جامعة أبو بكر بلقايد- تلمسان

University Abou Bakr Belkaïd of Tlemcen

Faculty of Technology

Department of Biomedical Engineering

Research Laboratory of Biomedical Engineering

Final Year Project Thesis

With a view to obtaining the degree

MASTER on Biomedical Engineering

Specialty: **Biomedical and Hospital Informatics**

Presented by: BERRAI Manal et KACIMI Khadra

Development of an Intelligent Online Response System (Chatbot) for Mental Illnesses and Severe Pathologies Based on Artificial Intelligence Models

Defended on June 29,2025 before the Jury

M.	LAZZOUNI M. EL Amine	*Dr*	University of Tlemcen	President
Mme.	HAMZACHRIF Souad	*MCA*	University of Tlemcen	Supervisor
M.	GAOUAR Adil	*MAA*	University of Tlemcen	Co-Supervisor
Mme.	ELAOUABER Zineb Aziza	*Dr*	University of Tlemcen	Co-supervisor
Mme.	BELAIDI Asma	*Dr*	University of Tlemcen	Examiner

Academic year 2024-2025

Acknowledgments

Above all, I thank Allah for His endless blessings and for granting me the strength and guidance to reach this stage of my journey.

My deepest gratitude goes to my beloved parents, whose love, sacrifices, and constant encouragement have been the foundation of everything I've achieved. I am also truly grateful to my brother Amin, my sisters Sara, Imen, and Safia, and to our little sunshine Malak — thank you all for surrounding me with love and support every step of the way.

A heartfelt thank you to my dear colleague and friend Khadra, who shared this journey with me — through every challenge and every triumph, you were there, and I couldn't have asked for a better partner in this adventure.

I would also like to express my sincere appreciation to our professors and supervisors, whose guidance, patience, and dedication played a vital role in the completion of this thesis.

Last but not least, I extend my warmest wishes to all my fellow students in the Biomedical Informatics department. May you all find success, happiness, and fulfillment in whatever paths you pursue.

— *Berrai Manal*

Acknowledgments

First and foremost, I thank Allah for His guidance and for helping me reach this point in my journey.

I would like to express my deepest gratitude to my parents, whom I hold in the highest respect and appreciation. Their constant support and concern for our success have always been my driving force. I also want to thank my brother Mohamed, and my sisters Meriem and Zineb, for always being there for me.

My sincere thanks go to my cousin Khadija, who has been my true support and motivation throughout this path.

I would also like to send my warmest regards and deepest respect to my colleague and friend Manal, for being part of this adventure, through all its sweet and bitter moments.

I am deeply grateful to my grandparents, aunts, and uncles for their prayers and continuous encouragement.

I extend my highest appreciation to the professors who supervised us during this thesis for their dedication and support.

I am also thankful to everyone who contributed to my journey in any way, near or far, and to all the teachers who accompanied me from primary school to university.

Finally, I send my warm regards to my classmates in the Biomedical Informatics department and ask Allah to grant them success and fulfillment in their lives.

— *Kacimi Khadra*

Abstract

Mental health issues such as depression and anxiety are affecting more people worldwide, with over 970 million cases reported. Many individuals do not receive proper care due to stigma, lack of professionals, and limited access to services.This thesis proposes a chat- bot system to help detect and respond to mental health conditions. It uses a multimodal approach that combines text input and visual features like facial expressions and eye movement to better understand the user's emotional state.The system integrates Clini- calBERT for text classification, Flan-T5 for generating responses, and Ft_Transformer for visual analysis. These outputs are fused using an XGBoost model for final classifi- cation.The proposed model achieves a classification accuracy of **95%**, which surpasses current state-of-the-art results in mental health detection tasks.This work offers a prac- tical and scalable tool to support mental health, especially in areas with limited access to professional care.

Keywords: Mental Health, Chatbot, Artificial Intelligence AI, Deep Learning DL, Multimodal Analysis, Natural Language Processing NLP, Transformer Models, Depres- sion Detection, XGBoost.

Résumé

Les troubles de santé mentale tels que la dépression et l'anxiété touchent un nombre croissant de personnes à travers le monde, avec plus de 970 millions de cas recensés. L'accès aux soins reste limité en raison de la stigmatisation, du manque de professionnels et de l'insuffisance des services disponibles.Ce mémoire propose un système de chatbot intelligent destiné à détecter et à accompagner les troubles mentaux. Il adopte une approche multimodale combinant données textuelles et indices visuels (expressions fa- ciales, mouvements oculaires) pour une évaluation plus précise de l'état émotionnel de l'utilisateur.Le système repose sur ClinicalBERT pour la classification textuelle, Flan-T5 pour la génération de réponses, et un modèle CNN-LSTM pour l'analyse visuelle. Ces in- formations sont fusionnées via un classifieur XGBoost.Le modèle atteint une précision de 95% en classification, dépassant les performances des méthodes de référence actuelles.Ce travail offre une solution accessible et efficace, en particulier pour les zones où l'accès aux soins reste limité.

Mots-clés : Santé mentale, Chatbot, Intelligence Artificielle (IA), Apprentissage pro- fond (AP), Analyse multimodale, Traitement automatique du langage naturel (TALN), Modèles Transformers, Détection de la dépression, XGBoost.

ملخص

تُعدّ اضطرابات الصحة النفسية مثل الاكتئاب والقلق من المشاكل المتزايدة على مستوى العالم، حيث تم تسجيل أكثر من 970 مليون حالة. ومع ذلك، يظل الوصول إلى الرعاية المناسبة محدودًا بسبب الوصمة الاجتماعية ونقص المتخصصين وضعف البنية التحتية للخدمات.
يقدّم هذا البحث نظام دردشة ذكيًا يهدف إلى اكتشاف ومتابعة حالات الاضطراب النفسي. ويعتمد على مقاربة متعددة الوسائط تجمع بين البيانات النصية والمؤشرات البصرية (مثل تعابير الوجه وحركة العين) لفهم أفضل للحالة العاطفية للمستخدم.
يعتمد النظام على نموذج **ClinicalBERT** لتصنيف النصوص، و **Flan-T5**لتوليد الردود، بالإضافة إلى نموذج **CNN-LSTM** لتحليل البيانات البصرية. ويتم دمج هذه المعلومات عبر مصنّف**.XGBoost**
حقق النموذج دقة تصنيف بلغت 95\%، متفوّقًا على أحدث الأساليب في هذا المجال.
يوفّر هذا العمل أداة فعالة وقابلة للتطبيق، خاصة في المناطق التي تعاني من نقص في خدمات الصحة النفسية.

كلمات مفتاحية : صحة نفسية ، الصحة العقلية ، اضطرابات عقلية ، اكتئاب ،حالة عاطفية ، ذكاء اضطناعي، تطبيق ردود

Contents

List of Figures . . . 3
List of Tables . . . 5
List of Abbreviations . . . 8

1 Mental Health Disorders **11**
1.1 Introduction . . . 11
1.2 Defining Mental Health: . . . 11
1.3 Disorders Symptoms and Severity . . . 11
1.3.1 Stress . . . 11
1.3.2 Anxiety Disorders . . . 12
1.3.3 Post-Traumatic Stress Disorder (PTSD) . . . 12
1.3.4 Depression . . . 13
1.3.5 Suicide . . . 14
1.4 Mental Disorders Diagnosis . . . 14
1.5 The Use of AI in Mental health Field . . . 15
1.6 Conclusion . . . 17

2 Artificial Intelligence and Generative Artificial Intelligence **18**
2.1 Introduction . . . 18
2.2 Artificial Intelligence . . . 18
2.3 Machine Learning . . . 18
2.4 Natural Language Processing (NLP) . . . 19
2.5 Deep Learning . . . 19
2.5.1 Deep Learning Techniques and Applications: . . . 19
2.5.1.1 Deep Networks for Supervised or Discriminative Learning . . . 19
2.5.1.2 Deep Networks for Generative or Unsupervised Learning: . . . 21
2.5.1.3 Deep Networks for Hybrid Learning: . . . 21
2.6 Generative AI . . . 21
2.6.1 Definition . . . 21
2.6.2 Types of Generative AI . . . 22
2.6.2.1 Text Generation . . . 22
2.6.2.2 Image Generation . . . 22
2.6.2.3 Multimodal Generation . . . 23
2.7 Transformers . . . 23
2.7.1 Definition . . . 23
2.7.2 The Transformer Architecture . . . 24
2.7.2.1 Encoder . . . 24
2.7.2.2 Decoder . . . 26
2.7.3 BERT (Bidirectional Encoder Representations from Transformers) . . . 27
2.7.3.1 Pre-training BERT . . . 27
2.7.3.2 Fine-tuning BERT . . . 28

2.7.3.3 Input / Output Representation . . . 28
2.7.3.4 BERT-based Models . . . 29
2.7.4 GPT (Generative Pre-trained Transformer) . . . 30
2.7.4.1 How GPT Works . . . 30
2.7.4.2 Different versions of the GPT model . . . 31
2.7.5 Flan-T5 . . . 32
2.8 Related Work and State of the Art . . . 33
2.9 Conclusion . . . 34

3 Methodology **35**
3.1 Introduction . . . 35
3.2 Dataset Description . . . 37
3.2.1 Textual data Description . . . 39
3.2.1.1 Label Distribution and Initial Imbalance . . . 39
3.2.2 Image Data Description . . . 39
3.3 Dataset Preprocessing . . . 40
3.3.1 Text Data Preprocessing and Preparation . . . 40
3.3.2 Image Data Preprocessing and Preparation . . . 44
3.4 Text Classification: Methodological Approach . . . 48
3.4.1 Model functionality: . . . 48
3.4.2 Binary Classification: . . . 50
3.4.3 Multiclass Classification: . . . 60
3.5 Image Classification: Methodological Approach . . . 67
3.5.1 Model functionality . . . 67
3.5.1.1 LSTM + CNN . . . 67
3.5.1.2 ResNet-1D . . . 68
3.5.1.3 FT-Transformer (Feature Tokenization Transformer) . . . 68
3.5.1.4 Classical Machine Learning Models . . . 69
3.5.2 Binary classification . . . 70
3.5.2.1 LSTM + CNN . . . 70
3.5.2.2 ResNet-1D . . . 71
3.5.2.3 FT-Transformer . . . 72
3.5.2.4 Classical Machine Learning Models . . . 73
3.5.3 Multiclass classification . . . 74
3.5.3.1 LSTM + CNN . . . 74
3.5.3.2 ResNet-1D . . . 75
3.5.3.3 FT-Transformer . . . 76
3.6 Multimodal Classification . . . 76
3.6.1 Early Fusion Approach . . . 76
3.6.2 Late Fusion Approach . . . 79
3.7 Generative Model . . . 80
3.7.1 Dataset Used . . . 80
3.7.2 Flan-T5 Architecture and Fine-Tuning Process . . . 80

4 Experimentation and discussion **82**
4.1 Introduction . . . 82
4.2 Text Classification Results . . . 82
4.2.1 Binary Classification . . . 82
4.2.1.1 Comparative Study . . . 85
4.2.2 Multiclass Classification . . . 86

4.2.2.1 Comparative study . . . 89
4.3 Image-Based Classification . . . 91
4.3.1 Binary Classification . . . 91
4.3.1.1 LSTM + CNN . . . 91
4.3.1.2 ResNet 1D . . . 92
4.3.1.3 FT Transformer . . . 94
4.3.1.4 Machine Learning Models (SVM , RF , LR , KNN) . . . 95
4.3.2 Multiclass Classification . . . 97
4.3.2.1 CNN + LSTM . . . 97
4.3.2.2 ResNet 1D . . . 99
4.3.2.3 FT Tranformer . . . 101
4.4 Multimodal-Based Classification . . . 103
4.4.1 Early fusion . . . 103
4.4.2 Late fusion . . . 104
4.5 Evaluation Metrics . . . 106
4.6 ChatBot Realisation . . . 106
4.6.1 Frontend Development . . . 106
4.6.2 Backend and Flan-T5 Integration . . . 109
4.7 Conclusion . . . 113

List of Figures

2.1 Deep Learning Techniques [1] . . . 20
2.2 The three gate of LSTM [2] . . . 21
2.3 Image generated from the prompt " a sunset over a mountain with a lake reflecting the sky" by PIXLR [3] . . . 23
2.4 The transformer-model architecture [4] . . . 24
2.5 The encoder architecture [4] . . . 25
2.6 The decoder architecture [4] . . . 26
2.7 Comparison of BERT base and BERT large [5] . . . 27
2.8 Input representation of BERT model [6] . . . 29
2.9 GPT Architecture [7] . . . 30
2.10 Table of comparison of different versions of GPT model [8]. . . . 32

3.1 System Workflow for Multimodal AI-Based Mental Illness Detection Chatbot 36
3.2 DAIC WOZ Dataset in Kaggle [9] . . . 37
3.3 PHQ8 Questionnaire [10] . . . 38
3.4 Binary classification dataset: class distribution before (left) and after (right) MixUp-based augmentation and balancing. . . . 44
3.5 Multiclass dataset: class distribution before (left) and after (right) augmentation and balancing. . . . 44
3.6 Binary classification image dataset: class distribution before (left) and after (right) Manifold MixUp-based augmentation and balancing. . . . 46
3.7 Multiclass classification image dataset: class distribution before (left) and after (right) Manifold MixUp-based augmentation and balancing. . . . 47
3.8 ClinicalBERT used across time-structured patient notes to estimate readmission probabilities. [11] . . . 49
3.9 BERTClassifier architecture for binary classification . . . 52
3.10 Architecture of the `RoBERTaClassifier` for binary classification . . . 55
3.11 Architecture of MentalBERTClassifier for binary classification . . . 58
3.12 ClinicalBERT Binary Classification Architecture . . . 59
3.13 Multiclass BERTClassifier Architecture . . . 61
3.14 RoBERTaClassifier architecture for multiclass classification (differences highlighted) . . . 63
3.15 MentalBERT architecture adapted for multiclass classification . . . 65
3.16 ClinicalBERT Multiclass Architecture Overview . . . 66
3.17 Architecture of FT Transformer [12] . . . 68
3.18 Early Fusion Workflow . . . 77
3.19 Schematic representation of XGBoost-based multimodal classification architecture . . . 78
3.20 Late Fusion Workflow . . . 79
3.21 Architecture of the Flan-T5 fine-tuning pipeline using a mental health conversation dataset. . . . 81

4.1 Confusion matrix of the BERT model for binary classification 83
4.2 Confusion matrix of the RoBERTa model on the validation set 83
4.3 Confusion matrix of the MentalBERT model with attention 84
4.4 Confusion matrix of the ClinicalBERT model (binary classification) . . . 85
4.5 Confusion matrix for the BERT multiclass classification model 86
4.6 Confusion matrix of the multiclass RoBERTa model 87
4.7 Confusion Matrix for Multiclass Classification with MentalBERT 88
4.8 Confusion Matrix for Multiclass Classification with ClinicalBERT 89
4.9 Confusion Matrix Binary CNN+LSTM (Gaze) 91
4.10 Confusion Matrix Binary CNN+LSTM (Features) 91
4.11 Confusion Matrix Binary ResNet 1D (Gaze) 93
4.12 Confusion Matrix Binary ResNet 1D (Features) 93
4.13 Confusion Matrix for XGBoost on Multiclass Depression Classification . 104
4.14 Home Page Interface 107
4.15 User Authentication Interfaces 108
4.16 Home screen of the chatbot interface 109
4.17 Example of conversation related to sadness and stress 110
4.18 Example of response to anxiety 110
4.19 Conceptual Modeling Data (CMD) 111
4.20 Unified Modling Language Diagram (UML) 112

List of Tables

2.1 BERT Variants and Their Use Cases [13] 30
2.2 Summary of Reviewed Studies on Depression Detection 34

3.1 Sample distribution before and after augmentation – Binary classification 43
3.2 Sample distribution before and after augmentation – Multiclass classification 43
3.3 Sample distribution before and after augmentation – Binary classification 46
3.4 Sample distribution before and after augmentation – Multiclass classification 46
3.5 Parameters and hyperparameters used for training the BERTClassifier model 51
3.6 Architecture of the BERTClassifier model 52
3.7 Model configuration and training hyperparameters for RoBERTaClassifier 54
3.8 General architecture of the `RoBERTaClassifier` model 55
3.9 Hyperparameter configuration for MentalBERTClassifier (binary) 57
3.10 Model architecture of the MentalBERT binary classifier 58
3.11 ClinicalBERT Hyperparameters for Binary Classification 59
3.12 ClinicalBERT Model Architecture (Binary) 60
3.13 Adjusted elements in the BERTClassifier architecture for multiclass classification 60
3.14 General Architecture of the Multiclass BERTClassifier Model 61
3.15 Modifications specific to multiclass RoBERTaClassifier training 62
3.16 Architectural differences from binary to multiclass RoBERTaClassifier . . 63
3.17 Detailed architecture of the MentalBERT multiclass model 64
3.18 Detailed architecture of the MentalBERT multiclass model 65
3.19 Hyperparameter Configuration – ClinicalBERT Multiclass 66
3.20 ClinicalBERT Multiclass – Model Layer Specifications 67
3.21 Comparison of Training Parameters and Hyperparameters of the CNN+LSTM Models on Gaze and Feature Datasets 70
3.22 Comparison of ResNet-1D Architectures and Training Settings for Binary Classification (Gaze vs. Features) 71
3.23 FT-Transformer hyperparameters for gaze and features datasets (binary classification). 72
3.24 Classical ML model configurations for gaze and features datasets (binary classification). 73
3.25 Comparison of CNN+LSTM Hyperparameters and Training Settings for Multiclass Classification (Gaze vs. Features) 74
3.26 Comparison of ResNet-1D Architecture and Training Settings for Multiclass Classification (Gaze vs. Features) 75
3.27 FT-Transformer hyperparameters for gaze and features datasets (multiclass classification). 76
3.28 XGBoost Hyperparameter Configuration 78
3.29 Overview of the Flan-T5 fine-tuning process 81

4.1 Classification report of the BERT model (binary classification) 82
4.2 Classification report of the RoBERTa model (binary classification) 83
4.3 Classification report for the MentalBERT + Attention model 84
4.4 Classification results of the ClinicalBERT model (binary task) 85
4.5 Comparison of binary classification performance across models 85
4.6 Classification Report for the BERT Multiclass Model 86
4.7 Classification report of the RoBERTa-based multiclass model 87
4.8 Classification Report for MentalBERT on Multiclass Task 88
4.9 Classification Report for ClinicalBERT on Multiclass Task 89
4.10 Multiclass Classification Performance Summary 89
4.11 Binary Classification Results of CNN+LSTM on Gaze and Features Datasets 91
4.12 Binary Classification Results of ResNet 1D on Gaze and Features Datasets 93
4.13 Performance Comparison of FT-Transformer on Gaze and Features Datasets (Binary Classification) 94
4.14 Confusion Matrices of FT-Transformer for Gaze and Features Datasets . 94
4.15 Vertical Comparison of Classical ML Models on Gaze and Features Datasets (Binary Classification) 95
4.16 Final Comparative Results of All Models on Gaze and Features Datasets (Binary Classification) 96
4.17 Classification Reports of CNN+LSTM Model on Gaze and Features Datasets (Multiclass Classification) 97
4.18 Confusion Matrices of CNN+LSTM Model on Gaze and Features Datasets (Multiclass Classification) 97
4.19 Classification Reports of ResNet 1D Model on Gaze and Features Datasets (Multiclass Classification) 99
4.20 Confusion Matrices of ResNet 1D Model on Gaze and Features Datasets (Multiclass Classification) 99
4.21 Classification Reports of FT Transformer Model on Gaze and Features Datasets (Multiclass Classification) 101
4.22 Confusion Matrices of FT Transformer Model on Gaze and Features Datasets (Multiclass Classification) 101
4.23 Overall Performance Comparison of Multiclass Classification Models on Gaze and Features Datasets 102
4.24 Classification Report for XGBoost on Multiclass Depression Detection . . 103
4.25 Overall Performance of the Late Fusion Model (Multiclass Classification) 104
4.26 Classification Report of the Late Fusion Model (Multiclass) 104
4.27 Confusion Matrix – Late Fusion (Multiclass) 105

List of Abbreviations

PTSD	Post Traumatic Stress Disorder
DAIC-WOZ	Distress Analysis Interview Corpus - Wizard of Oz
PHQ	Patient Health Questionnaire
AI	Artificial Intelligence
Gen AI	Generative Artificial Intelligence
CNN	Convolutional Neural Network
ML	Machine Learning
NLP	Natural Language Processing
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
BERT	Bidirectional Encoder Representations from Transformers
GPT	Generative Pre-trained Transformer

General Introduction

In recent decades, mental health has become a major public health concern, affecting individuals across all ages, cultures, and socio-economic backgrounds. The fast pace of modern life, marked by urbanization, digital dependency, economic precarity, and social isolation, has significantly increased the prevalence of psychological disorders. According to the World Health Organization (WHO), over **280 million people suffer from depression**, and **more than 700,000 die by suicide each year**. These numbers are not just individual tragedies—they represent a profound social and economic burden, with productivity losses due to untreated mental illness estimated at **over USD 1 trillion per year** [14]. In this context, mental health is no longer an isolated medical issue—it is a global emergency.

Mental health services around the world still face serious challenges. Many people lack access to proper care due to a shortage of professionals, distance from medical centers, financial limitations, or cultural barriers. On top of that, delays in diagnosis and inconsistent follow-up make the situation worse. All these issues highlight the urgent need for innovative and accessible solutions that can support traditional mental health care and offer early intervention.

Thanks to recent breakthroughs in **Artificial Intelligence (AI)**, especially in **Natural Language Processing (NLP)** and deep learning, new doors are opening for mental health support. Advanced models like **BERT**, **ClinicalBERT**, and **Flan-T5** can now understand and generate human language with impressive accuracy and sensitivity. These tools make it possible to detect early signs of psychological distress, offer scalable mental health screening while preserving user privacy, a key concern in this field.In parallel, the use of visual cues, such as **facial landmarks**, has become increasingly important. Subtle changes in facial expressions and movements can reveal valuable information about a person's emotional or mental state. By combining linguistic and facial features, AI systems can better recognize patterns linked to depression or anxiety, providing more comprehensive and accurate assessments. This multimodal approach strengthens early detection and paves the way for more responsive and accessible mental health interventions.

This thesis, titled "**Development of an Intelligent Online Response System (Chatbot) for Mental Illnesses and Severe Pathologies Based on Artificial Intelligence Models**", is situated at the intersection of AI, digital health, and mental health informatics, and proposes the development of an **intelligent online response system**—a chatbot—designed to assist individuals suffering from mental illness or emotional distress. The objective of this system is to interact with users in natural language, detect emotional and cognitive cues, and provide adaptive support. Unlike generic conversational agents, this chatbot is tailored to the mental health context and is capable of handling conditions ranging from mild stress and anxiety to more severe pathologies such as PTSD and suicidal ideation.

To better understand a user's mental state, our system combines both **text** (like conversations) and **visual caracteristics**. This **multimodal approach** makes the analysis more reliable and accurate. On the technical side, the chatbot uses **transformer-based**

models to handle visual sequences and for processing language. Additionally, **generative AI**, like **Flan-T5**, is used to create personalized and empathetic responses, making the interaction feel more natural and supportive.

To guide the reader through this work, the thesis is structured as follows:

- **Chapter 1** introduces key mental health disorders and their symptoms, explains how they are diagnosed, and highlights how artificial intelligence can help improve mental health care.
- **Chapter 2** presents the key concepts of Artificial Intelligence, including machine learning, deep learning, NLP, and transformers, and explores related works.
- **Chapter 3** outlines the methodology used in this research. It details the datasets, preprocessing steps, and the classification models applied to both text and visual data, including multimodal and generative approaches designed to detect and understand mental health conditions.
- **Chapter 4** presents the experimental results and discussions. It covers the performance of various models on text, image, and multimodal data for both binary and multiclass mental health classification, and highlights comparative findings and evaluation metrics.
- **Conclusion** summarizes the key contributions of the work, reflects on the performance of the proposed AI-based mental health system, and outlines future research directions to improve scalability, personalization, and ethical integration in real-world scenarios.

Chapter 1

Mental Health Disorders

1.1 Introduction

This chapter aims to examine the nature, complexity, and prevalence of significant mental health disorders, particularly stress, anxiety, depression, post-traumatic stress disorder (PTSD), and suicidality. Through doing so, it highlights not only the clinical challenges of diagnosing and treating such conditions but also the systemic limitations that frequently prevent intervention. Finally, this chapter offers the framework for understanding how modern technology, particularly artificial intelligence, might help deliver faster, accessible, and personalized mental health treatment.

1.2 Defining Mental Health:

The term "mental health" describes an individual's whole emotional, psychological, and social well-being. It affects people's everyday thoughts, feelings, and behaviors as well as how they manage stress, form bonds with others, and make decisions. [14, 15]. Mental health is a condition of balance and well-being that enables individuals to have meaningful and productive lives, rather than only the absence of disease.It is best viewed as a continuum, with high levels of mental health at one end and severe mental illness at the other. There are several degrees in between, including stress, emotional problems, and moderate disorders [16]. Furthermore, it is critical to differentiate between mental illness (typically referring to more severe or chronic diseases like depression or schizophrenia), mental disorders (clinical problems that impact everyday functioning), and mental discomfort (normal reactions to life events) [17]. By r Recognizing these differences helps improve care, reduce stigma, and support early and appropriate interventions.

1.3 Disorders Symptoms and Severity

This section focuses on the five key mental health conditions at the heart of this research: stress, anxiety, depression, post-traumatic stress disorder (PTSD), and suicidal ideation. Each of these disorders presents unique symptoms and severities, yet they are deeply interconnected and often co-occur in vulnerable individuals:

1.3.1 Stress

Stress is one of the most common and pervasive psychological responses in contemporary life. It can be broadly categorized into two types: *acute stress*, which results from

short-term challenges or perceived threats, and *chronic stress*, which emerges from prolonged exposure to stressors such as financial hardship, workplace pressure, or caregiving responsibilities [18].

While acute stress may have adaptive functions—triggering a "fight or flight" response to immediate danger—chronic stress can become toxic. It leads to sustained activation of the hypothalamic-pituitary-adrenal (HPA) axis and elevated cortisol levels, which in turn contribute to physical health issues such as cardiovascular disease, immune dysfunction, and metabolic disorders [19, 20]. Psychologically, chronic stress is a well-documented precursor to more severe mental health disorders including anxiety, depression, and PTSD [21].

Despite its prevalence, stress is often trivialized or misunderstood. In many cultures, high stress is normalized or even valorized as a marker of productivity. This normalization can obscure early signs of distress and delay preventive care, allowing stress to evolve silently into deeper psychological impairments. Therefore, accurate and timely identification of stress is essential—not only to improve mental health outcomes but also to prevent escalation into chronic psychiatric conditions.

1.3.2 Anxiety Disorders

Anxiety disorders are among the most prevalent mental health conditions worldwide, affecting approximately 301 million people as of 2019 [22]. These disorders encompass a range of conditions, the most common being Generalized Anxiety Disorder (GAD), Panic Disorder, and Social Anxiety Disorder (SAD). While each subtype presents distinct patterns, they share a core feature: an overwhelming sense of fear or apprehension that is disproportionate to actual threats [23].

Generalized Anxiety Disorder is characterized by persistent and excessive worry about various aspects of daily life, often accompanied by symptoms such as restlessness, fatigue, difficulty concentrating, irritability, and muscle tension. Individuals with *Panic Disorder* experience recurrent panic attacks—sudden surges of intense fear that trigger physical symptoms like chest pain, shortness of breath, or dizziness, often leading to emergency room visits due to misdiagnosis as cardiac events [24]. *Social Anxiety Disorder*, on the other hand, involves intense fear of social scrutiny or embarrassment, which can severely impair interpersonal relationships and occupational functioning.

Symptoms of anxiety disorders typically include excessive worry, heightened vigilance, sleep disturbances, somatic complaints (e.g., headaches or stomach pain), and behavioral avoidance. These manifestations can become chronic and disabling, especially if not addressed early.

Despite their prevalence, anxiety disorders remain underdiagnosed and undertreated. Part of the diagnostic complexity lies in their high comorbidity with other mental conditions—especially depression—and their tendency to manifest differently across individuals and cultures [25]. Moreover, many people with anxiety do not seek help due to stigma, lack of access to care, or minimization of their symptoms by others.

In light of their burden and subtle presentation, anxiety disorders require nuanced assessment strategies that can detect patterns beyond surface-level symptoms. This makes them a relevant and challenging target for intelligent mental health detection systems.

1.3.3 Post-Traumatic Stress Disorder (PTSD)

Post-Traumatic Stress Disorder (PTSD) is a severe and often chronic mental health condition that arises following exposure to traumatic events. These may include war, natural

disasters, physical or sexual assault, domestic violence, serious accidents, or forced displacement. Although it is natural to feel distressed after such events, PTSD is diagnosed when these symptoms persist for weeks or months and significantly impair daily functioning [26].

Common symptoms of PTSD include intrusive memories or flashbacks, nightmares, hypervigilance, exaggerated startle responses, emotional numbness, irritability, and difficulty concentrating. Individuals may also engage in avoidance behaviors—steering clear of places, people, or conversations that remind them of the trauma [27].

Certain populations are particularly at risk. Veterans returning from combat zones often report high rates of PTSD due to prolonged exposure to life-threatening situations and moral injury [28]. Refugees and displaced individuals, who frequently endure both the trauma of conflict and the hardship of resettlement, are another vulnerable group [29]. Survivors of childhood abuse or domestic violence are also disproportionately affected, with studies indicating that early-life trauma can alter neurobiological pathways and increase long-term vulnerability to mental illness [30].

PTSD presents unique diagnostic challenges because of its overlap with depression, anxiety, and substance use disorders. Moreover, cultural perceptions of trauma and emotional expression can shape how symptoms are reported and recognized. In many cases, PTSD remains undiagnosed or misunderstood, particularly in low-resource settings or among marginalized communities.

Addressing PTSD requires trauma-informed care and accurate early detection—two domains where artificial intelligence and digital mental health technologies are beginning to show promise.

1.3.4 Depression

Depression is one of the most prevalent and disabling mental health disorders globally. It is not merely a passing feeling of sadness or low mood but a serious medical condition that affects how individuals think, feel, and function. Clinically, depression encompasses affective symptoms (such as persistent sadness, hopelessness, or emotional numbness), cognitive symptoms (including negative thought patterns, impaired concentration, and indecision), and behavioral changes (such as withdrawal from social interaction, fatigue, and loss of interest in previously enjoyable activities) [31].

Depressive episodes can range in severity from mild to severe. While some individuals experience brief and situational periods of low mood, others suffer from Major Depressive Disorder (MDD), characterized by prolonged and intense symptoms that interfere with daily life for weeks or months [32]. MDD can be recurrent and is often comorbid with anxiety disorders, substance use, and suicidal ideation.

According to the World Health Organization, depression is the leading cause of disability worldwide, affecting over 280 million people across all age groups [32]. The burden is especially heavy among adolescents and older adults, and women are statistically more affected than men. In addition to its personal toll, depression contributes to social disintegration, lost productivity, and increased healthcare utilization.

Despite its high prevalence, depression remains underdiagnosed and undertreated, particularly in low- and middle-income countries. Stigma, lack of mental health resources, and disparities in access to care exacerbate this gap. Addressing depression at scale requires not only clinical interventions but also community awareness and scalable technological solutions, including AI-driven screening tools and personalized therapeutic approaches.

1.3.5 Suicide

Suicide represents one of the most tragic and preventable outcomes of mental illness. It exists along a continuum, beginning with suicidal ideation—recurrent thoughts about death or taking one's own life—progressing to suicide attempts, and in some cases, resulting in completed suicide [33].

Understanding this trajectory is critical. Suicidal ideation does not always lead to an attempt, and not all attempts result in death. However, each stage signifies a profound level of psychological distress requiring immediate attention and intervention.

Numerous risk factors contribute to suicidality, including mental disorders (especially depression, bipolar disorder, and PTSD), substance abuse, history of trauma, chronic illness, and social isolation. Gender differences are striking: while men are more likely to die by suicide, women report more suicidal thoughts and attempts [34]. Cultural and societal contexts also play a significant role, often shaping whether individuals seek help or suffer in silence. In many regions, suicide remains heavily stigmatized or even criminalized, creating barriers to prevention and care [35].

Globally, over 700,000 people die by suicide each year, making it one of the leading causes of death among adolescents and young adults [36]. Suicide is preventable, yet prevention efforts remain challenged by a lack of early detection systems, underfunded mental health services, and poor integration of mental health into primary care.

While strategies such as crisis hotlines, school-based interventions, and awareness campaigns have made a difference, they often fail to reach the most vulnerable. Innovative tools—such as AI-based risk prediction models—offer promising paths forward, though ethical and accuracy concerns remain.

Ultimately, addressing suicidality requires a multi-layered approach: clinical, societal, technological, and deeply human.

