Multimodal AI for Early Mental Health Detection

A major project report submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Technology

in

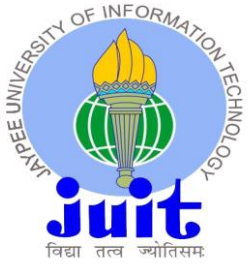
Computer Science & Engineering

Submitted by

Ojaswi Chauhan (221030035)
Rahul Rose (221030051)
Rahul Kumar (221030109)
Nishant Gautam (221030334)

Under the guidance & supervision of

Dr. Ravindara Bhatt



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology,

Waknaghat, Solan - 173234 (India)

December 2025

Table of Contents

Certificate

This is to certify that the major project report entitled **'Multimodal AI for Early Mental Health Detection'**, submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering**, in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat, is a bonafide project work carried out under my supervision during the period from July 2025 to December 2025.

I have personally supervised the research work and confirm that it meets the standards required for submission. The project work has been conducted in accordance with ethical guidelines, and the matter embodied in the report has not been submitted elsewhere for the award of any other degree or diploma.

Supervisor Name & Sign: Dr. Ravindara Bhatt

Date: Designation: Associate Professor

Place: Department: Computer Science & Engineering

Declaration

We hereby declare that the work presented in this major project report entitled **'Multimodal AI for Early Mental Health Detection'**, submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering**, in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat, is an authentic record of our own work carried out during the period from July 2025 to December 2025 under the supervision of **Dr. Ravindara Bhatt.**

We further declare that the matter embodied in this report has not been submitted for the award of any other degree or diploma at any other university or institution.

Name & Sign: Ojaswi Chauhan
Roll No.: 221030035
Date:

Name & Sign: Rahul Rose
Roll No.: 221030051
Date:

Name & Sign: Rahul Kumar
Roll No.: 221030109
Date:

Name & Sign: Nishant Gautam
Roll No.: 221030334
Date:

This is to certify that the above statement made by the candidates is true to the best of my knowledge.

Supervisor Name & Sign: Dr. Ravindara Bhatt
Date: Designation: Associate Professor
Place: Department: Computer Science & Engineering

List of Tables:

List of Figures:

List of Abbreviations, Symbols, or Nomenclature

1. **AI –** Artificial Intelligence
2. **ML –** Machine Learning
3. **DL –** Deep Learning
4. **NLP –** Natural Language Processing
5. **LSTM –** Long Short-Term Memory
6. **CNN –** Convolutional Neural Network
7. **TF-IDF –** Term Frequency–Inverse Document Frequency
8. **BERT –** Bidirectional Encoder Representations from Transformers
9. **MFCC –** Mel-Frequency Cepstral Coefficients
10. **API –** Application Programming Interface

11. **PHQ-9 –** Patient Health Questionnaire-9 (clinical depression assessment tool)
12. **BDI-II –** Beck Depression Inventory-II
13. **DAIC-WOZ –** Distress Analysis Interview Corpus Wizard-of-Oz (dataset)
14. **FER-2013 –** FaRelucial Expression Recognition 2013 (dataset)
15. **RAVDESS** – Ryerson Audio-Visual Database of Emotional Speech and Song
16. **TESS** – Toronto Emotional Speech Set
17. **F1-Score** – Harmonic Mean of Precision and Recall
18. **SVM** – Support Vector Machine

Abstract

The prevalence of mental health issues such as stress, anxiety, and depression has been steadily rising, with students and working professionals being the most affected groups. The prevention of the intensification of such problems greatly depends on the early identification stage. With this in mind, the **Multimodal AI for Early Mental Health Detection** project may well be the answer to the problem. The Project aims to support the early awakening of the users by absorbing and analyzing various forms of user-generated data (text, speech, and facial expressions) through machine learning and deep learning techniques.

The current development has only the text and audio modalities. Text inputs are handled with natural language processing techniques, and speech signals are subjected to Mel-Frequency Cepstral Coefficient (MFCC) extraction and are classified with a Long Short-Term Memory (LSTM) network to track the temporal emotional cues. The audio models were developed with the help of publicly available emotional-speech datasets like RAVDESS and TESS.

These processed components are combined in a Flask-based backend, thereby facilitating real-time prediction via a user-friendly web interface. The facial expression recognition part, which is supported by convolutional neural networks (CNNs), is the next step to completing the multimodal framework by adding the visual emotional cues.

The primary goal of this undertaking is to provide a working example of artificial intelligence as a helpful tool in realizing mental-health problems in their early stages through the integration of various behavioural signals. The clinical diagnosis itself is not intended to be replaced by the system but only to be supplemented, which indicates the degree of technological accomplishment of such approaches in helping people become mentally aware and, thus, prompting them to seek professional advice in time.

Chapter 1: Introduction

1.1 Introduction

It can be largely observed that the rate of mental-health problems has increased maily in the last few years. Generally, these issues develop step-by-step, thus identifying them at an early stage is indispensable if one wants to receive help on time. Unfortunately, a vast number of people do not recognise at all that their symptoms are in the initial stages and even those who do, are reluctant to share due to the stigma associated with it and their own lack of awareness. Devices that provide users with insights into changes in someone's behaviour and are easily reachable can really solve this problem.

