrodes in biomedical applications, particularly in electrocardiography (ECG) and electromyography (EMG) [**?**]. It serves as an interface between the skin and the electrode, facilitating the transmission of electrical signals while minimizing impedance and improving signal quality.Conductive gel is typically water-based and contains electrolytes to enhance conductivity. When applied to the skin beneath the electrode, it helps to reduce the skin-electrode interface impedance by filling in microscopic irregularities and ensuring uniform contact. This results in a more reliable and stable electrical connection, leading to clearer and more accurate recordings of physiological signals. In addition to improving signal quality, conductive gel also serves to protect the skin from irritation or discomfort caused by prolonged contact with electrodes. It is formulated to be non-drying and hypoallergenic, Made suitable for use by many patients, including those with sensitive skin.

3. Connector: Connectors play a critical role in biomedical instrumentation by facilitating the secure and reliable connection between various components, such as electrodes, cables, sensors, and monitoring devices. In electrocardiogram (ECG) monitoring, connectors are used to link the electrodes to the monitoring equipment, ensuring the transmission of electrical signals with minimal interference or signal loss.There are several types of connectors commonly used in biomedical applications, including snap connectors, banana plugs, and DIN connectors. Snap connectors are often found on disposable or reusable ECG electrodes, providing a quick and easy way to attach and detach the electrodes from the monitoring device. Banana plugs, with their simple and robust design, are commonly used for connecting cables to ECG machines or amplifiers. DIN connectors, characterized by their circular shape and multiple pins, are used in more specialized applications where additional functionalities or signal channels are required. Connectors are designed with features such as locking mechanisms, shielding, and insulation to ensure a secure and stable connection while minimizing the risk of signal interference or electrical hazards. Additionally, connectors may be color-coded or labeled to facilitate proper alignment and orientation during assembly, reducing the likelihood of errors and ensuring compatibility between components..

4. Lead Wire: Lead wires are essential components in biomedical instrumentation, particularly in electrocardiography (ECG) and other physiological monitoring systems. These wires serve as conduits for transmitting electrical signals between electrodes placed on the patient's body and the monitoring device or amplifier. Lead wires are typically made of flexible and durable materials such as insulated copper or silver conductors, allowing for easy maneuverability and reliable signal transmission without compromising patient comfort. They come in various lengths to accommodate different electrode placements and patient sizes, ranging from short lengths for limb electrodes to longer lengths for chest or back electrodes. Lead wires are equipped with connectors on both ends to facilitate secure and standardized connections between electrodes and monitoring equipment. These connectors may include snap connectors, banana plugs, or DIN connectors[**?**], depending on the specific requirements of the monitoring system and the type of electrodes being used.In addition to transmitting electrical signals, lead wires may also feature color-coded markings or labeling to facilitate proper electrode placement and connection, reducing the risk of errors during setup. Some lead wires may also incorporate shielding or insulation to minimize the risk of electromagnetic interference and ensure the integrity of the transmitted signals.

5. Electrode Size and Shape: Electrodes used in biomedical applications come in various sizes and shapes to accommodate different patient populations and electrode placements. Standard sizes for adult electrodes often range from 1 inch (2.5 cm) to 1.5 inches (3.8 cm) in diameter. Circular and oval shapes are common due to their versatility and ease of application, allowing for consistent contact with the skin. However, rectangular or square-shaped electrodes are also available and may be preferred for specific electrode placements or applications where a larger surface area is required.In addition to standard sizes for adults, pediatric electrodes are available in smaller sizes to suit the anatomical dimensions of infants and small children. These electrodes are typically designed with reduced diameters and may feature specialized shapes or designs to ensure optimal skin contact and comfort for pediatric patients. The selection of electrode size and shape depends on various factors, including the patient's age, body size, and the specific monitoring application. Healthcare professionals carefully consider these factors when choosing the appropriate electrodes to ensure accurate signal acquisition and patient comfort during diagnostic or monitoring procedures. Overall, the availability of electrodes in different sizes and shapes enhances the versatility and effectiveness of biomedical instrumentation across diverse patient populations and clinical settings.

6. Pre-gelled vs Dry Electrodes: Pre-gelled electrodes come pre-coated with a conductive gel or adhesive gel layer, which enhances electrical conductivity and ensures optimal skin contact upon application. These electrodes are convenient to use, as they eliminate the need for separate gel application and reduce preparation time during electrode placement. Pre-gelled electrodes are particularly advantageous for quick and easy setup in clinical settings where efficiency is crucial. They also help to minimize skin irritation or discomfort during prolonged monitoring sessions by providing a smooth and cushioned interface between the electrode and the skin.On the other hand, dry electrodes do not require any conductive gel or adhesive gel layer and rely on dry contact with the skin for signal acquisition. Instead of gel, dry electrodes utilize specialized materials or surface treatments to promote conductivity and maintain stable skin-electrode contact. Dry electrodes offer the advantage of being mess-free and non-sticky, making them more comfortable for patients, especially those with sensitive skin or allergies to gel ingredients. Additionally, dry electrodes are reusable and do not require frequent replacement, leading to cost savings over time. The choice between pre-gelled and dry electrodes depends on various factors, including the specific monitoring application, patient preferences, and clinical requirements. While pre-gelled electrodes offer convenience and immediate readiness, dry electrodes provide a comfortable and cost-effective alternative for long-term monitoring or patients with skin sensitivities. Healthcare professionals carefully consider these factors when selecting the most suitable type of electrode for each patient and clinical scenario, ensuring optimal signal quality and patient comfort during diagnostic or monitoring procedures.

7. Disposable vs Reusable: Disposable and reusable electrodes are two types of electrodes commonly used in biomedical applications for electrocardiography (ECG), Electromyography[**?**] (EMG), and other physiological monitoring systems. Each type has distinct characteristics and advantages, catering to different clinical needs and preferences. Disposable electrodes are designed for single-use applications and are intended to be discarded after a single patient interaction or monitoring session. These electrodes typically come gelled or attached with a conductive gel layer, eliminating the need for additional gel application and reducing setup time. Disposable electrodes offer convenience and hygiene benefits, as they help prevent cross-contamination between patients and reduce the risk of infection transmission in clinical settings. They are particularly suitable for short-term monitoring applications, such as emergency care, ambulatory monitoring, or routine diagnostic tests. On the other hand, reusable electrodes are designed for multiple uses and can be cleaned and sterilized between patient interactions. These electrodes are typically made of durable materials such as metal or plastic and may require separate gel application before each use. Reusable electrodes offer cost-effectiveness and environmental benefits, as they can be used repeatedly over an extended period, reducing waste and lowering overall healthcare costs. They are well-suited for long-term monitoring applications, such as inpatient care, outpatient clinics, or home healthcare settings, where frequent electrode replacement may not be feasible or economical.

8. Electrode Placement: Electrode placement is a critical aspect of electrocardiogram (ECG) monitoring, ensuring accurate signal acquisition and interpretation. Standardized locations are designated for each electrode based on the standard ECG lead system, which includes limb leads and chest leads. Limb leads are typically placed on the arms and legs, while chest leads are positioned on the chest wall [**?**] .Color coding is commonly used to match electrodes with their corresponding leads, facilitating accurate placement by healthcare providers. For example, electrodes for limb leads may be color-coded with red for the right arm, yellow for the left arm, green for the right leg, and black for the left leg. Similarly, chest electrodes are often color-coded to correspond with specific chest lead positions, such as V1 through V6. By adhering to standardized electrode placement and color coding, healthcare providers can ensure consistency and accuracy in ECG monitoring procedures. This not only streamlines the process of electrode application but also helps to prevent errors and misinterpretations of ECG data. Standardized placement and color coding contribute to the efficiency and reliability of ECG monitoring, ultimately enhancing patient care and diagnostic accuracy.

9. Compatibility: Compatibility is a crucial aspect of ECG electrodes, ensuring seamless integration with lead wires and compatibility with a wide range of ECG machines. Modern ECG electrodes often feature universal connectors designed to fit standard ECG lead wires commonly used in clinical settings. These universal connectors provide a standardized interface, allowing electrodes to be easily connected and disconnected from lead wires without the need for specialized adapters or connectors.Furthermore, ECG electrodes are designed to be compatible with a variety of ECG machines from different manufacturers. This compatibility ensures that healthcare providers have flexibility in choosing the appropriate electrodes for their specific monitoring equipment, regardless of the brand or model. Whether used with portable handheld devices, bedside monitors, or advanced diagnostic systems, ECG electrodes are engineered to deliver reliable signal acquisition and compatibility across diverse ECG machine platforms. By offering universal connectors and compatibility with a wide range of ECG machines, ECG electrodes facilitate efficient and interoperable monitoring procedures in clinical environments. Healthcare providers can confidently select electrodes that meet their performance and compatibility requirements, ensuring optimal signal quality and patient care during ECG monitoring and diagnostic procedures.