1.4 Mental Disorders Diagnosis

The diagnosis of mental health disorders has traditionally been guided by clinical expertise, structured interviews, and standardized psychological tools. Instruments such as the Patient Health Questionnaire (PHQ-9 or PHQ-8) are widely used to screen for depression and anxiety, offering quantifiable insights into emotional states [37,38]. Within the scope of this thesis, the PHQ-8 plays a key role in assessing depressive symptom severity in the DAIC-WOZ corpus. Diagnostic frameworks such as the DSM-5 and ICD-11 have further contributed to the standardization of mental disorder classification, enabling more consistent diagnoses across clinical contexts [39,40]. However, these systems are not without limitations. Their categorical nature may fail to reflect the continuum and complexity of mental states, and symptom overlap, cultural biases, or reliance on self-reporting can reduce diagnostic precision [41].

Beyond technical considerations, the diagnostic process is often influenced by deeply rooted human and systemic barriers. Stigma remains one of the most persistent challenges, discouraging individuals from seeking help due to fear of judgment or discrimination [42]. Access to mental health professionals is another major concern, particularly in low- and middle-income countries where resources are limited [43]. Even in wealthier regions, barriers such as long wait times, high costs, or lack of culturally sensitive care continue to restrict access to appropriate treatment.

Cultural beliefs and gender norms further shape how mental health is experienced and expressed. Men, for instance, may underreport distress due to societal expectations surrounding masculinity [44], while women's symptoms are sometimes misattributed to

social or hormonal factors, delaying accurate diagnosis. Clinically, the high prevalence of comorbidity—where conditions like depression, anxiety, and PTSD present overlapping symptoms—adds another layer of complexity [45]. These challenges highlight the risk of misdiagnosis or missed intervention, especially when rigid diagnostic tools are applied without contextual nuance.

Altogether, the combination of technical limitations and human factors underscores the urgent need for more adaptive, inclusive, and scalable diagnostic approaches. In this evolving landscape, artificial intelligence does not aim to replace traditional methods, but to complement and enhance them—providing earlier detection, greater reach, and a more individualized understanding of mental health.

1.5 The Use of AI in Mental health Field

Artificial Intelligence (AI) is opening new frontiers in mental health care, particularly through its ability to analyze large datasets and detect subtle behavioral patterns. One key application is **Natural Language Processing (NLP)**, where AI models analyze language used in interviews or on social media to identify linguistic markers of psychological distress [46, 47].

AI also supports clinicians through decision-aid systems that synthesize patient data to suggest diagnoses or treatment plans [48]. Audio-based tools that evaluate voice patterns, pauses, and tone are being developed to detect emotional states in real time.

Additionally, AI-driven chatbots such as **Woebot** and **Wysa** offer cognitive-behavioral support through text-based conversations, providing users with immediate coping tools and psychoeducation [49].

Digital Tools for Mental Health

A wide range of digital tools has emerged to address the growing demand for accessible mental health support. These include:

- **Mobile Applications:** Apps like *Calm* and *Headspace* offer guided meditations, mindfulness training, and stress-reduction exercises [50].
- **Online Therapy Platforms:** Services such as *BetterHelp* and *Talkspace* provide access to licensed therapists via chat or video, making therapy more flexible and discreet.
- **Real-Time Monitoring Tools:** Wearable devices and mobile features enable users to track mood, sleep, and activity levels, offering continuous feedback and early warning signs [51].
- **Virtual Peer Support:** Online communities and forums create spaces for shared experiences, emotional validation, and peer-guided coping strategies.

These technologies aim not to replace clinical care, but to complement it by offering timely, user-centered support—particularly in environments where traditional services are scarce or stigmatized.

Applications and Case Studies

- **Woebot:** An AI-powered chatbot that uses CBT techniques to reduce symptoms of depression and anxiety. Clinical trials have shown measurable improvements in users' well-being within short periods [49].
- **Wysa:** A widely used app offering interactive mental health support via journaling, mindfulness exercises, and conversation. It has been adopted across 65 countries and demonstrates positive outcomes based on user feedback [52].
- **Ellie (DAIC-WOZ):** A virtual interviewer designed to assess depression severity using multimodal data such as facial expressions, voice tone, and linguistic patterns [53].
- **Ginger.io:** A mobile platform that passively collects behavioral data (e.g., mobility, sleep, communication) to detect early signs of mental health decline and enable timely intervention [54].

These case studies illustrate how AI is not only enhancing existing models of care but also laying the groundwork for proactive, personalized, and data-informed mental health services. However, the implementation of these tools must be carefully governed to ensure ethical use, prevent algorithmic bias, and protect user privacy.

Benefits of AI in Mental Health

Artificial Intelligence (AI) holds immense promise in transforming mental health care across various dimensions, from accessibility and early detection to personalization and long-term monitoring. These benefits are especially relevant in addressing the limitations of traditional mental health systems, which often suffer from a shortage of qualified professionals, delayed diagnoses, and underreporting due to stigma.

1. Early Detection and Risk Prediction: AI-powered tools can identify subtle behavioral patterns that may signal early onset of mental disorders. Natural Language Processing (NLP) models can analyze speech, tone, and content to detect markers of depression, anxiety, or suicidality [46, 55]. For instance, machine learning algorithms have been trained to predict suicide risk by analyzing social media posts [56].

2. Scalability and Accessibility: Unlike human professionals who are constrained by time and location, AI-based solutions like mobile apps or chatbots are scalable and available 24/7. This is particularly valuable in low-resource settings or for individuals hesitant to seek in-person help [52]. Users can receive immediate support, psychoeducation, and coping strategies without barriers related to cost or stigma.

3. Personalized Interventions: AI enables the development of adaptive systems that tailor therapeutic recommendations to an individual's needs. By continuously learning from user interactions, these systems can suggest more effective interventions over time, enhancing therapeutic outcomes [57].

4. Augmenting Clinical Decision-Making: Clinicians can benefit from AI-assisted diagnostics, which provide data-driven insights from large patient datasets. These tools can support decisions about treatment plans, risk assessment, and prognosis, reducing the cognitive load on healthcare providers [58].

5. Continuous Monitoring: AI tools can monitor patients between sessions, offering clinicians real-time data on mood, behavior, and adherence. This helps in timely intervention and prevents escalation, particularly in chronic conditions or high-risk patients [59].

In sum, the integration of AI into mental health care can bridge critical gaps, improve diagnostic precision, and foster more proactive and patient-centered care systems.

1.6 Conclusion

Mental health stands today as one of the most pressing global health challenges, affecting individuals across all ages, cultures, and socioeconomic backgrounds. Disorders such as stress, anxiety, depression, PTSD, and suicidal ideation not only diminish quality of life but also impose substantial human, social, and economic costs. Traditional diagnostic and therapeutic frameworks, while essential, remain constrained by systemic limitations, stigma, and accessibility barriers.

As this chapter has explored, recent technological advancements — particularly in artificial intelligence — are beginning to reshape how we understand, detect, and support mental health. From virtual therapy tools to intelligent screening systems, AI introduces new possibilities for scalable, personalized, and proactive mental healthcare. Yet, its deployment must be guided by ethical vigilance, cultural sensitivity, and interdisciplinary collaboration.

By framing the current landscape of mental health and technology, this chapter lays the groundwork for the research that follows. The next sections will explore how this study aims to harness AI—specifically deep learning and transfer learning approaches—to develop an intelligent, context-aware support system for mental health detection and intervention. This endeavor is rooted not only in technical innovation but also in a deep commitment to human dignity, accessibility, and the ethical use of intelligent systems for social good.

Chapter 2

Artificial Intelligence and Generative Artificial Intelligence

2.1 Introduction

In recent years, the field of Artificial Intelligence (AI) has grown rapidly, transforming how machines interact with the world. This chapter provides an overview of AI and its main branches, including Machine Learning (ML), Natural Language Processing (NLP), and Deep Learning. It also explores advanced models such as Transformers, with a focus on well-known architectures like BERT, GPT, and Flan-T5.

A special section is dedicated to Generative Artificial Intelligence (Generative AI), which enables machines to create new content in the form of text, images, or multimodal outputs. This chapter explains how these models work, highlights their applications, and presents the challenges they bring.

The overall objective is to provide an organized and understandable knowledge of the construction and application of AI systems, particularly generative ones, in many fields.

2.2 Artificial Intelligence

Artificial Intelligence (AI), a branch of computer science dedicated to building robots with human-like thought and behavior, has emerged as a result of these advancements. One of the creators of artificial intelligence, John McCarthy, described AI as the science and engineering of creating intelligent devices, particularly clever computer programs. By researching how the human brain functions, how humans learn, make decisions, and solve problems, and using this knowledge to create intelligent software and systems, artificial intelligence (AI) seeks to mimic human-like thinking [60].

2.3 Machine Learning

Machine learning is a subfield of artificial intelligence (AI) that eliminates the need for manual programming by enabling computers to learn and improve over time based on experience. Various types of machine learning are used depending on the nature of the data and the problem to solve for exemple result [61]:

Supervised Learning:

is similar to learning from a teacher. A dataset that already contains the right answers (labels) is used to train the system. The system may use what it has learnt to predict new data after training.

Unsupervised Learning:

is a kind of machine learning in which the model is trained on unlabeled data, enabling it to independently identify patterns, structures, or groups.

Reinforcement learning:

consist of an agent gains knowledge by interacting with its surroundings, getting rewards or penalties for its actions, and gradually changing its behavior to get the optimum.

Semi-supervised learning:

is one of type of machine learning where the model is trained using a lot of unlabeled data and a little quantity of labeled data. The advantages of both supervised and unsupervised learning are combined in this method.

2.4 Natural Language Processing (NLP)

Is a subfield of artificial intelligence that enables computers to comprehend and communicate using human language. NLP allows robots to read, understand, and produce voice and text by fusing linguistic rules with machine learning and deep learning. Voice assistants (like Siri and Alexa), chatbots, GPS voice instructions, and search engines are all powered by this technology. Additionally, it is essential to generative AI, which helps models comprehend and react to human input [62].

2.5 Deep Learning

Is a technique for teaching computers to learn from data and make decisions independently, much like people do. In order to identify patterns, comprehend complicated data, and resolve issues without requiring detailed instructions, it makes use of layers of algorithms known as neural networks [63].It works by breaking down complex ideas into simpler ones, step by step, until it understands the bigger picture.
Each layer in a deep neural network builds on the one before it to increase accuracy. Data flows through the network in a process called forward propagation, leading to a final prediction. If the prediction is wrong, another process called backpropagation adjusts the network's internal settings to reduce errors. By repeating this cycle, the network gradually learns and becomes more accurate over time [64].

2.5.1 Deep Learning Techniques and Applications:

2.5.1.1 Deep Networks for Supervised or Discriminative Learning

The primary applications of this kind of deep learning method are in classification or supervised learning tasks. By simulating the probability of classes depending on the in-

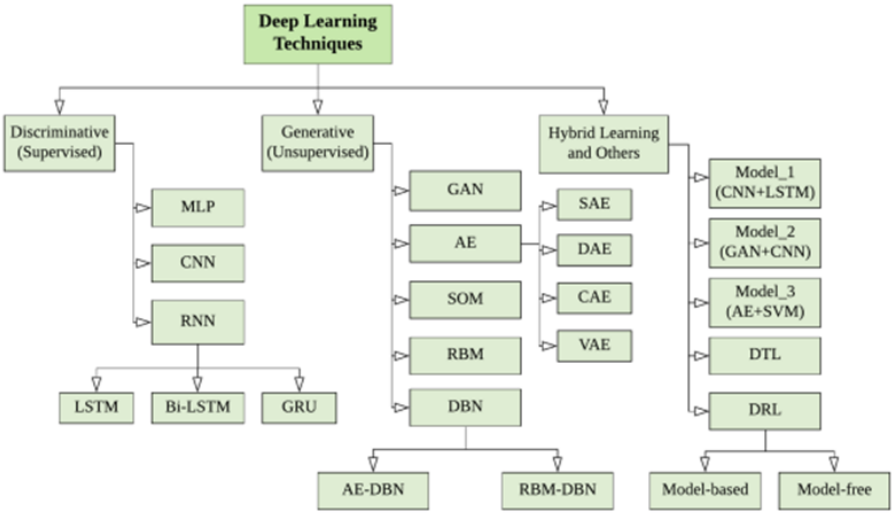

Figure 2.1: Deep Learning Techniques [1]

put data, discriminative deep architectures are made to efficiently differentiate between patterns.
Common examples of these architectures include Multi-Layer Perceptrons (MLPs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs), along with their various versions. We provide a brief overview of each of these techniques below:

Convolutional Neural Network (CNN or ConvNet): are an effective kind of deep learning model that does not require feature extraction; instead, it may automatically learn features from input data. CNNs handle data, particularly 2D structures like pictures, more effectively than classic neural networks by using layers like convolutions and pooling. Additionally, they employ strategies like dropout to enhance generality and minimize overfitting. CNNs are frequently utilized for applications including natural language processing, segmentation, medical imaging, and recognizing images. According to their design and learning strengths, popular CNN models such as VGG, AlexNet, Inception, ResNet, and Xception are each appropriate for a particular application [1].

Recurrent Neural Network (RNN): is a type of deep learning model that processes sequential input by converting it into another sequence. Examples of this type of data include words, phrases, and time series. According to complex principles of meaning and order, this kind of data is organized such that each component depends on the ones that came before it. RNNs function by simulating how people process these kinds of sequences, for as when translating text across languages [65].
There are several popular variants of recurrent networks, with Long Short-Term Memory (LSTM) being one of the most widely used due to its ability to handle long-term dependencies more effectively. It is A popular RNN variation created to address the vanishing gradient issue. It was first presented by Hochreiter et al. [66] and makes use of unique memory cells that have a lengthy storage period. These cells are controlled by three key gates: the input gate chooses which new information to store, the forget gate chooses which old information to delete, and the output gate chooses what to transmit.

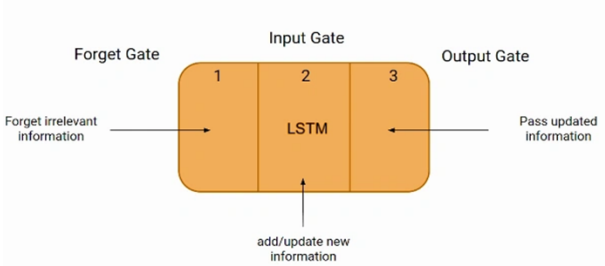

Figure 2.2: The three gate of LSTM [2]

2.5.1.2 Deep Networks for Generative or Unsupervised Learning:

This kind of deep learning methodology focuses on identifying complex links and patterns wthin data, which is particularly helpful for data generation and analysis or for discovering underlying structures. Generative deep architectures are primarily employed in unsupervised learning tasks such as feature extraction or data generation as they do not require labeled outputs during training. To enhance the performance of classification models, these models may also be used as an effective preprocessing step for supervised learning. Popular generative models include Generative Adversarial Networks (GANs), Autoencoders (AEs), Self-Organizing Maps (SOMs), and others, along with their various extensions [1].

2.5.1.3 Deep Networks for Hybrid Learning:

Because of their adaptability, generative models can learn from both labeled and unlabeled data. On the other hand, discriminative models need labeled data to train, yet they outperform other models in supervised tasks. Hybrid deep learning models are based on the premise that we could use the capabilities of both kinds by integrating them into a single framework. Typically, these hybrid models combine two or more deep learning components, which may be discriminative or generative. Generally, hybrid models may be divided into three groups based on how they are combined:

- Hybrid Model 1: combines many generative or discriminative models, such as CNN+LSTM or AE+GAN In order to extract more robust and rich characteristics.
- Hybrid Model 2: Employs a discriminative model for classification after first using a generative model to develop useful representations. DBN+MLP, GAN+CNN, and AE+CNN are a few examples.
- Hybrid Model 3: combines a conventional (non-deep learning) classifier, such CNN+SVM or AE+SVM, with a generative or discriminative model [1].

2.6 Generative AI

2.6.1 Definition

Generative Artificial Intelligence (GenAI) is a new class of AI algorithms that use progressed machine learning techniques to generate complicated, original content in a variety of types, such as natural language text, human-like speech, dynamic video sequences, images, and functional computer code, all based on simple user prompts or instructions. These systems are notable by their impressive ability to generate entirely new materials

that display originality, coherence, and contextual awareness, more than just processing or analyzing available materials. These tools have demonstrated great performance across a wide range of standardized assessments and professional standards, with outcomes that match or sometimes surpass human skills [67].
Large datasets are readily available, but significant advancements in deep learning models are also responsible for the growth of generative AI. Generative Adversarial Networks (GANs), first presented by academics at the University of Montreal in 2014, are based on two models collaborating: one creates material (such as photos), while the other determines if it appears realistic. Diffusion models were created a year later by Stanford and UC Berkeley academics. Tools like Stable Diffusion, which creates visuals from text prompts, are based on these models, which progressively tame noisy data to yield realistic results. The transformer architecture, which Google unveiled in 2017, completely changed language models like ChatGPT. Through the analysis of word relationships inside sentences, transformers are able to comprehend context and generate content that makes sense. These developments are only a handful of the potent technological advancements that underpin generative AI today [68].

2.6.2 Types of Generative AI

Generative AI models may produce text, graphics, code, and video, among other kinds of content. By modifying their algorithms or structures, researchers can modify these models for certain tasks or domains. Both generic and task-specific generative AI research are highlighted in this section.

2.6.2.1 Text Generation

Text generation models use input prompts to generate text that seems human. Large datasets of books, articles, code, webpages, and other materials are used to train these models. These models have a wide range of key applications, including Automatic content generation, Text summarization, Chatbots and assistants...
Through training on vast text collections, including books, papers, and webpages, an AI text generator learns how language works. The model is able to comprehend language, word meaning, and the relationships between concepts thanks to this learning process. The transformer design used by the majority of contemporary text generators, such as GPT, enables the system to recognize the relationships between words, even when they are far separated in a phrase. The program is able to generate coherent and fluid phrases by learning to predict the next word during training. Users can engage with the model once it has been trained by entering a prompt; this serves as the basis for the AI's creation of new text. After converting the prompt into a format that the model can comprehend, it uses what it has learnt to construct a response word by word. The model employs a variety of generating approaches to provide accurate and well-structured outputs. The system may do a last check after text generation to make sure the result is clear, pertinent, and in line with user expectations [69].

2.6.2.2 Image Generation

is a technique that creates images based on textual prompts, enabling users to describe what they want in words and have an AI produce a corresponding visual. This technology has wide-ranging applications, including photo searching, editing, art creation, computer-aided design, image reconstruction. Advanced algorithms, like those used in diffusion models or Generative Adversarial Networks (GANs), analyze the semantics of the text

and use learned patterns from training data to synthesize visuals that represent the scene, object, or concept described [70].

Figure 2.3: Image generated from the prompt " a sunset over a mountain with a lake reflecting the sky" by PIXLR [3]

2.6.2.3 Multimodal Generation

By identifying patterns in existing data, generative AI generates new material, including text, pictures, audio, and music. This is furthered by Multimodal Generative AI, which combines many data kinds, such as text, pictures, sound, and video, into a single, cohesive framework. Multiple formats may be understood and generated by multimodal models, in contrast to unimodal models that can only interpret one type of input. For example, they are able to create voice from video, convert a written input into a picture, and describe an image with words. Because of their adaptability, they are particularly helpful for real-world applications, allowing humans and robots to connect more naturally and human-like. Multimodal generative AI is based on the capacity to employ specialized architectures to analyze and combine many forms of data. These usually consist of many encoders and decoders, where the encoders convert all input types into numerical embeddings and the decoders use those embeddings to produce the necessary output. A text-to-image model is a typical example, in which one part encodes the text and another uses it to create a picture [71].

2.7 Transformers

2.7.1 Definition

Vaswani et al. (2017) created a sort of deep learning model called Transformers. For a variety of natural language processing (NLP) tasks, they are generally acknowledged as the best approach. In contrast to conventional models, which employ a sequential data processing procedure, Transformers employ a technique known as self-attention. They can now examine every input element at once, which increases their efficiency for complicated tasks.

In transformers, the encoder and the decoder are the two primary parts. The output is produced by the decoder using the representations that the encoder has created after processing the input data [72].

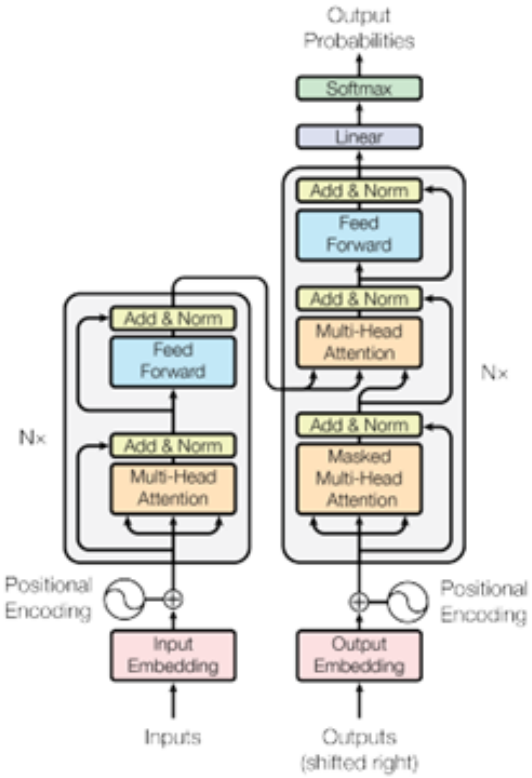

Figure 2.4: The transformer-model architecture [4]

The Transformer was originally developed for sequence-to-sequence tasks like translation, but its architecture has since evolved into three main types. Encoder-only models (e.g., BERT, RoBERTa, DistilBERT) use bidirectional attention for tasks like classification and entity recognition. Decoder-only models (e.g., GPT) generate text by predicting the next word using only past context. Encoder-decoder models (e.g., the original Transformer, BART, T5) handle tasks like translation and summarization [73].

2.7.2 The Transformer Architecture

This framework comprises two main components:

2.7.2.1 Encoder

- Word Embeddings: Word embeddings transform input tokens into dense vectors of fixed size d_model , where semantically similar words are represented by vectors close to each other in a continuous space, enabling the model to capture meaning and relationships effectively.

- Positional Encoding: Positional encodings are added to the embeddings to provide information about the relative positions of the words in the sequence.
These encodings are calculated using sinusoidal functions, defined as:

$$PE(pos, 2i) = \sin\left(\frac{pos}{1000^{2i/d_{\text{model}}}}\right)$$

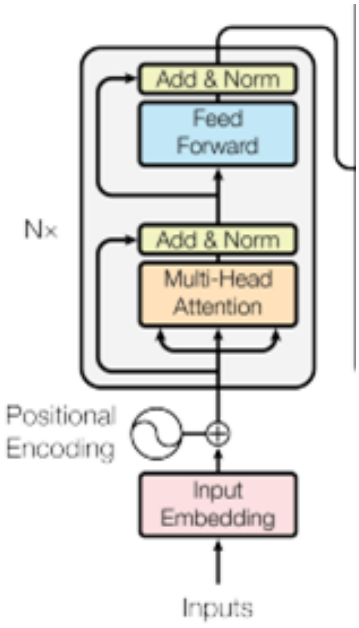

Figure 2.5: The encoder architecture [4]

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{1000^{2i/d_{\text{model}}}}\right)$$

The positional encoding vector is then added to the word embedding to produce the final input.

- Multi-head attention: This mechanism calculates the relationships between tokens by assessing their importance and the strength of their connections. Mathematically, these relationships are represented as distances and angles between vectors in a multidimensional space. It works by using three key vectors:

- **Query (Q)**: The query indicates the type of information we're searching for from other segments of the sequence while processing a word or token.
- **Key (K)**: Keys help match the right information to the Query. Each token produces a Key that describes what kind of information it contains.
- **Value (V)**: This is the actual content or information that gets passed forward.

The process involves the following steps:
1. Matrix Multiplication: The query and key vectors are multiplied to produce a score matrix that measures the relationships between words.

2. Scaling: The scores are divided by the square root of their dimension $\sqrt{d_k}$ to stabilize the gradients.

3. Softmax: A softmax function is applied to the scores to obtain attention weights, which prioritize relevant words.

4. Weighting: The attention weights are multiplied by the value vectors, creating the output for that particular attention head.

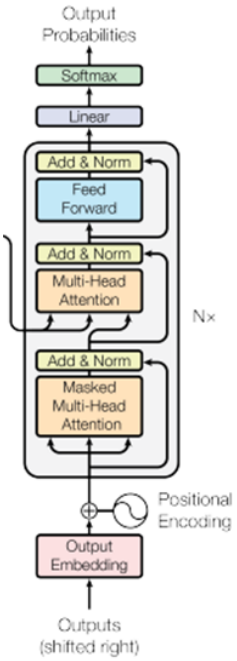

Figure 2.6: The decoder architecture [4]

5. Combining the Attention Heads: Several attention heads are working at the same time, each concentrating on a different aspect of the sequence. The outputs from all heads are then combined and passed through a linear layer to generate the final result.

- Feedforward Network (FNN): Each position in the sequence undergoes a process of linear transformation, ReLU activation, and another linear transformation, with unique parameters for the feedforward network in each encoder layer.

- Residual Connection and Normalization (Add and Norm): A residual connection helps the model maintain crucial information that could be lost during processing by appending a sub-layer's input to its output. The output is normalized with this addition to stabilize and expedite training. The following layer receives this merged output after that [74].

2.7.2.2 Decoder

The decoder consists of 6 identical layers, each with three sub-layers:

- Masked Multi-Head Self-Attention: This sub-layer is used in the decoder of the Transformer model to ensure that each position in the sequence can only attend to the previous positions, not future ones, by applying a mask that blocks access to these future words during the attention calculation. This mask sets the attention scores for future positions to a very low value ($-\infty$).

- Multi-Head Attention: it allows the decoder to focus on different parts of the encoder's output. It uses query vectors from the decoder and key-value vectors from the

encoder, applying scaled dot-product attention in multiple parallel heads. The results from each head are combined to help the decoder generate the next token based on the encoder's entire input sequence.

- Feed-Forward Network (FNN): Similar to the encoder, each position in the sequence is passed through a two layer fully connected network with ReLU activation. This layer improves the model's capacity to identify non-linear patterns by teaching it intricate relationships inside each sequence position [4].

Depending on the application and the input data, transformers have different topologies, particular network designs, and training goals. Among the transformers we find the following:

2.7.3 BERT (Bidirectional Encoder Representations from Transformers)

The BERT model, developed by Google in 2018, is built on a multi-layered Transformer encoder and was pre-trained using text from the Book Corpus (800 million words) and Wikipedia (2,500 million words) [75]. To comprehend complicated semantic linkages between word tokens and sequential sentence dependencies, it employs an encoder-only transformer architecture with attention mechanisms.
BERT comes in two primary versions: base and big. With the exception of using varying quantities of parameters, their design is completely similar. In comparison to BERT base, BERT big has 3.09 times as many parameters to adjust. 24 transformer layers, 16 attention layers, and 340 million parameters were employed in the development of BERT big, whereas BERT base had 12 transformer layers, 12 attention layers, and 110 million parameters.

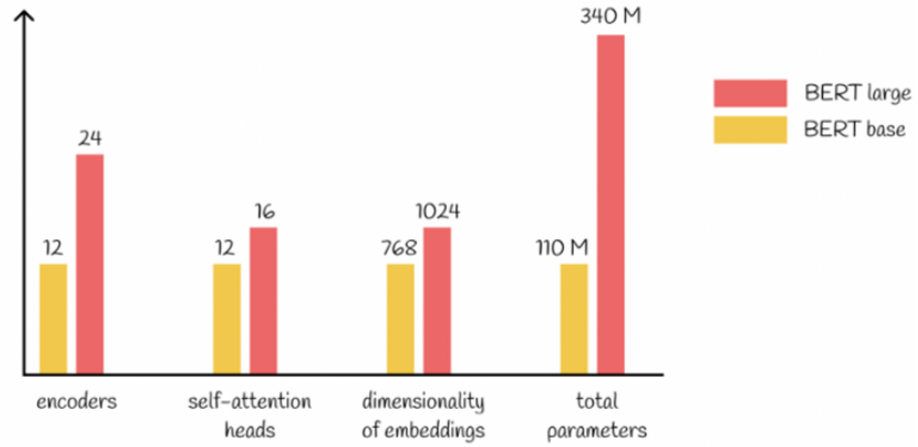

Figure 2.7: Comparison of BERT base and BERT large [5]

2.7.3.1 Pre-training BERT

Bert is trained using two primary objectives: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP):

a. Masked Language Model (MLM): in this phase, 15% of the words in a phrase are hidden (or "masks"), and the model must estimate what those words are. The model is only trained to predict these masked words, not the whole sentence. BERT handles masked words in numerous approaches to increase the model's flexibility:
- In 80% of cases, the special token [MASK] is used in place of the word: The sky is blue → The sky is [MASK].
- In 10% of the time, a random word from the dataset is used in its stead: The sky is blue → The sky is short.
- In 10% of the time, the term remains unchanged: The sky is blue → The sky is blue.

In order to improve the model's ability to handle real-world sentences during fine-tuning, this helps the model become accustomed to words appearing in various ways.

b. Next Sentence Prediction (NSP): This phase aids the model in comprehending the relationships between phrases. Given two phrases (A and B), BERT has to determine if B actually follows A.
- Sentence B is the actual sentence that comes after A in the original text 50% of the time (labeled IsNext).
- B is a random text from the dataset (labeled NotNext) in the remaining 50% of cases. This facilitates BERT's acquisition of sentence associations, which is helpful for tasks like answering questions and inferring real language [76].

2.7.3.2 Fine-tuning BERT

The BERT model begins with pre-trained parameters, which are subsequently adjusted using labeled data from particular tasks like text classification, question-answering, or text similarity [77].

2.7.3.3 Input / Output Representation

BERT can handle a wide range of language tasks. It must have a consistent method for handling inputs, whether they are a single sentence or two sentences, in order to do its task effectively. Any continuous text is called a "sentence". It could be a complete sentence, a portion of a sentence, or even many sentences together. The whole collection of tokens (words or word fragments) that we input into BERT is called a "sequence". There Are Two Primary Steps in BERT's Input Representation:

- Tokenization: With a technique called WordPiece, each word is divided into smaller components (referred to as *tokens*). This aids BERT in comprehending uncommon or difficult words.
- Embedding: Every token is transformed into a vector of numbers that BERT can comprehend.

Special tokens are used by BERT to clearly comprehend input. Used mostly for classification tasks, the [CLS] token is always added at the beginning and represents the entire input. The [SEP] token is appended at the end of each sentence and is used to divide two sentences. Additionally, BERT combines three different kinds of embeddings for each token: the segment embedding which indicates whether the word is from Sentence A or B; the position embedding which informs BERT the word's location in the sentence; and the token embedding which captures the meaning of the word. BERT fully comprehends

the structure and meaning of the input thanks to this combination [77].

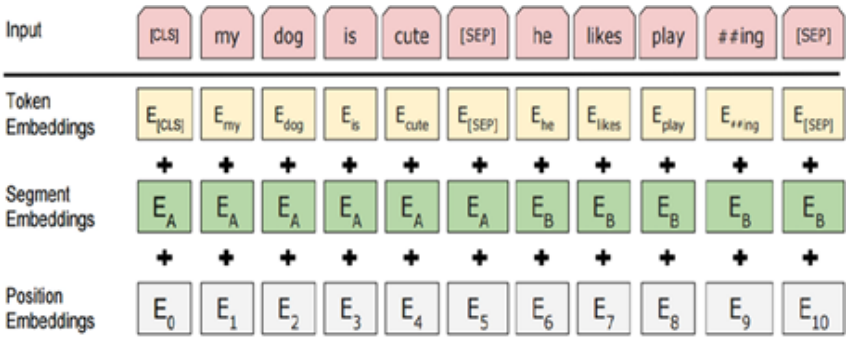

Figure 2.8: Input representation of BERT model [6]

2.7.3.4 BERT-based Models

Presented in Table 2.1

Model Name	Purpose / Use Case
PatentBERT	Classifies patents.
DocBERT	Used for document classification.
BioBERT	Specialized for biomedical text mining.
VideoBERT	Learns from unlabeled video data (e.g., YouTube).
SciBERT	Focused on scientific texts.
G-BERT	Uses medical codes and graphs to make medical recommendations.
TinyBERT (Huawei)	A compressed version of BERT trained via distillation. It's 7.5x smaller and 9.4x faster than BERT-base.
DistilBERT (Hugging Face)	A smaller, faster, and cheaper version of BERT created by simplifying its architecture.
ALBERT	A lighter BERT that uses less memory and trains faster.
SpanBERT	Improves BERT's ability to predict text spans.
RoBERTa	Trained longer and on more data to boost performance.

Table 2.1 – Continued from previous page

Model Name	Purpose / Use Case
ELECTRA	Produces high-quality text representations with a more efficient training approach.
ClinicalBERT	A version of the BERT model specially trained on clinical texts from electronic health records [77].

Table 2.1: BERT Variants and Their Use Cases [13]

2.7.4 GPT (Generative Pre-trained Transformer)

The sophisticated language model known as Generative Pre-trained Transformer, or GPT, was created by OpenAI [73]. Understanding and producing human-like writing is its primary objective. GPT has made it possible for computers to interact with humans in a far more meaningful and natural way.
In their 2017 publication "Attention is All You Need" [4] researchers first presented the transformer, a unique design upon which GPT is based. By using a mechanism of self-attention, transformers, as opposed to previous versions that read words one at a time, enable GPT to comprehend the meaning of each word based on its relationship with every other word in a phrase, regardless of where those words are written [78].