Digital communication has become so popular that in a way people have to show their feelings unintentionally through a language they use, a wave in their voice or their facial gestures. However, a breakthrough in Artificial Intelligence (AI) and Machine Learning (ML) has opened doors for the new era of technology which automatically and significantly can understand the signals that are given by a person without any human interaction. Whereas single-modality systems can only partially reveal the emotional state of an individual, multimodal systems that fuse different data can provide a holistic view of the person's feelings.

This AI-powered multimodal project is designed to use AI-based methods to analyze texts, speeches and later facial expressions. For example, the natural-language processing system in the project uses TF-IDF and BERT model fine-tuning to get not only the general surface level of the textual data but also to find the deep contextual meaning of the text. Speech plays a great role in communication and hence here they first use Mel-Frequency Cepstral Coefficients (MFCC) to get the main features out of the speech and then they use the Long Short-Term Memory (LSTM) neural network to locate the changes in the emotion over the speech. Addition of a facial-expression recognition feature will mark the completion of the multimodal pipeline in the next phase.

The tool is designed to help rather than replace doctors, and its role would be that of a facilitator encouraging users to think about, and perhaps realize, their poor mental health, and urging them to get professional help when necessary. The idea of pin-pointing the

mental state through technological means by actively employing AI in analyzing the multiple signals from the behavioral and speech domains is quite fascinating, indeed which could make huge advances in mental-health care available to everyone in an accessible and friendly manner. Although the system is not designed to make medical diagnoses, it is basically an aid tool that prompts users to think about their psychological health and get a professional if needed. Using several behavioural signs and state-of-the-art AI techniques, the initiative is a showcase of the potential of tech to localize and even prefigure the onset of mental health problems in a way that is both easily understandable and approachable by the general public.

1.2 Problem Statement

Even though the general public is more aware of mental health issues than before, the problem of the detection of these issues at an early stage remains. Most times, clinical diagnosis is made when a person decides to seek the help of a professional, and this is usually too late. The existing AI solutions are single-minded and hence only focus on one type of data, such as text or speech. However, human emotions are far more complicated than that.

There is no widely available system that brings together multiple data sources (text, voice, and facial cues) and works in real time to give an indication of someone's mental state. This is the gap we want to address with our project.

1.3 Objectives

- Using AI and machine learning to recognize the early symptoms of mental health problems (e.g., stress, anxiety, and depression) is one of the goals.
- Another goal is the research and understanding of text, audio, and video/image data for the identification of hidden emotional and behavioral patterns.
- It is about designing a multimodal system that merges the features of the voice, the facial expressions, and the written text for more significant precision.

- Creating a user-friendly web application where people can easily interact with the system is another thing.
- Giving the right-time feedback or, in some cases, alerts that can motivate people to get professional help at the early stages is another point on the list.

1.4 Significance and Motivation of the Project Work

Mental-health difficulties of the kind stress, anxiety, and depression may continue to affect a person without his/her being aware within a long time and without any obvious early signs. Unfortunately, many people do not seek help immediately since the stigma attached to the problem, their own lack of knowledge, and the mistaken idea that their symptoms are transient. This turns early detection into a necessity because, in fact, if interventions are carried out in time, the risks of the escalation of such problems will be reduced to the minimum. Digital means of communication which are more and more favored by people give the possibility of watching over one's emotional condition through the behavioural changes that are apparent in texts, speech as well as facial expressions and in the language we use in our everyday lives.

The importance of this enterprise is revealed by the fact that it employs AI to facilitate the understanding of such changes without human intervention automatically. It is indicative of a person's emotional state when ideas are expressed through words, or voice, or face. Thanks to significant improvements in natural language processing and deep learning, detecting these patterns can also be achieved in a very meaningful and non-interventionist way. Here the textual modality is powered by transformer-based models like BERT which can facilitate a much higher level of the context understanding of the texts than that of the standard NLP approaches. In like manner, the speech signal is extracted with MFCCs and then the LSTM networks which are good at modeling the changes of the nature over time in the speech and hence emotional states are used for the evaluation. Fusing all the modalities allows the technology to be more robust and complete in comparison to the reliance of a single channel only.

The project is the outcome of a strong desire fixed on helping the society and also the field of technology. On the side of the society, the issue of mental health among university students and young workers is gathering momentum while at the same time no readily

available screening tools for early detection are in place. AI-enabled technology helps create awareness about one's symptoms and thus acts as an early agent of support making the user have the initiative to consult professionals before his/her symptoms exacerbate. The activity, on the other hand, is capable of attracting, the attention of computer science

theorists and practitioners in various fields such as natural language processing, speech processing, machine learning, and deep learning and Web development. The use of a multimodal structure, punctuated by a Flask-based simple GUI for interaction, is a perfect show-off of the power implied in the new generation of AI techniques when dealing with everyday life well-being issues.