10. Hygiene and Storage: Proper hygiene and storage are essential to maintain the integrity and performance of ECG electrodes throughout their shelf life.To ensure optimal hygiene and prevent contamination, ECG electrodes are often individually packaged in hygienic packaging materials. This packaging helps maintain sterility and prevents exposure to environmental contaminants, ensuring that the electrodes are clean and safe for use in clinical settings.Additionally, proper storage conditions are essential for preserving the quality of ECG electrodes. Electrodes should be stored in a cool, dry place away from direct sunlight, as exposure to heat and moisture can degrade the electrode materials and compromise their performance. Storing electrodes in a controlled environment helps prevent the gel from drying out or the adhesive from losing its effectiveness over time, ensuring that the electrodes maintain their adhesive properties and conductivity for reliable signal acquisition. By adhering to hygienic packaging practices and proper storage guidelines, healthcare providers can ensure that ECG electrodes remain clean, sterile, and effective for use in patient monitoring and diagnostic procedures. This helps uphold the highest standards of patient safety and quality of care in clinical environments.

Fig. 3.14: ECG Electrode

3.12 Block Diagram of project

1. ECG Leads:

- An ECG lead is an electrical conductor that connects the body to an electrocardiogram (ECG) machine, facilitating the measurement and recording of the heart's electrical activity. These leads are essential components of ECG monitoring systems, providing valuable insights into the heart's rhythm, rate, and conduction pathways. There are two main types of ECG leads: limb leads and chest leads. Limb leads are placed on the limbs of the body and record the electrical activity in the frontal plane, while chest leads, also known as precordial leads, are positioned on the chest wall to capture the electrical activity in the horizontal plane. By analyzing the signals obtained from these leads, healthcare professionals can diagnose various cardiac conditions, monitor the effectiveness of treatments, and assess the overall function of the heart, contributing to the management of cardiovascular health.

2. AD8232 Sensor:

- The AD8232 sensor [**?**] is a highly integrated, single-lead electrocardiogram (ECG) front-end module designed for monitoring cardiac electrical activity in portable and wearable devices. This compact and versatile sensor incorporates essential components such as instrumentation amplifiers, lead-off detection circuitry, and right leg drive (RLD) amplifiers[**?**], providing high-quality ECG signal acquisition while minimizing power consumption. The AD8232 sensor features a low-pass filter to attenuate noise and interference, ensuring accurate ECG signal capture even in noisy environments. Additionally, its built-in right leg drive circuitry helps reduce common-mode interference and improve common-mode rejection ratio, enhancing the signal-to-noise ratio of the acquired ECG signals. With its small form factor, low power consumption, and robust performance, the AD8232 sensor is The AD8232 sensor is ideal for a variety of medical and healthcare applications including remote patient monitoring, security monitoring and Holter monitoring..

3. ESP8266 (WiFi Module):

- The ESP8266 is a highly versatile and compact WiFi module developed by Espressif Systems, renowned for its affordability, simplicity, and robust performance. Integrating a powerful 32-bit microcontroller unit (MCU) with a built-in WiFi radio transceiver, the ESP8266 enables seamless wireless connectivity for a diverse range of embedded applications. Supporting IEEE 802.11 b/g/n WiFi standards, the module facilitates easy integration into WiFi networks, allowing devices to communicate with each other and connect to the Internet. With its extensive GPIO pins [**?**], native TCP/IP protocol stack, and low-power consumption features, the ESP8266 serves as a versatile platform for developing devices, home automation systems, smart appliances, and DIY electronics projects, It is becoming a popular choice among hobbyists, developers and professional developers.

4. UbidotS (Cloud Service):

- The ESP8266 module sends the processed ECG data to the Ubidots cloud service. Data Storage and Processing: Ubidots receives, stores, and processes the ECG data in the cloud. It provides a platform for real-time data visualization, analysis, and monitoring.

5. Ubidots App :

- Ubidots is a leading cloud-based Internet of Things (IoT) platform that empowers users to easily collect, store, visualize, and analyze sensor data in real-time. With its intuitive interface and comprehensive suite of features, Ubidots enables users to build scalable IoT applications and solutions quickly and efficiently. From data collection and storage to visualization and analysis, Ubidots provides a seamless end-to-end solution for IoT development and deployment. Its ordered dashboards, powerful analytics tools, and integration options allow users to gain valuable insights from their IoT data and drive informed decision-making. Whether for industrial monitoring, environmental sensing, smart cities, or consumer IoT applications, Ubidots provides the tools and capabilities needed to Unleash the full potential of IoT and drive innovation across industries and markets.

- The patient's ECG signals are captured by the electrodes (ECG Leads).

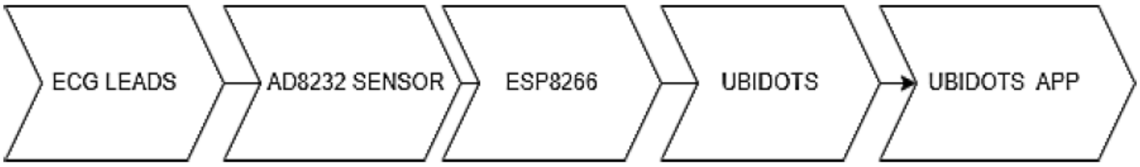


Fig. 3.15: Block Diagram

- The AD8232 Sensor processes these signals, making them suitable for digital processing.

- The ESP8266 module establishes a wireless connection to the internet and sends the processed ECG data to the Ubidots cloud service.

- Ubidots stores, analyzes, and visualizes the ECG data in real-time on its platform.

- The Ubidots app, accessible through web or mobile devices, provides a user-friendly interface for monitoring the patient's ECG graphs and vital statistics.

This project enables continuous, real-time monitoring of a patient's ECG signals remotely. Healthcare providers or caregivers can access the Ubidots app to keep track of the patient's heart health, receive alerts for any abnormalities, and take necessary actions promptly.

3.13 Experimental Configuration

We've designed a system using three ECG leads to capture important aspects of the heart's electrical activity. Previous research suggests that this arrangement is enough to get the key details we need. The trick is to place the electrodes in a triangle shape around the heart, as shown in Figure 1 of our system. We wanted to make sure our system is both stable and accurate, so we tested it on a healthy volunteer. This volunteer wore the three electrodes in the triangle setup while doing different activities. The system worked well, showing clear heart rate changes and waveform patterns during rest, light

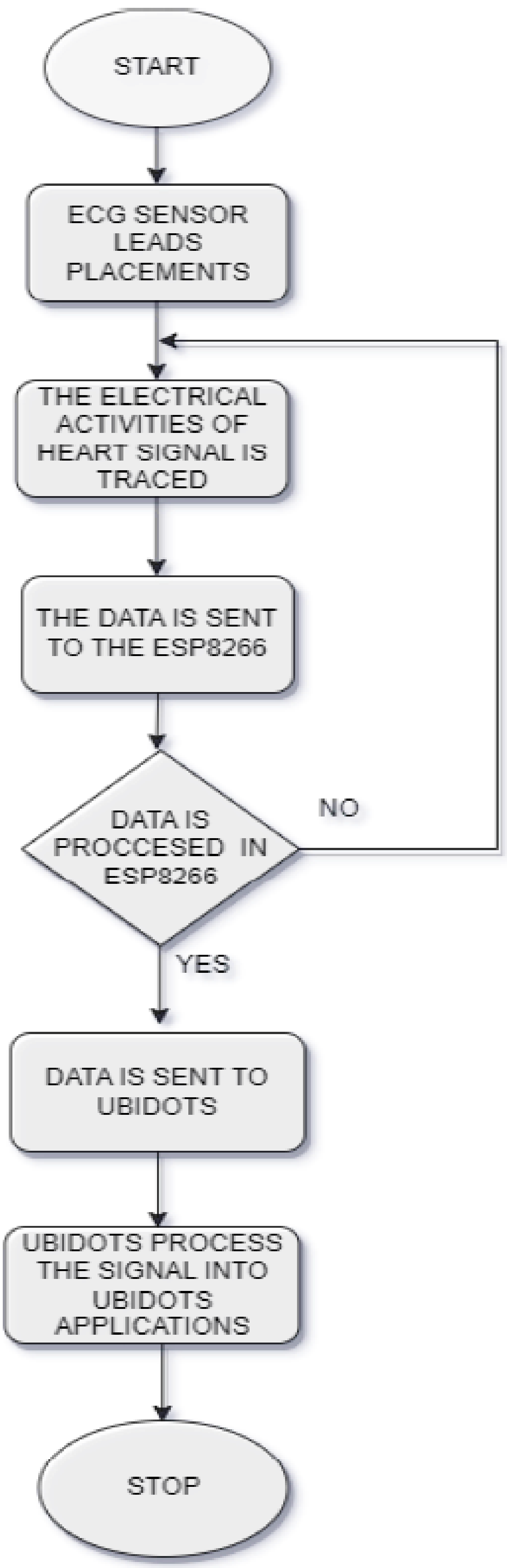


Fig. 3.16: Process Flow

exercise, and deep breathing. This setup seems promising for keeping an eye on heart health, especially in everyday settings. The next step? We plan to test it with more people to see how well it works for different folks.