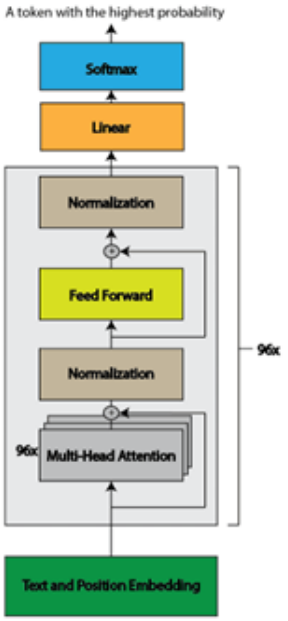

Figure 2.9: GPT Architecture [7]

2.7.4.1 How GPT Works

Transformer-based neural networks are used in GPT models to learn and produce human language. In order to learn how to predict the next word in a phrase based on the

preceding words, the model must first be pre-trained on a huge quantity of text using unsupervised learning. This process is known as language modeling. Thus, the model is better able to comprehend meaning, context, and grammar. Following pre-training, GPT may be tuned on smaller, task-specific datasets to enhance performance on specific applications such as content creation or text categorization. GPT generates text by predicting the most likely next word, one word at a time. This enables it to provide replies that are both contextually appropriate and coherent [8].

2.7.4.2 Different versions of the GPT model

Since its first release in 2018, OpenAI's Generative Pre-trained Transformer (GPT) has undergone several significant enhancements, each of which has added new features, enhanced performance, and more sophisticated language comprehension. These models from GPT-1 to the most recent GPT-4o have revolutionized the way that computers communicate with humans by enabling the generation of information, enabling multimodal functions like picture and audio processing, and powering tools like ChatGPT.

GPT-1: The first GPT model was released by OpenAI in 2018 as a proof of concept. It shown the initial potential for chatbots and content production by being able to produce text and respond to queries. But it had trouble with lengthy, complicated responses and frequently provided wrong answers.

GPT-2: With 1.5 billion parameters, GPT-2 which OpenAI released in 2019 was much more potent than GPT-1 and could produce lengthier, coherent writing [7].

GPT-3: The third generation of OpenAI's model, GPT-3, was far more efficient than GPT-2 since it was trained on 175 billion parameters. It was trained on a variety of data, such as the Common Crawl dataset (it is an open-source collection of web data gathered by a non-profit organization called Common Crawl) and Wikipedia. Its capacity to write computer code and perform at specific activities like creative writing and storytelling are two of its most notable qualities.

GPT-4: The latest model from OpenAI is called GPT-4. Being a large multimodal model (LMM), it has the ability to read both text and visual inputs. The most sophisticated version of the GPT model, it performs at a level comparable to humans on a range of academic and professional standards. It provides enhanced functionality, less biases, and higher-quality material [79].

GPT-4o: which was introduced in May 2024, is multilingual and multimodal. It is more adaptable since it can comprehend and produce written words, visuals, and sounds. Additionally, it doubles the speed of text creation and reduces expenses by 50% compared to GPT4 Turbo [7].

Model	Tokens	Size	Parameters	Dataset	Year	Features	Input Type	Drawbacks
GPT-1	-	12-layer decoder	117M parameters	Books corpus	2018	Used mostly for language modelling tasks and it is transformer based	A sequence of tokens and words	Limited Capacity, Limited Data, Cannot perform complex tasks, Limited applications
GPT-2	-	10 times the size of GPT-1	1.5B parameters	Downstream task datasets	2019	Text generation capabilities are improved and a chance for misuse	A sequence of tokens and words	Limited Control, Limited Data Diversity, Expensive computational requirements, Risk of improper information
GPT-3	4096 and 2049 tokens	100 times larger than GPT-2	175B parameters	Common Crawl	2020	Good NLP capabilities, language translation, summarization and generation of text	A sequence of tokens and words and images and tables	Limited Control, Limited Data Diversity, Lack of explanation, Ethical concerns
GPT-3.5	maximum token limit of 4096 tokens	96 layers	similar or larger number of parameters like GPT-3	-	2022	Improves user experience by delivering more precise and contextually relevant information	The input type typically consists of text data	Limited resources to train,Data Bias,Lack of Explainability,Limited Contextual Understanding,High Inference Latency
GPT-4	8192 and 32768 tokens	-	100T parameters	-	2023	Creative and technical writing tasks	A sequence of tokens and words and images and tables	-

Figure 2.10: Table of comparison of different versions of GPT model [8].

2.7.5 Flan-T5

Is a large transformer-based language model developed by Google in late 2022, in the paper "Scaling Instruction-Finetuned Language Models" [50], based on the T5 architecture. It can effectively process text thanks to its 12 transformer layers. It's one of Google's biggest models, with over 20 billion parameters, and it was trained on a vast quantity of data, including books, pages, and articles. Flan T5 comes in several sizes and is excellent for a variety of natural language applications, including question answering, content summarization, and text classification. It learns by predicting missing words in sentences and by using a technique that helps it understand the meaning of text more deeply [80].

2.8 Related Work and State of the Art

In recent years, numerous studies have explored the potential of natural language processing (NLP) and machine learning (ML) for detecting depression based on clinical interviews or user-generated texts. This section presents a synthesis of representative contributions in the field, focusing on methodologies, datasets, and reported outcomes.

Lorenzoni et al. [81] conducted a comprehensive experimental analysis of traditional ML classifiers applied to transcribed clinical interviews from the DAIC-WOZ dataset. After selecting 148 training and 37 test samples, they extracted 27 linguistic and sentiment features such as average response time, speaking rate, and sentiment scores. Their results showed that Random Forest and XGBoost achieved the highest accuracy (83.8%), outperforming Support Vector Machines (64.8%) and a baseline model.

In a different line of work, Jafari et al. [82] developed a Persian-language mental health chatbot capable of emotion and stress detection, leveraging datasets such as ArmanEmo and Reddit/Twitter corpora. Using models like XLM-RoBERTa, ParsBERT, and BiLSTM, they achieved promising results with F1-scores of 75.39% (emotion detection), 93.43% (toxicity validation), and 98% accuracy (stress classification). However, their pilot trials with only 9 users highlighted the need for larger-scale testing.

Danner et al. [83] proposed a transformer-based pipeline for depression detection using GPT-3.5 and BERT on DAIC-WOZ and Extended-DAIC datasets. Their findings emphasized the potential of large language models (LLMs), with GPT-3.5 yielding the highest F1-score (0.78) and outperforming fine-tuned BERT baselines.

Lau et al. [84] explored depression severity estimation via parameter-efficient tuning of pretrained models. Using prefix-tuning, they reduced trainable parameters and minimized overfitting risks in small datasets. Their approach achieved state-of-the-art results on DAIC-WOZ with RMSE of 4.67 and MAE of 3.80, outperforming both multimodal and fully fine-tuned baselines.

Burdisso et al. [85] investigated the ethical implications of using therapist prompts in text-based models. They compared models trained solely on participant responses versus therapist cues. While models trained on prompts achieved higher F1-scores (up to 0.90 with ensemble methods), the study highlighted risks of overfitting to scripted questions and called for more interpretable systems.

Multimodal approaches have also gained traction. Gimeno-Gómez et al. [86] built a transformer-based system that processes non-verbal features—facial expressions, gestures, gaze, and audio cues—from video datasets (e.g., D-Vlog, DAIC-WOZ). Their model achieved F1-scores up to 0.78 on D-Vlog and outperformed prior work on all benchmarks.

Similarly, Aghaei and Khodaei [87] integrated attention mechanisms in a cross-modal LSTM architecture for audio-video fusion. Their model surpassed unimodal baselines with an RMSE of 5.2 on DAIC-WOZ.

Patapati [88] introduced a tri-modal architecture combining GPT-4 with BiLSTM encoders for text, audio, and video. Using leave-one-subject-out validation, the model achieved an impressive 91.01% accuracy and 85.95% F1-score, showing the power of LLMs when fused with temporal behavioral features.

Ahmed et al. [89] proposed an uncertainty-aware multimodal framework using EEG, facial landmarks, audio, and text. With dropout and probabilistic layers, their model remained robust to missing modalities and achieved F1-scores exceeding 94% across several datasets, including DAIC-WOZ.

Lastly, Flores et al. [90] focused on temporal facial dynamics such as eye gaze and action units, extracted via OpenFace. Their CNN-based models, enhanced with attention, achieved up to 0.81 F1-score on DAIC-WOZ, reinforcing the predictive power of nonverbal

cues in depression screening.

These studies collectively underline the growing importance of personalized, multimodal, and interpretable AI solutions for mental health monitoring and diagnosis.

Author(s)	**Model Type**	**Modalities Used**	**Dataset(s)**	**Key Results / Contributions**
Jafari et al. (2025)	XLM-R, ParsBERT, BiLSTM	Text (Persian), Emotion, Toxicity	ArmanEmo, Reddit, Twitter	F1: 75.3% (emotion), 93.4% (toxicity); chatbot prototype
Lorenzoni et al. (2024) sentiment feature study	Random Forest, XGBoost	Textual features	DAIC-WOZ	Accuracy up to 83.8%; linguistic
Burdisso et al. (2024)	Ensemble, Transformer	Text (therapist prompts)	DAIC-WOZ	F1 up to 0.90; ethics of using therapist prompts
Patapati (2024)	GPT-4 + BiLSTM	Text, Audio, Video	DAIC-WOZ	Accuracy = 91.01%, F1 = 85.95%; trimodal architecture
Gimeno-Gómez et al. (2024)	Transformer-based	Non-verbal (face, gaze, audio)	D-Vlog, DAIC-WOZ	F1 up to 0.78; strong on multimodal
Danner et al. (2023)	GPT-3.5, BERT	Text	DAIC-WOZ, Extended-DAIC	GPT-3.5 outperforms; F1 = 0.78
Ahmed et al. (2023)	Uncertainty-aware fusion	EEG, Audio, Text, Video	DAIC-WOZ + others	F1 > 94%; robust to missing modalities
Aghaei & Khodaei (2023)	Attention + LSTM	Audio-Visual	DAIC-WOZ	RMSE = 5.2; cross-modal performance
Lau et al. (2023)	Prefix-tuned Transformers	Text	DAIC-WOZ	RMSE = 4.67; MAE = 3.80; efficient tuning
Flores et al. (2022)	CNN + Attention	Facial dynamics (OpenFace)	DAIC-WOZ	F1 = 0.81; focused on eye gaze + AUs

Table 2.2: Summary of Reviewed Studies on Depression Detection

2.9 Conclusion

By allowing machines to comprehend, produce, and communicate with human language in complex ways, generative AI has greatly improved the science of natural language processing. These models, which are usually based on transformer models, are able to provide insightful and well-organized replies by using attention processes to identify contextual linkages in text. Generative AI systems are capable of learning intricate linguistic patterns through extensive dataset training, which allows them to carry out a variety of activities, including producing unique material and responding to queries. These models are capable of handling large-scale language problems well because they combine position-based learning, self-attention, and parallel processing. All things considered, generative AI is a noteworthy development in artificial intelligence that enhances human-computer interaction and encourages innovation in a variety of fields, such as healthcare, education, and customer service.

Chapter 3

Methodology

3.1 Introduction

The proposed system, titled "**Development of an Intelligent Online Response System (Chatbot) for Mental Illnesses and Severe Pathologies Based on Artificial Intelligence Models**", is a sophisticated multimodal chatbot designed to detect various mental health conditions such as depression, anxiety, and post-traumatic stress disorder (PTSD) by analyzing user interactions in the form of dialogues between patients and the chatbot. This detection is achieved through the integration of multimodal AI models that combine textual and visual features for accurate classification, as well as generative AI components responsible for dynamically generating clinically relevant questions during the conversation.

At its core, the system relies on the Distress Analysis Interview Corpus - Wizard of Oz (DAIC-WOZ) dataset, which provides aligned transcripts and facial video recordings of clinical interviews. The architecture of the proposed system follows a modular and interpretable pipeline that includes the following core stages showing in the figure below 3.1 :

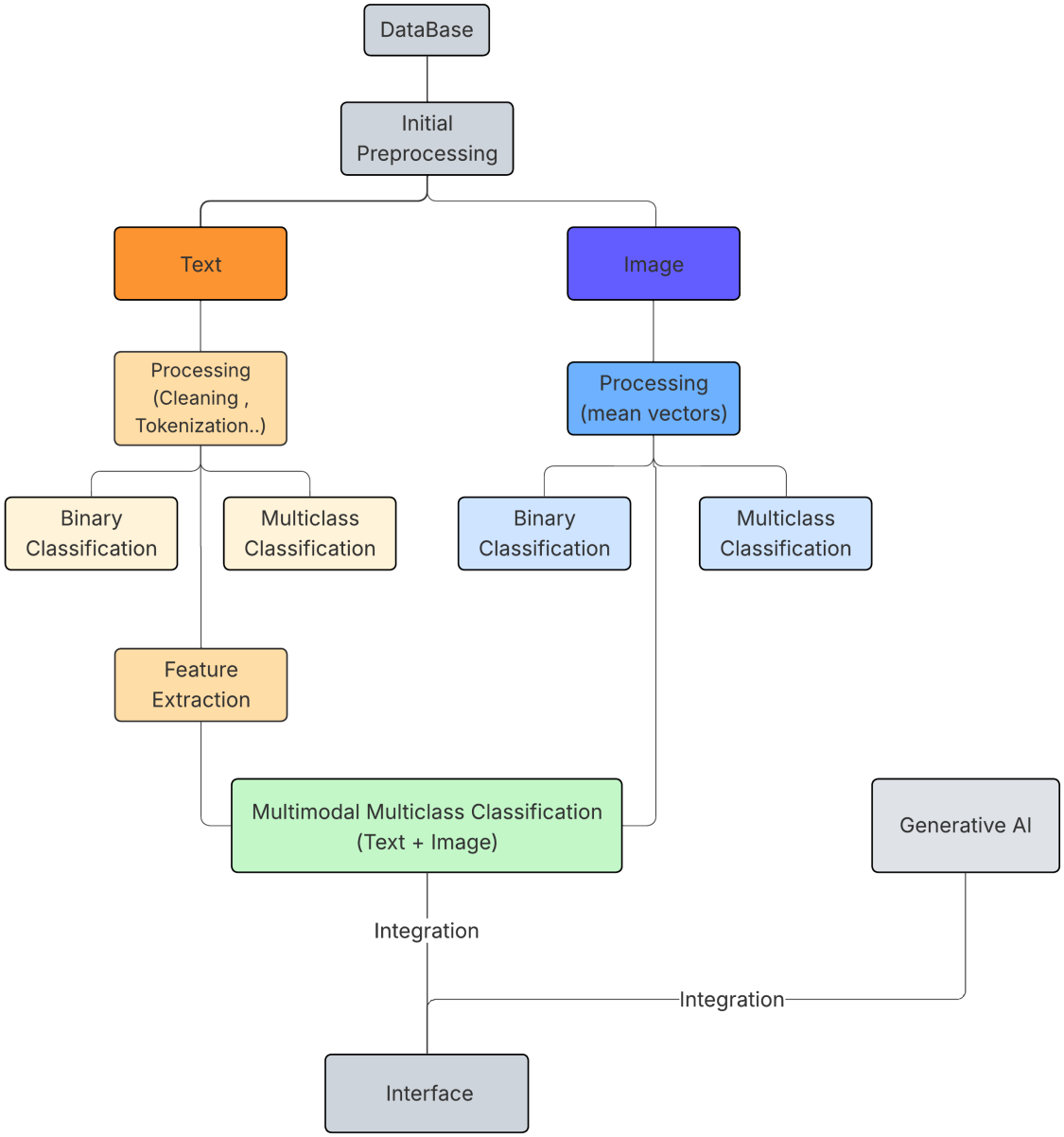

Figure 3.1: System Workflow for Multimodal AI-Based Mental Illness Detection Chatbot

3.2 Dataset Description

The first and most essential step in building the system is data preprocessing. At this stage, the raw multimodal data from the DAIC-WOZ dataset is carefully prepared and organized to support effective training of the machine learning models. Since the dataset includes different types of information such as written transcripts and facial video data this step is crucial to bring all these elements into a consistent and usable format for feature extraction, classification, and analysis.

The system utilize the Distress Analysis Interview Corpus - Wizard of Oz (DAIC-WOZ) dataset, a publicly available resource hosted on Kaggle [9], designed for the development and evaluation of mental health diagnostic systems. The dataset includes clinical interviews aimed at detecting psychological conditions such as depression, anxiety, and post-traumatic stress disorder (PTSD).

Figure 3.2: DAIC WOZ Dataset in Kaggle [9]

The interviews are conducted in a controlled setting with a virtual interviewer named Ellie, an animated character whose responses are remotely controlled by a human operator. This setup is intended to simulate a realistic and emotionally engaging interaction while maintaining consistency across sessions.

The DAIC-WOZ dataset consists of the following components:

- Audio files, although not used directly in this study.
- Textual transcripts of the spoken dialogue between the participant and Ellie.
- PHQ-8 questionnaire responses serving as the clinical ground truth labels for depression severity.
- Facial landmark features, which is extracted frame-by-frame using OpenFace and Non-verbal behavior annotations, including head movements and gaze directions.

In total, the dataset comprises 189 interview sessions, with session durations ranging from 7 to 33 minutes (average 16 minutes). However, sessions 342, 394, 398, and 460 were removed from the analysis as they were marked for exclusion due to data quality issues or incomplete information.

Also participants 373, 402, 416, 417, 444, 451, 458, 480 were removed for one of the following reasons:

- Excluded sessions: Some sessions may have been excluded due to technical issues or non-compliance with data collection protocols.

- Interrupted sessions: Some sessions may have been interrupted before completion, resulting in incomplete data that cannot be used for analysis.
- Missing transcripts: For some participants, transcription files were missing, making it impossible to extract the textual features required for analysis.

So, our final dataset contains **181** participants after cleaning.

PHQ-8: The Patient Health Questionnaire-8 (PHQ-8) is a standardized, clinically validated self-report measure used to assess and screen for depression symptoms in people.(in the figure 3.3)

Over the *last 2 weeks*, how often have you been bothered by any of the following problems?	PHQ-8 BFRSS conversion	Not at all 0 - 1 day	Several days 2 - 6 days	More than half the days 7 - 11 days	Nearly every day 12 - 14 days
1. Little interest or pleasure in doing things		0	1	2	3
2. Feeling down, depressed, or hopeless		0	1	2	3
3. Trouble falling or staying asleep, or sleeping too much		0	1	2	3
4. Feeling tired or having little energy		0	1	2	3
5. Poor appetite or overeating		0	1	2	3
6. Feeling bad about yourself—or that you are a failure or have let yourself or your family down		0	1	2	3
7. Trouble concentrating on things, such as reading the newspaper or watching television		0	1	2	3
8. Moving or speaking so slowly that other people could have noticed. Or the opposite—being so fidgety or restless that you have been moving around a lot more than usual		0	1	2	3

Interpretation of Total Score/Total Score Depression Severity: 0–4 None, 5–9 Mild depression, 10–14 Moderate depression, 15–19 moderately severe depression, 20–24 severe depression.

Figure 3.3: PHQ8 Questionnaire [10]

Each of the eight items in the PHQ-8 asks the respondent to reflect on how often they have been bothered by specific symptoms over the past two weeks, using the following 4-point Likert scale:

- **0** – Not at all
- **1** – Several days
- **2** – More than half the days
- **3** – Nearly every day

The eight questions cover the following depressive symptom domains:

1. Little interest or pleasure in doing things
2. Feeling down, depressed, or hopeless
3. Trouble falling or staying asleep, or sleeping too much
4. Feeling tired or having little energy
5. Poor appetite or overeating
6. Feeling bad about yourself—or that you are a failure or have let yourself or your family down

7. Trouble concentrating on things, such as reading or watching television
8. Moving or speaking so slowly that other people could have noticed, or the opposite: being so fidgety or restless that you were moving around more than usual

The total PHQ-8 score ranges from **0** to **24**, obtained by summing the individual scores of the eight items. The interpretation is generally as follows:

- **0–4**: None to minimal depression
- **5–9**: Mild depression
- **10–14**: Moderate depression
- **15–19**: Moderately severe depression
- **20–24**: Severe depression

A threshold score of **10 or higher** is commonly used in clinical and research settings to indicate the presence of clinically significant depressive symptoms.

3.2.1 Textual data Description

Textual data is extracted from two key fields: `Ellie_Transcripts` and `Participant_Transcripts`, with the latter being the primary focus. Extensive preprocessing was necessary due to the presence of non-standard linguistic elements, transcription errors, and inconsistent formatting - all of which are typical in spoken dialogue datasets collected in clinical or semi-structured settings.

In total, over 189 unique participant interviews were analyzed in this textual study, spanning a range of psychological profiles. The diversity of language used - from short, hesitant answers to longer narrative segments - makes this corpus especially valuable for training natural language processing (NLP) models in the mental health domain.

3.2.1.1 Label Distribution and Initial Imbalance

The initial dataset presented a binary label column based on PHQ-8 scores, with values mapped as follows: `0` indicating non-depressed participants and `1` for those classified as at risk of depression. This labeling strategy was designed to simplify the classification task and align with clinically recognized thresholds for screening depression.

However, an immediate challenge emerged: label imbalance. Approximately 70% of participants were labeled as `0`, with only 30% falling into the `1` category. This imbalance is common in clinical datasets, particularly those involving voluntary participation, and poses risks of bias during model training.

To better understand the data distribution, exploratory visualization techniques such as bar plots and pie charts were applied, revealing class skewness.

The imbalance in class representation necessitated specific pre-processing strategies, including data augmentation and resampling, which are discussed in subsequent sections.

3.2.2 Image Data Description

Our system's image modality is based on visual characteristics collected from DAIC-WOZ interview videos. These characteristics give essential nonverbal behavioral informations, such as eye movement, face orientation, and expressions motion, which are critical for determining mental health status. To successfully handle this visual information, we use two output files: "$CLNFgaze.txt$" and "$CLNFfeatures.txt$".

A. Gaze Data (XXX CLNF gaze.txt): This file provides detailed information about eye gaze direction and head orientation at each frame of the interview. Each record contains:

- **Timestamp:** Time in seconds when the frame was captured.
- **Confidence:** A float between 0 and 1 indicating how reliable the detection is.
- **Success:** Binary value (1 = Depressed, 0 = Non Depressed).
- **Eye Gaze Vectors:**
  o $x0, y0, z0$: Gaze vector of the left eye in world coordinates.
  o $x1, y1, z1$: Gaze vector of the right eye in world coordinates.
- **Head-Relative Eye Vectors:**
  o x_h0, y_h0, z_h0: Left eye vector in the head's local coordinate system.
  o x_h1, y_h1, z_h1: Right eye vector in the head's local coordinate system.

Hence, each valid frame contributes four key 3D vectors that describe the participant's gaze direction both in absolute space and relative to their head.

B. Facial Landmarks Data (XXX CLNF features.txt): This file captures the 2D positions of 68 facial landmarks for each frame, reflecting real-time facial expressions:

- **Timestamp:** Synchronized with gaze data.
- **Confidence and Success:** Same definitions as in gaze data.
- **x0–x67:** Horizontal coordinates of facial landmarks in pixel space.
- **y0–y67:** Vertical coordinates corresponding to each x.

These landmarks are used to track facial movement patterns such as smiling, frowning, or eye blinking over time.

3.3 Dataset Preprocessing

3.3.1 Text Data Preprocessing and Preparation

Prior to any classification task or data augmentation strategy, a rigorous text preprocessing pipeline was applied to the participant transcripts from the DAIC-WOZ dataset. This step was essential to ensure consistency in the data, eliminate transcription noise, and prepare the input format compatible with downstream transformer-based models such as BERT and SBERT. Below, we describe each preprocessing operation applied.

1. Transcript Cleaning and Token Normalization

- The multiclass version of the dataset included five levels of severity:
  - Class 0: No depression
  - Class 1: Mild
  - Class 2: Moderate
  - Class 3: Moderately Severe

  - Class 4: Severe

- Classes 2–4 were under-represented compared to Classes 0 and 1.

- **Handling missing and irrelevant data:** Initially, we identified and removed records that contained null values in either the participant transcripts or associated labels. Utterances with fewer than a defined number of words (e.g., less than 3 tokens) were excluded as they lacked sufficient semantic content for downstream modeling. Moreover, non-informative system tags such as `[laughter]`, `[long pause]`, and artifacts like `[noise]` were stripped from the transcripts to minimize noise.

- **Removal of Dialogue Artifacts:** The original transcripts contained dialogue tags (e.g., "Participant:", "Ellie:") and special characters (e.g., "–", "[noise]", "...") which were systematically removed using regular expressions to ensure semantic clarity.

- **Standardization of Case:** All text was converted to lowercase to ensure case-invariant embedding computation.

- **Correction of Transcription Errors:** Leveraging a manually curated dictionary of common misrecognized phrases and abbreviations (e.g., "imma" to "I'm going to", "gonna" to "going to"), the transcripts were corrected using rule-based substitutions. This step improved embedding coherence by aligning non-standard expressions with expected lexical forms.

- **Whitespace Normalization:** Redundant spaces were eliminated to avoid token misalignment during embedding and tokenization.

2. Sentence Structuring and Filtering

- **Concatenation of Utterances:** Since most participant responses consisted of short, fragmented utterances, we grouped sequences of consecutive replies into aggregated dialogue blocks per participant, increasing context length without including interviewer content.

- **Participant-Only Filtering:** All turns spoken by the virtual agent "Ellie" were excluded. Only participant responses were retained to ensure that the model focused solely on the mental state expressions of the user.

- **Length Filtering:** Transcripts with fewer than 20 tokens were discarded from training to avoid poor-quality inputs. This was particularly crucial for SBERT and BERT-based models, which rely on contextual richness.

3. Lemmatization and Stopword Handling

- **Lemmatization:** All remaining tokens were lemmatized using spaCy's English model to reduce inflected forms to their base lemma (e.g., "talked" to "talk", "going" to "go"), ensuring vocabulary uniformity across responses.

- **Stopword Management:** Standard stopwords (e.g., "the", "and", "is") were retained in the final input. This choice was based on empirical findings in mental health NLP that suggest that even function words may carry stylistic or emotional significance (e.g., high use of "I" in depressed speech).

4. Label Extraction and Structuring

- **Binary Labels:** Based on PHQ-8 scores, a binary label was assigned:
  - `0`: Non-depressed (PHQ-8 score < 10)
  - `1`: Depressed (PHQ-8 score 10)
- **Multiclass Labels:** For finer granularity, we assigned depression severity levels from 0 to 4 based on established PHQ-8 intervals. These were stored under the `classes` column for use in multiclass experiments.
- **Metadata Storage:** Each cleaned and labeled transcript was saved along with its corresponding `participant_id`, `Gender`, `PHQ-Score`, and `PHQ-Binary` into a structured DataFrame used in all further stages.

5. Final Dataset Statistics

The final preprocessed text corpus consisted of 181 participant samples after filtering. Each entry consisted of a cleaned textual block and its corresponding metadata. The distribution across labels (binary and multiclass) was visualized and logged to inform the augmentation strategy (see Section 3.3.1).

6. Example – Before and After Preprocessing

- **Original:** "Participant: i'm...i'm not sure. like, it's been...kinda hard, you know? uhm, maybe i'm just tired or something..."
- **After Preprocessing:** "i am not sure it has been kind of hard you know maybe i am just tired or something"

This preprocessing pipeline was applied to each participant transcript, generating a clean and normalized version used for both binary and multiclass classification tasks. It significantly improved text consistency and reduced vocabulary noise, leading to better feature extraction and downstream model performance. In this part, we describe the precise augmentation and balancing approaches used in the textual modality, with a focus on binary and multiclass classification cases. The goal was to correct the class imbalance while maintaining the semantic integrity of the original dataset.

7.Augmentation Methodology

- We applied a class-aware **MixUp** technique on embeddings extracted via Sentence-BERT (SBERT).
- **SBERT** is a variant of BERT fine-tuned using a Siamese network architecture to produce semantically meaningful sentence embeddings. Unlike standard BERT, SBERT enables efficient comparison and manipulation of sentence-level vectors, making it particularly well-suited for data augmentation tasks.
- These sentence embeddings capture deep semantic information from the original text, allowing for realistic interpolation between examples.
- The MixUp method generates synthetic samples by linearly interpolating embeddings of two samples x_i and x_j from the same class.

$$\tilde{x} = \lambda x_i + (1 - \lambda)x_j, \quad \tilde{y} = \lambda y_i + (1 - \lambda)y_j$$

- The coefficient λ controls the degree of mixing between the two embeddings. The parameter α in the Beta distribution determines the shape of the distribution — a smaller α yields values of λ closer to 0 or 1 (thus creating samples more similar to one of the originals), while a larger α yields more blended samples.
- In our experiments, we used $\alpha = 0.4$, a value that ensures enough variation in interpolation while maintaining semantic coherence between generated embeddings and original texts.
- Augmentation was applied exclusively on the minority class (i.e., `Depressed` in binary, and classes 2–4 in multiclass), and stratified by gender in the binary setting to preserve demographic balance.
- We applied **class-conditional MixUp** to only the minority classes (2, 3, and 4).

1.Augmentation Balancing

- After MixUp, we applied **random oversampling** to the minority class to fully match the number of samples in the majority class.
- This ensured a 1:1 ratio between depressed and non-depressed samples.

Step	**Before Augmentation**	**After Augment+balancing**
Class 0 (Non-depressed)	125	438
Class 1 (Depressed)	56	406
Total	181	844

Table 3.1: Sample distribution before and after augmentation – Binary classification

Class Label	**Before**	**After Augmentation & Balancing**
Class 0 (None)	94	486
Class 1 (Mild)	58	486
Class 2 (Moderate)	24	486
Class 3 (Mod. Severe)	20	486
Class 4 (Severe)	52	486
Total	248	2430

Table 3.2: Sample distribution before and after augmentation – Multiclass classification

- Post augmentation, we applied **random oversampling** across all classes to achieve a uniform sample count per class.
- This step ensured balanced class representation without relying solely on synthetic data.

This approach had several advantages:

- It generated smooth variations of existing samples, helping the model generalize better.
- It avoided overfitting by reducing reliance on exact training instances.

- It was label-preserving, as we applied MixUp only within the same class during binary classification.

After augmentation and MixUp-based oversampling, the dataset exhibited improved class balance and diversity, strengthening the robustness and fairness of downstream classifiers.

2.Visualization and Observations

- Distribution plots confirmed that class frequencies became nearly uniform after augmentation and balancing.
- The use of MixUp helped mitigate overfitting by introducing nuanced interpolations within minority classes while preserving contextual semantics.

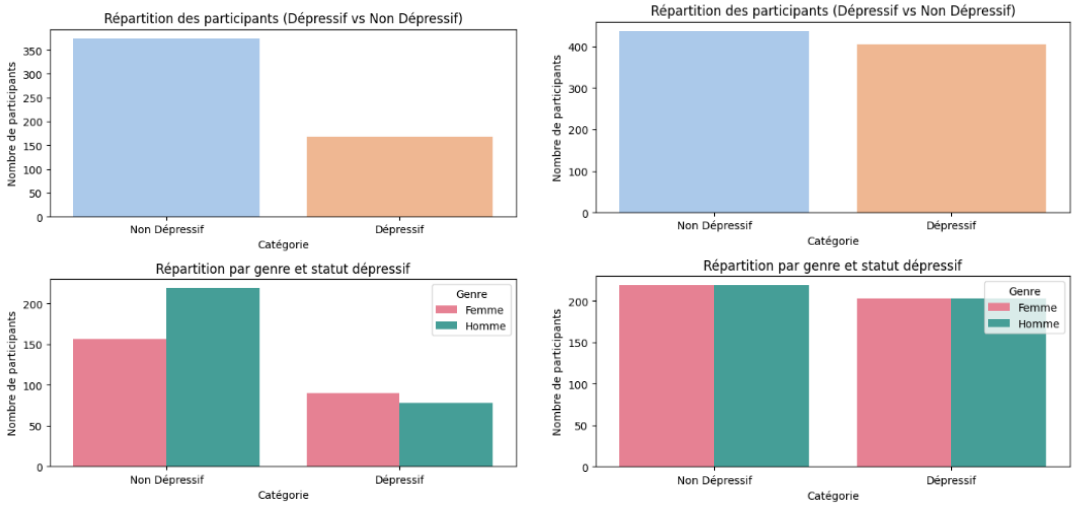

Figure 3.4: Binary classification dataset: class distribution before (left) and after (right) MixUp-based augmentation and balancing.

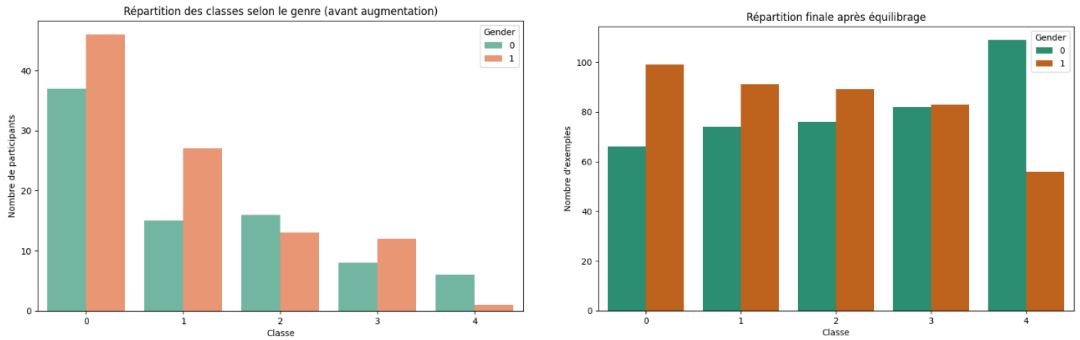

Figure 3.5: Multiclass dataset: class distribution before (left) and after (right) augmentation and balancing.