The aim of the venture is to aid early mental-health awareness through technology, which is an easy, convenient, and hybrid behavioural-indicators strategy capable of producing combined results rather than isolated ones.

1.5 Organization of Project Report

1. **Chapter 1: Introduction:**

   This chapter details the extent of the background of the project, the issue, the aims, and the impulse of creating a multimodal AI system for the early detection of mental health. It also briefly mentions the importance of the first two modalities, i.e., text and speech, and then the inclusion of facial cues and the project scope.

2. **Chapter 2: Literature Survey:**

   This chapter reviews previous studies and existing systems related to mental-health detection using text, audio, and vision. It summarizes important research findings, common datasets, model architectures, and identifies key gaps—such as limitations of single-modality approaches and lack of real-time multimodal systems—that the present work aims to address.

3. **Chapter 3: System Development:**

   This chapter explains the technical development of the system. It covers requirement analysis, system design, architecture, dataset details, text preprocessing using tokenization, audio preprocessing using MFCC extraction, and the training of both the BERT-based text model and the LSTM-based audio model.

4. **Chapter 4: Testing:**

   This chapter describes how the system components were executed together and implemented to work correctly. Other than that, this shows whether the evaluation parameters were calculated efficiently and correctly. It talks about the working of APIs and other things responsible for smooth operation of model.

5. **Chapter 5: Results and Evaluation:**

   This chapter gives insights about the entire integration of the above-mentioned chapters with key findings and learning about stuff and model. It presents graphs and important conclusions and results of the entire workflow but facts and precision score as well as accuracy. It shows the in total result so far.

6. **Chapter 6: Conclusions and Future Scope:**

   The final chapter summarizes the major contributions of the project and outlines the limitations encountered during development. It also presents the future scope, including potential technical improvements, possible addition of visual modality, real-world deployment possibilities, and opportunities for expanding the multimodal framework for more robust mental-health support.

7. **References:** Provide all the supporting research papers and data.

Chapter 2: Literature Survey

2.1 Overview of Relevant Literature

In the last few years, detecting early signs of mental health issues such as depression, stress, or anxiety using AI has become a highly active area of research. Different studies have explored text data (like social media posts or transcripts), audio features (like tone of voice), and visual signals (like facial expressions). Here we summarize some of the most relevant works in each area.

2.1.1 Text-based studies:

Social media has been a popular source for detecting mental health problems because people often express their feelings openly online. Zeberga et al. [9] built a hybrid model using BERT embeddings and Bi-LSTM networks to detect depression and anxiety in Reddit and Twitter posts, reporting impressive results with ~98% accuracy. Hasan et al. [10] compared transformer models (BERT, RoBERTa) with classical LSTM approaches and found that while transformers were usually better, LSTMs combined with BERT embeddings could still perform competitively when computing resources are limited. Sevinç [11] proposed a new training method called LatentGLoss, which uses a teacher–student style architecture, showing that carefully designed loss functions can improve classification of mental health text.

Other recent text-based work also highlights this trend. Sawhney et al. [25] used multimodal sentiment and emotion features from Reddit posts to identify suicidal ideation, while Yates et al. [26] evaluated large-scale Reddit corpora for depression detection and stressed the importance of balanced and annotated datasets.

2.1.2 Audio and speech-based studies:

Voice and speech patterns are also strong indicators of mental health. Al Hanai et al. [1] demonstrated that using LSTM models on audio and text sequences from interviews (DAIC-WOZ dataset) can capture depression cues over time. Morales [2] compared different modalities (verbal, acoustic, visual) and showed that combining them is almost

always better than relying on one. More recently, Wang et al. [3] presented Speechformer-CTC, a transformer-based model that leverages long-range dependencies in speech and tested it on datasets like Callyope-GP and the Androids Corpus, achieving strong results.

2.1.3 Vision-based studies:

Facial expressions are another powerful signal of emotional and mental state. Early work by Almaev and Valstar [4] used handcrafted spatio-temporal features (LGBP-TOP) for recognizing facial expressions. Later, Ipinze Tutuianu et al. [5] benchmarked modern deep facial expression recognition protocols on balanced datasets collected "in the wild," improving reliability across demographics. Widely used datasets such as FER-2013 [6] and AffectNet [19] have powered CNN-based models for facial emotion recognition. For example, Upadhyay and Sharma [7] demonstrated that hybrid CNN approaches combining Haar cascades with deep networks can achieve ~70% accuracy on FER-2013.

2.1.4 Datasets and shared tasks:

The AVEC challenges [8] played a central role in standardizing evaluation for depression and emotion recognition. Similarly, datasets like DAIC-WOZ [12], FER-2013 [6], Callyope-GP [13], Androids Corpus [18], and AffectNet [19] are repeatedly cited as reliable benchmarks. These datasets vary in size and modality, but together they allow researchers to compare methods more fairly.