Typical ECG signals are made up of five types of waves: P surge, T surge, Q surge, R surge, and S surge. These waves intervals are commonly utilised to diagnose a number of heart conditions. Among all the features of these waves, four are most commonly used in medical diagnosis.

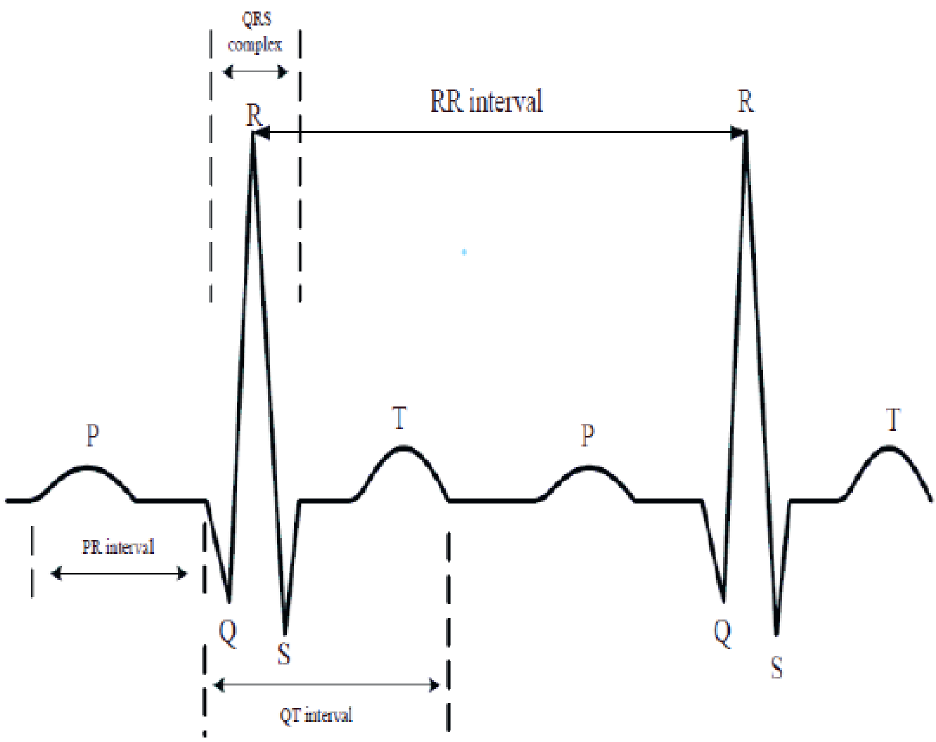


Fig. 3.17: Standard ECG Signal

- **RR Break:** The RR break, also known as the RR interval, is a crucial measure in analyzing an ECG signal, particularly because it signifies the time between two consecutive R waves. The R wave, being a prominent feature of the ECG waveform, is often used as a reference point due to its consistency and clear identification. The RR interval reflects the heart's rhythm, with variations indicating irregularities that might signal underlying heart condition[**?**]. By measuring the duration between R waves, healthcare providers can assess the regularity and stability of the heart's electrical activity. In cases of arrhythmia or other cardiac abnormalities, the RR interval may show irregular patterns, providing valuable diagnostic information for appropriate medical intervention.

- **PR Break:** The PR break, known as the PR interval, is a fundamental component of an ECG signal, indicating the time between the beginning of the P wave and the start of the QRS complex. This interval provides insight into the conduction time from the atria (upper chambers of the heart) to the ventricles (lower chambers). Essentially, it measures the duration it takes for the electrical impulse generated by the sinoatrial (SA) node to travel through the atria, pass through the atrioventricular (AV) node, and reach the ventricles to trigger their contraction. A normal PR interval reflects a healthy electrical conduction system, ensuring coordinated heart contractions. Deviations from the normal range can indicate conditions such as heart block, where the impulse is delayed or blocked in its path, affecting the heart's rhythm and efficiency. Monitoring the PR interval helps clinicians assess the integrity of the heart's electrical pathways and identify potential cardiac abnormalities.

- **QT Break:** The QT break, referred to as the QT interval, is a critical measure on an ECG representing the time from the beginning of the Q wave to the end of the T wave. This interval signifies the duration of both ventricular depolarization (contraction) and relaxation of the heart. It reflects the time it takes for the heart's electrical activity to stimulate the ventricles to contract and then recover to their resting state. A normal QT interval indicates proper coordination and timing of these electrical events. However, if the QT interval extends beyond its expected value, it may indicate a higher risk of ventricular arrhythmias, such as ventricular fibrillation. Ventricular fibrillation is a dangerous and potentially life-threatening condition where the heart's rhythm becomes chaotic, leading to inadequate blood flow and sudden cardiac arrest. Monitoring the QT interval is crucial in assessing the risk of these serious cardiac events, guiding clinicians in preventive measures and appropriate treatment strategies.

- **QRS Break:** The QRS break, or the QRS complex, is a key component of an ECG signal that reflects the electrical activity during ventricular depolarization, the phase when the ventricles contract. It consists of three main waves: the Q wave, the R wave (the largest deflection), and the S wave. The QRS complex represents the rapid spread of electrical impulses through the ventricles, leading to their contraction and the ejection of blood from the heart. Analyzing the shape, duration, and amplitude of the QRS complex provides valuable information about the heart's health and function. Certain conditions, such as electrolyte imbalances or drug toxicities, can manifest as changes in the QRS complex. For example, an enlarged QRS complex might indicate a blockage in the heart's electrical pathways, while a widened or prolonged QRS complex could suggest conditions like bundle branch blocks. These insights help clinicians diagnose and manage various cardiac disorders, allowing for timely interventions and appropriate treatment plans.

3.14 Key Challenges of ECG Monitoring System

The ECG monitoring systems, as discussed in this paper, are constructed from a diverse array of components, frameworks, and technologies. This diversity and heterogeneity in ECG sensor-based architectures present several challenges, as highlighted by numerous researchers in the field. A multitude of obstacles can arise, including interoperability issues between different sensor types and data formats. The integration of various sensors and devices into a cohesive system can pose technical complexities, leading to compatibility issues and difficulties in data synchronization. Furthermore, the management More ECG data, often generated on the fly, requires robust storage solutions and efficient data processing mechanisms. Additionally, ensuring the security and privacy of sensitive patient data in these systems remains a paramount concern. Addressing these challenges necessitates collaborative efforts among researchers, engineers, and healthcare professionals to develop standardized protocols, seamless integration frameworks, and robust data management strategies for reliable and effective ECG monitoring systems [**?**].

There are a variety of obstacles that can be encountered, including the following:

1. Signal Quality Issues: One of the primary challenges with ECG monitoring devices is ensuring consistent and reliable signal quality. Factors such as poor electrode contact with the skin, motion artifacts, and external interference can degrade the signal. These issues can lead to inaccuracies in the recorded ECG data, making it challenging to obtain precise diagnostic information.

2. Difficulties with Durability Monitoring: Monitoring devices, especially those used in ambulatory or continuous monitoring settings, need to withstand daily wear and tear. Durability becomes crucial to ensure that the devices remain functional and accurate over extended periods. Factors such as battery life, water resistance, and robustness against physical damage are key considerations.

3. Issues with the Size of ECG Signal Data: ECG signals generate a significant amount of data, especially in continuous monitoring applications. Storing and managing these large datasets pose challenges in terms of storage capacity and data transfer. Efficient compression techniques and optimized data transmission protocols are essential to handle the sizeable ECG signal data efficiently.

4. Visualization-Related Difficulties: Interpreting ECG graphs and trends is critical for healthcare providers to make accurate diagnoses and treatment decisions. However, the complexity of ECG data visualization, especially in real-time monitoring, can be a challenge. Clear and intuitive graphical representations are needed to convey information effectively without overwhelming the user.