3.3.2 Image Data Preprocessing and Preparation

1. Frame Filtering:

Only high-quality frames are retained by applying two conditions:

- $Confidence >= 0.5$: Ensures reliable detection.
- $Success == 1$: Filters out frames with failed tracking or corrupted data.

This step guarantees that only valid and meaningful visual cues are processed.

2. Vector Averaging (Temporal Aggregation):

Rather than processing each frame independently, we compute the mean vector for each feature across all valid frames in a session. This includes:

- Averaging the 3D eye gaze vectors.
- Averaging the 2D facial landmarks $(x0 - x67, y0 - y67)$.

This results in a single, compact feature vector per participant session, capturing their average facial orientation, eye contact behavior, and overall facial dynamics throughout the interview.

3. Augmentation of image data

We used Manifold Mixup to balance our dataset and make the model more robust. Since the original data was imbalanced, this technique creates better training examples by more intelligently combining features.
Manifold Mixup is a data augmentation approach that expands on the previously stated conventional Mixup method 3.3.1. Mixup utilizes linear interpolation directly to the input data, whereas Manifold Mixup uses the same interpolation approach to hidden representations in the neural network, including intermediary layer outputs. Since smoother decision limits are encouraged in deeper feature spaces, the model is better able to generalize. Our methodology ensures that each new sample maintains significant semantic structure while increasing the diversity of training data by augmenting the data according to the class label, whether for binary or multiclass classification.

1.Augmentation Methodology

- Augmentation focused on balancing the classes:
  - Each new sample was created by interpolating a depressed and a non-depressed example.
  - This ensured that most synthetic data represented the minority (depressed) class, helping to reduce imbalance.
- A Beta distribution with parameter $\alpha = 0.3$ was used to generate the mixing coefficient (λ).
- Gender was assigned randomly (50% male, 50% female) to maintain demographic diversity.
- A total of **500** new samples were generated and assigned unique $participant_ids$ starting after the last ID in the original dataset.
- Finally, the augmented data was combined with the original dataset, creating a balanced version ready for training, The result is shown in the table mentioned 3.3.2.

Step	Before	After Augment+balancing
Class 0 (Non-depressed)	125	398
Class 1 (Depressed)	56	283
Total	181	681

Table 3.3: Sample distribution before and after augmentation – Binary classification

- For each synthetic sample:
  - Two feature vectors were randomly selected from different classes.
  - Features were linearly interpolated using $X_{\text{mix}} = \lambda \cdot X_1 + (1 - \lambda) \cdot X_2$.
  - The corresponding PHQ_{Score}, PHQ_{Binary}, and Class were also computed through interpolation and rounding.
- The final augmented dataset was created by concatenating the original and synthetic samples into one balanced DataFrame,The table below displays the result.

Class Label	Before	After Augmentation & Balancing
Class 0 (None)	94	168
Class 1 (Mild)	58	153
Class 2 (Moderate)	24	147
Class 3 (Mod. Severe)	20	130
Class 4 (Severe)	52	83
Total	248	681

Table 3.4: Sample distribution before and after augmentation – Multiclass classification

2.Visualization and Observations Both binary (Depressed vs. Non-Depressed) and multiclass (5 severity levels) distribution plots (before and after augmentation) clearly show better class balance.

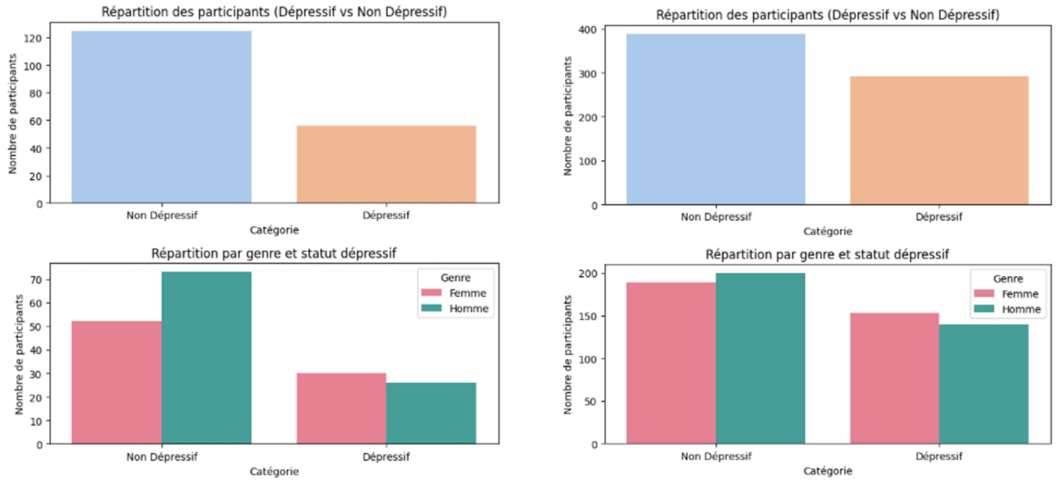

Figure 3.6: Binary classification image dataset: class distribution before (left) and after (right) Manifold MixUp-based augmentation and balancing.

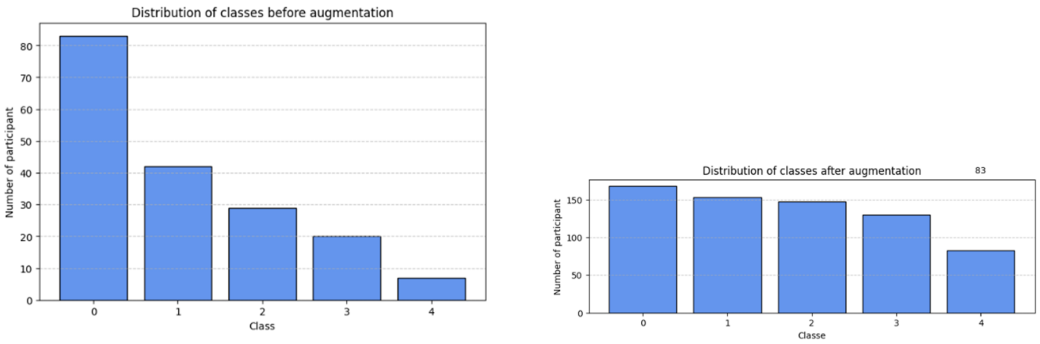

Figure 3.7: Multiclass classification image dataset: class distribution before (left) and after (right) Manifold MixUp-based augmentation and balancing.

3.4 Text Classification: Methodological Approach

3.4.1 Model functionality:

The core of our text classification pipeline is built upon transformer-based models, a class of deep learning architectures that has revolutionized natural language processing (NLP) in recent years. Introduced by Vaswani et al. in 2017 [4], the Transformer architecture relies heavily on a self-attention mechanism to capture contextual relationships between tokens in a sentence, allowing models to process text sequences in parallel rather than sequentially.

We explored multiple Transformer variants in our experiments, including:

- **BERT (Bidirectional Encoder Representations from Transformers):**
  BERT serves as the foundational model upon which many other architectures build. It leverages a deep bidirectional Transformer encoder, allowing it to learn context from both the left and right of a token simultaneously. Pre-trained on large-scale English corpora such as Wikipedia and BookCorpus, BERT captures rich general-purpose language representations. For our experiments, we applied the base version (BERT-base).

- **RoBERTa (A Robustly Optimized BERT Approach):**
  RoBERTa builds upon the BERT architecture but improves it through a more extensive and dynamic pre-training methodology. Specifically, it removes the Next Sentence Prediction (NSP) objective and instead focuses on longer training over larger corpora (Common Crawl, OpenWebText, and more). RoBERTa benefits from larger batch sizes and longer sequences during training, enhancing its ability to model complex sentence structures and dependencies—particularly useful for nuanced dialogue data such as mental health interviews.

- **ClinicalBERT (Domain-Specific Fine-Tuned BERT):**
  ClinicalBERT extends the general-purpose BERT by further training it on clinical texts, specifically from the MIMIC-III database, which includes intensive care unit (ICU) notes. This model is particularly adept at understanding medical terminology, abbreviations, and structure typical of clinical notes and patient interviews. Instead of only learning general language patterns, ClinicalBERT incorporates domain-specific semantics, which makes it highly relevant for psychiatric evaluation and dialogue-based diagnosis. As illustrated in Figure 3.8, ClinicalBERT can process longitudinal clinical notes and assign risk probabilities at each stage, demonstrating its temporal awareness and diagnostic power in real-world clinical applications.

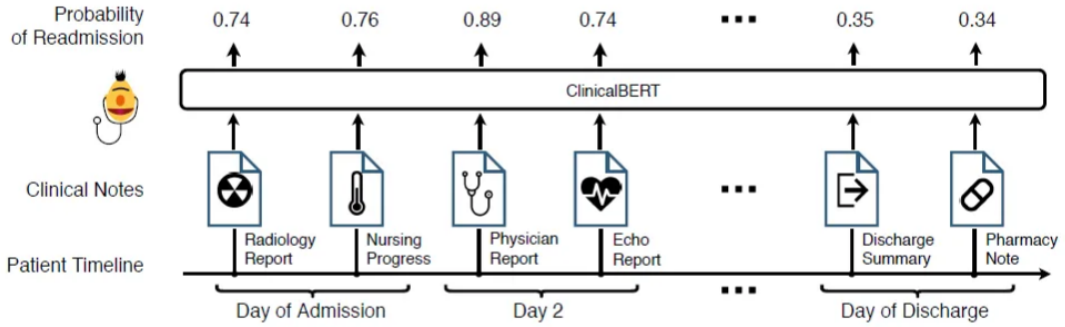

Figure 3.8: ClinicalBERT used across time-structured patient notes to estimate readmission probabilities. [11]

- **MentalBERT (Emotion and Clinical-Text Focused Variant)**
  MentalBERT is a specialized variant of BERT pre-trained and fine-tuned on corpora related to mental health, including Reddit mental health discussions, clinical narratives, and emotion-rich texts. Its vocabulary and weight initialization are specifically adapted to capture mental state indicators such as emotional tone, psychiatric terminology, and conversational dynamics. Unlike traditional models trained solely on medical notes or general language, MentalBERT excels in identifying subtle affective shifts and latent psychological cues often embedded in naturalistic dialogues.

3.4.2 Binary Classification:

BERT-base:

1.Model Description : The proposed model relies on the **BERT (Bidirectional Encoder Representations from Transformers)** architecture in its pretrained version `bert-base-uncased`, adapted here for binary classification of depressive disorders. The full architecture, named `BERTClassifier`, consists of three main components:

- **BERT Encoder**: The core of the model is the `BertModel` module from the `transformers` library. This model includes 12 transformer encoder layers, each with multi-head self-attention mechanisms. Attention dynamically weighs each word in the text based on its global context within the sequence, capturing long-range dependencies and contextual meaning.
- **[CLS] Token**: At the end of the BERT encoding process, the representation of the special [CLS] token is extracted. This vector (of size 768) represents the overall encoding of the input sequence and serves as a contextual summary of the text.
- **Dropout & Classification Layer**: A dropout rate of 0.5 is applied to the [CLS] token output to prevent overfitting. This representation is then passed through a linear layer (`Linear(768, 2)`) producing two logits, indicating the probability of belonging to either the *healthy* or *depressed* class.

The model was trained using the `AdamW` optimizer with an initial learning rate of 10^{-5} and L2 regularization (`weight decay`) of 0.01. A `ReduceLROnPlateau` scheduler was used to dynamically adjust the learning rate in case of performance stagnation on the validation set. An *early stopping* mechanism (patience = 15) was implemented to prevent overfitting.

Training was conducted on a balanced and preprocessed dataset, with an 80% / 20% split between training and validation. The best-performing model was saved during training based on validation performance.

Element	Value	Description
`Base model`	`bert-base-uncased`	Pretrained BERT model of uncased type (case-insensitive).
`Number of classes`	2	Binary classification (0 = healthy, 1 = depressed).
`Token max_length`	512	Maximum length of tokenized sequences (truncated/padded if necessary).
`Batch size`	32	Number of samples processed per training iteration.
`Dropout rate`	0.5	Dropout rate applied for regularization.
`Loss function`	CrossEntropyLoss	Suitable loss function for multi-class classification.
`Optimizer`	AdamW	Adam optimizer with separate L2 weight decay management.
`Learning rate`	1×10^{-5}	Initial learning rate for weight updates.
`Weight decay`	0.01	L2 regularization applied to weights to reduce overfitting.
`Scheduler`	ReduceLROnPlateau	Automatically reduces the learning rate upon validation loss stagnation.
`Scheduler patience`	2	Number of epochs with no improvement before reducing learning rate.
`Scheduler factor`	0.5	Learning rate reduction factor.
`Number of epochs`	100	Maximum number of training epochs.
`Early stopping patience`	15	Training is stopped early if no validation loss improvement.
`Train/val split`	80% / 20%	Data split between training and validation sets.
`Execution device`	GPU if available	The model is trained on CUDA if available, otherwise on CPU.

Table 3.5: Parameters and hyperparameters used for training the BERTClassifier model

2.Model Architecture:

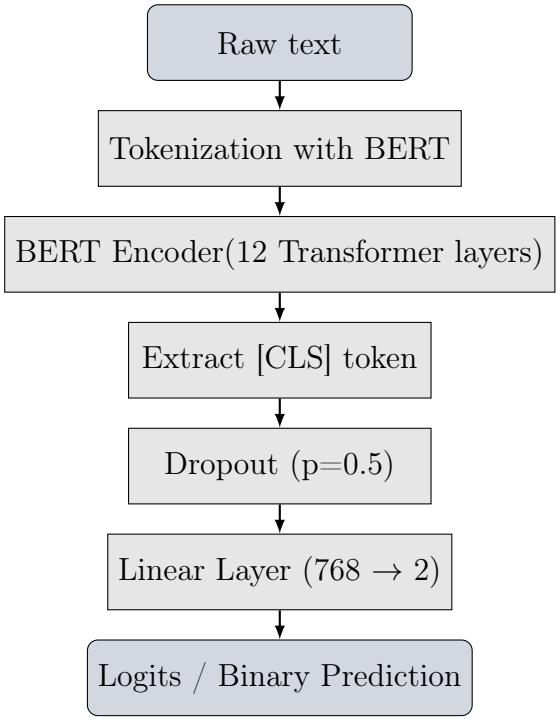

Figure 3.9: BERTClassifier architecture for binary classification

Layer	**Output Dimensions**	**Parameters**	**Description**
`BertModel (base-uncased)`	(batch_size, 512, 768)	~109M	Transformer encoder based on multi-head attention (12 layers), pretrained on a large corpus.
`Dropout (p=0.5)`	(batch_size, 768)	0	Regularization layer to prevent overfitting by randomly disabling neurons.
`Linear (768 → 2)`	(batch_size, 2)	1,538	Binary classification layer producing logits for the two classes (0 or 1).

Table 3.6: Architecture of the BERTClassifier model

Note : It is important to note that attention is not added manually in our architecture as it is natively integrated into the BERT model. Each BERT layer includes a multi-head attention mechanism that allows the model to weight tokens differently depending on their contextual importance. These attention mechanisms are central to the transformer architecture and enable it to effectively capture long-range dependencies between words.

RoBERT-base:

1.Model description: The architecture implemented for binary classification is based on the **RoBERTa** model, a robustly optimized variant of BERT developed by Facebook AI. We employed the `roberta-base` version, consisting of 12 transformer layers and 125 million parameters. RoBERTa is well-known for its ability to capture complex contextual representations thanks to its pretraining on a diverse and large-scale corpus using byte-level BPE tokenization.

The proposed classifier, named `RoBERTaClassifier`, is constructed with the following components:

- **RoBERTa Encoder** : The core of the model uses `RobertaModel` to encode input sequences. The self-attention mechanism embedded within each of the 12 layers allows dynamic weighting of tokens based on their contextual importance.
- **[CLS] Token Extraction** : The model captures the contextual summary of the entire sentence by extracting the embedding of the first token in the sequence (position 0 of `last_hidden_state`), which serves as a proxy for global representation.
- **Dropout & Linear Layer** : A dropout layer (rate = 0.3) is applied to reduce overfitting by randomly deactivating neurons during training. The resulting vector is then passed through a linear layer of size $768 \rightarrow 2$ to produce logits for binary classification.

Training was conducted using the `AdamW` optimizer, known for its decoupled weight decay, ensuring more effective regularization. A learning rate of 2×10^{-5} was chosen along with weight decay of 0.01. To dynamically adjust learning progress, a `ReduceLROnPlateau` scheduler was integrated, which reduces the learning rate by half if the validation F1-score plateaus for 3 consecutive epochs.

Furthermore, an *early stopping* mechanism was employed with a patience of 10 epochs to prevent overfitting. The model was trained for up to 50 epochs, with performance monitored through both loss and classification metrics on a validation set comprising 30% of the full dataset.

This configuration ensures a balanced trade-off between generalization, performance, and training stability—critical for downstream tasks in mental health detection.

Component	Value	Description
`Base Model`	`roberta-base`	Pretrained transformer model using byte-level BPE tokenizer, trained on a large English corpus.
`Classification Type`	Binary (2 classes)	Predicts depression status: 0 = no depression, 1 = depression.
`Max Sequence Length`	512	Maximum number of tokens per input text after tokenization.
`Batch Size`	32	Number of examples processed in one forward/backward pass.
`Dropout Rate`	0.3	Regularization method to prevent overfitting by randomly deactivating 30% of neurons.
`Optimizer`	AdamW	Adaptive optimizer with decoupled weight decay for stability and generalization.
`Learning Rate`	2×10^{-5}	Initial learning rate for gradient descent updates.
`Weight Decay`	0.01	L2 regularization coefficient to penalize large weights.
`Loss Function`	CrossEntropyLoss	Suitable for classification tasks with mutually exclusive labels.
`Scheduler`	ReduceLROnPlateau	Lowers learning rate when validation F1-score plateaus.
`Scheduler Patience`	3	Epochs with no F1 improvement before reducing learning rate.
`Scheduler Factor`	0.5	Learning rate is halved when triggered.
`Min Learning Rate`	1×10^{-7}	Prevents scheduler from reducing learning rate to zero.
`Early Stopping`	Enabled	Stops training if validation F1 does not improve for 10 consecutive epochs.
`Epochs`	50	Maximum number of training iterations over the full dataset.
`Validation Split`	30%	Dataset split used for model validation.
`Execution Device`	CUDA or CPU	GPU used if available for faster training, otherwise CPU fallback.

Table 3.7: Model configuration and training hyperparameters for RoBERTaClassifier

2.Model Architecture:

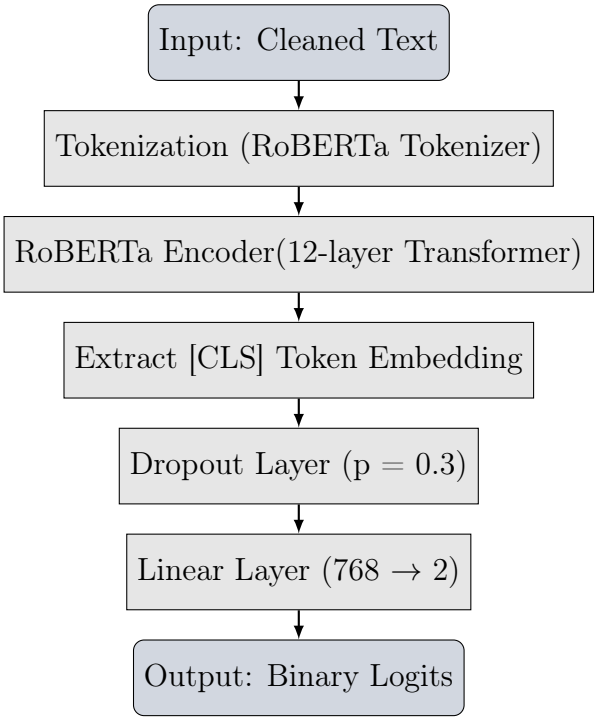

Figure 3.10: Architecture of the `RoBERTaClassifier` for binary classification

Layer	**Output Shape**	**Parameters**	**Description**
`RobertaModel (roberta-base)`	(batch_size, 512, 768)	~125M	Transformer encoder with 12 layers and multi-head self-attention, pretrained on a large corpus.
`[CLS] token extraction`	(batch_size, 768)	0	The hidden state of the first token is extracted to represent the whole sentence.
`Dropout (p=0.3)`	(batch_size, 768)	0	Regularization layer to prevent overfitting by randomly deactivating neurons.
`Linear (768 → 2)`	(batch_size, 2)	1,538	Final classification layer that outputs logits for two classes: 0 (no depression) or 1 (depression).

Table 3.8: General architecture of the `RoBERTaClassifier` model

MentalBERT

1. Model Description: In this study, we implemented a binary classification model based on the **MentalBERT** architecture. The variant employed is `mental/mental-bert-base-uncased` which builds upon the standard BERT-base architecture with 12 transformer layers and a hidden size of 768.

To enhance the model's capacity to focus on the most semantically relevant parts of an input sequence, a **multi-head self-attention layer** was integrated post-encoding. This mechanism allows the model to weigh tokens differently depending on their contribution to the overall context of mental health indicators.

The architecture of the proposed `MentalBERTClassifier` consists of the following components:

- **MentalBERT Encoder**: Utilizes the pretrained `mental-bert-base-uncased` to extract deep contextual representations from input sequences.
- **Self-Attention Layer**: A multi-head attention mechanism (8 heads) captures long-range dependencies by re-encoding the hidden states.
- **[CLS] Token Pooling**: The output corresponding to the first token (position 0) is extracted from the attention layer output to serve as a global representation of the input.
- **Dropout and Classification Layer**: A dropout regularization layer ($p = 0.3$) is applied to prevent overfitting. This is followed by a fully connected layer (Linear: $768 \rightarrow 2$) that outputs class logits for binary classification.

The model was trained using the `AdamW` optimizer with an initial learning rate of 2×10^{-5} and a weight decay of 0.01. A learning rate scheduler (`ReduceLROnPlateau`) was used to reduce the learning rate by a factor of 0.5 if the validation loss stagnated for 3 consecutive epochs.

To prevent overfitting, *early stopping* was applied with a patience of 10 epochs. The model was trained for a maximum of 50 epochs using a stratified 80/20 split on the dataset. Performance was monitored using accuracy, F1-score, and loss metrics on the validation set. The best-performing model was saved and used for final evaluation.

This configuration was implemented using a combination of modern deep learning libraries including `transformers`, `torch`, and `sklearn`, ensuring both robustness and reproducibility.

Component	Value	Description
`Pretrained model`	mental-bert-base-uncased	Domain-specific BERT variant trained on mental health corpora.
`Max sequence length`	512	Tokenized sequence length with truncation/padding.
`Batch size`	32	Mini-batch size for training and validation.
`Dropout rate`	0.3	Applied post-attention for regularization.
`Learning rate`	2×10^{-5}	Initial step size for gradient descent.
`Weight decay`	0.01	L2 regularization to penalize large weights.
`Optimizer`	AdamW	Optimizer with decoupled weight decay.
`Loss function`	CrossEntropyLoss	Suitable for binary classification tasks.
`Scheduler`	ReduceLROnPlateau	Lowers learning rate when validation loss stagnates.
`Scheduler patience`	3	Epochs without improvement before reduction.
`Early stopping`	Patience = 10	Training stops after 10 stagnant epochs.
`Number of epochs`	50	Maximum number of training epochs.
`Validation split`	20%	Proportion of data held out for validation.
`Device`	GPU/CPU	CUDA if available, otherwise CPU fallback.

Table 3.9: Hyperparameter configuration for MentalBERTClassifier (binary)

2.Model Architecture:

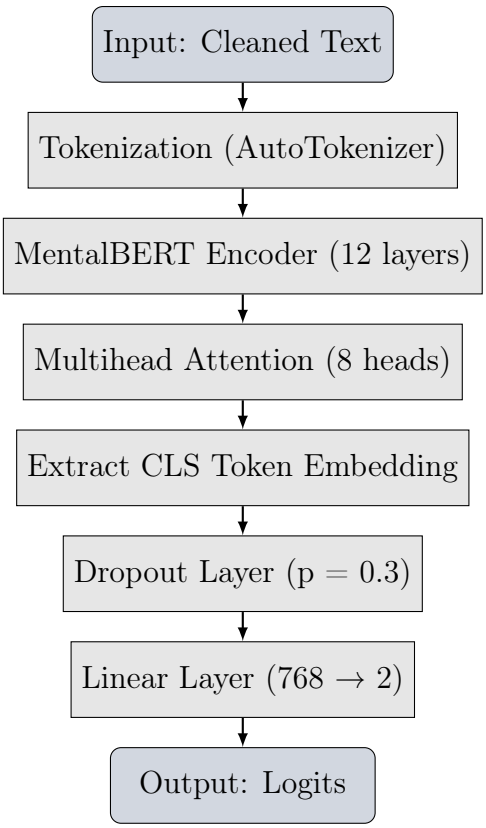

Figure 3.11: Architecture of MentalBERTClassifier for binary classification

Layer	**Output Shape**	**Description**
`Input Text`	(string)	Cleaned free-text interview input.
`Tokenizer`	(batch_size, 512)	Converts raw text into input IDs and attention masks.
`MentalBERT Encoder`	(batch_size, 512, 768)	Transformer model pretrained on mental health data.
`Multihead Attention`	(batch_size, 512, 768)	Captures interactions among token embeddings.
`[CLS] Embedding`	(batch_size, 768)	First token used as sentence-level summary.
`Dropout (p = 0.3)`	(batch_size, 768)	Randomly masks neurons for regularization.
`Linear (768 → 2)`	(batch_size, 2)	Outputs logits for two classes: no depression or depression.

Table 3.10: Model architecture of the MentalBERT binary classifier

ClinicalBERT:

1.Model description: ClinicalBERT is a domain-adapted variant of BERT, pre-trained on clinical notes from the MIMIC-III database, making it highly suitable for mental health text classification. In this binary classification setting, we employed a model based on `Bio_ClinicalBERT` and added a multi-head self-attention mechanism followed by a dropout and linear layer for binary prediction ("No Depression" vs "Depression").

- **Input:** Preprocessed clinical utterances from the `Cleaned_Participant` column.
- **Tokenizer:** ClinicalBERT tokenizer.
- **Embedding:** Last hidden state of ClinicalBERT + attention pooling.
- **Output:** Binary prediction (logits over 2 classes).

Hyperparameter	**Value**
Model	Bio_ClinicalBERT (HuggingFace)
Token Max Length	512
Batch Size	32
Learning Rate	2×10^{-5}
Weight Decay	0.01
Dropout Rate	0.3
Epochs	100
Early Stopping Patience	30
Optimizer	AdamW
Scheduler	ReduceLROnPlateau
Loss Function	CrossEntropyLoss

Table 3.11: ClinicalBERT Hyperparameters for Binary Classification

2.Model Architecture:

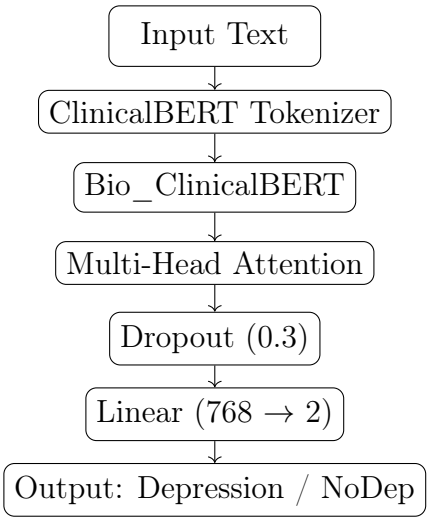

Figure 3.12: ClinicalBERT Binary Classification Architecture

Layer	Output Shape	Description
Input Text	-	Raw clinical utterance
Tokenizer	(512)	Converts text to tokens padded to max length
Bio_ClinicalBERT	(batch, 512, 768)	Contextual embeddings per token
Multi-Head Attention	(batch, 512, 768)	Focuses on relevant tokens
Pooling [CLS]	(batch, 768)	First token output used as summary
Dropout	(batch, 768)	Regularization (p=0.3)
Linear	(batch, 2)	Classification layer

Table 3.12: ClinicalBERT Model Architecture (Binary)

3.4.3 Multiclass Classification:

BERT-base:

1.Model Description : Building upon the previously introduced `BERTClassifier` architecture, we adapted the final layers of the model to support a five-class classification task. All components related to the BERT encoder, tokenization, and attention mechanisms remain unchanged, as described in the binary classification section.

The modifications for the multiclass setting are as follows:

- **Output Layer**: The classification head was changed from `Linear(768 → 2)` to `Linear(768 → 5)` to accommodate five output classes.
- **Dropout Rate**: The dropout rate was reduced from 0.5 to 0.3 to balance regularization with the increased complexity of the task.
- **Data Split**: A stratified 70% / 30% train-validation split was used to preserve class distributions across the five severity levels.
- **Early Stopping**: The early stopping patience was lowered to 5 epochs due to faster overfitting observed during experimentation.
- **Learning Rate**: Increased to 2×10^{-5} to better adapt to the complexity of multiclass training.

Item	Multiclass Value	Notes
`Dropout Rate`	0.3	Reduced compared to binary model (0.5).
`Output Layer`	Linear(768 → 5)	Adjusted for 5-class prediction.
`Number of Classes`	5	Categories from 0 (healthy) to 4 (severe).
`Learning Rate`	2×10^{-5}	Slightly higher than in binary setting.
`Early Stopping Patience`	5	Shorter due to faster convergence.
`Train/Validation Split`	70% / 30%	Stratified sampling used.

Table 3.13: Adjusted elements in the BERTClassifier architecture for multiclass classification

2.Model Architecture:

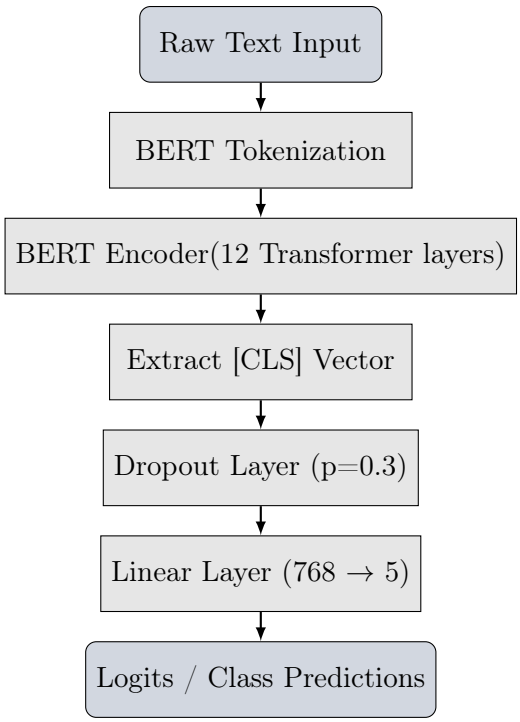

Figure 3.13: Multiclass BERTClassifier Architecture

Layer	**Output Shape**	**Parameters**	**Description**
`BertModel (bert-base-uncased)`	(batch_size, 768)	~110M	Pretrained encoder with 12 Transformer layers. Includes native multi-head self-attention mechanisms for contextual word representations.
`Dropout (p=0.3)`	(batch_size, 768)	0	Regularization layer to prevent overfitting by randomly zeroing neurons.
`Linear (768 → 5)`	(batch_size, 5)	3,845	Final classification layer producing logits for the 5 mental health classes.

Table 3.14: General Architecture of the Multiclass BERTClassifier Model

RoBERT-base:

1.Model description: Building upon the binary RoBERTaClassifier architecture previously described, we adapted the output layer to handle five classes corresponding to the severity levels of depression (0 = healthy to 4 = severe). All other architectural and training elements remain consistent with the binary setup, except for the following changes:

- The final **linear layer** was modified from `Linear(768 → 2)` to `Linear(768 → 5)`.
- A **CrossEntropyLoss** function suitable for multiclass classification was retained.
- The model was trained over a maximum of 100 epochs with `early stopping` (patience = 5) based on validation loss.
- The input **sequence length** was fixed to 128 tokens to reduce training time while preserving performance.
- A stratified train/validation split of 70%/30% was used to ensure class distribution consistency.

Parameter	Value	Remarks
`Output Layer`	Linear(768 → 5)	Adapted to multiclass output (5 mental health categories).
`Max Sequence Length`	128	Reduced from 512 to speed up training.
`Epochs`	100	Maximum training cycles.
`Early Stopping Patience`	5	Stops if validation loss stagnates for 5 epochs.
`Train/Validation Split`	70% / 30%	Stratified to maintain class balance.