2.2 Key Gaps in the Literature

Even though progress has been significant, there are some gaps that remain:

1 **Limited and imbalanced datasets:** Labelling and evaluation differences: Different researchers use different tools for the assessment (PHQ-9 [20], BDI-II [21]) and also metrics (accuracy, F1-score, CCC, RMSE), thus, it is a very complicated task to determine the results directly from the comparison of these works [2], [8].

2 **Differences in labelling and evaluation:** Although multimodal methods indicate stable enhancement [2], [3],] it is not possible to find a single most effective way to

combine text, audio, and vision data. The advantages and disadvantages of early, late, and hybrid fusion strategies exist.

3 **Fusion of modalities:** While multimodal approaches show consistent improvements [2], [3], there is no single best practice for combining text, audio, and vision data. Early, late, and hybrid fusion strategies each have strengths and weaknesses.

4 **Generalizability:** Models trained on controlled datasets (FER-2013 [6]) often perform poorly in noisy, real-world conditions [5]. Robustness across culture, language, and demographic factors is still a major challenge.

5 **Ethical and privacy issues:** Most of the technical works are aimed at accuracy improvement, however, very few are the discussions regarding ethical deployment, user consent, and protection of the sensitive mental health data.

Chapter 3: System Development

3.1 Requirements and Analysis

The development of the proposed multimodal mental-health detection system requires a combination of hardware, software, and functional capabilities to ensure accurate processing of text and audio inputs. Since the system uses deep-learning models and real-time inference through a web interface, both computational resources and appropriate libraries are essential.

Table 3.1: Functional and Non-Functional Requirements

Category	Requirement
Functional	• Users can enter text inputs (and later upload audio/image). • Data is preprocessed (text cleaning, tokenization, audio feature extraction, face detection). • AI/ML models classify mental health state (normal, stress, anxiety, depression). • Predictions and user inputs are stored securely in a database. • Web interface displays predictions clearly.
Non-Functional	• Predictions must be near real-time. • System must be scalable for multimodal inputs. • Interface must be user-friendly and accessible.

Table 3.2: Software and Hardware Requirements

Requirement	Details
Hardware	Laptop/PC with at least 8GB RAM, Intel i5 or higher, optional GPU for model training (NVIDIA CUDA-enabled preferred).
Operating System	Windows 10/11, Linux (Ubuntu recommended), or macOS.
Programming Language	Python 3.10+ (core development).
Frameworks	Flask (backend API), TensorFlow/PyTorch (ML/DL modeling).
Libraries	pandas, numpy, scikit-learn, joblib, nltk/spacy (NLP), librosa (audio features), OpenCV (face detection), matplotlib (visualization).
Frontend	HTML, CSS, JavaScript (basic web UI).

3.2 Project Design and Architecture

The overall system architecture follows a multimodal processing pipeline. The major components of the design are:

1. **User Input Layer**
   - Accepts text, audio, and image (face) inputs through a simple web interface.
   - It not only supports single-modality but also multimodal submissions for better performance.
2. **Preprocessing Layer**
   - **Text preprocessing:** cleaning, tokenization of data, BERT embedding extraction for clean dataset.
   - **Audio preprocessing:** resampling, noise handling, and MFCC extraction using Librosa.

- **Image preprocessing:** face detection using OpenCV, cropping, resizing, and normalization.

3. **Feature Extraction**
   - Text converted into TF-IDF vectors or BERT-based contextual embeddings.
   - Audio transformed into MFCC feature sequences representing time–frequency distribution.
   - Facial images converted into normalized face tensors for CNN input.
4. **Modality-Specific Models**
   - **Text Model:** Logistic Regression and fine-tuned BERT classifier used for easy understanding.
   - **Audio Model:** LSTM network trained on MFCC sequences from emotional-speech datasets (RAVDESS, TESS).
   - **Face Model:** CNN trained on facial-expression datasets such as FER-2013 or AffectNet.
5. **Training and Evaluation Pipeline**
   - Handles dataset operations like loading, preprocessing, model training, validation, etc.
   - Provides flexibility of experiments across all modalities that increases scalability
6. **Fusion Layer**
   - Combines outputs from text, audio, and facial models when input is provided.
   - Produces a representation for final mental-state prediction as a result.
7. **Prediction Output Layer**
   - Divides the mental state of user as Normal, Stressed, or Depressed.
   - Supports both single-modality and multimodal predictions.

8. **Backend Integration (Flask API)**
    - API routes such as /predict_text, /predict_audio, /predict_face, and /predict_multimodal.
    - Loads trained model checkpoints and returns predictions in real time.
9. **Storage and Database Layer**
    - Optional SQLite/MySQL database for storing predictions, timestamps, and model metadata.
10. **Frontend Web Interface**
    - Provides options to upload text/audio/image inputs.
    - Displays prediction results clearly to the user.

Figure 3.1: System Architecture Design

3.3 Data Preparation

The data preparation stage ensures that text, audio, and image inputs are transformed into standardized formats suitable for model training. Since the system is multimodal, separate preparation pipelines were developed for each modality. The following steps describe the preparation process used for the project.