5. Difficulties with System Integration: ECG monitoring devices often need to integrate with other healthcare systems, such as Electronic Health Records (EHR) or platforms. Achieving seamless integration requires compatibility with different data formats, communication protocols, and interoperability standards. Ensuring that data from ECG devices can be easily shared, accessed, and analyzed within broader healthcare systems is crucial for efficient patient care.

3.15 Addressing the Challenges

1. Signal Quality Enhancement: Advanced signal processing algorithms can help mitigate noise and artifacts in ECG signals, improving overall signal quality. Use of high-quality electrodes and proper skin preparation techniques can enhance electrode-skin contact, reducing signal interference.

2. Improved Device Durability: Designing monitoring devices with robust materials and construction can enhance durability. Incorporating features such as waterproofing, shock resistance, and long-lasting batteries improves device reliability.

3. Efficient Data Management: Employing data compression algorithms and cloud-based storage solutions can address the challenges of large ECG signal datasets. Implementing data encryption and secure transmission protocols ensures data security and privacy.

4. Enhanced Visualization Tools: Developing user-friendly interfaces with intuitive ECG waveform displays aids healthcare providers in interpreting data effectively. Real-time trend analysis and alarm systems can help highlight abnormalities for timely intervention.

5. Interoperability and System Integration: Adhering to standard data formats such as HL7 (Health Level Seven) or DICOM (Digital Imaging and Communications in Medicine) promotes interoperability. Developing Application Programming Interfaces (APIs) and middleware solutions facilitate seamless integration with existing healthcare IT systems.

By addressing these challenges through technological innovations, improved device design, and standardized protocols, ECG monitoring devices can become more reliable, user-friendly, and effective tools in the realm of healthcare, enabling better patient care and outcomes.

Continuous remote monitoring through wireless connectivity options like Bluetooth or WiFi allows healthcare providers to track patients ECG signals in real-time, facilitating early detection of abnormalities and timely intervention. Personalized alert systems, tailored to individual patient profiles and medical history, ensure that relevant notifications are delivered promptly, enhancing patient management and outcomes. Integration of artificial intelligence (AI) algorithms enhances diagnostic accuracy and predictive capabilities by analyzing ECG data in real-time, aiding healthcare providers in making informed clinical decisions. Wearable form factors such as smartwatches or patches improve patient comfort and compliance with long-term monitoring protocols, enabling continuous monitoring during daily activities. E-medicine integration enables remote consultation and diagnosis, particularly beneficial in poor areas, enhancing access to specialized cardiac care and reducing healthcare disparities. Long-term data trends analysis tools track changes in patients cardiac health over time, guiding proactive treatment decisions based on subtle changes or collapse in cardiac function. Patient education materials and interactive features within ECG monitoring devices promote patient engagement and empowerment, leading to better adherence to treatment plans and improved health outcomes.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 ECG Monitoring Through Ubidots

The project successfully implements the AD8232 sensor for ECG signal analysis. Data is sent to an IoT platform, allowing easy at-home heart rate monitoring for patients. Healthcare providers can remotely assess heart health through the platform. The project aims for a user-friendly system, emphasizing real-time data transmission. The circuit diagram illustrates the sensor's integration for efficient ECG analysis [**?**]
The AD8232 sensor, crucial for this monitoring system, features nine connection points that allow for soldering pins, wires, or other connectors. Among these, the LO+, LO-, OUTPUT, SDN, 3.3V, and GND pins are essential for operation alongside the Nodemcu ESP8266 microcontroller, enabling data transmission and reception. Specifically, five general pins must be connected to the microcontroller: the output links to analog pin A0 of the Nodemcu[**?**], while LO- and LO+ are attached to pins D6 and D5, respectively. Finally, the sensor is powered by connecting the supply pin to 3.3V and grounding the device for proper functionality within the monitoring setup.

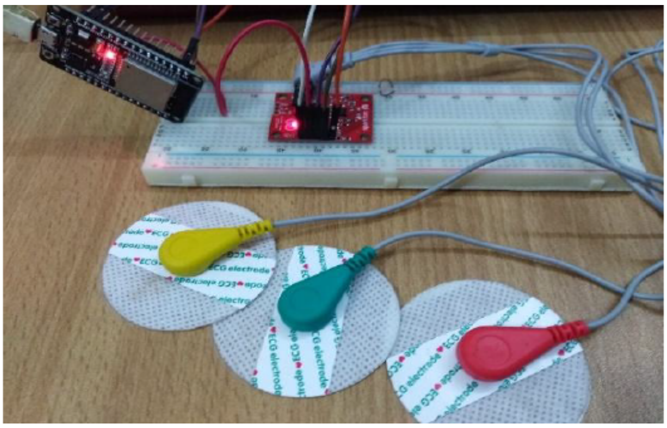


Fig. 4.1: Prototype of Project

As shown in Figure 4.2, the data is published into the Ubidots, and it is viewed in serial monitor once the MQTT connection is secured and connects successfully.

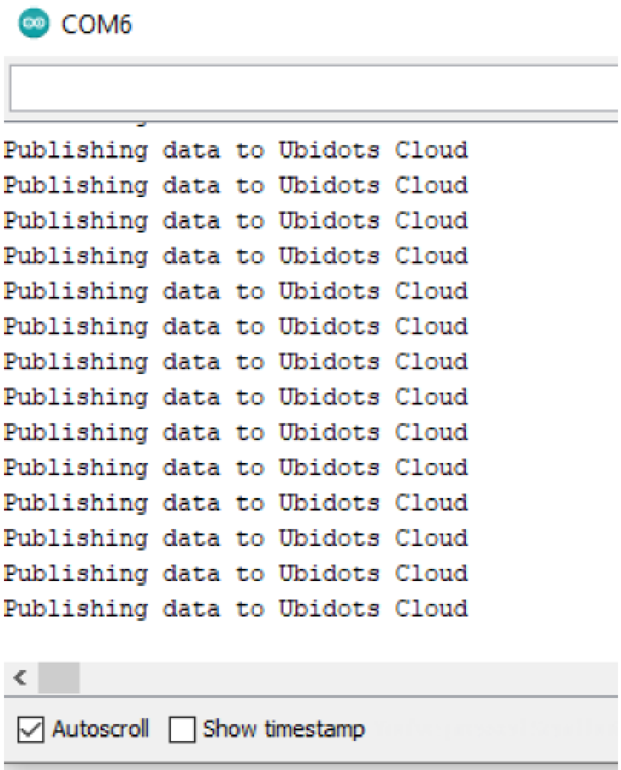


Fig. 4.2: The publishing of data into Ubidots

The process of ECG signal analysis involves identifying specific points within the ECG waveform to understand the heart's electrical activity. The R-peak, which represents the peak of the QRS complex, is a crucial point in the ECG signal. It signifies the ventricular depolarization, the moment when the main pumping chambers of the heart contract to push blood out. The R-peak is recognized as the highest point in each cycle of the ECG waveform, and it serves as a key reference point for analyzing heart rate and rhythm. The QRS complex, comprising the Q, R, and S waves, provides important information about the electrical conduction through the ventricles. The span of the QRS complex is determined from the trigger point before the R-peak to the one after it. The Q wave represents the initial downward deflection, the R wave is the main upward deflection, and the S wave follows the R wave as a downward deflection. These waves collectively show the spread of electrical impulses across the ventricles, leading to their contraction. Additionally, the T and S waves are crucial components of the

ECG waveform. The T wave represents the repolarization of the ventricles, the moment when they relax and prepare for the next heartbeat. It is located after the QRS complex, usually in the upward direction. The S wave, on the other hand, is the downward deflection immediately following the R wave, indicating the end of ventricular depolarization. "Real-time data" in this context refers to the immediate processing and display of ECG information for quick user response. In this system, as soon as the ECG signals are captured and analyzed, the results are presented to the user without delay. This allows for continuous and up-to-date monitoring of the heart's electrical activity, providing timely insights into heart rate, rhythm, and potential abnormalities. Healthcare providers can promptly assess the ECG data, make informed decisions, and intervene if necessary for effective heart health management. The system's ability to offer continuous, real-time data presentation ensures that any changes or irregularities in the ECG signals can be promptly detected and addressed, enhancing patient care and monitoring.