Table 3.15: Modifications specific to multiclass RoBERTaClassifier training

2. Model Architecture:

Component	Modification	Comment
`Final Linear Layer`	$768 \rightarrow 5$	Outputs 5 logits instead of 2
`Input Sequence Length`	128 tokens	Compared to 512 in binary setup

Table 3.16: Architectural differences from binary to multiclass RoBERTaClassifier

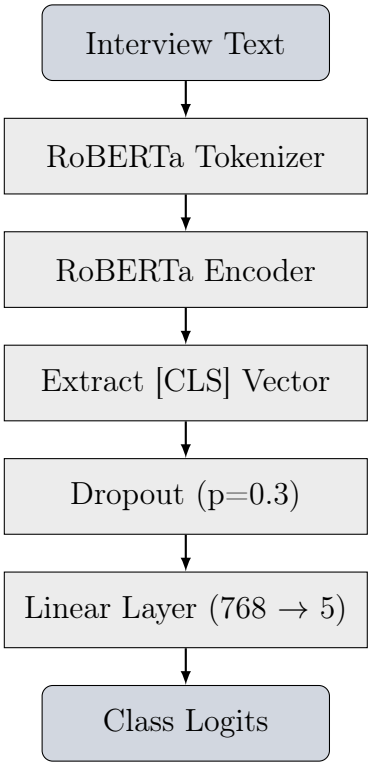

Figure 3.14: RoBERTaClassifier architecture for multiclass classification (differences highlighted)

MentalBERT:

1.Model description: Extending the binary classification setup, the `MentalBERTClassifier` model was adapted to predict five discrete categories of depression severity (0 = healthy, ..., 4 = severe). The overall architecture remains based on the `mental/mental-bert-base-uncased` model introduced in the binary section, leveraging its deep contextual understanding specific to mental health discourse.

Only the output layer was modified to project the pooled sentence representation to a five-dimensional vector. This change enables multiclass classification using a softmax activation. The dropout rate and other regularization mechanisms were preserved to ensure consistency across both tasks.

Training was conducted using the same loss function (`CrossEntropyLoss`), with stratified train-validation splitting to maintain class distribution. A ReduceLROnPlateau scheduler and early stopping (patience = 5) were integrated to optimize convergence while mitigating overfitting risks.

This multiclass formulation allows a more nuanced clinical interpretation of language features, useful in distinguishing varying levels of depressive intensity.

Layer	**Output Shape**	**Description**
`Input Text`	(raw string)	Participant utterance.
`Tokenized Input`	(batch_size, 128)	Tokenized sequence with padding and attention masks.
`MentalBERT Encoder`	(batch_size, 128, 768)	Outputs contextualized embeddings.
`Pooled Output`	(batch_size, 768)	Embedding from the [CLS] token.
`Dropout`	(batch_size, 768)	Dropout layer with rate 0.3.
`Linear Layer`	(batch_size, 5)	Dense layer mapping to 5 output classes.
`Softmax`	(batch_size, 5)	Converts logits to class probabilities.

Table 3.17: Detailed architecture of the MentalBERT multiclass model

2.Model Architecture:

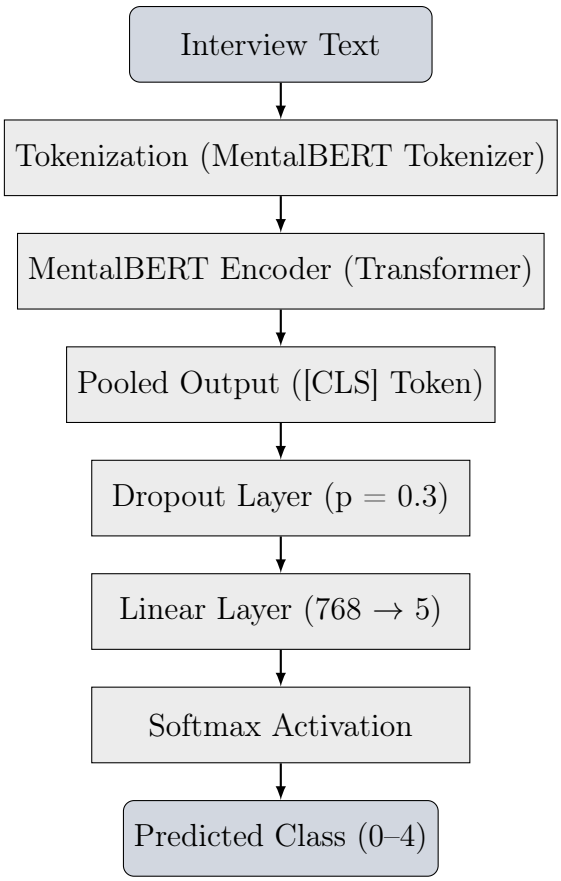

Figure 3.15: MentalBERT architecture adapted for multiclass classification

Layer	**Output Shape**	**Description**
`Input Text`	(raw string)	Participant utterance.
`Tokenized Input`	(batch_size, 128)	Tokenized sequence with padding and attention masks.
`MentalBERT Encoder`	(batch_size, 128, 768)	Outputs contextualized embeddings.
`Pooled Output`	(batch_size, 768)	Embedding from the [CLS] token.
`Dropout`	(batch_size, 768)	Dropout layer with rate 0.3.
`Linear Layer`	(batch_size, 5)	Dense layer mapping to 5 output classes.
`Softmax`	(batch_size, 5)	Converts logits to class probabilities.

Table 3.18: Detailed architecture of the MentalBERT multiclass model

ClinicalBERT:

1.Model description: Following the binary classification setting described previously, we extended the use of ClinicalBERT to address the multiclass classification task, aiming to distinguish between five mental health severity levels. The same pretrained model (`emilyalsentzer/Bio_ClinicalBERT`) and preprocessing logic were retained, but with adaptations in the classification layer and training configuration to accommodate the new output structure.

The architecture builds upon the base ClinicalBERT encoder, removing the attention layer used in the binary variant to simplify the flow and directly applying a fully connected layer with 5 output units corresponding to the 5 classes. The activation relies on the [CLS] token pooled output from ClinicalBERT. Dropout was maintained at 0.3 to limit overfitting.

Parameter	**Value**
Model	`emilyalsentzer/Bio_ClinicalBERT`
Max Sequence Length	128
Batch Size	32
Learning Rate	2×10^{-5}
Optimizer	AdamW
Loss Function	CrossEntropyLoss
Dropout	0.3
Epochs	100
Early Stopping Patience	5
Weight Decay	0.01
Scheduler	ReduceLROnPlateau

Table 3.19: Hyperparameter Configuration – ClinicalBERT Multiclass

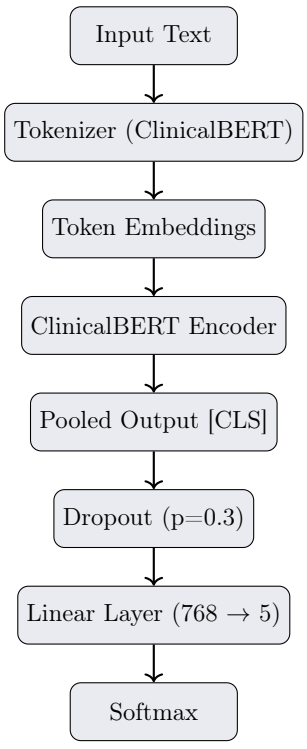

Figure 3.16: ClinicalBERT Multiclass Architecture Overview

Layer	Output Shape	Parameters	Description
ClinicalBERT (Encoder)	(batch_size, 768)	108M	Outputs [CLS] embedding
Dropout (p=0.3)	(batch_size, 768)	0	Regularization layer
Linear	(batch_size, 5)	3,845	Maps embeddings to class logits
Softmax	(batch_size, 5)	0	Converts logits to probabilities

Table 3.20: ClinicalBERT Multiclass – Model Layer Specifications

2.Model Architecture:

note : This variant preserves the encoder's contextual power while adapting the output head for multiclass prediction. Training used early stopping based on validation loss to mitigate overfitting, and performance was evaluated via accuracy, loss curves, and a detailed classification report including precision, recall, and F1-scores.

3.5 Image Classification: Methodological Approach

3.5.1 Model functionality

This section presents the modeling techniques used to classify psychological depress levels using the image modality. Unlike traditional computer vision approaches that operate on raw pixel-level data, our methodology leverages pre-extracted facial features and gaze data numerical descriptors that summarize visual behavior during clinical interviews. These features are derived from two different output files: the Gaze data and the Facial Landmark Features file. It is important to note that the gaze file is an extension of the feature file, with the latter includes the maximum number of extracted visual descriptors, making it a richer and more comprehensive source of information.
Our technique has placed more focus on the feature file throughout the study since utilising as many relevant parameters as feasible is crucial to attaining accurate mental health classification.To deal with the structural, and tabular characteristics of the retrieved visual information, a variety of modeling techniques are used. These models include transformer-based tabular models, deep learning architectures, and traditional machine learning classifiers.

3.5.1.1 LSTM + CNN

This hybrid model processes structured feature vectors from visual input by combining Convolutional Neural Networks (CNNs) with Long Short-Term Memory (LSTM) units. It Applied to both Gaze and Landmark data separately.

The CNN component concentrates on finding local spatial patterns throughout the feature space (e.g., facial landmarks or gaze vectors),while the LSTM component is kept to capture any possible sequential structure inside the structured feature dimensions themselves, even though the input data is temporally aggregated. Even with non-sequential data, this combination aims to improve the model's capacity to understand intricate inter-feature correlations.

3.5.1.2 ResNet-1D

Our data is essentially tabular and one-dimensional, comprising of extracted gaze and face coordinate characteristics (e.g., x_0, y_0, z_0), even though ResNet was first created for image analysis tasks like classification based on 2D pixel data (e.g., ResNet50). In this situation, using a conventional 2D ResNet would not be the best or most relevant choice. To address this, instead of using the usual ResNet for image data, we utilize ResNet 1D as our data consists of one-dimensional feature sequences (such as gaze and face landmarks). This model is appropriate for structured inputs because it uses 1D convolutional layers that function over the sequence of extracted features.
The architecture is composed of residual blocks, which include skip connections that add the input of a layer directly to its output. This approach enables the model to train deeper networks efficiently by addressing vanishing gradient problems and making it easier to understand complex patterns.
Additionally, Similar to spatial pooling in image-based models, 1D pooling methods are also employed to minimize dimensionality throughout the sequence while preserving important information.

3.5.1.3 FT-Transformer (Feature Tokenization Transformer)

this sophisticated deep learning architecture, presented in the in the paper "Revisiting Deep Learning Models for Tabular Data" [91], was created especially for tabular data, adapting the Transformer model, which was initially created for natural language processing, to structured datasets. FT-Transformer use self-attention to capture complex correlations between features (columns) inside each data sample, in contrast to more conventional techniques like neural networks or gradient boosting.

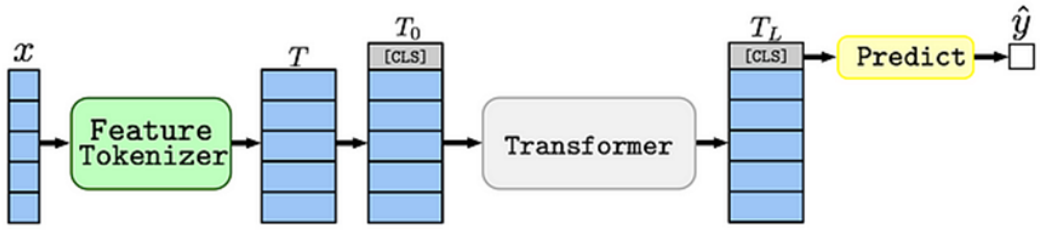

Figure 3.17: Architecture of FT Transformer [12]

Key Components of FTTransformer: [12]

- **Feature Embedding:** The model could see each feature as a token, much like words in a sentence, because both numerical and category characteristics are incorporated into vector representations.

- **Transformer Encoder Layers :** Several Transformer encoder layers make up the FTTransformer's core. The two primary parts of each layer are feed-forward neural networks and multi-head self-attention.
  By focusing on distinct parts of the input sequence, multi-head self-attention enables the model to capture feature dependencies, and feed-forward neural networks analyze the attention mechanism's output to produce feature-wise transformations.

- **Feature-wise Processing:** The FTTransformer concentrates on changing each feature separately across all rows rather than working on the sequence as a whole. This method makes it possible for the model to learn feature dependencies and interactions in a more straightforward and understandable way.
- **Output Layer:** A final output layer processes the encoded features after they have been aggregated for binary or multiclass classification tasks.

3.5.1.4 Classical Machine Learning Models

1. Support Vector Machine (SVM) is a supervised machine learning technique for classification and regression problems. It works by determining the optimal decision boundary, also known as a hyperplane, that maximizes the margin between data points of various classes. SVM has become known for its accuracy and adaptability, even with smaller datasets, and is particularly useful in high-dimensional domains.

2. Random Forest (RF) is a powerful ensemble learning approach for applications involving regression and classification. In order to increase accuracy and decrease overfitting, it constructs many decision trees during training and aggregates their results. A random subset of the data and characteristics is used to train each tree, adding variation and enhancing the overall robustness and dependability of the model.

3. Logistic Regression (LR) is a method for supervised machine learning that is used to problems with classification. It predicts the probability that an input belongs to a certain class, as opposed to linear regression, which predicts continuous values. When it comes to binary categorization, the result might fall into one of two categories, like Yes/No. It transforms inputs into a probability value between 0 and 1 using the sigmoid function [92].

4. K-Nearest Neighbors (KNN) is an easy-to-understand machine learning algorithm. It works by comparing a new data point to its "k" closest neighbors in the training set (based on distance). The projection is based on the most prevalent class among these neighbors. For example, the new participant is likely to be classed as "depressed" if the majority of others in close proximity are. KNN just memorizes the input and makes decisions during prediction; it doesn't learn from it during training.

3.5.2 Binary classification

3.5.2.1 LSTM + CNN

Item	Gaze Dataset	Features Dataset
`Input Features`	13	136
`Convolution Layers`	2 Layers (64, 128)	2 Layers (64, 128)
`Kernel Size`	3	3
`Activation Functions`	ReLU, Sigmoid	LeakyReLU ($\alpha = 0.01$), Sigmoid
`Regularization`	None	L2 ($\lambda = 0.001$)
`Pooling Layers`	MaxPooling1D (pool size=2)	MaxPooling1D (pool size=2)
`LSTM Layers`	2 Layers (64 $\rightarrow$ 32 units)	2 Layers (64 $\rightarrow$ 32 units)
`Batch Normalization`	Yes	Yes
`Dropout Rates`	0.2, 0.3	0.2, 0.3, 0.3
`Dense Layers`	Dense(64) $\rightarrow$ Dense(1)	Dense(64) $\rightarrow$ Dense(1)
`Optimizer`	Adam	Adam
`Loss Function`	Binary Crossentropy	Binary Crossentropy
`Learning Rate Scheduler`	None	ReduceLROnPlateau
`Early Stopping`	Patience = 15	Patience = 7
`Model Checkpoint`	Save best on val_loss	Save best on val_loss
`Train/Test Split`	80% / 20%	75% / 25%
`Batch Size`	20	25
`Epochs`	150	150

Table 3.21: Comparison of Training Parameters and Hyperparameters of the CNN+LSTM Models on Gaze and Feature Datasets

To classify participants into depressed and non-depressed classes using visual data, a unified deep learning architecture was designed by integrating **1D Convolutional Neural Networks (CNN)** with **Long Short-Term Memory (LSTM)** layers. This model was applied to two different input sources: **gaze data** and **facial landmark features**. While the core structure remains similar, certain adaptations were made depending on the nature and dimensionality of each dataset.

The model begins with an **input layer** that accepts either:

- 13 normalized gaze features — including 3D gaze coordinates, 3D head pose, and gender indicator, reshaped to $(samples, 13, 1)$; or
- 136 facial landmark features — consisting of 68 X-coordinates and 68 Y-coordinates of facial keypoints, reshaped to $(samples, 136, 1)$.

Next, two **Conv1D layers** are applied with 64 and 128 filters respectively and a fixed kernel size of 3. These layers extract local spatial patterns across the feature dimensions. In the gaze model, standard `ReLU` activation is used, while the facial landmark model utilizes `LeakyReLU` ($\alpha = 0.01$) to enhance learning stability in high-dimensional input.

Batch normalization is employed after each convolutional block to ensure training stability. To reduce the spatial dimension and computational load, **MaxPooling1D** is applied (pool size = 2) in both models. **Dropout** regularization is also introduced after each convolution and pooling operation—set to 0.2 for gaze data, and increased to 0.3 for the richer feature dataset to mitigate overfitting risks.

Following the convolutional blocks, two stacked **LSTM layers** (64 $\rightarrow$ 32 units) process the feature sequences. These layers are capable of modeling ordered dependencies in the input—even though the data is not strictly temporal. In the facial landmark model,

L2 regularization ($\lambda = 0.001$) is applied to both LSTM and dense layers for additional generalization control.

The final layers include a **Dense layer** with 64 units and either `ReLU` (gaze) or `LeakyReLU` (features), followed by **Batch Normalization** and additional dropout (0.3). The output layer is a single neuron with `sigmoid` activation that provides a binary classification score.

Both models are compiled using the **Adam optimizer** and **binary crossentropy loss**. Evaluation metrics include `Accuracy`, `Precision`, and `Recall`. For the facial landmark model, a `ReduceLROnPlateau` scheduler is added to lower the learning rate when the validation loss plateaus.

Training strategies for both models include:

- `EarlyStopping` to terminate training if validation performance does not improve (patience = 15 for gaze, and 7 for features).
- `ModelCheckpoint` to save only the best-performing version of the model based on validation loss.

Data was split into training and testing subsets with stratification: 80%/20% for gaze and 75%/25% for facial landmark features. Training was conducted for a maximum of 150 epochs, with batch sizes of 20 (gaze) and 25 (features), respectively.

Finally, the implementation relied on the following libraries: `TensorFlow (Keras)` for model construction, `Scikit-learn` for preprocessing and splitting, and `Seaborn` and `Matplotlib` for result visualization, including confusion matrices.

3.5.2.2 ResNet-1D

Item	Gaze Dataset	Features Dataset
Input Features	13	136
Initial Conv Layer	Conv1D(64, kernel=7)	Conv1D(64, kernel=7)
Residual Blocks	3 blocks: 64 → 128 → 256	3 blocks: 64 → 128 → 256
Activation Function	ReLU	ReLU
Regularization	L2 (Conv, Dense)	L2 (Conv, Dense)
Batch Normalization	Yes	Yes
Dropout Rates	0.5, 0.5	0.3, 0.3
Dense Layers	Dense(256) → Dense(128) → Dense(1)	Dense(256) → Dense(128) → Dense(1)
Output Activation	Sigmoid	Sigmoid
Optimizer	Adam (LR = 0.0005)	Adam (LR = 0.0005)
Loss Function	Binary Crossentropy	Binary Crossentropy
Class Weighting	Yes	Yes
Early Stopping	Patience = 10	Patience = 10
Model Checkpoint	Yes	Yes
Train/Test Split	80% / 20%	75% / 25%
Batch Size	32	64
Epochs	150	300

Table 3.22: Comparison of ResNet-1D Architectures and Training Settings for Binary Classification (Gaze vs. Features)

For binary classification, a one-dimensional ResNet-inspired architecture was applied to both gaze and facial feature datasets. This model leverages residual connections to enable deeper and more robust representation learning while addressing vanishing gradients.

Each input sample is first passed through a **Conv1D layer** with 64 filters and a kernel size of 7, followed by **Batch Normalization** and `ReLU` activation. The core structure

consists of three stacked **residual blocks**, each composed of two convolutional layers with identity shortcuts. The number of filters increases progressively across blocks: 64, 128, and 256.

Post-convolutional processing includes **Global Average Pooling** and two dense layers (256 and 128 units), each followed by **dropout regularization** to mitigate overfitting. Finally, a single neuron with a `sigmoid` activation function produces the binary output.

Both models apply `L2 regularization` on convolutional and dense layers and use the **Adam optimizer** with a reduced learning rate of 5×10^{-4}. The `binary crossentropy` loss is used in conjunction with `class weights` to address data imbalance. Training is monitored using **early stopping** and **model checkpointing** based on validation loss.

Evaluation metrics include `accuracy`, `precision`, and `recall`, along with confusion matrix visualization and classification reports.

3.5.2.3 FT-Transformer

Parameter	**Gaze Dataset**	**Features Dataset**
Number of features	13	136
Token embedding size (d_{token})	64	64
Number of Transformer blocks (n_{blocks})	3	3
Feedforward hidden size	128	128
Attention dropout	0.2	0.2
FFN dropout	0.1	0.1
Loss function	BCEWithLogitsLoss	BCEWithLogitsLoss
Optimizer	Adam	Adam
Learning rate	1×10^{-3}	1×10^{-3}
Epochs	80	250
Batch size	32	32
Library used	rtdl (PyTorch)	rtdl (PyTorch)

Table 3.23: FT-Transformer hyperparameters for gaze and features datasets (binary classification).

For both gaze and feature datasets, the FT-Transformer model was instantiated using the rtdl.FTTransformer function with identical architectural settings. The model accepts only numerical inputs and does not utilize categorical features. Each input feature is projected into a latent space of dimension (d_token = 64). The architecture includes three transformer blocks ($n\_blocks = 3$), each composed of multi-head attention (with $attention\_dropout = 0.2$) and feed-forward layers with a hidden size of ($ffn\_d\_hidden = 128$) and ($ffn\_dropout = 0.1$). A CLS token is used, and only its representation is extracted for binary prediction ($last\_layer\_query\_idx = [-1]$). The final output layer has a single unit ($d\_out = 1$) and uses a sigmoid activation. During training, the inputs are normalized using StandardScaler, and the data is split into training and testing sets using stratified sampling. The model is optimized using the Adam optimizer with a learning rate of 10^{-3}, and the loss is computed using BCEWithLogitsLoss(is a loss function that combines a sigmoid activation with binary cross-entropy loss). The gaze model is trained for 80 epochs, while the feature model is trained for 250 epochs, both with a batch size of 32. Evaluation is performed periodically using accuracy, precision, recall, F1-score, and the confusion matrix.

3.5.2.4 Classical Machine Learning Models

Parameter	Gaze Dataset	Features Dataset
Number of features	13	136
Normalization	StandardScaler	StandardScaler
Train/Test Split	80% / 20% (Stratified)	80% / 20% (Stratified)
SVM Kernel	Linear	Linear
Random Forest Estimators	100	100
Logistic Regression	max_iter = 500	max_iter = 500
KNN Neighbors	5	5

Table 3.24: Classical ML model configurations for gaze and features datasets (binary classification).

For binary classification using classical machine learning, four standard models, Support Vector Machine (SVM), Random Forest, Logistic Regression, and K-Nearest Neighbors (KNN), were applied to both gaze and features datasets. Each dataset was preprocessed by selecting relevant features, followed by normalization using StandardScaler. The data was then split into training and testing sets with an 80/20 ratio while maintaining class balance via stratification. For the gaze dataset, 13 features were used, whereas the features dataset included 136 attributes derived from landmark coordinates. All models shared common hyperparameter settings: SVM with a linear kernel, Random Forest with 100 trees, and KNN with $k = 5$ neighbors. Logistic Regression used default parameters for the gaze dataset and an increased iteration limit (max_iter=500) for the features dataset to ensure convergence. Model performance was evaluated using accuracy, precision, recall, and F1-score. These classical methods serve as a baseline to compare against more complex deep learning architectures.

3.5.3 Multiclass classification

3.5.3.1 LSTM + CNN

Item	Gaze Dataset	Features Dataset
`Input Features`	13	136
`Convolution Layers`	2 Layers (128, 256)	2 Layers (128, 256)
`Kernel Size`	3	3
`Activation Functions`	ReLU, ReLU	ReLU, LeakyReLU
`Regularization`	L2 ($\lambda = 0.001$)	L2 ($\lambda = 0.001$)
`Pooling Layers`	MaxPooling1D (pool size=2)	MaxPooling1D (pool size=2)
`LSTM Layers`	2 Layers (128 $\rightarrow$ 64)	2 Layers (128 $\rightarrow$ 64)
`Batch Normalization`	Yes	Yes
`Dropout Rates`	0.2, 0.3	0.2, 0.5, 0.5
`Dense Layers`	Dense(128) $\rightarrow$ Dense(5)	Dense(128) $\rightarrow$ Dense(5)
`Output Activation`	Softmax	Softmax
`Optimizer`	AdamW (LR = 1e-4)	Adam
`Loss Function`	Categorical Crossentropy	Categorical Crossentropy
`Class Weighting`	No	Yes
`Learning Rate Scheduler`	ReduceLROnPlateau	None
`Early Stopping`	Patience = 10	Patience = 10
`Model Checkpoint`	Yes	Yes
`Train/Test Split`	75% / 25%	75% / 25%
`Batch Size`	32	15
`Epochs`	500	500

Table 3.25: Comparison of CNN+LSTM Hyperparameters and Training Settings for Multiclass Classification (Gaze vs. Features)

For the multiclass classification task, we extended our CNN+LSTM architecture—originally designed for binary classification—by adapting the output structure and certain training strategies to predict across five mental health classes (0 to 4).

The architecture remains structurally similar for both gaze and facial feature datasets: it begins with two **Conv1D layers** (128 and 256 filters) to extract local patterns, followed by **MaxPooling1D**, **Batch Normalization**, and progressively deeper **dropout regularization** to mitigate overfitting.

Sequential temporal patterns are captured via **LSTM layers** (128 then 64 units), after which a dense layer with 128 neurons prepares the data for final classification.

For gaze data, `ReLU` is used throughout, with `AdamW` optimizer (LR = 1e-4) and a `ReduceLROnPlateau` scheduler. No class weighting was applied. In contrast, the facial features model applies `LeakyReLU` in deeper layers, uses standard `Adam` optimizer, and integrates `class weights` to address imbalanced classes.

The final output layer consists of 5 neurons activated by a **softmax** function, trained using `categorical crossentropy`. Evaluation metrics included `CategoricalAccuracy`, `Precision`, `Recall`, `AUC`, and `F1-score`. EarlyStopping and ModelCheckpoint callbacks were used in both models to prevent overfitting and retain the best model.

3.5.3.2 ResNet-1D

Item	Gaze Dataset	Features Dataset
Input Features	13	136
Initial Conv Layer	Conv1D(64, kernel=9)	Conv1D(64, kernel=9)
Residual Blocks	3 blocks: 64 → 128 → 256	3 blocks: 64 → 128 → 256
Block Activation	LeakyReLU + ReLU	ReLU
Regularization	L2 + Dropout(0.3)	Dropout(0.3)
Pooling Strategy	GlobalAveragePooling1D	GlobalAveragePooling1D
Dense Layers	Dense(256, leaky_relu) → Dense(128, leaky_relu)	Same
Output Layer	Dense(5, softmax)	Dense(5, softmax)
Loss Function	Categorical Crossentropy	Categorical Crossentropy
Optimizer	Adam	Adam
Learning Scheduler	ReduceLROnPlateau	None
Class Weights	Yes	Yes
Early Stopping	Patience = 10	Patience = 10
Batch Size	32	15
Train/Test Split	80% / 20%	70% / 30%
Epochs	300	200

Table 3.26: Comparison of ResNet-1D Architecture and Training Settings for Multiclass Classification (Gaze vs. Features)

To extend the binary classification approach, a multiclass version of the ResNet-1D architecture was developed and applied to both gaze and facial landmark datasets. The classification task involved five distinct categories, making it necessary to adapt the output layer and training settings accordingly.

Both models retain the core residual structure previously introduced. The input passes through an initial `Conv1D` layer with 64 filters and kernel size 9, followed by three **residual blocks** with increasing depth (64, 128, and 256 filters). These blocks combine standard convolutional operations with shortcut connections to ensure gradient flow and feature reuse.

After spatial feature extraction, **global average pooling** is used, followed by two dense layers (256 and 128 units), each employing the `leaky_ReLU` activation and **dropout** for regularization. The output layer comprises 5 neurons with a `softmax` activation, suitable for multiclass prediction.

In training, categorical crossentropy is used as the loss function, and class imbalance is handled via computed `class weights`. An `Adam` optimizer is applied in both models, with the gaze model benefiting additionally from `learning rate reduction on plateau`. The learning dynamics are monitored using `early stopping` and `model checkpointing` to prevent overfitting and retain optimal parameters.

Performance is evaluated using multiclass `accuracy`, `precision`, and `recall`, along with confusion matrix visualization and classification reports to assess class-wise model behavior.

3.5.3.3 FT-Transformer

Parameter	Gaze Dataset	Features Dataset
Number of classes	5	5
Number of features	13	136
Token embedding size (d_{token})	64	64
Number of Transformer blocks (n_{blocks})	3	3
Feedforward hidden size (ffn_d_hidden)	128	128
Attention dropout	0.2	0.2
FFN dropout	0.1	0.1
Residual dropout	0.0	0.0
Output layer dimension (d_{out})	5	5
Loss function	CrossEntropyLoss	CrossEntropyLoss
Optimizer	Adam	Adam
Learning rate	1×10^{-3}	1×10^{-3}
Epochs	400	100
Batch size	32	32
Early stopping patience	10	10

Table 3.27: FT-Transformer hyperparameters for gaze and features datasets (multiclass classification).

For multiclass classification, the FT-Transformer architecture was reused with the same internal structure as in binary classification but adapted to output five classes. The input features were standardized using StandardScaler and stratified into training and testing sets. The model receives numerical features without categorical encoding, and projects them into a latent space of dimension 64. It consists of three transformer blocks composed of multi-head attention with a dropout rate of 0.2, followed by feedforward layers of size 128 and dropout rate of 0.1. The residual connections are preserved with no additional dropout. Only the last token embedding (CLS token) is extracted to predict class labels via a dense layer with softmax activation. Class imbalance is addressed using weighted CrossEntropyLoss. An Adam optimizer with a learning rate of 10^{-3} is used for optimization. Early stopping is implemented to prevent overfitting, triggered after 10 epochs of non-improvement in training loss. The gaze model is trained for up to 400 epochs, while the features model stops after a maximum of 100 epochs. Model evaluation uses classification metrics and confusion matrices.

3.6 Multimodal Classification

To make our classification system more accurate, we applied multimodal learning by combining features from both text and image data. We used two main approaches to merge this information: Early Fusion and Late Fusion. Below, we explain each approach along with how it works, and the overall process.

3.6.1 Early Fusion Approach

In the early fusion strategy, we combined the features from both modalities before feeding them into the classifier. This allows the model to learn from the interactions between text and image features directly.

1.Data Preparation for Early Fusion

- Text Modality:
  - We applied data augmentation using MixUp to address class imbalance. Then, we extracted meaningful feature representations
  - using ClinicalBERT, a transformer model fine-tuned for clinical and mental health data.
- Image Modality:
  - The image features (e.g., gaze coordinates or Facial Landmarks data) were pre-extracted and further augmented using Manifold Mixup to improve generalization and balance.

2.Early Fusion Workflow

We concatenated the extracted text features from ClinicalBERT with the image features. The resulting unified feature vector was used as input to the XGBoost model for multiclass classification tasks, the early fusion showing in the figure 3.18.

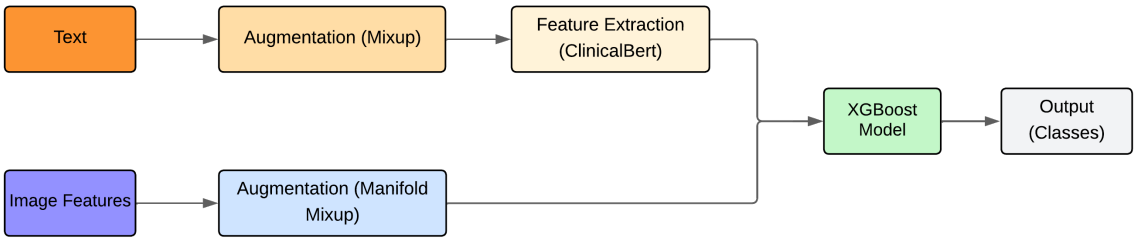

Figure 3.18: Early Fusion Workflow

3.XGBoost Architecture Description

XGBoost (Extreme Gradient Boosting) is a scalable and efficient gradient boosting algorithm widely adopted for structured data tasks. In the context of our multimodal approach, it was chosen for its ability to handle heterogeneous features—specifically, textual embeddings (from SBERT) and image-derived "features" while maintaining robust performance and interpretability. Unlike deep learning architectures composed of layers of neurons, XGBoost is an ensemble of decision trees built sequentially. Each tree attempts to correct the errors of the previous one using gradient descent on a loss function. In multiclass settings, XGBoost learns to approximate the class probabilities for each category using the softmax function.