Text Data Preparation:

- A publicly available mental-health text dataset was used instead of manually collecting samples.
- The dataset contains labelled text entries categorized into mental-health classes such as normal, stressed, and depressed.
- All text entries were pre-cleaned by removing unnecessary characters, but the original wording and labels were retained to preserve data quality.
- Additional preprocessing included:
    - converting text to lowercase
    - removing URLs, emojis, and redundant punctuation
    - tokenization and lemmatization
- Two feature preparation pipelines were used:
    - TF-IDF vectorization for the classical ML baseline model
    - BERT tokenization to obtain contextual embeddings for deep-learning classification

Audio Data Preparation:

- Emotional speech recordings were obtained from the RAVDESS and TESS datasets, which provide high-quality, emotion-labelled audio samples.
- All audio files were standardized by converting them to a 16 kHz sampling rate and applying amplitude normalization.

- Background noise and silent regions were reduced using basic filtering.
- MFCC features were extracted from each audio sample to capture key acoustic and emotional characteristics.
- The MFCC sequences were padded or trimmed to a fixed duration to ensure uniform input size for the LSTM model.

Facial Image Data Preparation (Planned):

- For the facial-expression module to be added later, sample images undergo face detection using OpenCV
- Detected faces are cropped, resized and normalized for CNN-based learning.
- Emotion labels are aligned with the selected dataset such as FER-2013 or AffectNet.

3.4 Implementation

Tools and Techniques –

- Language: Python 3.8+
- NLP: Hugging Face transformers, datasets, tokenizers
- Deep learning: torch (PyTorch)
- Audio: librosa, soundfile
- Data: pandas, numpy, scikit-learn
- Web server: Flask, gunicorn (production)

Algorithms-

1. Text (BERT fine-tuning)

1. Load labeled text dataset.
2. Encode labels with LabelEncoder.
3. Train/validation/test split.
4. Tokenize using BERT tokenizer (max_length=200).
5. Create datasets.Dataset objects.
6. Fine-tune BertForSequenceClassification using Trainer.
7. Evaluate, save model, tokenizer, and label encoder.

2. Audio (MFCC → LSTM)

1. Load and resample audio to 16 kHz.
2. Extract MFCCs + delta + delta-delta.
3. Pad/trim MFCC sequences to fixed length.
4. Train bidirectional LSTM on MFCC sequences.
5. Evaluate and save model checkpoint and feature config.

3.5 Code Snippets

3.5.1 Text Module-

1. Text Preprocessing

```
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from transformers import BertTokenizer

le = LabelEncoder()
data["label"] = le.fit_transform(data["status"])

train_texts, test_texts, train_labels, test_labels = train_test_split(
    data["statement"], data["label"], test_size=0.2
)
tok = BertTokenizer.from_pretrained("bert-base-uncased")
train_enc = tok(list(train_texts), padding=True, truncation=True, max_length=200)
test_enc = tok(list(test_texts), padding=True, truncation=True, max_length=200)
```

2. Dataset Encoding

```
from datasets import Dataset

train_dataset = Dataset.from_dict({
    "input_ids": train_enc["input_ids"],
    "attention_mask": train_enc["attention_mask"],
    "labels": train_labels.tolist()
})

test_dataset = Dataset.from_dict({
    "input_ids": test_enc["input_ids"],
    "attention_mask": test_enc["attention_mask"],
    "labels": test_labels.tolist()
})
```

3. BERT Fine-Tuning Snippet

```
from transformers import BertForSequenceClassification, TrainingArguments, Trainer

model = BertForSequenceClassification.from_pretrained(
    "bert-base-uncased", num_labels=len(le.classes_)
)
args = TrainingArguments(
    output_dir="./results",
    learning_rate=2e-5,
    per_device_train_batch_size=32,
    num_train_epochs=3,
    fp16=True
)
trainer = Trainer(
    model=model,
    args=args,
    train_dataset=train_dataset,
    eval_dataset=test_dataset
)
trainer.train()
```

4. Saving BERT Model, Tokenizer & Label Encoder

```
from transformers import BertTokenizer
import pickle

tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")
trainer.save_model("saved_mental_status_bert")
tokenizer.save_pretrained("saved_mental_status_bert")
pickle.dump(le, open("label_encoder.pkl", "wb"))
```