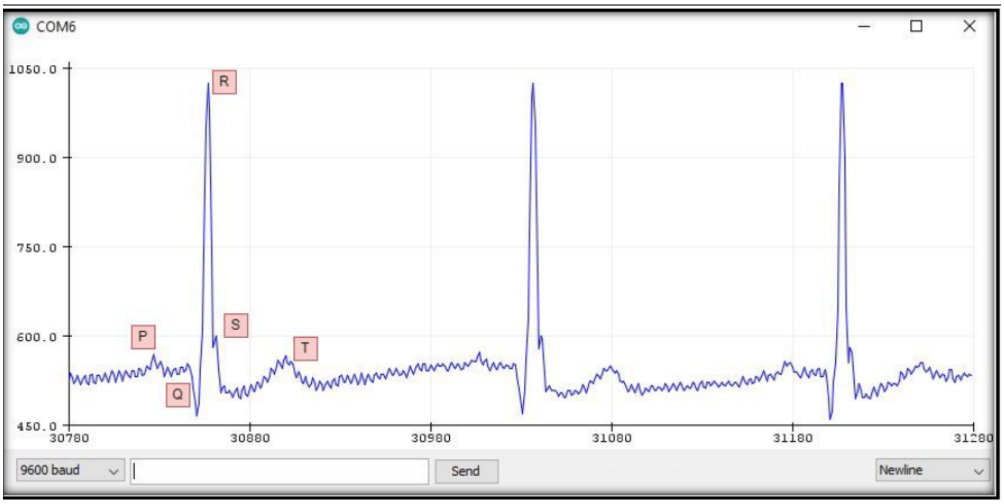


Fig. 4.3: The Signal in Serial Monitor

The dashboard of Ubidots, as depicted in Figure 4.4, serves as a visual representation of the heart signals captured and analyzed from the sensor data. Here, users can observe the ECG signals generated from the heart of subject 1, showcasing the characteristic waves of P, Q, R, S, and T in their normal, steady-state positions. Each of these waves corresponds to a specific phase of the cardiac cycle, providing valuable insights into the heart's electrical activity [**?**].

- The P wave represents the atrial depolarization, the contraction of the atria to push blood into the ventricles.

- The QRS complex, consisting of the Q, R, and S waves, signifies the ventricular depolarization, the contraction of the main pumping chambers of the heart.

- The T wave reflects the ventricular repolarization, the relaxation and resetting of the ventricles for the next heartbeat.

Interpreting these waves is crucial as it allows healthcare providers to understand the heart's rhythm, rate, and overall function. An ECG tracing can reveal if the heart is beating too slowly (bradycardia), too fast (tachycardia), or irregularly (arrhythmia). This information is vital for diagnosing various cardiac conditions, monitoring treatment effectiveness, and identifying potential abnormalities. Furthermore, the data transmitted to Ubidots can also be extracted and analyzed in Excel, providing a more detailed and ordered view of the ECG signals. Healthcare professionals can use this data to track trends over time, compare multiple recordings, and make informed decisions about patient care.

Fig. 4.4: Dashboard of Ubidots

4.2 Depression Detection Through ML Model

The machine learning model achieved an accuracy of 84% on the training dataset and 77.34% on the test dataset. This performance indicatesMore ECG data, often generated on the fly. However, the slight drop in accuracy from training to testing suggests that the model might be slightly providing training to the training data, meaning it may have learned to memorize patterns specific to the training set rather than capturing unspecified patterns. To address this, techniques such as regularization, data augmentation, or adjusting model complexity could be explored to improve the model's performance on unseen data. Additionally, further analysis of misclassified instances in the test set could provide insights into areas where the model struggles and guide further refinement efforts.[**?**]

```
validation = model.predict(X_test)
```

```
validation1 = model.predict(X_train)
```

```
from sklearn.metrics import accuracy_score
accuracy_score(y_train, validation1)
```

```
0.843316447368421
```

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test, validation)
```

```
0.7736
```

Fig. 4.5: Result of ML Model

In the realm of future research endeavors, the exploration of alternative machine learning and deep learning techniques stands poised as a promising avenue for enhancing prediction accuracy. Leveraging the expansive landscape of methodologies, including ensemble methods, convolution neural networks, and recurrent neural networks, holds the potential to unlock deeper insights and finer predictions. Furthermore, the acquisition of a comprehensive dataset, rich in diverse instances, presents an opportunity to fortify experimental endeavors. By amassing a wealth of data, researchers can delve into the nuances of complex patterns, thereby refining models to unprecedented levels of precision and robustness. This multifaceted approach not only promises to elevate predictive performance but also to push the boundaries of understanding within the field, fostering innovation and advancement.

```
In [24]: train = pd.DataFrame()
         train['label'] = y_train
         train['text'] = X_train

         def predict_category(s, train=X_train, model=model):
             pred = model.predict([s])
             return pred[0]

In [25]: predict_category("i wanna shot myself")

Out[25]: 0

In [26]: predict_category("i Kill you")

Out[26]: 0

In [27]: predict_category("I'm cute")