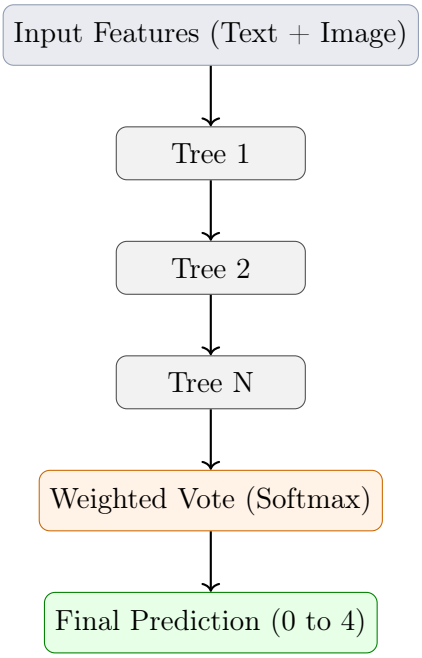

Figure 3.19: Schematic representation of XGBoost-based multimodal classification architecture

Hyperparameters Used

Hyperparameter	Value	Description
`objective`	`multi:softprob`	Enables multiclass probability prediction using softmax.
`num_class`	5	Number of classes (0: healthy, ..., 4: severe).
`eval_metric`	`mlogloss, merror`	Monitors both log loss and classification error.
`eta`	0.1	Learning rate that scales contribution of each new tree.
`max_depth`	6	Maximum tree depth for base learners.
`subsample`	0.8	Fraction of samples used for each tree to prevent overfitting.
`colsample_bytree`	0.8	Fraction of features used in each boosting iteration.
`seed`	42	Random seed for reproducibility.
`num_boost_round`	100	Maximum number of boosting rounds.

Table 3.28: XGBoost Hyperparameter Configuration

Training Strategy

The model was trained using stratified 80-20 train/validation split to preserve class balance. Class distributions were also equilibrated via MixUp in earlier stages. Evaluation was based on macro-F1 and accuracy, ensuring robustness across all five categories. Early stopping was applied if validation metrics plateaued to avoid overfitting.

3.6.2 Late Fusion Approach

The late fusion approach involves processing each modality independently through separate models and then combining their predictions to make a final decision. This approach is useful when the modalities have very different feature spaces and structures.

1.Models Used in Late Fusion

- Text Modality:
  - ClinicalBERT: for clinical language feature extraction and classification.
  - RoBERTa: for complementary representation of textual input.
- Image Modality:
  - FT-Transformer: a transformer model adapted for tabular data (i.e., gaze and facial landmarks features).

2.Fusion Technique: Majority Voting

We adopted majority voting to aggregate predictions from the three models. Majority Voting is a simple way for combining predictions from several models. Each model makes its own prediction, and the final result is the class that most models agree on. This decreases the possibility of one model making an incorrect decision, thereby increasing overall accuracy and system reliability.

3.Late Fusion Workflow

Each model (ClinicalBERT, RoBERTa, and FT-Transformer) makes an independent prediction. These predictions are then passed through a voting mechanism, and the class with the majority of votes is selected as the final prediction, the late fusion showing in the figure 3.20.

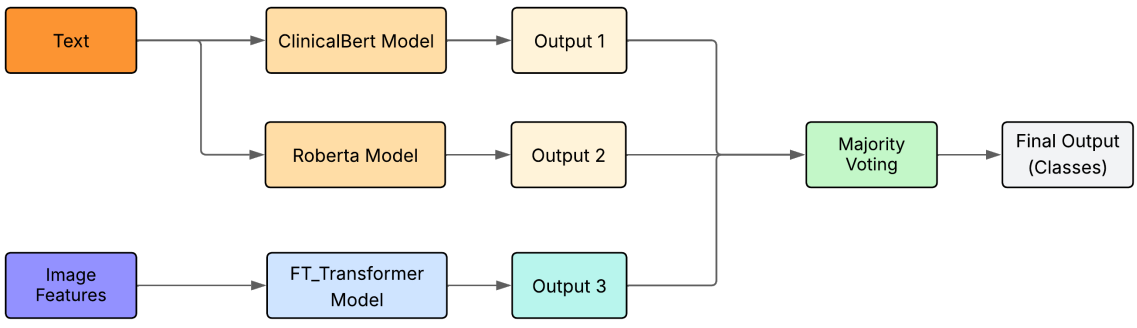

Figure 3.20: Late Fusion Workflow

The following points summarize the dataset configuration, voting mechanism and system setup:

- A merged dataset named dataset_augmented_equilibrated was used as the general base for the late fusion strategy. This dataset contains both textual data (from participants' transcripts) and facial landmarks features. It was first divided into training and testing sets using an 80/20 stratified split. The test set was then split

into two separate modalities: a text subset and a feature subset. Each of these subsets was used to re-evaluate the previously trained models ClinicalBERT and RoBERTa for the textual part, and FT-Transformer for the image based features. The final predictions from these models were aggregated using majority voting to produce the ensemble decision.

- The fusion process was implemented using a majority voting mechanism. For each sample, predictions from the three models were collected, and the most frequent class label among them was selected as the final decision.

3.7 Generative Model

3.7.1 Dataset Used

To fine-tune our generative model, we used a dataset titled **"NLP - Mental Health Conversations"** This dataset was retrieved from Kaggle [93]. This dataset includes conversation examples where users express emotional or psychological distress (e.g., anxiety, depression, loneliness) and receive empathetic responses.

Each record in the dataset is structured as a pair: `(prompt, response)`, where:

- **Prompt**: Represents the user's input message, typically expressing a concern or emotional challenge.
- **Response**: Corresponds to a written reply, often supportive, empathetic, and human-centered.

After cleaning and formatting, the dataset contained several thousand usable pairs to train an emotionally intelligent chatbot.

3.7.2 Flan-T5 Architecture and Fine-Tuning Process

We chose the **Flan-T5-Base** model, a sequence-to-sequence (Seq2Seq) language model developed by Google and available through Hugging Face. This model is particularly well-suited for conditional text generation tasks.

Our training pipeline involved the following steps:

1. **Data preprocessing**: Tokenizing both the prompts and responses with padding and truncation (max 512 tokens).
2. **Model loading**: Initializing the pre-trained Flan-T5-Base model.
3. **Fine-tuning**: Training the model for 100 epochs with a batch size of 32.
4. **Model saving**: The final fine-tuned model was saved locally for deployment via a Flask API.
5. **Response generation**: Manual testing with example prompts to evaluate the coherence and empathy of the generated text.

The main objective was not quantitative performance, but rather to produce clinically appropriate and emotionally relevant responses in natural language.

Summary Table

Component	Description
Dataset	NLP Mental Health Conversations (Kaggle)
Data format	Pairs: (Prompt, Response)
Model	`google/flan-t5-base`
Task	Empathetic text generation (chatbot)
Max sequence length	512 tokens
Number of epochs	100
Batch size	32
Goal	Generate relevant, empathetic responses to mental health-related user inputs
Model output files	`pytorch_model.bin`, `tokenizer.json`, `config.json`

Table 3.29: Overview of the Flan-T5 fine-tuning process

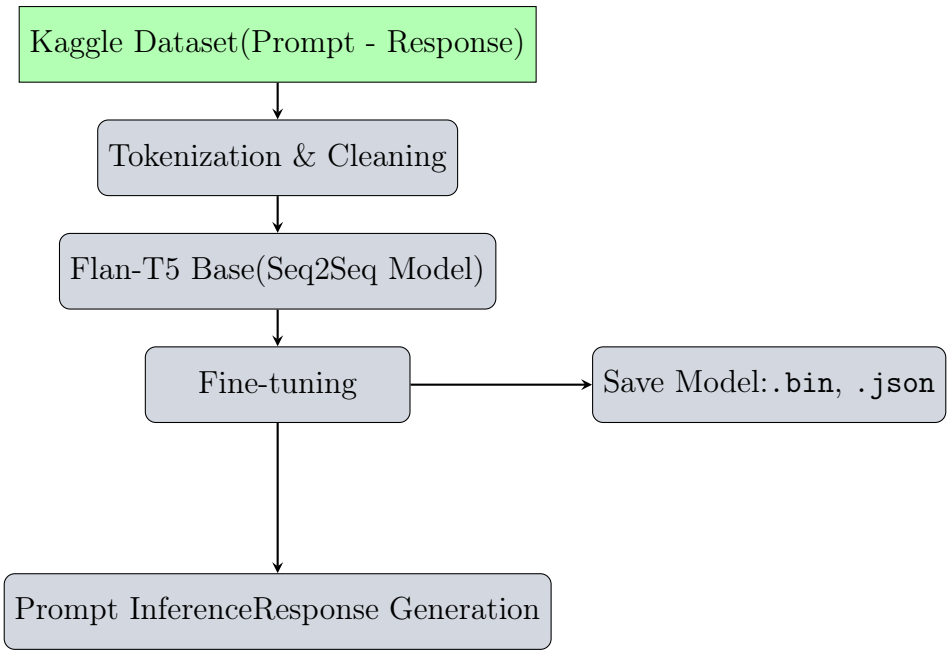

Figure 3.21: Architecture of the Flan-T5 fine-tuning pipeline using a mental health conversation dataset.

Chapter 4

Experimentation and discussion

4.1 Introduction

This section defines the experimental procedure executed to evaluate compare several machine learning and deep learning models across diverse data modalities (image, text, and multimodal). the ultimate objective of these research is the creating a chatbot interface capable of interacting with users and detecting potential mental health issues. After testing and comparing all the models, we selected the most effective one and combined it with the FLAN-T5 generative language model. This final step allows us to create a chatbot that can not only detect symptoms of depression but also engage in fluent and meaningful conversations with patient.

4.2 Text Classification Results

4.2.1 Binary Classification

BERT-base:

The **BERTClassifier** achieved an overall accuracy of **87%** on the validation set. Class "0" (non-depressed participants) was predicted with 90% precision and 86% recall, while class "1" (depressed) achieved 88% recall and 84% precision. The **macro-average F1-score** is well-balanced (0.87), indicating the model's solid ability to distinguish both classes, even with slightly uneven support. The confusion matrix shows that errors are moderately and symmetrically distributed, suggesting a robust and generalizable model.

Class	Precision	Recall	F1-score	Support
0 (Healthy)	0.90	0.86	0.88	93
1 (Depressed)	0.84	0.88	0.86	77
Overall Accuracy	0.87 (170 samples)			
Macro Average	0.87	0.87	0.87	170
Weighted Average	0.87	0.87	0.87	170

Table 4.1: Classification report of the BERT model (binary classification)

RoBERT-base:

The RoBERTa-based classifier achieved a global accuracy of **93%** on the validation set. The model shows balanced performance across both classes, with precision and recall

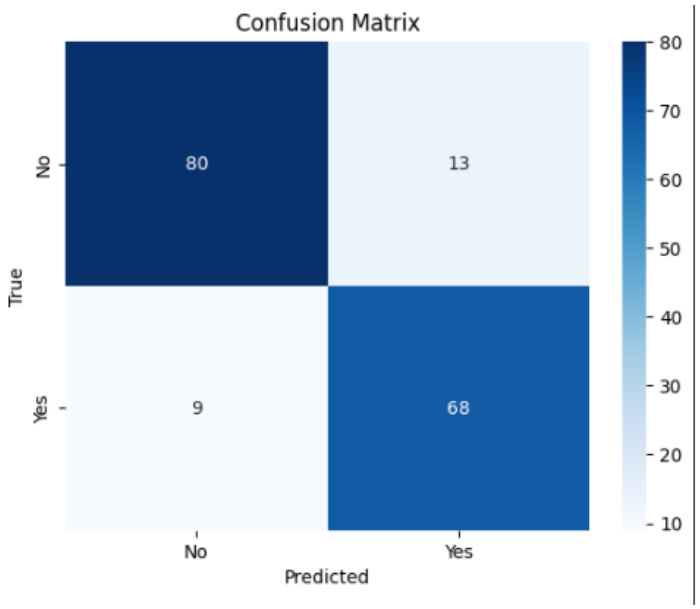

Figure 4.1: Confusion matrix of the BERT model for binary classification

values above 0.93. This suggests that the model can reliably distinguish between depressed and non-depressed individuals. The confusion matrix indicates a well-balanced prediction behavior, with very few misclassified instances, confirming the robustness of the architecture for binary mental health classification.

Class	Precision	Recall	F1-score	Support
0 (No Depression)	0.94	0.93	0.94	132
1 (Depression)	0.93	0.93	0.93	122
Accuracy	0.93 (254 samples)			
Macro Average	0.93	0.93	0.93	254
Weighted Average	0.93	0.93	0.93	254

Table 4.2: Classification report of the RoBERTa model (binary classification)

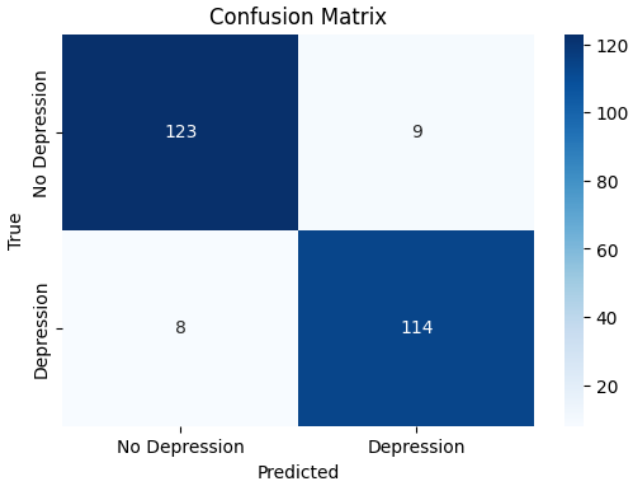

Figure 4.2: Confusion matrix of the RoBERTa model on the validation set

MentalBERT:

The confusion matrix presented in Figure 4.3 reveals an exceptionally high level of performance by the MentalBERT model with attention. Both classes, *No Depression* and *Depression*, were predicted with a precision and recall of 0.98, indicating a highly balanced classification. The accuracy of 98% confirms the model's reliability in detecting depressive symptoms from textual data. This strong performance demonstrates that the attention-enhanced contextual encoding significantly improves the understanding of mental health discourse, even in subtle cases.

Class	Precision	Recall	F1-score	Support
No Depression	0.98	0.98	0.98	88
Depression	0.98	0.98	0.98	82
Accuracy		0.98		
Macro Average	0.98	0.98	0.98	170
Weighted Avg	0.98	0.98	0.98	170

Table 4.3: Classification report for the MentalBERT + Attention model

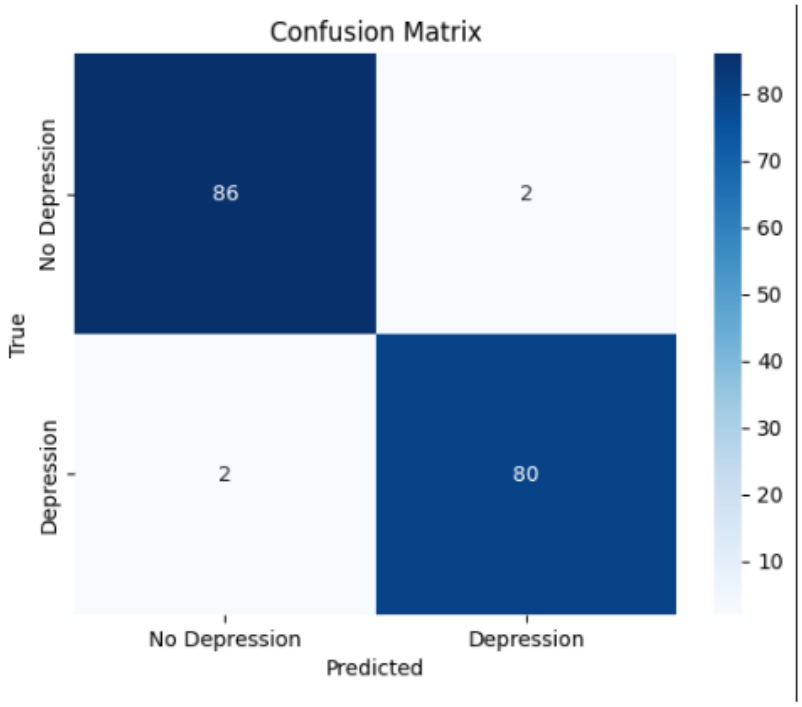

Figure 4.3: Confusion matrix of the MentalBERT model with attention

ClinicalBERT:

The ClinicalBERT model demonstrated strong performance in distinguishing between depressed and non-depressed individuals, achieving an overall accuracy of 89%. The precision for the "No Depression" class (0.91) was slightly higher than for the "Depression" class (0.80), indicating a small imbalance in prediction confidence. Similarly, recall was higher for the majority class (0.94 vs. 0.75), suggesting the model is more sensitive to detecting the absence of depression. Nonetheless, the weighted F1-score of 0.89 confirms a balanced trade-off between precision and recall, making ClinicalBERT a robust and reliable model for binary mental health classification.

Class	Precision	Recall	F1-score	Support
0 - No Depression	0.91	0.94	0.93	126
1 - Depression	0.80	0.75	0.78	44
Overall Accuracy	0.89			
Macro Average	0.86	0.84	0.85	170
Weighted Average	0.89	0.89	0.89	170

Table 4.4: Classification results of the ClinicalBERT model (binary task)

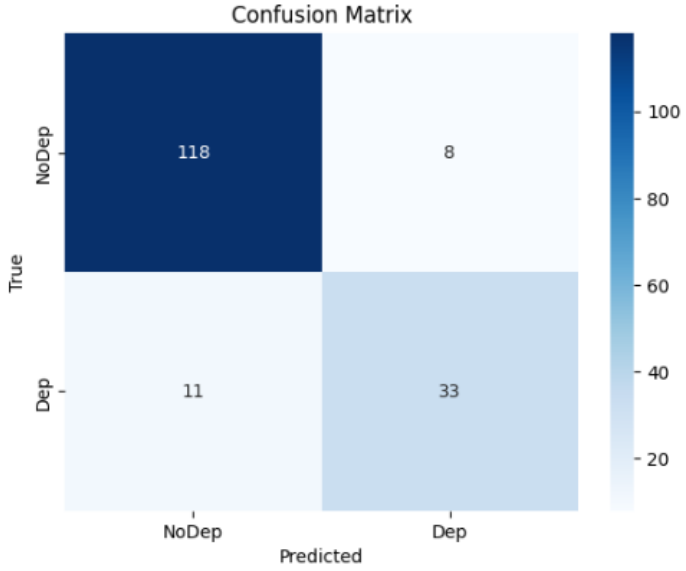

Figure 4.4: Confusion matrix of the ClinicalBERT model (binary classification)

4.2.1.1 Comparative Study

Model	Accuracy	Precision	Recall	F1-score
BERT (base)	0.93	0.93	0.93	0.93
RoBERTa (base)	0.93	0.93	0.93	0.93
MentalBERT	0.98	0.98	0.98	0.98
ClinicalBERT	0.89	0.86	0.84	0.85

Table 4.5: Comparison of binary classification performance across models

The comparative evaluation of transformer-based models for binary mental health classification reveals notable differences in performance depending on the model architecture and pretraining corpus. Both BERT and RoBERTa, as general-purpose models, delivered strong and consistent results with an F1-score of 0.93, confirming their reliability as robust baselines. In contrast, MentalBERT—pretrained on domain-specific mental health data—significantly outperformed the baselines, achieving a near-perfect F1-score of 0.98. Interestingly, adding a multi-head attention mechanism to MentalBERT did not further improve its performance, indicating that the base model is already highly effective in capturing relevant patterns for this task. ClinicalBERT, despite its specialization in clinical text, underperformed with an F1-score of 0.85. This suggests a potential mismatch between the nature of its pretraining data (structured clinical notes) and the conversational, patient-centered data used in this study. Overall, the findings highlight the

substantial benefits of domain adaptation and suggest that models pretrained on contextually relevant corpora, such as MentalBERT, are best suited for tasks involving nuanced understanding of mental health expressions.

4.2.2 Multiclass Classification

BERT-base:

The BERTClassifier model achieved an impressive global accuracy of **95.97%** on the validation set across five severity levels of mental health. The performance was particularly strong for the more severe categories (Moderately Severe and Severe), with perfect scores (1.00) in precision, recall, and F1-score, indicating no misclassifications.

The macro-averaged F1-score of **0.9597** shows that the model performs consistently well across all classes, regardless of their support size. The weighted average confirms the same trend, showing robustness even with slight class imbalances.

The confusion matrix (Figure 4.5) visually confirms that the predictions are highly accurate, with very few off-diagonal errors. This suggests that the model generalizes well and captures nuanced distinctions between severity levels—particularly valuable in the context of mental health diagnosis and support.

Class	Precision	Recall	F1-score	Support
0 - Healthy	0.9231	0.9600	0.9412	50
1 - Mild	0.9000	0.9184	0.9091	49
2 - Moderate	0.9787	0.9200	0.9485	50
3 - Mod. Severe	1.0000	1.0000	1.0000	50
4 - Severe	1.0000	1.0000	1.0000	49
Accuracy	0.9597 (248 samples)			
Macro Average	0.9604	0.9597	0.9597	248
Weighted Average	0.9604	0.9597	0.9598	248

Table 4.6: Classification Report for the BERT Multiclass Model

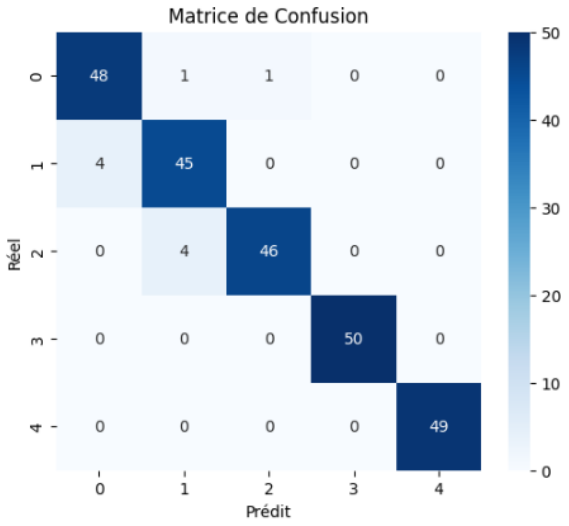

Figure 4.5: Confusion matrix for the BERT multiclass classification model

RoBERT-base:

The RoBERTa-based model achieved strong performance across all five classes, with an overall accuracy of **94.5%**. The model demonstrated excellent precision and recall scores for severe depressive categories, especially for `Moderate` and `Severe` cases (F1-scores: 0.96 and 0.99, respectively), which are critical in clinical settings.

The confusion matrix (Figure 4.6) reveals that the model correctly classifies the majority of samples, with only a few misclassifications. Notably, the `Healthy` class had slightly lower recall (0.85), indicating that some non-depressed samples were misclassified as mildly or moderately affected — a common trade-off in mental health prediction tasks where sensitivity is prioritized.

The macro-averaged and weighted-averaged metrics confirm the model's balanced behavior across all classes, highlighting its robustness and generalization capability despite class balance constraints.

This level of performance supports the model's integration into a broader mental health assessment system, offering both accuracy and clinical relevance.

Class	Precision	Recall	F1-score	Support
0 - Healthy	1.0000	0.8485	0.9180	33
1 - Mild	0.9118	0.9394	0.9254	33
2 - Moderate	0.9167	1.0000	0.9565	33
3 - Mod. Severe	0.9394	0.9394	0.9394	33
4 - Severe	0.9706	1.0000	0.9851	33
Accuracy	0.9455			
Macro avg	0.9477	0.9455	0.9449	165
Weighted avg	0.9477	0.9455	0.9449	165

Table 4.7: Classification report of the RoBERTa-based multiclass model

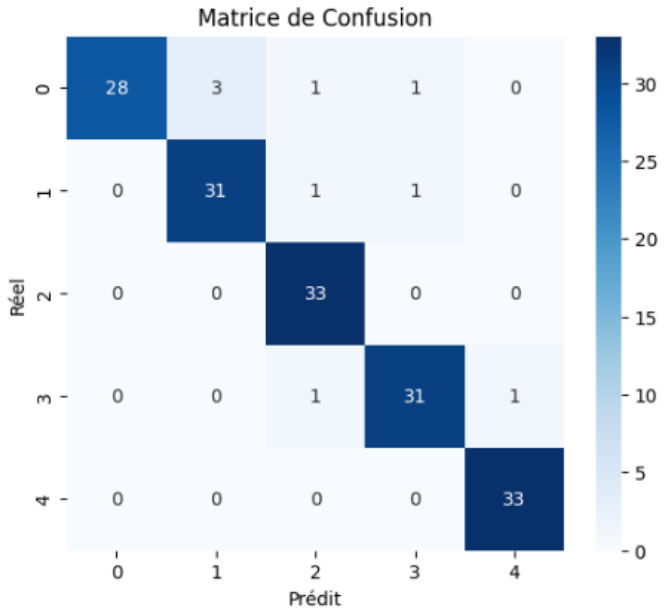

Figure 4.6: Confusion matrix of the multiclass RoBERTa model

MentalBERT:

The confusion matrix and classification report indicate that the model performs very reliably across all five severity levels of depression. The highest recall (1.00) was achieved for classes *Moderate* and *Severe*, while the *Healthy* class had slightly lower recall (0.88), suggesting some mild cases might have been misclassified as non-pathological. Overall, the model demonstrates robust generalization and balanced precision-recall tradeoffs, confirming its suitability for nuanced multiclass mental health classification.

Class	Precision	Recall	F1-Score
0 - Healthy	1.00	0.88	0.94
1 - Mild	0.97	0.94	0.95
2 - Moderate	0.89	1.00	0.94
3 - Mod. Severe	0.97	0.97	0.97
4 - Severe	0.97	1.00	0.99
Accuracy		0.96	
Macro Avg	0.96	0.96	0.96
Weighted Avg	0.96	0.96	0.96

Table 4.8: Classification Report for MentalBERT on Multiclass Task

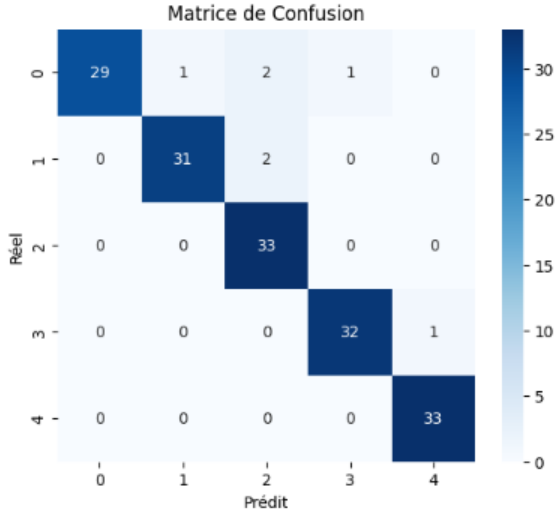

Figure 4.7: Confusion Matrix for Multiclass Classification with MentalBERT

ClinicalBERT:

The evaluation results highlight ClinicalBERT's robust performance across all five classes. The model achieves perfect or near-perfect recall for moderate and severe categories, reflecting strong sensitivity to more critical depression levels. Notably, it maintains high precision even for mildly affected individuals. The confusion matrix reveals minimal misclassification, and the macro-averaged scores confirm balanced performance, making ClinicalBERT a reliable option for fine-grained mental health assessment.

Class	Precision	Recall	F1-Score
0 - Healthy	0.91	0.94	0.93
1 - Mild	1.00	0.91	0.95
2 - Moderate	0.97	1.00	0.99
3 - Mod. Severe	0.97	0.97	0.97
4 - Severe	0.97	1.00	0.99
Accuracy		0.96	
Macro Avg	0.96	0.96	0.96
Weighted Avg	0.96	0.96	0.96

Table 4.9: Classification Report for ClinicalBERT on Multiclass Task

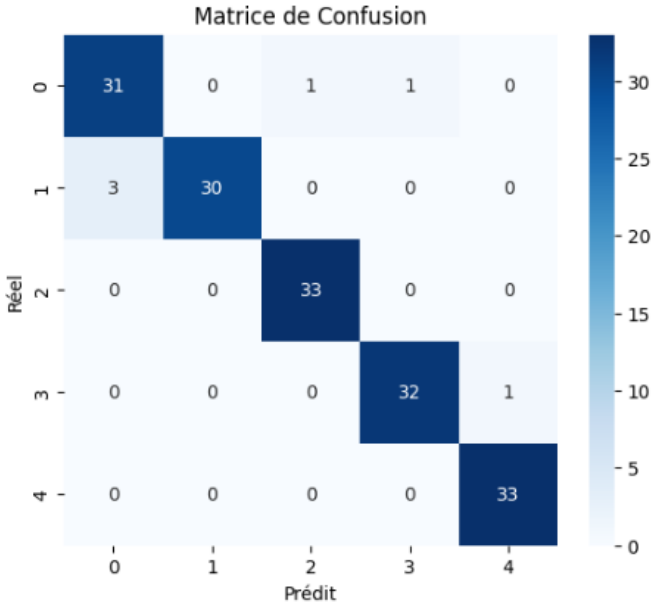

Figure 4.8: Confusion Matrix for Multiclass Classification with ClinicalBERT

4.2.2.1 Comparative study

Model	Accuracy	Precision (Macro)	Recall (Macro)	F1-Score (Macro)
BERT	0.9576	0.9602	0.9576	0.9573
RoBERTa	0.9636	0.9645	0.9636	0.9635
MentalBERT	0.9576	0.9602	0.9576	0.9573
ClinicalBERT	0.9636	0.9645	0.9636	0.9635

Table 4.10: Multiclass Classification Performance Summary

The comparative evaluation of the four transformer-based models—BERT, RoBERTa, MentalBERT, and ClinicalBERT—reveals subtle but meaningful differences in their ability to handle multiclass mental health classification. Both BERT and MentalBERT exhibit solid and nearly identical performance, achieving an accuracy of 95.76% with macro F1-scores around 0.957. These results reflect the effectiveness of general and domain-adapted language models in capturing mental health-related linguistic nuances. However, RoBERTa and ClinicalBERT outperform their counterparts slightly, each reaching an accuracy of 96.36% and macro F1-scores close to 0.964. RoBERTa's success likely

stems from its training on larger corpora and more advanced pretraining strategies, while ClinicalBERT benefits from medical-domain pretraining, which seems especially suited to mental health content. In summary, although all models demonstrate high-level performance, RoBERTa and ClinicalBERT stand out as the most reliable choices for fine-grained severity detection in mental health applications.

4.3 Image-Based Classification

This section presents the experimental results of image-based classification models applied to both the gaze and features datasets. The models discussed here have already been introduced in the methodology chapter and are evaluated for both binary and multiclass classification tasks. The goal is to assess the performance of each model using various metrics, including accuracy, precision, recall, and F1-score, and to compare their effectiveness across different configurations and dataset types.

4.3.1 Binary Classification

4.3.1.1 LSTM + CNN

Dataset	Class	Precision	Recall	F1-score	Support
Gaze	Healthy (0)	0.88	0.87	0.87	76
	Depressed (1)	0.84	0.85	0.85	61
	Accuracy	**0.86 (137 samples)**			
Features	Healthy (0)	0.70	0.76	0.73	98
	Depressed (1)	0.63	0.56	0.59	73
	Accuracy	**0.67 (171 samples)**			

Table 4.11: Binary Classification Results of CNN+LSTM on Gaze and Features Datasets

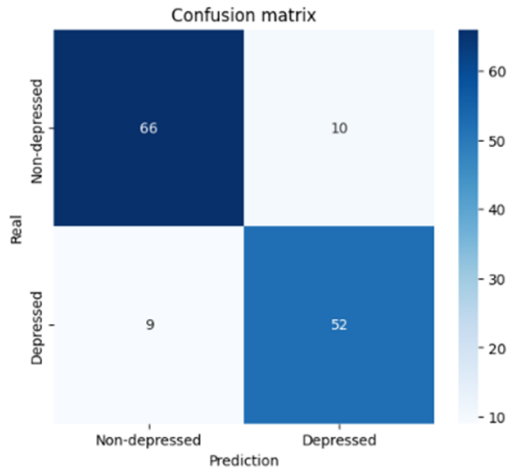

Figure 4.9: Confusion Matrix Binary CNN+LSTM (Gaze)

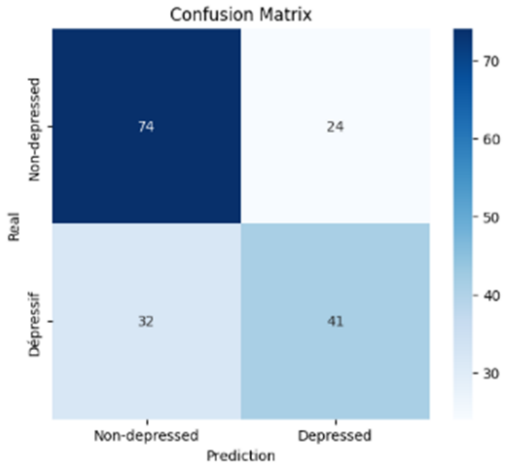

Figure 4.10: Confusion Matrix Binary CNN+LSTM (Features)

Table 4.11 summarizes the classification performance of the CNN+LSTM model applied to both the gaze and features datasets. The corresponding confusion matrices provide further insight into class-specific prediction behaviors.

- **Overall Accuracy:**
  - The model achieved a significantly higher accuracy on the **gaze dataset** (86%) compared to the **features dataset** (67%), indicating that gaze dataset are more informative for distinguishing between healthy and depressed individuals in binary classification.