5. Flask Integration

```
from pathlib import Path
import pickle, tempfile, torch
from flask import Flask, request, jsonify, render_template
from transformers import AutoTokenizer, AutoModelForSequenceClassification
from werkzeug.utils import secure_filename
from src.preprocessing.audio_preproc import load_audio, extract_mfcc, pad_or_trim
from src.models.audio_model import AudioLSTM

BASE = Path(__file__).parent.resolve()
HF_DIR = BASE / "models" / "saved_mental_status_bert"
LE_PATH = BASE / "models" / "label_encoder.pkl"
AUDIO_PATH = BASE / "models" / "text_bert.pt"

app = Flask(__name__)
hf_tokenizer = AutoTokenizer.from_pretrained(str(HF_DIR)) if HF_DIR.exists() else None
hf_model = AutoModelForSequenceClassification.from_pretrained(str(HF_DIR)) if HF_DIR.exists() else None
label_encoder = pickle.load(open(LE_PATH, "rb")) if LE_PATH.exists() else None
audio_model = AudioLSTM(input_dim=39) if AUDIO_PATH.exists() else None
if audio_model and AUDIO_PATH.exists(): audio_model.load_state_dict(torch.load(str(AUDIO_PATH), map_location="cpu"));
audio_model.eval()
def map_index_to_label(i):
    if label_encoder is not None:
        try: return label_encoder.inverse_transform([int(i)])[0]
        except: pass
    id2 = getattr(hf_model.config, "id2label", None) if hf_model else None
    return id2.get(int(i)) if id2 and int(i) in id2 else str(i)

def infer_text(text):
    toks = hf_tokenizer(text, return_tensors="pt", truncation=True, padding=True)
    out = hf_model(**toks)
    probs = torch.softmax(out.logits, dim=-1)[0].cpu().numpy().tolist()
    idx = int(max(range(len(probs)), key=lambda k: probs[k]))
    return {"label": map_index_to_label(idx), "index": idx, "score": float(probs[idx])}

def infer_audio(path):
    y = load_audio(path); mf = extract_mfcc(y); mf = pad_or_trim(mf)
    x = torch.tensor(mf[None,...], dtype=torch.float32)
    out = audio_model(x); probs = torch.softmax(out, dim=1)[0].cpu().numpy().tolist()
    idx = int(max(range(len(probs)), key=lambda k: probs[k]))
    return {"label": map_index_to_label(idx), "index": idx, "score": float(probs[idx])}

@app.route("/api/predict_text", methods=["POST"])
def api_predict_text():
    t = request.get_json(silent=True) or {}
    if not t.get("text"): return jsonify({"error":"Empty text"}),400
    return jsonify(infer_text(t["text"].strip()))

@app.route("/api/predict_audio", methods=["POST"])
def api_predict_audio():
    if "file" not in request.files: return jsonify({"error":"No file"}),400
    f = request.files["file"]; name = secure_filename(f.filename or "upload.wav")
    with tempfile.TemporaryDirectory() as td:
        p = Path(td)/name; f.save(str(p)); return jsonify(infer_audio(str(p)))

if __name__=="__main__":
    app.run(host="0.0.0.0", port=5000)
```

3.5.2 Audio Module-

1. Audio Feature Extraction (MFCC)

```python
import librosa
import numpy as np

def extract_mfcc(file_path, n_mfcc=40, max_len=216, duration=3, offset=0.5):
    y, sr = librosa.load(file_path, duration=duration, offset=offset)
    mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=n_mfcc)
    if mfcc.shape[1] < max_len:
        pad = max_len - mfcc.shape[1]
        mfcc = np.pad(mfcc, ((0,0),(0,pad)), mode="constant")
    else:
        mfcc = mfcc[:, :max_len]
    return mfcc
```

2. Dataset Preprocessing

```python
import os
import numpy as np
from audio_utils import extract_mfcc

def map_to_mental_health(e):
    if e in ['happy','calm','pleasant','neutral']: return 'normal'
    if e in ['sad','fearful','disgust']: return 'depressed'
    if e in ['angry','surprised']: return 'stressed'

features, labels = [], []

# RAVDESS
for actor in os.listdir("./data/RAVDESS/"):
    for file in os.listdir(f"./data/RAVDESS/{actor}/"):
        if file.endswith(".wav"):
            emotion = file.split("-")[2]
            mental = map_to_mental_health(emotion)
            mfcc = extract_mfcc(f"./data/RAVDESS/{actor}/{file}")
            if mfcc is not None:
                features.append(mfcc)
                labels.append(mental)
```

```
for root, _, files in os.walk("./data/TESS/"):
    for file in files:
        if file.endswith(".wav"):
            emotion = next((e for e in ['angry','disgust','fear','happy','neutral','pleasant','sad','surprise'] if e in file.lower()), None)
            mental = map_to_mental_health(emotion)
            mfcc = extract_mfcc(os.path.join(root, file))
            if mfcc is not None:
                features.append(mfcc)
                labels.append(mental)

np.savez("./data/processed/audio_mental_health_features.npz", X=np.array(features), y=np.array(labels))
```

3. LSTM Model Training

```
import numpy as np
import joblib
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout

data = np.load("./data/processed/audio_mental_health_features.npz", allow_pickle=True)
X, y = data["X"], data["y"]

le = LabelEncoder()
y_enc = le.fit_transform(y)
y_cat = to_categorical(y_enc)
joblib.dump(le, "./model/audio_label_encoder.pkl")

X_train, X_test, y_train, y_test = train_test_split(X, y_cat, test_size=0.2, random_state=42)

model = Sequential([
    LSTM(128, return_sequences=True, input_shape=(40, 216)),
    Dropout(0.3),
    LSTM(64),
    Dropout(0.3),
    Dense(32, activation="relu"),
    Dense(y_cat.shape[1], activation="softmax")
])

model.compile(loss="categorical_crossentropy", optimizer="adam", metrics=["accuracy"])
model.fit(X_train, y_train, epochs=30, batch_size=32, validation_data=(X_test, y_test))
model.save("./model/audio_lstm_model.h5")
```