Out[27]: 1
```

Fig. 4.6: Predictions of ML Model

This project aimed to harness sentiment analysis as a tool for detecting depression, utilizing a dataset sourced from Kaggle. The dataset underwent meticulous processing and cleansing to ensure its readiness for analysis. Subsequently, the dataset was partitioned into 80 % for training and 20 % for testing purposes. Employing renowned machine learning algorithms, such as Naive Bayes, the experiments unfolded, yielding compelling results. Notably, the Naive Bayes technique emerged as a frontrunner, showcasing a classification accuracy exceeding 84 %. This achievement underscores the efficacy of sentiment analysis in discerning indicators of depression, marking a significant stride in leveraging computational techniques for mental health assessment.

Our project's website serves as an interactive platform where users can assess and improve their mental well-being through the innovative application of machine learning. Upon visiting the site, users encounter a series of meticulously crafted questions designed to delve into various aspects of their emotional and mental health. These questions cover a wide range of topics, including mood fluctuations, energy levels, quality of sleep, patterns of social interaction, and more. Users provide their responses,

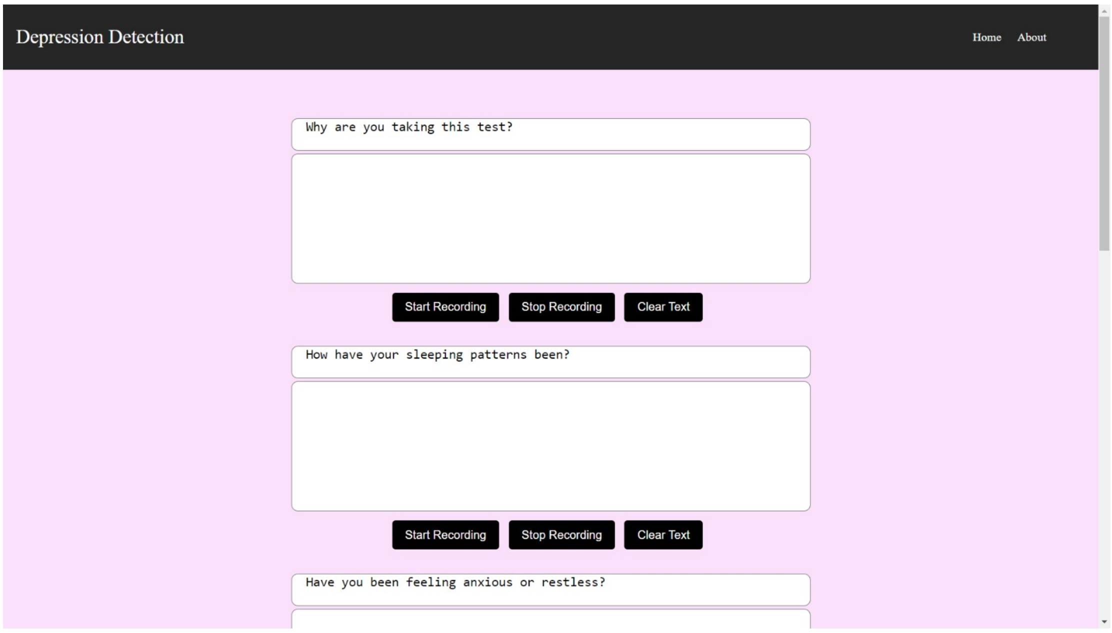


Fig. 4.7: Web Page of our Project

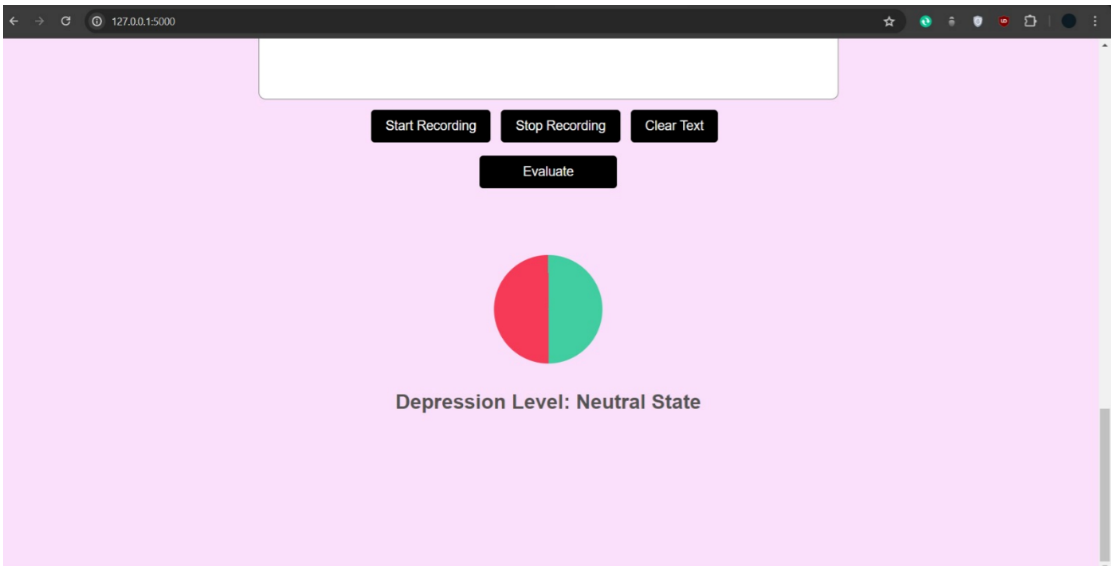


Fig. 4.8: Evaluation of Depression

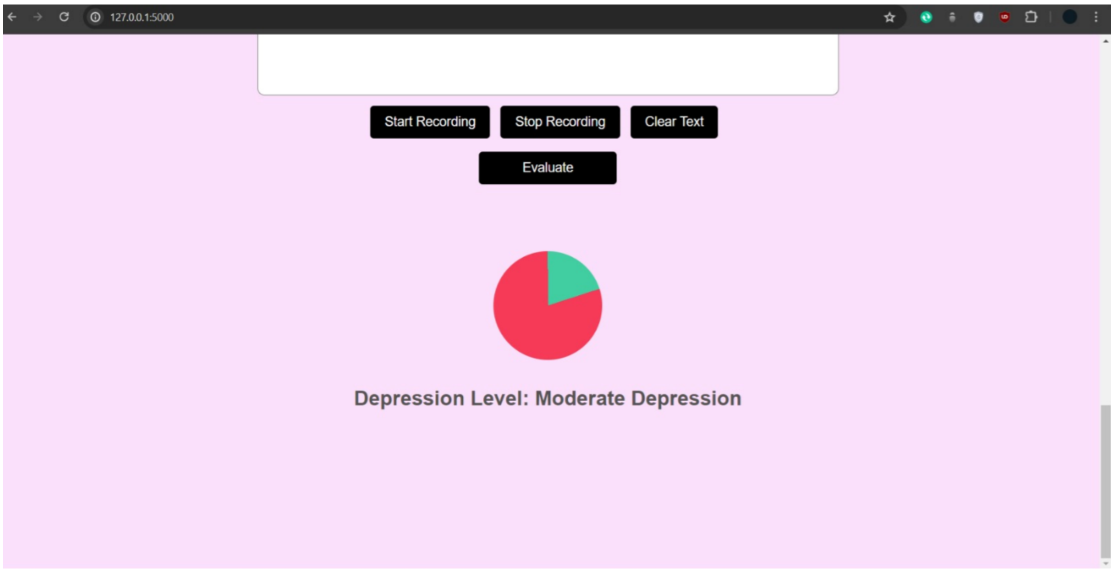


Fig. 4.9: Result of Depression Detection

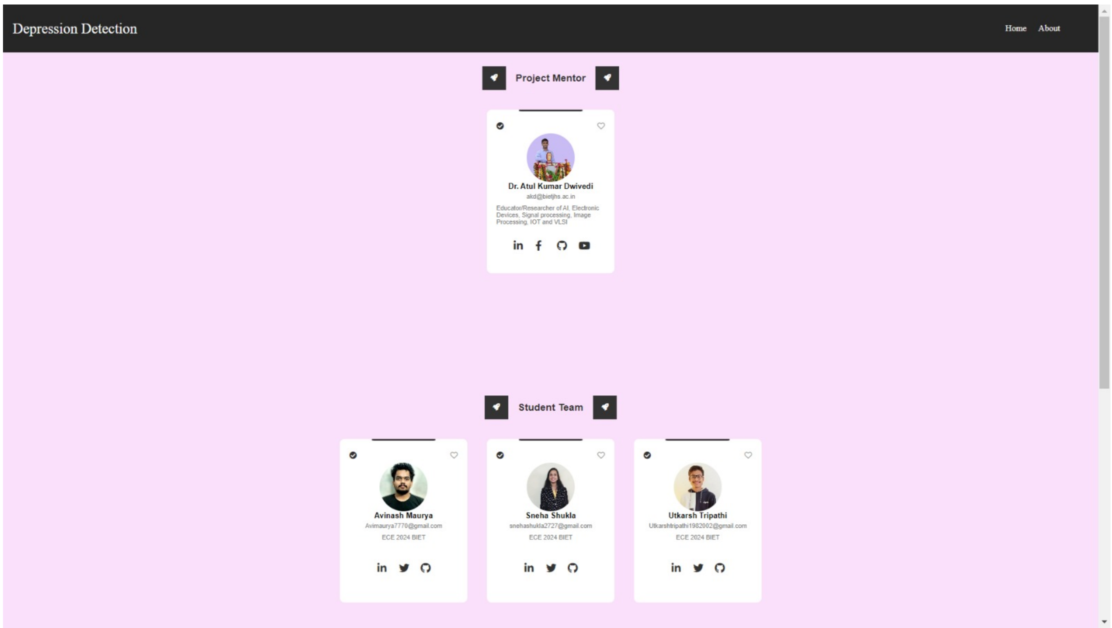


Fig. 4.10: About section of web page

which are then meticulously analyzed by sophisticated machine learning algorithms. Through this analysis, the platform generates personalized insights and guidance tailored to each user's unique mental health profile. These insights not only offer valuable self-awareness but also empower users to make informed decisions to enhance their overall well-being. Additionally, the platform may provide recommendations for further resources, such as articles, exercises, or professional support, based on the individual's responses and identified needs. By leveraging the power of machine learning, our platform aims to revolutionize mental health care, offering accessible, personalized support to users worldwide.

Upon completing the questionnaire, users responses undergo thorough analysis by a machine learning model to ascertain their depression level. This analysis generates a score on a scale from 0 to 1, indicating varying degrees of depression severity. Scores falling between 0 and 0.15 denote severe depression, while those between 0.15 and 0.35 suggest moderate depression. A range of 0.35 to 0.65 indicates a neutral state, whereas scores from 0.65 to 0.85 signify a stable condition. Finally, a score between 0.85 and 1 represents an optimal mental state. This nuanced scoring system enables users to gain precise insights into their mental health status, guiding them towards appropriate interventions and support tailored to their individual needs.

The web page provides users with a user-friendly platform to evaluate their mental health status independently and obtain personalized feedback through machine learning analysis. This streamlined approach offers individuals a convenient and accessible means to assess and track their mental well-being. By leveraging advanced algorithms, the platform delivers valuable insights and guidance tailored to each user's responses, empowering them to monitor and manage their mental health effectively. This innovative approach represents a significant step forward in mental health care, offering individuals the tools and support they need to prioritize their well-being in a proactive and informed manner.

4.3 Result of Audio based Depression Detection

The Convolution Neural Network (CNN) model implemented for audio-based depression detection exhibited a notable achievement with an accuracy of 82% in correctly classifying individuals as either depressed or non-depressed based on acoustic features extracted from speech recordings. This accuracy signifies a promising step forward in leveraging machine learning techniques for mental health assessment. However, despite this moderate success, several avenues for enhancing the model's performance exist.

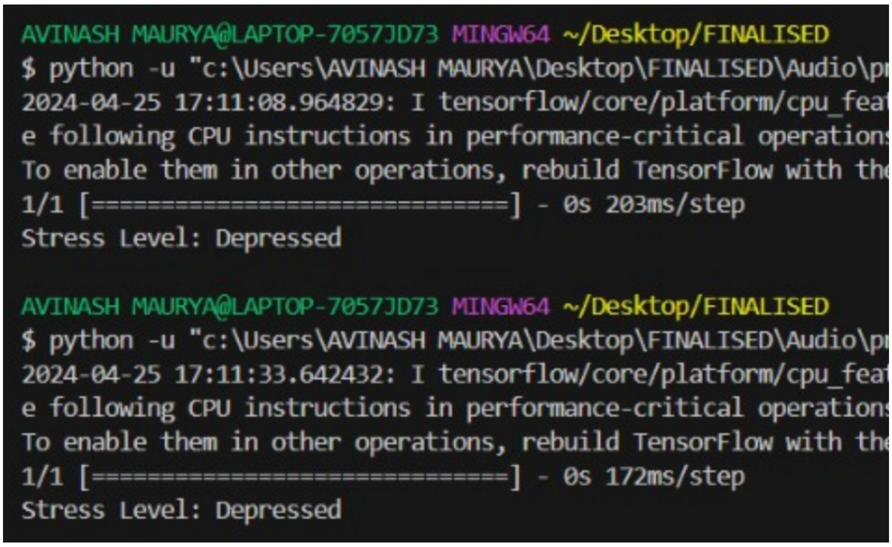


Fig. 4.11: Prediction of Audio Based Model

The Convolution Neural Network (CNN) model developed using TensorFlow and Keras libraries for audio classification achieved promising results. After thorough data processing, which involved extracting features like Mel-frequency cepstral coefficients (MFCCs) from audio files and augmenting the dataset with various techniques, the model was constructed with convolution and pooling layers followed by dense layers. During training, the model was put together using categorical cross-entropy loss and the Adam optimizer, achieving a significant accuracy rate. Evaluation on the test data yielded impressive metrics such as accuracy and F1 score, indicating the model's robustness in classifying emotions from audio recordings. The confusion matrix visualization further illustrated the model's performance, providing insights into its ability to accurately predict different emotional states. Overall, this comprehensive script show-cases the effectiveness of deep learning techniques in audio classification tasks and highlights the potential for further advancements in this domain [**?**].

Firstly, refining the feature extraction process could involve delving deeper into the acoustic characteristics of speech that correlate with depression, potentially incorporating advanced signal processing techniques. Secondly, optimizing the network architecture might entail experimenting with different CNN configurations or exploring hybrid models that combine CNN with other types of feedforward networks. Moreover, augmenting the size and diversity of the training dataset could address potential biases and improve the model's robustness across various demographic groups. Additionally, while the 82% accuracy suggests the model's potential utility, further research is warranted to validate its effectiveness in real-world clinical settings and to ascertain its specification across different populations, considering factors Including differences in age, gender, cultural background and language. This comprehensive evaluation is essential for ensuring the CNN model's reliability and practical applicability in supporting mental health diagnosis and intervention strategies.

The CNN model demonstrated a 82% accuracy in discerning between depressed and non-depressed individuals using audio features from speech recordings, indicating a moderate level of effectiveness. However, there are opportunities for improvement in refining feature extraction, optimizing the network architecture, and enlarging the training dataset. Further research is necessary to evaluate the model's suitability across various populations and in clinical contexts. The figure provides a visual representation of how the CNN model categorizes individuals into depressed and non-depressed groups.

CHAPTER 5

CONCLUSION

In our project, we embarked on a multifaceted journey to revolutionize mental health assessment and remote patient monitoring by integrating sentiment analysis and real-time ECG monitoring. Through rigorous experimentation, we achieved significant milestones with implications for mental health care. Leveraging machine learning techniques, we trained models to discern depression indicators from textual data, with the Naive Bayes technique achieving an impressive classification accuracy of over 84%. Simultaneously, our venture into real-time ECG monitoring, facilitated by the AD8232 sensor and Internet integration[**?**], provided healthcare providers with timely insights into patients' cardiovascular health, enabling seamless at-home monitoring. This integration of sentiment analysis and ECG monitoring lays the groundwork for a holistic approach to remote patient care, with future research poised to explore further fusion of sentiment analysis with physiological data and the development of user-friendly interfaces to enhance accessibility.

Furthermore, our project expanded its scope by integrating audio-based depression detection through Convolution Neural Network[**?**] (CNN) technology, achieving an impressive accuracy rate of 82%. This addition not only bolstered our efforts in mental health assessment but also showcased the potential of advanced deep learning techniques in identifying depression based on acoustic features extracted from speech recordings. Our project represents a groundbreaking convergence of technology and healthcare, paving the way for a brighter, healthier future for individuals worldwide. By combining sentiment analysis, ECG monitoring, and audio-based depression detection, we laid a solid foundation for a comprehensive- personalized approach to remote patient care, with the potential to revamp the landscape of mental health assessment and cardiovascular care, ultimately improving the lives of countless individuals.

APPENDIX A

Important Codes/Analysis(1)

A.1 Jupyter Coding