- **Gaze Dataset Analysis:**
  - **Healthy (Class 0):** Precision = 0.88, Recall = 0.87, F1-score = 0.87
  - **Depressed (Class 1):** Precision = 0.84, Recall = 0.85, F1-score = 0.85
  - The model performs robustly on both classes with balanced metrics, indicating reliable detection for both healthy and depressed participants.
  - **Confusion Matrix:**
    - Among the 76 individuals labeled as healthy, the model correctly identified 66 of them, while only 10 were mistakenly classified as depressed.
    - For the 61 individuals who were actually depressed, 52 were correctly detected, and 9 were incorrectly predicted as healthy.
    - These results show a well-balanced classification, with a fairly even distribution of errors across both classes, indicating that the model does not favor one class over the other.
- **Features Dataset Analysis:**
  - **Healthy (Class 0):** Precision = 0.70, Recall = 0.76, F1-score = 0.73
  - **Depressed (Class 1):** Precision = 0.63, Recall = 0.56, F1-score = 0.59
  - The model performs moderately well on healthy participants but shows notable weakness in identifying depressed individuals.
  - **Confusion Matrix:**
    - Out of the 98 participants who were actually healthy, the model correctly identified 74, but mistakenly classified 24 as depressed.
    - Among the 73 depressed individuals, only 41 were correctly detected, while 32 were incorrectly labeled as healthy.
    - This imbalance highlights a tendency of the model to overlook depressed cases, which is particularly concerning in clinical applications where failing to detect depression can have serious consequences.
- **Comparative Insights:**
  - Overall, the model performs better on the gaze dataset, achieving higher precision and recall for both healthy and depressed classes.
  - Predictions based on gaze features are more balanced and accurate, with fewer misclassifications compared to the static facial features.
  - In contrast, the results on the features dataset show a high number of missed depressed cases (false negatives), which could be problematic in a mental health context where early and accurate detection is essential.
  - These findings indicate that gaze data, which captures dynamic eye movement and head motion, provides more meaningful patterns for distinguishing between healthy and depressed individuals than facial landmarks alone.

4.3.1.2 ResNet 1D

The ResNet 1D model yielded strong results on both datasets, but a comparative analysis reveals notable differences:

Dataset	Class	Precision	Recall	F1-score	Support
Gaze	Healthy (0)	0.87	0.96	0.91	76
	Depressed (1)	0.94	0.82	0.88	61
	Accuracy	**0.90 (137 samples)**			
Features	Healthy (0)	0.90	0.78	0.84	98
	Depressed (1)	0.75	0.89	0.81	73
	Accuracy	**0.82 (171 samples)**			

Table 4.12: Binary Classification Results of ResNet 1D on Gaze and Features Datasets

- **Overall Accuracy:** The ResNet 1D model performed better on gaze data, achieving an accuracy of **90%**, compared to **82%** on the features dataset. This reinforces the idea that gaze offer richer and more discriminative information for distinguishing between healthy and depressed individuals than facial characteristics.
- **Healthy Class (0):**
  - On the **gaze dataset**, the model showed excellent recall (**0.96**), indicating its strong ability to correctly identify healthy individuals, with only a few false positives.
  - In the **features dataset**, while the precision was slightly higher at **0.90**, recall dropped to **0.78**, meaning more healthy individuals were misclassified as depressed.
- **Depressed Class (1):**
  - On the **gaze dataset**, the model showed high precision (**0.94**) but a slightly lower recall (**0.82**), indicating that while most predicted depressed cases were correct, a few actual depressed individuals were missed.
  - In contrast, the **features dataset** showed a higher recall (**0.89**), capturing more actual depressed cases, but at the expense of lower precision (**0.75**), meaning more false positives were introduced.

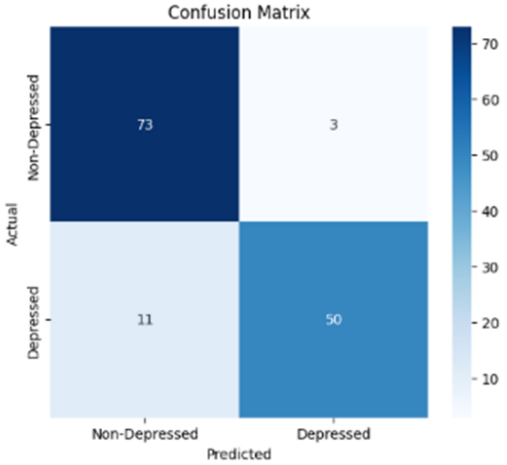

Figure 4.11: Confusion Matrix Binary ResNet 1D (Gaze)

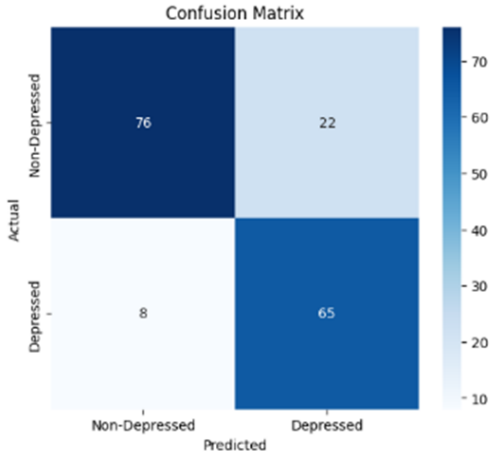

Figure 4.12: Confusion Matrix Binary ResNet 1D (Features)

- **Confusion Matrix:**
  - On the **gaze data**, the model correctly classified most individuals, with only a small number of misclassifications, only 3 healthy individuals misclassified and 11 depressed missed.
  - On the **features data**, misclassifications were more common — especially in predicting healthy individuals (22 false positives) and some false negatives for depressed cases (8 missed).
- Overall, while both models perform well, the ResNet 1D applied to gaze data clearly provides more accurate and balanced predictions.

4.3.1.3 FT Transformer

Metric	Gaze Dataset	Features Dataset
Accuracy	0.8686	0.8394
Precision	0.8413	0.8600
Recall	0.8689	0.7414
F1-score	0.8548	0.7963

Table 4.13: Performance Comparison of FT-Transformer on Gaze and Features Datasets (Binary Classification)

Gaze Dataset	Predicted	
Actual	Healthy (0)	Depressed (1)
Healthy (0)	66	10
Depressed (1)	8	53

Features Dataset	Predicted	
Actual	Healthy (0)	Depressed (1)
Healthy (0)	72	7
Depressed (1)	15	43

Table 4.14: Confusion Matrices of FT-Transformer for Gaze and Features Datasets

The FT-Transformer model shows solid and reliable performance on both the gaze and features datasets for binary depression classification.

- On the **gaze dataset**, the model achieved an impressive accuracy of **87%**. Its high recall of **0.87** means it successfully identified most of the depressed and non-depressed individuals. The precision and F1-score were also well-balanced, confirming that the model made few mistakes and maintained consistent predictions.
- On the **features dataset**, the model also delivered strong results with an accuracy of **84%**. While its precision was slightly higher (**0.86**), indicating fewer false positives, the lower recall of **0.74** shows that it missed a number of actual depressed cases, suggesting a more conservative prediction behavior.

- **The confusion matrices**: For the gaze data, it correctly identified 66 out of 76 healthy individuals and 53 out of 61 depressed individuals, resulting in only 10 false positives and 8 false negatives. In the features dataset, it accurately predicted 72 out of 79 healthy cases and 43 out of 58 depressed ones, with 7 false positives and 15 false negatives. These results confirm the model's strong overall performance, especially with gaze data, which led to fewer misclassifications in both classes and thus offered a better balance for detecting depression.

4.3.1.4 Machine Learning Models (SVM , RF , LR , KNN)

Model	Metric	Gaze Dataset	Features Dataset
SVM	Accuracy	0.50	0.62
	Precision	0.37	0.54
	Recall	0.16	0.24
	F1-score	0.23	0.33
Random Forest	Accuracy	0.89	0.86
	Precision	0.87	0.78
	Recall	0.89	0.88
	F1-score	0.88	0.83
Logistic Regression	Accuracy	0.53	0.61
	Precision	0.46	0.50
	Recall	0.26	0.30
	F1-score	0.33	0.37
K-Nearest Neighbors	Accuracy	0.63	0.69
	Precision	0.60	0.59
	Recall	0.51	0.67
	F1-score	0.55	0.63

Table 4.15: Vertical Comparison of Classical ML Models on Gaze and Features Datasets (Binary Classification)

The classical machine learning models produced notably different results across the two datasets:

- **Random Forest (RF)** clearly stood out as the best-performing model for both datasets. It achieved an excellent accuracy of **89%** on gaze data and **86%** on features, along with strong precision, recall, and F1-scores. This consistency highlights its robustness and ability to generalize well across modalities.
- **Support Vector Machine (SVM)** struggled significantly, especially on the gaze dataset where it achieved only **50%** accuracy and very low recall (**0.16**), suggesting poor generalization and underfitting. Although it performed slightly better on features (**62%** accuracy), its recall remained low (**0.24**), limiting its reliability for detecting depressed individuals.
- **Logistic Regression (LR)** showed weak results in both cases, with accuracy hovering around **53–61%**, and low recall scores, indicating its limited capacity to capture complex decision boundaries inherent in both gaze and facial features.

- **K-Nearest Neighbors (KNN)** provided moderate performance. On gaze, its accuracy was **63%**, while on features it improved to **69%**, with acceptable F1-scores in both. However, its performance was still well below that of Random Forest.

Final Comparative Results

Model	Dataset	Accuracy	Precision	Recall	F1-score
CNN + LSTM	Gaze	0.86	0.86	0.86	0.86
CNN + LSTM	Features	0.67	0.67	0.66	0.66
ResNet-1D	Gaze	0.90	0.91	0.89	0.89
ResNet-1D	Features	0.82	0.83	0.83	0.82
FT-Transformer	Gaze	0.87	0.84	0.87	0.85
FT-Transformer	Features	0.84	0.86	0.74	0.80
Random Forest	Gaze	0.89	0.87	0.89	0.88
Random Forest	Features	0.86	0.78	0.88	0.83
SVM	Gaze	0.50	0.37	0.16	0.23
SVM	Features	0.62	0.54	0.24	0.33
Logistic Regression	Gaze	0.53	0.46	0.26	0.33
Logistic Regression	Features	0.61	0.50	0.30	0.37
KNN	Gaze	0.63	0.60	0.51	0.55
KNN	Features	0.69	0.59	0.67	0.63

Table 4.16: Final Comparative Results of All Models on Gaze and Features Datasets (Binary Classification)

Among all models, the **ResNet-1D applied to gaze data** achieved the highest performance in binary classification, with an accuracy of **90%** and balanced precision, recall, and F1-score around **0.89–0.91**, making it the most effective model for binary classification of depression using gaze data.

4.3.2 Multiclass Classification

4.3.2.1 CNN + LSTM

Class	Precision	Recall	F1-score	Support
0 (Healthy)	0.61	0.55	0.57	42
1 (Mild)	0.56	0.50	0.53	38
2 (Moderate)	0.68	0.68	0.68	37
3 (Moderately Severe)	0.68	0.70	0.69	33
4 (Severe)	0.71	0.95	0.82	21
Accuracy	0.64 (171 samples)			
Macro Avg	0.65	0.67	0.66	171
Weighted Avg	0.64	0.64	0.64	171

Class	Precision	Recall	F1-score	Support
0 (Healthy)	0.40	0.27	0.32	44
1 (Mild)	0.34	0.60	0.44	48
2 (Moderate)	0.41	0.32	0.36	37
3 (Moderately Severe)	0.60	0.32	0.41	38
4 (Severe)	0.84	0.93	0.89	29
Accuracy	0.47 (196 samples)			
Macro Avg	0.52	0.49	0.48	196
Weighted Avg	0.49	0.47	0.46	196

Table 4.17: Classification Reports of CNN+LSTM Model on Gaze and Features Datasets (Multiclass Classification)

Gaze Dataset	0	1	2	3	4
0 (Healthy)	23	7	7	5	0
1 (Mild)	9	19	4	4	2
2 (Moderate)	1	7	25	2	2
3 (Mod. Severe)	4	1	1	23	4
4 (Severe)	1	0	0	0	20

Features Dataset	0	1	2	3	4
0 (Healthy)	12	23	7	2	0
1 (Mild)	12	29	4	2	1
2 (Moderate)	6	14	12	4	1
3 (Mod. Severe)	0	18	5	12	3
4 (Severe)	0	1	1	0	27

Table 4.18: Confusion Matrices of CNN+LSTM Model on Gaze and Features Datasets (Multiclass Classification)

Interpretation: The CNN+LSTM model was evaluated on both the **Gaze** and **Features** datasets for multiclass classification, aiming to distinguish between five levels of depression severity. The results show clear differences in performance between the two modalities, as detailed below.

- **Overall Accuracy:** The model performed significantly better on the **Gaze** dataset, achieving an accuracy of **64%**, compared to only **47%** on the **Features** dataset. This indicates that gaze patterns provided more reliable parameters for distinguishing between classes.

- **Macro and Weighted Averages:**

  - On the Gaze dataset, macro and weighted averages hovered around **0.64–0.66**, reflecting a fairly balanced performance across classes.
  - On the Features dataset, these values dropped to around **0.46–0.49**, showing the model struggled to generalize across the full range of depression categories using only facial features.

- **Class-wise Analysis:**

  - **Severe cases (Class 4)** were well-identified in both datasets, with especially high F1-scores: **0.82 (Gaze)** and **0.89 (Features)**. The model consistently detects critical cases, which is important in clinical applications.
  - **Moderate to Moderately Severe cases (Classes 2 and 3)** saw strong F1-scores on Gaze (both around **0.68–0.69**), while on Features, performance dropped to **0.36** and **0.41**, respectively. This suggests that eye movements better capture nuanced depressive behaviors than facial landmarks.
  - **Mild and Healthy classes (Classes 0 and 1)** were the most confused in both datasets. In Gaze, F1-scores were **0.57** (Healthy) and **0.53** (Mild), while in Features, they fell to **0.32** and **0.44**. The overlap in symptoms for these early stages of depression likely contributes to the confusion.

- **Confusion Matrix Observations:**

  - In the Gaze dataset, misclassifications were mostly concentrated between neighboring classes. For example, some healthy individuals were misclassified as mild or moderate, and vice versa, which aligns with real-world clinical overlap.
  - In the Features dataset, the confusion was more widespread. For instance, many Class 0 (Healthy) participants were predicted as Class 1 (Mild) or Class 2 (Moderate), and the same pattern held across other classes. This indicates less discriminative power in features alone.

4.3.2.2 ResNet 1D

Class	Precision	Recall	F1-score	Support
0 (Healthy)	0.63	0.50	0.56	34
1 (Mild)	0.47	0.71	0.56	31
2 (Moderate)	0.50	0.62	0.55	29
3 (Moderately Severe)	0.50	0.19	0.28	26
4 (Severe)	0.88	0.88	0.88	17
Accuracy	0.56 (137 samples)			
Macro Avg	0.60	0.58	0.57	137
Weighted Avg	0.57	0.56	0.55	137

Class	Precision	Recall	F1-score	Support
0 (Healthy)	0.58	0.42	0.48	53
1 (Mild)	0.68	0.36	0.47	58
2 (Moderate)	0.37	0.59	0.45	44
3 (Moderately Severe)	0.52	0.62	0.57	45
4 (Severe)	0.83	0.97	0.89	35
Accuracy	0.56 (235 samples)			
Macro Avg	0.59	0.59	0.57	235
Weighted Avg	0.59	0.56	0.55	235

Table 4.19: Classification Reports of ResNet 1D Model on Gaze and Features Datasets (Multiclass Classification)

Gaze Dataset	0	1	2	3	4
0 (Healthy)	17	11	5	1	0
1 (Mild)	2	22	5	2	0
2 (Moderate)	2	8	18	1	0
3 (Mod. Severe)	6	5	8	5	2
4 (Severe)	0	1	0	1	15

Features Dataset	0	1	2	3	4
0 (Healthy)	22	6	14	9	2
1 (Mild)	9	21	22	5	1
2 (Moderate)	2	4	26	12	0
3 (Mod. Severe)	4	0	9	28	4
4 (Severe)	1	0	0	0	27

Table 4.20: Confusion Matrices of ResNet 1D Model on Gaze and Features Datasets (Multiclass Classification)

Interpretation:

- **Overall Accuracy:** Both datasets achieved an identical accuracy of **56%**, showing that the model maintained consistent general performance across modalities.
- **Gaze Dataset Analysis:**

- The model performed best on **Class 4 (Severe)**, with high precision and recall (both **0.88**), indicating good sensitivity to severe cases.
- **Class 1 (Mild)** also showed relatively strong recall (**0.71**), though with moderate precision (**0.47**), suggesting some over-prediction.
- In contrast, the model struggled with **Class 3 (Moderately Severe)**, where recall dropped to **0.19**, leading to a low F1-score of **0.28**. This indicates difficulty in distinguishing this intermediate class from others.

- **Features Dataset Analysis:**
  - As with gaze, the model showed its best performance on **Class 4 (Severe)**, with a precision of **0.83** and an excellent recall of **0.97**, highlighting robustness in detecting the most critical condition.
  - **Class 3 (Moderately Severe)** achieved a better F1-score (**0.57**) compared to the Gaze dataset, indicating slightly better recognition of mid-range severity in this modality.
  - However, **Class 0 (Healthy)** and **Class 1 (Mild)** were often confused with other categories, as reflected by their lower F1-scores of **0.48** and **0.47**, respectively.

- **Confusion Matrices Insight:**
  - In the Gaze dataset, several moderate and mildly depressed participants were misclassified into adjacent classes, especially for Class 3 (Moderately Severe), where predictions were spread across almost all categories.
  - In the Features dataset, the confusion was particularly notable for Classes 0 and 1. For instance, many healthy individuals (Class 0) were predicted as moderate or moderately severe, indicating blurred boundaries in facial feature patterns for early depression stages.

4.3.2.3 FT Tranformer

Class	Precision	Recall	F1-score	Support
0 (Healthy)	0.77	0.61	0.68	34
1 (Mild)	0.58	0.74	0.65	31
2 (Moderate)	0.71	0.79	0.75	29
3 (Moderately Severe)	0.83	0.57	0.68	26
4 (Severe)	0.76	0.94	0.84	17
Accuracy	0.71 (137 samples)			
Macro Avg	0.73	0.73	0.72	137
Weighted Avg	0.73	0.71	0.71	137

Class	Precision	Recall	F1-score	Support
0 (Healthy)	0.73	0.54	0.62	35
1 (Mild)	0.68	0.66	0.67	39
2 (Moderate)	0.57	0.66	0.61	30
3 (Moderately Severe)	0.75	0.70	0.72	30
4 (Severe)	0.76	1.0	0.86	23
Accuracy	0.69 (157 samples)			
Macro Avg	0.70	0.71	0.70	157
Weighted Avg	0.69	0.69	0.68	157

Table 4.21: Classification Reports of FT Transformer Model on Gaze and Features Datasets (Multiclass Classification)

Gaze Dataset	0	1	2	3	4
0 (Healthy)	21	9	4	0	0
1 (Mild)	3	23	4	1	0
2 (Moderate)	1	3	23	2	0
3 (Mod. Severe)	2	3	1	15	5
4 (Severe)	0	1	0	0	16

Features Dataset	0	1	2	3	4
0 (Healthy)	19	8	5	2	1
1 (Mild)	2	26	8	3	0
2 (Moderate)	4	2	20	2	2
3 (Mod. Severe)	1	2	2	21	4
4 (Severe)	0	0	0	0	23

Table 4.22: Confusion Matrices of FT Transformer Model on Gaze and Features Datasets (Multiclass Classification)

Interpretation:

- **Overall Accuracy:** The model reached **71%** accuracy on the Gaze dataset and **69%** on the Features dataset, indicating stable and reliable performance across both modalities.

- **Gaze Dataset Observations:**
  - The model achieved its best results on **Class 4 (Severe)**, with a high F1-score of **0.84** and nearly perfect recall (**0.94**), showing strong sensitivity to critical depression cases.
  - **Classes 1 to 3** showed balanced precision and recall, especially **Class 2 (Moderate)** with an F1-score of **0.75**.
  - **Class 0 (Healthy)** had decent precision (**0.77**) but lower recall (**0.61**), meaning some healthy participants were misclassified as mildly or moderately depressed.
- **Features Dataset Observations:**
  - Again, **Class 4 (Severe)** was the best identified class, with perfect recall (**1.00**) and strong F1-score of **0.86**.
  - **Class 3 (Moderately Severe)** also performed well with F1-score of **0.72**, showing improved handling of complex intermediate cases.
  - **Classes 0 to 2** showed lower recall than precision, particularly in **Class 0 (Healthy)**, where recall dropped to **0.54**, implying a tendency to confuse healthy subjects with mild or moderate depression.
- **Confusion Matrix Insight:**
  - Most of the errors made by the model involved confusion between neighboring classes — for example, mistaking mild cases for moderate ones. This is understandable, as the symptoms of depression often evolve gradually, making it harder to draw strict boundaries.
  - Importantly, the model rarely confused very different classes, such as healthy individuals with those who are severely depressed. This suggests that the model is reliable for distinguishing between clearly distinct conditions.

Final Comparative Results

Model	Dataset	Accuracy	Precision	Recall	F1-score
CNN + LSTM	Gaze	0.64	0.65	0.67	0.66
CNN + LSTM	Features	0.47	0.52	0.49	0.48
ResNet 1D	Gaze	0.56	0.60	0.58	0.57
ResNet 1D	Features	0.56	0.59	0.59	0.57
FT Transformer	Gaze	**0.71**	**0.73**	**0.73**	**0.72**
FT Transformer	Features	**0.69**	**0.70**	**0.71**	**0.70**

Table 4.23: Overall Performance Comparison of Multiclass Classification Models on Gaze and Features Datasets

Interpretation:

- Among all models, the **FT Transformer** achieved the best performance on both datasets. It reached an accuracy of **71%** on Gaze data and **69%** on Features data, along with the highest precision, recall, and F1-score.

- **CNN+LSTM** performed moderately well on the Gaze dataset (**64%** accuracy) but showed a clear drop in performance on the Features dataset (**47%** accuracy), indicating that it may struggle to capture complex feature patterns from tabular inputs.
- **ResNet 1D** showed consistent but relatively moderate results across both datasets with **56%** accuracy. It provided balanced performance but was less effective than FT Transformer, especially in capturing subtler class distinctions.
- Overall, the FT Transformer proved to be the most suitable model for multiclass depression classification in both modalities, demonstrating stronger generalization and more reliable class separation.

4.4 Multimodal-Based Classification

4.4.1 Early fusion

The XGBoost model demonstrated strong performance across all five depression severity classes, achieving an overall accuracy of 95%. Perfect precision and recall scores were observed for classes 2 (Moderate), 3 (Moderate Severe), and 4 (Severe), while class 0 (Healthy) exhibited slightly lower precision. The macro-averaged F1-score of 0.95 confirms balanced classification, and the confusion matrix shows minimal misclassification. These results suggest that tree-based boosting methods can provide competitive performance, even when compared to large transformer models, particularly when the dataset is well-structured and preprocessed.

Class	**Precision**	**Recall**	**F1-score**
0 - Healthy	0.81	1.00	0.89
1 - Mild	1.00	0.75	0.86
2 - Moderate	1.00	1.00	1.00
3 - Mod. Severe	1.00	1.00	1.00
4 - Severe	1.00	1.00	1.00
Accuracy		0.95	
Macro Average	0.96	0.95	0.95
Weighted Avg	0.96	0.95	0.95

Table 4.24: Classification Report for XGBoost on Multiclass Depression Detection

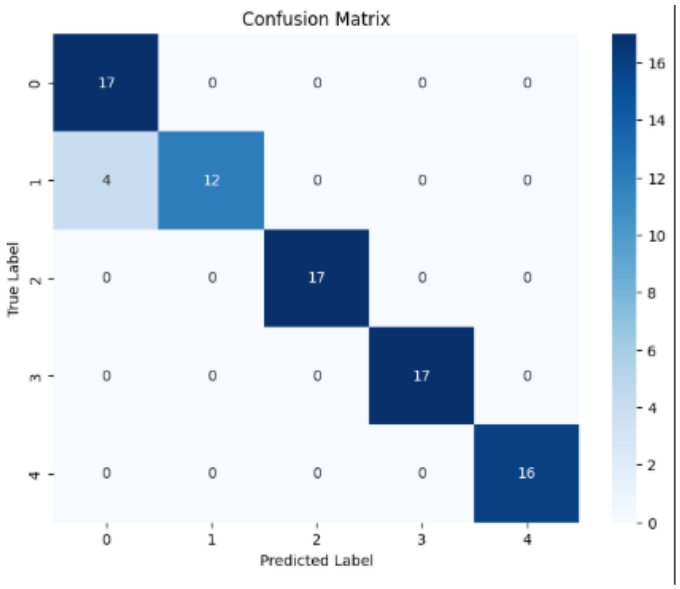

Figure 4.13: Confusion Matrix for XGBoost on Multiclass Depression Classification

4.4.2 Late fusion

In this experiment, we applied a majority voting strategy to combine predictions from three independently trained models: **ClinicalBERT** and **RoBERTa** for the text modality, and **FT-Transformer** for image-based features. Unlike previous binary tasks, this fusion focused on **multiclass classification**, aiming to distinguish between five severity levels of depression, ranging from 0 (healthy) to 4 (severe). The fusion mechanism selects the most frequently predicted class among the three models.
The table below summarizes the overall performance of the ensemble model:

Metric	Value
Accuracy	0.8313
Precision	0.8407
Recall	0.8313
F1-score	0.8325

Table 4.25: Overall Performance of the Late Fusion Model (Multiclass Classification)

A detailed breakdown of the performance per class is given below:

Class	Precision	Recall	F1-score	Support
0 (Healthy)	0.60	0.71	0.65	17
1 (Mild)	0.85	0.69	0.76	16
2 (Moderate)	0.93	0.82	0.87	17
3 (Moderately Severe)	0.89	0.94	0.91	17
4 (Severe)	0.94	1.00	0.97	16
Macro Average	0.84	0.83	0.83	83
Weighted Avg.	0.84	0.83	0.83	83

Table 4.26: Classification Report of the Late Fusion Model (Multiclass)

The confusion matrix below illustrates the prediction distribution across the five classes:

Actual \Predicted	**0**	**1**	**2**	**3**	**4**
0 (Healthy)	12	2	0	2	1
1 (Mild)	4	11	1	0	0
2 (Moderate)	3	0	14	0	0
3 (Mod. Severe)	1	0	0	16	0
4 (Severe)	0	0	0	0	16

Table 4.27: Confusion Matrix – Late Fusion (Multiclass)

Interpretation: The late fusion approach demonstrated solid overall performance, with an accuracy of **83%** and consistent scores across all key metrics (precision, recall, and F1-score around 0.83). This confirms that merging information from both text and image modalities leads to reliable classification of depression severity levels.

Looking closer at the per-class results, the model performed exceptionally well on the most severe cases. **Class 4 (Severe)** was identified with perfect accuracy (**100% precision and recall**), indicating that the model is highly dependable for detecting the most critical individuals. Similarly, it handled **moderate to moderately severe levels (Classes 2 and 3)** with high confidence, achieving F1-scores above **0.87**, which suggests robustness in identifying nuanced stages of depression.

Some difficulties appeared in identifying the healthy individuals (Class 0), as a few were mistakenly classified as mildly or moderately depressed. This led to a lower precision of 0.60, which could be explained by overlapping symptoms between non-depressed and slightly affected individuals. However, this minor issue doesn't overshadow the model's overall balance and reliability.

Overall, the late fusion strategy effectively combined insights from text and image data, resulting in a robust and trustworthy model for accurately predicting different levels of depression severity.

4.5 Evaluation Metrics

- **Accuracy:** calculates the percentage of cases with accurate predictions out of all the predictions. It provides a general sense of the model's accuracy rate. It has the following mathematical definition:

$$Recall = \frac{\text{Correct classifications}}{\text{Total classifications}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}} \tag{4.1}$$

- **Precision:** shows the percentage of all cases projected as positive that were accurately predicted as positive. It indicates how well the model predicts a positive class.

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.2}$$

- **Recall:** (also known as sensitivity) calculates the percentage of real positive cases that the model accurately detected. It demonstrates how well the model can identify affirmative situations.

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.3}$$

- **F1-Score:** The F1 score is the harmonic mean of precision and recall, giving a single score between 0 (worst) and 1, given by:

$$F1 \text{ score} = 2\frac{(\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}} \tag{4.4}$$

- **Confusion Matrix:** is a table that displays the number of predictions that were true positive, false positive, true negative, and false negative. It facilitates the visualization of the model's performance in many areas.

- **Loss function:** measures the variation between the actual labels during training and predicted values. The learning and optimization process is guided by a smaller loss, which shows that the model's predictions are close to the real results.

4.6 ChatBot Realisation

4.6.1 Frontend Development

To provide users with an intuitive and responsive interface, the frontend of the chatbot was developed using modern web technologies:

- **React** was used as the core JavaScript framework for building dynamic and component-based UI.

- **Vite** was chosen as the development environment and bundler for its fast Hot Module Replacement (HMR) and optimized builds.

- **TypeScript** improved code readability and robustness by introducing strong typing.

- **Tailwind CSS** enabled rapid styling and responsive design.

- **shadcn/ui**, a modern UI component library, was integrated for ready-to-use accessible components.

Home Page Interface

The homepage of the platform offers users a clear and intuitive interface to access mental health support services. It allows users to:

- Start a new conversation or log in to an existing account.
- Create a free and secure account quickly.
- Choose their preferred language using the language selector icon.
- Benefit from a specialized conversational assistant designed for mental disorder detection.
- Engage in confidential and secure conversations, ensuring privacy and anonymity.
- Receive compassionate and empathetic support aimed at improving their emotional well-being.

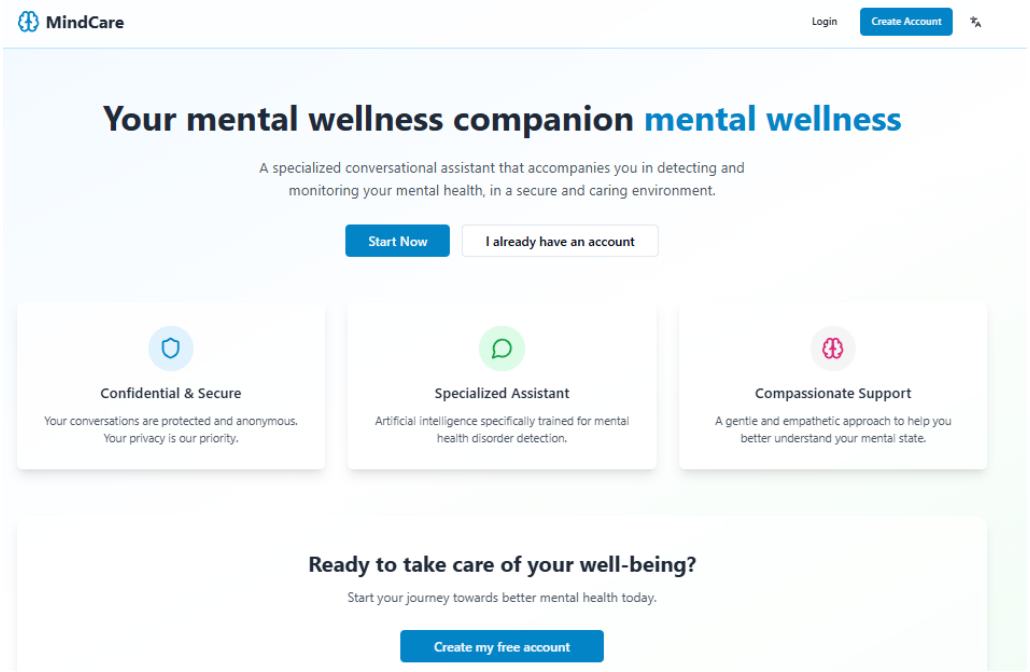

Figure 4.14: Home Page Interface

User Authentication Interfaces

The MindCare platform provides a user-friendly and secure authentication system composed of two primary interfaces:

- **Login Interface:**
  - Allows users to access their personal space with their email and password.
  - Includes a "Forgot password?" link for password recovery.
  - Displays a legal note informing users that by logging in, they accept the terms of use and privacy policy.
- **Registration Interface:**

- Allows new users to create an account by entering their full name, email, and a secure password.
- Requires users to confirm their password to avoid typos.
- Includes a mandatory checkbox to accept the terms of use and privacy policy before account creation.
- Provides a link to redirect users who already have an account to the login page.