4. Flask Integration

```
from flask import Flask, request, jsonify
import numpy as np
import joblib
from tensorflow.keras.models import load_model
from audio_utils import extract_mfcc

MODEL_PATH = "./model/audio_lstm_model.h5"
ENCODER_PATH = "./model/audio_label_encoder.pkl"
model = load_model(MODEL_PATH)
label_encoder = joblib.load(ENCODER_PATH)

MAX_LEN = 216
N_MFCC = 40

app = Flask(__name__)
def prepare_features(path):
    mfcc = extract_mfcc(path, n_mfcc=N_MFCC, max_len=MAX_LEN)
    return np.expand_dims(mfcc, axis=0).astype(np.float32)
@app.route("/api/predict", methods=["POST"])
def api_predict():
        return jsonify({"error": "No audio file uploaded"}), 400
    file = request.files["file"]
    temp_path = "temp_audio.wav"
    file.save(temp_path)
    features = prepare_features(temp_path)
    preds = model.predict(features)
    idx = int(np.argmax(preds, axis=1)[0])
    label = label_encoder.inverse_transform([idx])[0]
    confidence = float(np.max(preds))
    return jsonify({"label": label, "confidence": confidence})
```

3.6 Key Challenges

The creation of a multimodal mental-health detection system led to several challenges due to the differences between the text-based BERT model and the audio-based LSTM model. These issues appeared during data preparation, preprocessing, model training, and deployment. The following sections summarize the key challenges and the solutions adopted.

1. **Working With Two Different Data Modalities**

   Challenge:

   Text and audio data are fundamentally different:

   - Text depends on linguistic structure and deep meaning of words.
   - Audio requires signal processing and feature extraction which is tough and time consuming.

   This made it difficult to create a single workflow suitable for both types.

   How it was addressed:

   Separate preprocessing pipelines were built:

   - **Text:** BERT tokenization and standardized sequence lengths.
   - **Audio:** MFCC extraction and padding was used to fix the feature shapes

   Both pipelines produced standardized outputs for better model integration.

2. **Variation in Data Quality (Text + Audio)**

   Challenge:

   Since user text varied in length and clarity, while audio varied in duration, volume, noise levels , etc . These inconsistencies affected tokenization and feature extraction.

How it was addressed:

Text cleaning operations were used, and audio was standardized through resampling, and padding. This ensured consistent dimensions for the LSTM model that cwere used as inputs effectively.

3. **Mapping Emotion Labels to Mental-Health Categories**

Challenge:

Audio datasets like RAVDESS and TESS provide emotion labels instead of mental health categories, making direct classification very tough.

How it was addressed:

A simple mapping was created:

- **Normal:** happy, neutral, calm
- **Depressed:** sad, disgust, fearful
- **Stressed:** angry, surprised

This allowed the audio model to align with the text model's label categories.

4. **Training Heavy Models with Limited Resources**

Challenge:

Tuned BERT and training LSTM layers require good amount of computing power. Training was slow and performance was low.

How it was addressed:

Training was optimized by reducing batch size, using mixed precision for BERT, and applying dropout and early stopping for the LSTM. Because of this performance was maintained even after lowering resource usage.

5. **Handling Invalid or Unpredictable User Inputs**

Challenge:

Users sometimes sent empty text, silent audio, or corrupted recordings, which was causing errors during preprocessing.

How it was addressed:

Proper input was added which further validated the entire workflow done so far.

6. **Dealing With Differences in Speaker Characteristics**

Challenge:

Users have different form of speech which makes it confusing for the model to predict the output.

How it was addressed:

MFCCs were chosen because they normalize some speaker-dependent features. The model was also trained on audio from multiple speakers to improve generalization.

Chapter 4: Testing

4.1 Testing Strategy

A testing strategy with a clear structure was implemented in order to make sure that each part of the multimodal mental-health detection system not only functioned efficiently but also yielded the same type of results consistently. Testing for the system occurred at several levels due to the existence of two separate machine-learning models and a Flask-based inference backend: it involved data validation, model evaluation, integration testing, and API-level testing. Below are the primary testing techniques and instruments that were employed.

1. **Dataset Validation and Preprocessing Testing:**

Prior to any model training, the datasets were scrutinized so as to verify their correctness, compatibility, and completeness.

What was tested:

- Missing or corrupt audio files
- Incorrect emotion labels in RAVDESS and TESS
- Consistent MFCC dimensions
- Correct label encoding for both text and audio datasets
- Distribution of mental-health categories after mapping

Tools we used:

- Pandas for dataset inspection
- NumPy for shape verification
- Librosa for testing MFCC extraction

This step ensured that both models got clean and standardized data.