```
import numpy as np
import pandas as pd from matplotlib
import pyplot as plt from sklearn.model_selection
import train_test_split from sklearn.naive_bayes
import GaussianNB, MultinomialNB from sklearn.metrics
import accuracy_score, confusion_matrix, classification_report
from sklearn.feature_extraction.text
import TfidfVectorizer from sklearn.pipeline
import make_pipeline
import string
import nltk
import re
import seaborn as sns

dataset_columns = ["target", "ids", "date", "flag", "user", "text"]
dataset_encode = "ISO-8859-1"

df = pd.read_csv("training.1600000.processed.noemoticon.csv",
encoding=dataset_encode, names=dataset_columns)
df.head()

df.drop(['ids', 'date', 'flag', 'user'], axis=1, inplace=True)
df['target'].value_counts()

# Remove punctuation
def remove_punctuation(text):
    no_punct = [words for words in text if words not in string.punctuation]
    words_wo_punct = ''.join(no_punct)
    return words_wo_punct
```

```
df['clean_text'] = df['text'].apply(lambda x: remove_punctuation(x))
df.head()

# Remove hyperlink
df['clean_text'] = df['clean_text'].str.replace(r"http\S+", "")

# Remove emoji
df['clean_text'] = df['clean_text'].str.replace('[^\w\s#@/:%.,_-]', '',
flags=re.UNICODE)

# Convert all words to lowercase
df['clean_text'] = df['clean_text'].str.lower()
df.head()

# Tokenization
nltk.download('punkt')

def tokenize(text):
    split = re.split("\W+", text)
    return split

df['clean_text_tokenize'] = df['clean_text'].apply(lambda x: tokenize(x.lower()))

import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

stopword = nltk.corpus.stopwords.words('english')

def remove_stopwords(text):
    text = [word for word in text if word not in stopword]
    return text

df['clean_text_tokenize_stopwords'] =
df['clean_text_tokenize'].apply(lambda x: remove_stopwords(x))
df.head(10)
```

```
new_df = pd.DataFrame()
new_df['text'] = df['clean_text']
new_df['label'] = df['target']
new_df['label'] = new_df['label'].replace(4, 1)
print(new_df.head())
print('Label:␣\n', new_df['label'].value_counts())

from sklearn.model_selection import train_test_split

X = new_df['text']
y = new_df['label']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.05,
random_state=42)
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
y_train.value_counts()

model = make_pipeline(TfidfVectorizer(), MultinomialNB())
model.fit(X_train, y_train)

validation = model.predict(X_test)
validation1 = model.predict(X_train)

from sklearn.metrics import accuracy_score

accuracy_score(y_train, validation1)

from sklearn.metrics import accuracy_score

accuracy_score(y_test, validation)

cf_matrix = confusion_matrix(y_test, validation)
sns.heatmap(cf_matrix / np.sum(cf_matrix), annot=True, fmt='.2%', cmap='Greens')
print(classification_report(y_test, validation))

train = pd.DataFrame()
```

```
train['label'] = y_train
train['text'] = X_train

def predict_category(s, train=X_train, model=model):
    pred = model.predict([s])
    return pred[0]