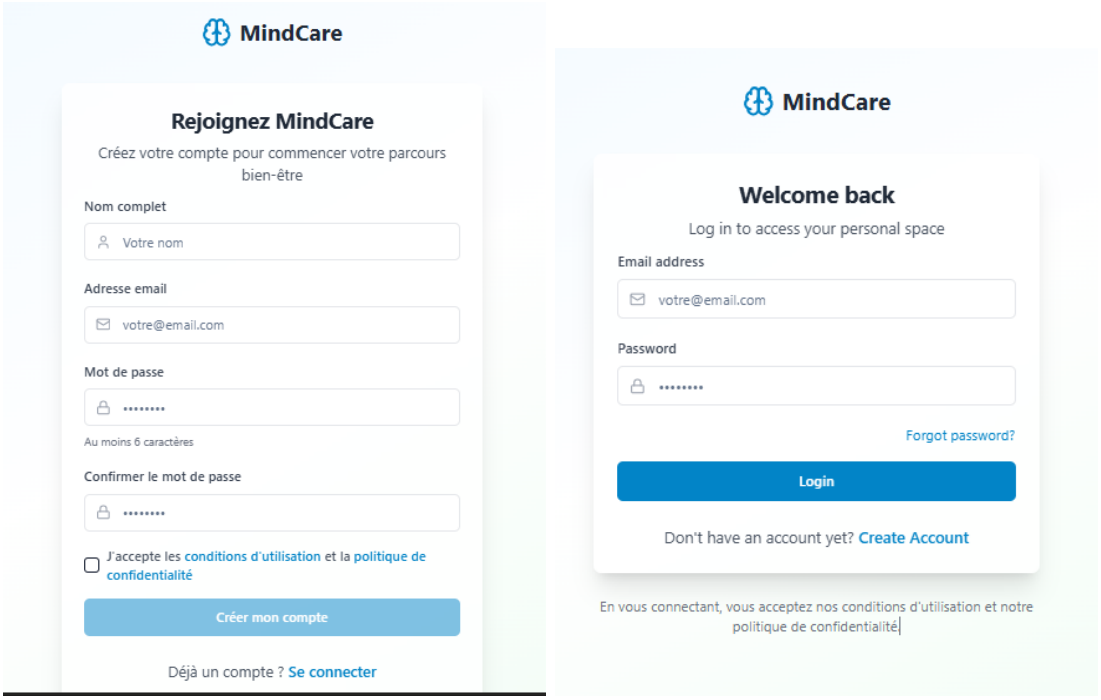

Figure 4.15: User Authentication Interfaces

Home screen of the chatbot interface

The user interface allows users to:

- Start new conversations and view recent messages.
- Receive a daily wellness tip.
- Chat securely and confidentially with the assistant.

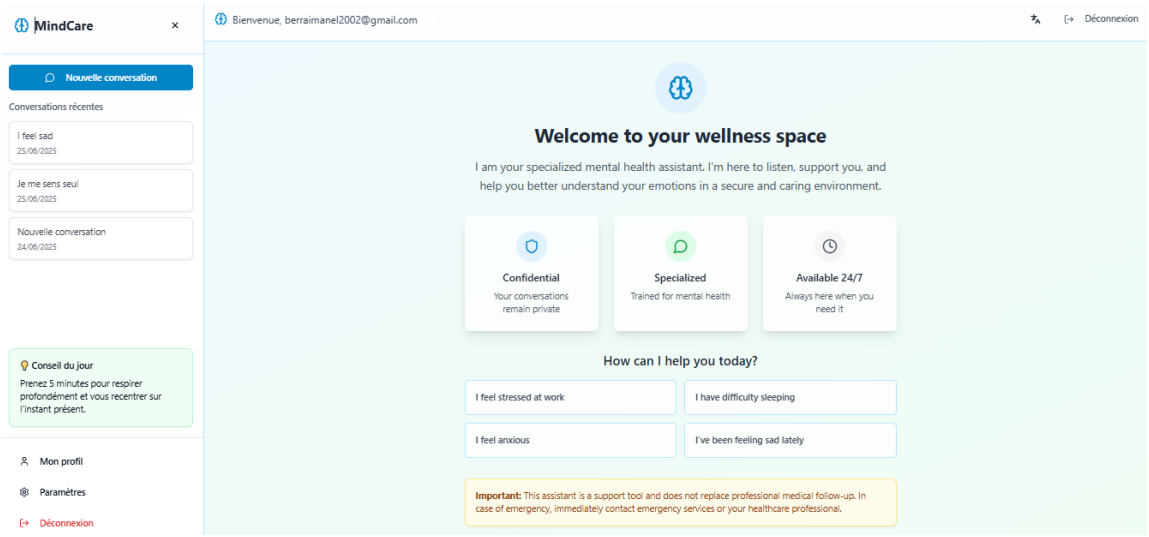

Figure 4.16: Home screen of the chatbot interface

4.6.2 Backend and Flan-T5 Integration

The backend of the system was developed using **Flask**, a lightweight Python web framework. It manages the communication between the frontend and the machine learning model.

The Flan-T5 model was fine-tuned on a Kaggle dataset of mental health conversations, formatted as prompt-response pairs. Once the model was trained, it was saved locally using:

```
model.save_pretrained("path_to_dir")
tokenizer.save_pretrained("path_to_dir")
```

The saved model files (`pytorch_model.bin`, `config.json`, and tokenizer files) were then loaded inside the Flask server. Every user message sent from the frontend is transmitted via an API call to Flask, which processes the input and returns a generated, context-aware, and empathetic response from the Flan-T5 model.

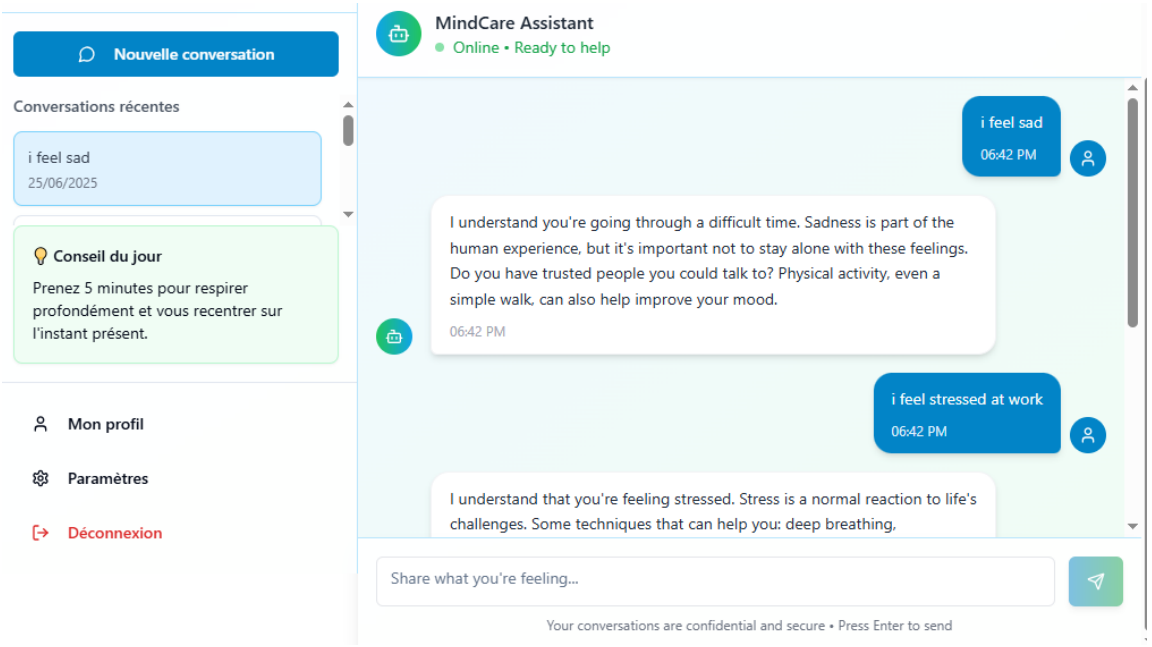

Figure 4.17: Example of conversation related to sadness and stress

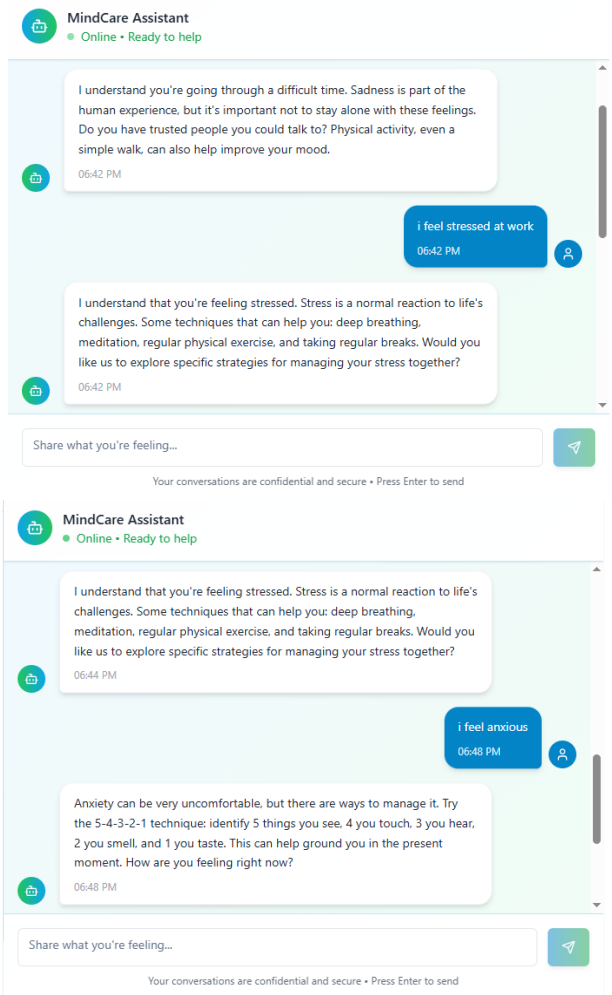

Figure 4.18: Example of response to anxiety

Conception Diagrams of The Website

CMD :

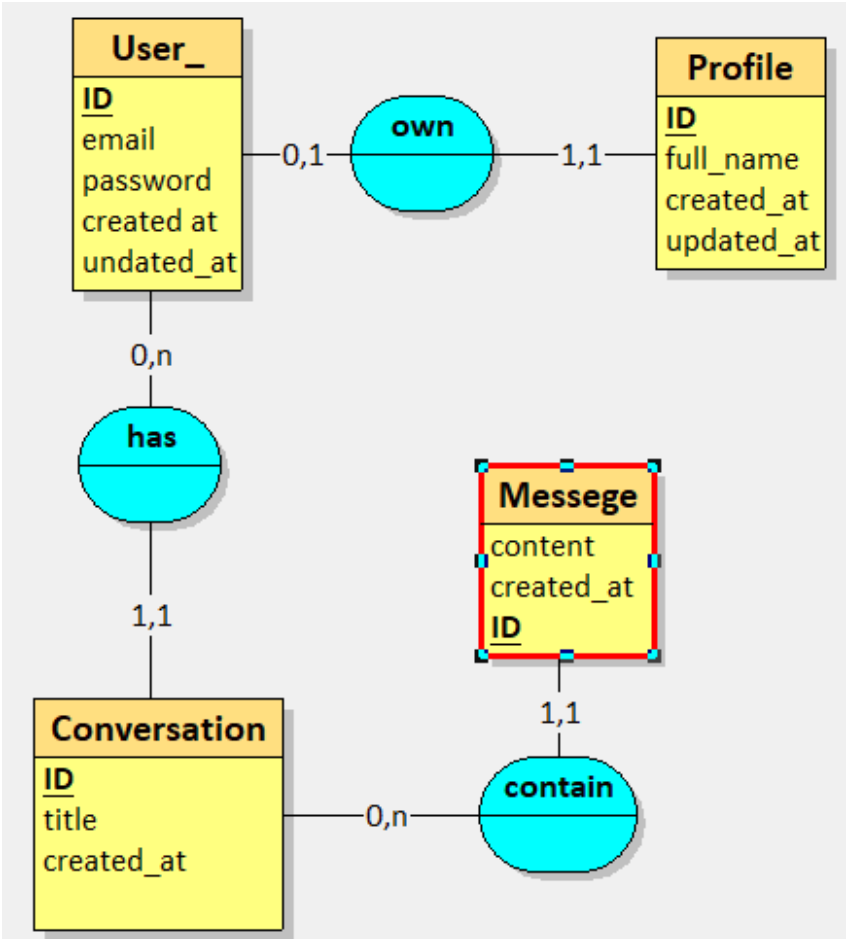

Figure 4.19: Conceptual Modeling Data (CMD)

UML :

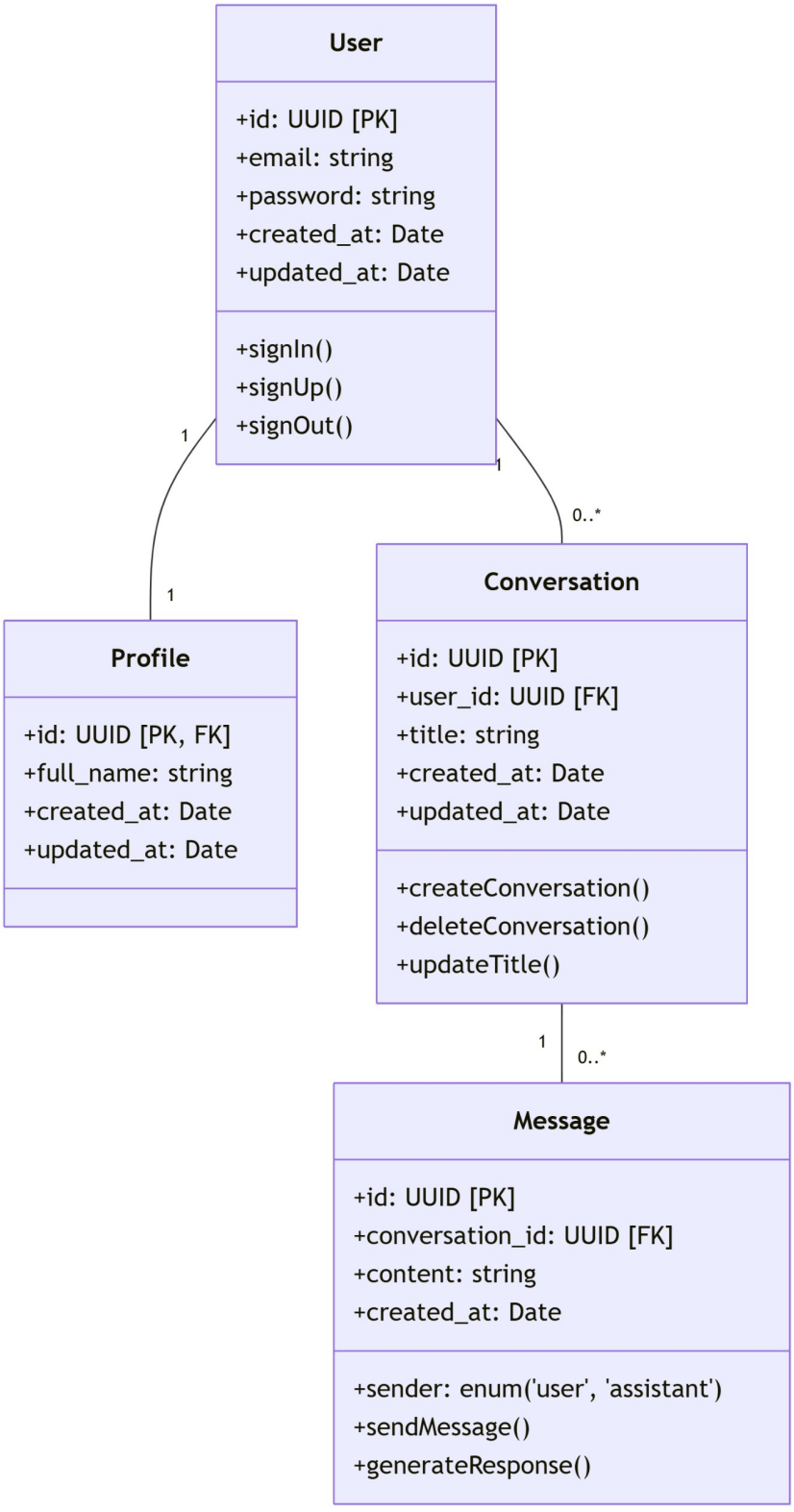

Figure 4.20: Unified Modling Language Diagram (UML)

Database Integration

To manage users and their conversation history, **Supabase** (a PostgreSQL-based backend-as-a-service) was used. SQL queries were executed to store, retrieve, and filter user messages securely. The integration was performed within VS Code using REST API calls.

4.7 Conclusion

This chapter presented a series of experiments exploring various architectures for depression detection using textual, visual, and multimodal data. Binary and multiclass classification tasks were conducted using transformer-based models, domain-specific variants, and structured-feature methods. Image-based classifiers were also evaluated separately, followed by fusion strategies for multimodal integration. Finally, a generative chatbot was introduced to demonstrate the practical potential of severity-aware conversational agents. Overall, the experimental results confirm the added value of specialized models, data fusion, and task-specific designs in enhancing predictive performance in mental health applications.

General Conclusion

This thesis explored the creation of an intelligent online chatbot designed to support people dealing with mental health challenges. Our work began with a deep understanding of how serious and widespread mental health issues have become, and how new technologies—especially artificial intelligence—can help fill the gaps in care.

We designed a system made of three main parts: a model that detects the user's mental health state, a chatbot that communicates in a natural and supportive way, and a simple interface that anyone can use. Among the models tested, the **XGBoost classifier** achieved excellent performance, with an accuracy of **95%**, showing its strength in analyzing and classifying complex emotional and behavioral data. The use of **Flan-T5**, a powerful generative AI model, allowed our system to create meaningful and empathetic responses that feel more human—offering real support to users during vulnerable moments.

We also focused on making the system easy to use. The user interface was designed to be clear and minimal, so that anyone—regardless of age, background, or technical skills—can feel comfortable interacting with it. In mental health, feeling safe and understood is as important as the technology behind the scenes.

While this project achieved promising results, there is still much we can do to improve and expand it:

- **Add real-time facial emotion monitoring** using camera input, allowing the system to better understand how users feel by analyzing their expressions.
- **Integrate more languages**, so that users from different linguistic and cultural backgrounds can benefit from the chatbot.
- **Develop mobile and offline versions** of the system, especially for regions with limited internet access.
- **Collaborate with mental health professionals** to ensure the chatbot follows psychological best practices and can be trusted in real-life situations.
- **Improve personalization**, so that the system can adapt its tone and responses to different user needs and emotional states over time.

To conclude, this thesis is not only a step forward in technology—it is also a step toward more inclusive, compassionate, and accessible mental health support. By combining smart AI tools with human-centered design, we hope to contribute to a future where no one has to face mental health struggles alone.

Bibliography

[1] I. H. Sarker. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2(6), 2021.

[2] S. Saxena. What is lstm? introduction to long short-term memory. https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/, May 2025.

[3] Pixlr. Générateur d'images ia: Transformer du texte en images, art génératif et photos générées, 2024.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems (NeurIPS)*, 2017.

[5] Vladislav Efimov. Large language models: Bert – bidirectional encoder representations from transformer. *Towards Data Science*, January 20 2025.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019.

[7] E. Zvornicanin and E. Zvornicanin. Comparison between bert and gpt-3 architectures | baeldung on computer science, January 2024.

[8] G. Yenduri et al. Gpt (generative pre-trained transformer)— a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *IEEE Access*, 12:54608–54649, 2024.

[9] Daicwoz. DAIC-WOZ Dataset, 2024.

[10] Table 1: Patient health questionnaire (phq-8) scoring and interpretation, n.d.

[11] Eleventh Hour Enthusiast. Clinicalbert and bluebert, November 2024. Accessed June 2025.

[12] A. Khoeini. Fttransformer: Transformer architecture for tabular datasets, 2024.

[13] C. Hashemi-Pour and B. Lutkevich. BERT language model. https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model, 2024.

[14] World Health Organization. Mental health: Strengthening our response, 2022. Accessed: 2025-05-29.

[15] American Psychological Association. Mental health, 2023. Accessed: 2025-05-29.

[16] Corey L. M. Keyes. The mental health continuum: From languishing to flourishing in life. *Journal of Health and Social Behavior*, 43(2):207–222, 2002.

[17] Vikram Patel et al. The lancet commission on global mental health and sustainable development. *The Lancet*, 392(10157):1553–1598, 2018.

[18] American Psychological Association. Stress: The different kinds of stress, 2023. Accessed: 2025-05-29.

[19] Bruce S. McEwen. Protective and damaging effects of stress mediators. *New England Journal of Medicine*, 338(3):171–179, 1998.

[20] Neil Schneiderman, Gail Ironson, and Scott D. Siegel. Stress and health: Psychological, behavioral, and biological determinants. *Annual Review of Clinical Psychology*, 1:607–628, 2005.

[21] Leandro D. Godoy et al. Stress, depression, and anxiety in university students: A cross-sectional study. *Journal of Depression and Anxiety*, 7(1):1–5, 2018.

[22] World Health Organization. Mental disorders: Key facts, 2022. Accessed: 2025-05-29.

[23] American Psychiatric Association. What are anxiety disorders?, 2023. Accessed: 2025-05-29.

[24] Michelle G. Craske and Murray B. Stein. Anxiety disorders. *Dialogues in Clinical Neuroscience*, 11(3):319–336, 2009.

[25] Dan J. Stein et al. Global prevalence and burden of anxiety disorders: A systematic review and meta-regression. *Psychological Medicine*, 47(11):2037–2049, 2017.

[26] American Psychiatric Association. What is posttraumatic stress disorder (ptsd)?, 2023. Accessed: 2025-05-29.

[27] Bessel A Van der Kolk. Posttraumatic stress disorder: the neurobiological impact of psychological trauma. *Dialogues in Clinical Neuroscience*, 15(3):263–278, 2013.

[28] Ronald C. Kessler et al. Posttraumatic stress disorder in the national comorbidity survey. *Archives of General Psychiatry*, 52(12):1048–1060, 1995.

[29] Mina Fazel, Jeremy Wheeler, and John Danesh. Mental health of displaced and refugee children resettled in high-income countries: risk and protective factors. *The Lancet*, 365(9467):1309–1314, 2005.

[30] Charles B. Nemeroff. Neurobiology of childhood trauma and abuse: implications for mental health and child welfare policy. *Development and Psychopathology*, 16(3):553–576, 2004.

[31] David J. Kupfer. The increasing medical burden in bipolar disorder. *JAMA*, 293(20):2528–2530, 2005.

[32] World Health Organization. Depression, 2021. Accessed: 2025-05-29.

[33] E. David Klonsky and Alexis M. May. The three-step theory (3st): A new theory of suicide rooted in the "ideation-to-action" framework. *International Journal of Cognitive Therapy*, 9(2):114–129, 2016.

[34] Matthew K. Nock, Guilherme Borges, Evelyn J. Bromet, Christine B. Cha, Ronald C. Kessler, and Sing Lee. Cross-national prevalence and risk factors for suicidal ideation, plans and attempts. *The British Journal of Psychiatry*, 192(2):98–105, 2008.

[35] Lakshmi Vijayakumar. Suicide in women. *Indian Journal of Psychiatry*, 57(Suppl 2):S233–S238, 2015.

[36] World Health Organization. Suicide, 2021. Accessed: 2025-05-29.

[37] James Morrison. *Interviewing Children and Adolescents: Skills and Strategies for Effective DSM-IV Diagnosis*. Guilford Press, 2012.

[38] Kurt Kroenke, Tara W. Strine, Robert L. Spitzer, Janet B.W. Williams, Joseph T. Berry, and Ali H. Mokdad. The phq-8 as a measure of current depression in the general population. *Journal of Affective Disorders*, 114(1–3):163–173, 2009.

[39] American Psychiatric Association. *Diagnostic and Statistical Manual of Mental Disorders (DSM-5)*. American Psychiatric Publishing, 5th edition, 2013.

[40] World Health Organization. International classification of diseases, 11th revision (icd-11), 2022. Accessed: 2025-05-29.

[41] Jerome C. Wakefield. The concept of mental disorder: diagnostic implications of the harmful dysfunction analysis. *World Psychiatry*, 6(3):149–156, 2007.

[42] Sarah Clement, Oliver Schauman, Tanya Graham, Federica Maggioni, Sara Evans-Lacko, Nino Bezborodovs, Craig Morgan, Nicolas Rüsch, June S. Brown, and Graham Thornicroft. What is the impact of mental health-related stigma on help-seeking? a systematic review of quantitative and qualitative studies. *Psychological Medicine*, 45(1):11–27, 2015.

[43] World Health Organization. Mental health atlas 2017, 2018. Accessed: 2025-05-29.

[44] Paul M. Galdas, Francine Cheater, and Paul Marshall. Men and health help-seeking behaviour: literature review. *Journal of Advanced Nursing*, 49(6):616–623, 2005.

[45] Harvey A. Whiteford, Louisa Degenhardt, Jürgen Rehm, Amanda J. Baxter, Alize J. Ferrari, Holly E. Erskine, Fiona J. Charlson, Rosana E. Norman, Abraham D. Flaxman, Nicole Johns, et al. Global burden of disease attributable to mental and substance use disorders: findings from the global burden of disease study 2010. *The Lancet*, 382(9904):1575–1586, 2013.

[46] Rafael A Calvo, David N Milne, Mohammad Shahid Hussain, and Helen Christensen. Natural language processing in mental health applications using non-clinical texts. *Natural Language Engineering*, 23(5):649–685, 2017.

[47] Sharath Chandra Guntuku, David B Yaden, Margaret L Kern, Lyle H Ungar, and Johannes C Eichstaedt. Language of adhd in adults on social media. *Journal of Attention Disorders*, 2017.

[48] Thomas Davenport and Ravi Kalakota. The ai-powered mental health revolution. *Harvard Business Review*, 2019.

[49] Kathleen Kara Fitzpatrick, Alison Darcy, and Molly Vierhile. Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (woebot): a randomized controlled trial. *JMIR Mental Health*, 4(2):e19, 2017.

[50] Joseph Firth, John Torous, Jennifer Nicholas, Rebekah Carney, Simon Rosenbaum, and Jerome Sarris. The efficacy of smartphone-based mental health interventions for depressive symptoms: a meta-analysis of randomized controlled trials. *World Psychiatry*, 18(3):325–336, 2019.

[51] John A. Naslund, Kelly A. Aschbrenner, Ricardo Araya, Lisa A. Marsch, Jürgen Unützer, and Vikram Patel. Digital technology for treating and preventing mental disorders in low-income and middle-income countries: a narrative review of the literature. *The Lancet Psychiatry*, 7(6):486–500, 2020.

[52] Becky Inkster, Sneha Sarda, and Venkatesh Subramanian. Digital health management: A case study of the wysa app. *BJPsych Open*, 4(5):427–432, 2018.

[53] Jonathan Gratch, Ron Artstein, Gale M Lucas, Giota Stratou, Stefan Scherer, Amir Nazarian, Rachel Wood, Jill Boberg, David DeVault, Stacy Marsella, David Traum, and Louis-Philippe Morency. The distress analysis interview corpus of human and computer interviews. In *Proceedings of LREC*, 2014.

[54] Ian Barnett, John Torous, Patrick Staples, Luis Sandoval, Matcheri Keshavan, and Jukka-Pekka Onnela. Predicting psychiatric hospital readmission with mobile phone data. *NPJ Digital Medicine*, 1(1):1–8, 2018.

[55] Nicholas Cummins, Stefan Scherer, Jarek Krajewski, Sarah Schnieder, Julien Epps, and Thomas F Quatieri. A review of depression and suicide risk assessment using speech analysis. *Speech Communication*, 71:10–49, 2015.

[56] Glen Coppersmith, Rebecca Leary, Patrick Crutchley, and Alex Fine. Clpsych 2015 shared task: Depression and ptsd on twitter. In *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology*, pages 31–39, 2015.

[57] Katrin Kliegl, Sabine Huber, Philip Lindner, Eva Walther, and Yannik Terhorst. Adaptive learning systems for health coaching: A framework and illustrative case. *JMIR Mental Health*, 6(6):e13716, 2019.

[58] Adrian B R Shatte, David M Hutchinson, and Samantha J Teague. Machine learning in mental health: a scoping review of methods and applications. *Psychological Medicine*, 49(9):1426–1448, 2019.

[59] John Torous, Mark E Larsen, Colin Depp, Theodore D Cosco, Ian Barnett, Matthew K Nock, and Joseph Firth. Digital phenotyping for mental health of college students: a clinical review. *Depression and Anxiety*, 38(6):573–586, 2021.

[60] TutorialsPoint. Artificial intelligence tutorial, March 2025.

[61] A. Alam. What is machine learning? Zenodo (CERN European Organization for Nuclear Research), 2023.

[62] C. Stryker and J. Holdsworth. Natural language processing. what is nlp (natural language processing)? https://www.ibm.com/think/topics/natural-language-processing, June 2025.

[63] Nasser Sami Z.-E. P. S. C. Itil. The 7 subfields of artificial intelligence! https://www.linkedin.com/pulse/7-subfields-artificial-intelligence-nasser-sami-2vyae/, February 2024.

[64] J. Holdsworth and M. Scapicchio. Deep learning. what is deep learning? https://www.ibm.com/think/topics/deep-learning, June 2025.

[65] Qu'est-ce qu'un rnn ? – présentation des réseaux neuronaux récurrents – aws. https://aws.amazon.com/fr/what-is/recurrent-neural-network/, n.d.

[66] S. Arifin, A. Wijaya, R. Nariswari, I. G. A. A. Yudistira, S. Suwarno, F. Faisal, and D. Wihardini. Long short-term memory (lstm): Trends and future research potential. *International Journal of Emerging Technology and Advanced Engineering*, 13(5):24–34, 2023.

[67] K. Singh. Principles of generative ai: A technical introduction. https://www.cmu.edu/intelligentbusiness/expertise/genai-principles.pdf, n.d. Carnegie Mellon University.

[68] Adam Zewe. Explained: Generative ai. https://news.mit.edu/2023/explained-generative-ai-1109, November 2023. MIT News | Massachusetts Institute of Technology.

[69] A. Kumar. Ai text generator: Working, facts, and uses. https://pwskills.com/blog/ai-text-generator/, May 2023. PW Skills Blog.

[70] S. Ramzan, M. M. Iqbal, and T. Kalsum. Text-to-image generation using deep learning. In *The 7th International Electrical Engineering Conference*, page 16. MDPI, Jul 2022.

[71] J. B. Chandran Jogith. A comprehensive overview of multimodal generative ai, 2025. Accessed: 2025-04-14.

[72] M. Belcaid. Comparison of transformer-based and convolutional neural network-based (cnn) models for remote sensing image classification. https://run.unl.pt/bitstream/10362/150959/1/TGEO0284_E.pdf, 2023.

[73] Lewis Tunstall, Leandro von Werra, and Thomas Wolf. *Natural Language Processing with Transformers, Revised Edition*. O'Reilly Media, 2022.

[74] Thanh Tam Nguyen. Machine translation with transformers (by n. t. vu & pavel denisov). Master's thesis, University of Stuttgart, 2019.

[75] Gayathri Siva. Bert — bidirectional encoder representations from transformer. *Medium*, January 4 2022.

[76] Shubham Baranwal. Understanding bert - towards ai. *Medium*, December 13 2021. Accessed: 2025-06-05.

[77] W&B. An introduction to bert and how to use it. https://wandb.ai/mukilan/BERT_Sentiment_Analysis/reports/An-Introduction-to-BERT-And-How-To-Use-It--VmlldzoyNTIyOTA1, June 8 2025.

[78] GeeksforGeeks. Introduction to generative pretrained transformer (gpt). https://www.geeksforgeeks.org/introduction-to-generative-pre-trained-transformer-gpt/, July 2024.

[79] J. Schulze. What is gpt? gpt-3, gpt-4, and more explained. https://www.coursera.org/articles/what-is-gpt, 2024. Coursera, Oct. 28, 2024.

[80] Accubits Technologies Inc. Flan t5. https://accubits.com/large-language-models-leaderboard/flan-t5/, 2023. Accubits, June 2, 2023.

[81] Giuliano Lorenzoni, Cristina Tavares, Natalia Nascimento, Paulo Alencar, and Donald Cowan. Assessing ml classification algorithms and nlp techniques for depression detection: An experimental case study. *arXiv preprint arXiv:2404.04284*, 2024.

[82] Sadegh Jafari, Erfan Zare, et al. Psychological health chatbot, detecting and assisting patients in their path to recovery. 2025.

[83] Michael Danner, Bakir Hadzic, et al. Advancing mental health diagnostics: Gpt-based method for depression detection. In *SICE Annual Conference*. IEEE, 2023.

[84] Clinton Lau, Xiaodan Zhu, and Wai-Yip Chan. Automatic depression severity assessment with deep learning using parameter-efficient tuning. *Frontiers in Psychiatry*, 14:1160291, 2023.

[85] Sergio Burdisso et al. Daic-woz: On the validity of using the therapist's prompts in automatic depression detection from clinical interviews. In *Proceedings of the 6th Clinical NLP Workshop*, 2024.

[86] David Gimeno-Gómez et al. Reading between the frames: Multi-modal depression detection in videos from non-verbal cues. *arXiv preprint arXiv:2401.02746*, 2024.

[87] Alireza Afzal Aghaei and Nadia Khodaei. Automated depression recognition using multimodal machine learning: A study on the daic-woz dataset. *Computational Modeling in Cognitive and Affective Neuroscience*, 2023.

[88] Santosh V. Patapati. Integrating large language models into a tri-modal architecture for automated depression classification. *arXiv preprint arXiv:2407.19340*, 2024.

[89] Sabbir Ahmed, Md Abu Yousuf, Md Mofizur Monowar, Abdullah Hamid, and Moayad O. Alassafi. Taking all the factors we need: A multimodal depression classification with uncertainty approximation. *IEEE Access*, 11:99847–99861, 2023.

[90] Ricardo Flores et al. Temporal facial features for depression screening. In *Proceedings of the 2022 ACM UbiComp Conference*, pages 488–493, 2022.

[91] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko. Revisiting deep learning models for tabular data. *arXiv preprint arXiv:2106.11959v2*, 2021.

[92] GeeksforGeeks. Logistic regression in machine learning, 2025.

[93] TheDevastator. Nlp - mental health conversations, 2023. Accessed: 2025-06-25.