predict_category("i␣wanna␣shot␣myself")
predict_category("i␣love␣you")
```

APPENDIX B

Important Codes/Analysis(2)

B.1 Code For ESP 8266

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

#define WIFISSID "Mynetwork"   // Put your WifiSSID here
#define PASSWORD "12348765"   // Put your wifi password here
#define TOKEN "BBUS-guLu2VyebqiOKpvpZGSsS4y2VfUjfa"
#define MQTT_CLIENT_NAME "MyProject"

/****************************************
 * Define Constants
 ****************************************/
#define VARIABLE_LABEL "myecg" // Assing the variable label
#define DEVICE_LABEL "esp8266" // Assig the device label

#define SENSOR A0 // Set the A0 as SENSOR

char mqttBroker[]  = "industrial.api.ubidots.com";
char payload[100];
char topic[150];
// Space to store values to send
char str_sensor[10];

/****************************************
 * Auxiliar Functions
 ****************************************/
WiFiClient ubidots;
PubSubClient client(ubidots);

void callback(char* topic, byte* payload, unsigned int length) {
```

```
  char p[length + 1];
  memcpy(p, payload, length);
  p[length] = NULL;
  Serial.write(payload, length);
  Serial.println(topic);
}

void reconnect() {
    // Loop until we're␣reconnected
␣␣␣␣while(!client.connected())␣{
␣␣␣␣␣␣Serial.println("Attempting␣MQTT␣connection...");

␣␣␣␣␣␣␣␣//␣Attempt␣to␣connect
␣␣␣␣␣␣␣␣if␣(client.connect(MQTT_CLIENT_NAME,␣TOKEN,␣""))␣{
␣␣␣␣␣␣␣␣␣␣␣␣Serial.println("Connected");
␣␣␣␣␣␣␣␣}␣else␣{
␣␣␣␣␣␣␣␣␣␣␣␣Serial.print("Failed,␣rc=");
␣␣␣␣␣␣␣␣␣␣␣␣Serial.print(client.state());
␣␣␣␣␣␣␣␣␣␣␣␣Serial.println("␣try␣again␣in␣2␣seconds");
␣␣␣␣␣␣␣␣␣␣␣␣//␣Wait␣2␣seconds␣before␣retrying
␣␣␣␣␣␣␣␣␣␣␣␣delay(2000);
␣␣␣␣␣␣␣␣}
␣␣␣␣}
}


/****************************************
␣*␣Main␣Functions
␣****************************************/
void␣setup()␣{
␣␣␣␣Serial.begin(115200);
␣␣␣␣WiFi.begin(WIFISSID,␣PASSWORD);
␣␣␣␣//␣Assign␣the␣pin␣as␣INPUT
␣␣␣␣pinMode(SENSOR,␣INPUT);

␣␣␣␣Serial.println();
␣␣␣␣Serial.print("Waiting␣for␣WiFi...");
```

```
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(500);
    }

    Serial.println("");
    Serial.println("WiFi Connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
    client.setServer(mqttBroker, 1883);
    client.setCallback(callback);
}

void loop() {
    if (!client.connected()) {
        reconnect();
    }

    sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
    sprintf(payload, "%s", ""); // Cleans the payload
    sprintf(payload, "{\"%s\":", VARIABLE_LABEL); // Adds the variable label

    float myecg = analogRead(SENSOR);

    /* 4 is minimum width, 2 is precision; float value is copied onto str_sensor*/
    dtostrf(myecg, 4, 2, str_sensor);

    sprintf(payload, "%s {\"value\": %s}}", payload, str_sensor); // Adds the value
    Serial.println("Publishing data to Ubidots Cloud");
    client.publish(topic, payload);
    client.loop();
    delay(10);
}
```

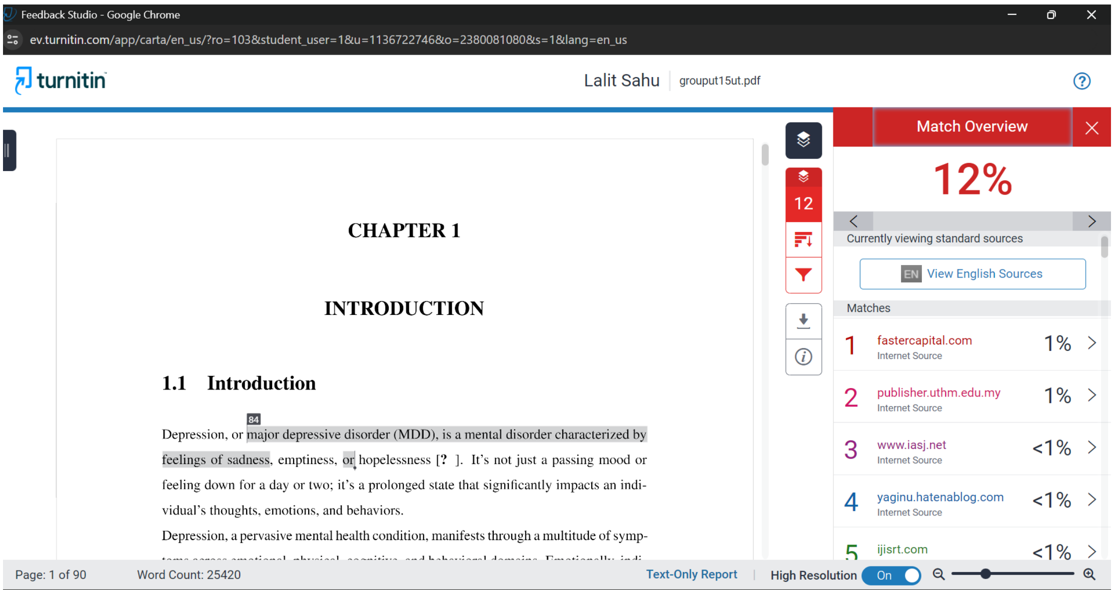
APPENDIX C

Weekly Report

W. N.	Week Dates	Work Done
1	15-09-23 to 20-09-23	We all did the research work
2	01-10-23 to 15-10-23.	We all collected the base paper
3	04-11-23	We all give our introduction presentation to the Faculty member
4	06-01-24 to 22-01-24	We extracted the required dataset from various websites for the use of our project
5	16-02-24 to 17-02-24	We all together made our required models and tested it through our dataset
6	15-03-24 to 22-03-24	We all together completed the project
7	08-04-24 to 25-04-24	We prepared a brief report of the project

APPENDIX D

Plagiarism Report



grouput15ut.pdf

ORIGINALITY REPORT

12%	10%	4%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	fastercapital.com Internet Source	1%
2	publisher.uthm.edu.my Internet Source	1%
3	www.iasj.net Internet Source	<1%
4	yaginu.hatenablog.com Internet Source	<1%
5	ijisrt.com Internet Source	<1%
6	essay365.x10.mx Internet Source	<1%
7	bspace.buid.ac.ae Internet Source	<1%
8	Submitted to Monash University Student Paper	<1%
9	www.ijcrt.org Internet Source	<1%

10	Submitted to University of Keele Student Paper	<1 %
11	mindthegraph.com Internet Source	<1 %
12	Submitted to Liverpool John Moores University Student Paper	<1 %
13	Submitted to Heriot-Watt University Student Paper	<1 %
14	Submitted to srmist Student Paper	<1 %
15	www.researchsquare.com Internet Source	<1 %
16	open-innovation-projects.org Internet Source	<1 %
17	appliedmechanics.asmedigitalcollection.asme.org Internet Source	<1 %
18	Hossain, Md. Razuan. "Bio-Inspired Memory Device Based Physical Reservoir Computing System to Solve Temporal and Classification Problems", The University of Mississippi, 2023 Publication	<1 %
19	Subedi, Gaurab. "Multimodal Learning: Generating Precise Chest X-Ray Report on	<1 %

Thorax Abnormality", University of South Dakota, 2023
Publication

20	raw.githubusercontent.com Internet Source	<1 %
21	www.journaltocs.ac.uk Internet Source	<1 %
22	Submitted to Manchester Metropolitan University Student Paper	<1 %
23	dokumen.pub Internet Source	<1 %
24	Submitted to Queen's University of Belfast Student Paper	<1 %
25	Submitted to University of Mosul Student Paper	<1 %
26	engrxiv.org Internet Source	<1 %
27	old.evaluationengineering.com Internet Source	<1 %
28	repositorio.utdt.edu Internet Source	<1 %
29	www.frontiersin.org Internet Source	<1 %

microbenotes.com

30	Internet Source	<1 %
31	Submitted to Erasmus University of Rotterdam Student Paper	<1 %
32	Submitted to University of KwaZulu-Natal Student Paper	<1 %
33	Submitted to University of Leeds Student Paper	<1 %
34	du.diva-portal.org Internet Source	<1 %
35	Rinki Chatterjee, Rajeev Kumar Gupta, Bhavana Gupta. "Depression Detection from Social Media Posts Using Multinomial Naive Theorem", IOP Conference Series: Materials Science and Engineering, 2021 Publication	<1 %
36	aiforsocialgood.ca Internet Source	<1 %
37	datasciencedojo.com Internet Source	<1 %
38	Submitted to The British College Student Paper	<1 %
39	Ton Duc Thang University Publication	<1 %

40	core.ac.uk Internet Source	<1 %
41	enoumen.com Internet Source	<1 %
42	vocal.media Internet Source	<1 %
43	Submitted to Canterbury Christ Church University Student Paper	<1 %
44	I. Fosic, D. Zagar, K. Grgic. "Network traffic verification based on a public dataset for IDS systems and machine learning classification algorithms", 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), 2022 Publication	<1 %
45	Submitted to University of Bucharest Student Paper	<1 %
46	assets.researchsquare.com Internet Source	<1 %
47	utahhealthsciences.net Internet Source	<1 %
48	www.geeksforgeeks.org Internet Source	<1 %
49	Submitted to University of the West Indies Student Paper	