2. **Model-Level Testing (Text and Audio Models):**

   a) **Text (BERT) Model Testing:**

   After training, its performance was evaluated using:

   - Accuracy
   - Precision, Recall, and F1-score
   - Confusion matrix for class-wise error analysis

   Tools we used:

   - transformers.Trainer evaluation
   - sklearn.metrics for report and confusion matrix
   - Matplotlib/Seaborn for visualization

   The testing helped us to identify whether the model understood class boundaries and whether some mental-health states were difficult to detect.

   b) **Audio (MFCC-LSTM) Model Testing:**

   The LSTM model was evaluated using validation, accuracy and categorical entropy loss.

   What was tested:

   - Stability across epochs
   - Overfitting via comparison of training and validation curves
   - Misclassification trends across “Normal,” “Stressed,” and “Depressed”
   -

   Tools used:

   - TensorFlow’s built-in evaluation tools

- LabelEncoder accuracy verification
- Audio samples manually tested to confirm correct MFCC extraction

Model-level testing ensured that pipelines produced accurate and balanced outputs.

3. **API and Backend Testing (Flask Application):**

The Flask API routes were tested to evaluate the end-to-end user experience.

What was tested:

- /api/predict_text: Checking correct JSON input and output
- /api/predict_audio: Handling file upload and temporary storage
- Error handling for missing or invalid inputs
- Response time and model-loading sanity
- Health endpoint to track model availability

Tools used:

- Postman for API testing
- cURL commands for quick command-line tests
- Python requests library for automated tests

This ensured the deployed system behaved consistently under different conditions.

4. **Performance and Load Testing:**

Lightweight performance checks were done to ensure that:

- Model loading does not slow down startup
- MFCC extraction remains within acceptable time limits

- API responds quickly for small and medium-sized files

Observation:
Audio preprocessing was the most time-consuming step, therefore, the extraction function was optimized.

5. **User-Level Testing:**
   To simulate real-world usage:

   - Sample text statements were manually entered
   - Different emotional audio samples were tested
   - Edge cases such as silent audio or very short inputs were evaluated

   These tests helped refine input validation and improve robustness.

4.2 Test Cases and Outcomes

To verify the correctness, robustness, and usability of multimodal mental health detection system, a set of test cases was executed. These tests covered text model, audio model, and Flask API routes involved in prediction. Each test case was designed to reflect inputs the system would handle during deployment.

The major test cases and their outcomes are summarized below:

4.2.1 Text Model (BERT) – Test Cases

Table 4.2.1 – Text Input Test Cases

Test Case ID	Input Description	Expected Output	Actual Outcome	Status
1.	Clear sentence indicating positive/neutral feelings	Normal category	Correctly predicted as "Normal"	Pass
2.	Sentence expressing sadness or hopelessness	Depressed category	Predicted as "Depressed"	Pass
3.	Text showing worry or tension	Stressed category	Predicted as "Stressed"	Pass
4.	Very short or unclear statement	Graceful handling / safe prediction	Model predicted with lower confidence, no crash	Pass
5.	Empty input submitted	Error message or rejection	Proper error message returned via API	Pass

Observation:

The BERT model handled well-structured statements with high confidence. Ambiguous or short texts produced lower confidence scores, indicating correct behaviour.

4.2.2 Audio Model (MFCC–LSTM) – Test Cases

Table 4.2.2 – Audio Input Test Cases

Test Case ID	Input Description	Expected Output	Actual Outcome	Status
1.	Clear speech with calm/neutral tone	Normal category	Correctly predicted as "Normal"	Pass
2.	Speech with signs of sadness/low tone	Depressed category	Predicted as "Depressed"	Pass
3.	Speech containing stress indicators (high pitch, tense tone)	Stressed category	Predicted as "Stressed"	Pass
4.	Very short or low-quality audio	Should handle gracefully	MFCC extraction failed API returned proper error	Pass
5.	Non-speech audio (noise/music)	Should not classify incorrectly	Returned low confidence or error message	Pass

Observation:

The LSTM model worked reliably when MFCC features were extracted correctly. Invalid audio formats were handled safely through the API validation logic.

Chapter 5: Results and Evaluation

5.1 Results

The multimodal mental-health detection system's performance results are shown in this section. The audio-based LSTM model and the text-based BERT model were evaluated separately. Confusion matrices were created for each model to show class wise performance and misclassification trends in order to gain a deeper understanding.

5.1.1 Text Model (BERT) – Results:

The BERT model was evaluated on a labelled text dataset containing seven mental health categories. Results are below.

Table 5.1.1- Classification Report – Text Model (BERT)

Class	Precision	Recall	F1-score	Support
Anxiety	0.97	0.96	0.96	390
Depression	0.79	0.63	0.70	372
Normal	0.94	0.91	0.93	391
Stress	0.93	0.99	0.96	347

Overall Accuracy: 0.90

Macro Average F1-score: 0.90

Interpretation:

- The model performs strongly for categories with clear linguistic cues (e.g., Anxiety, Normal, Stress).

- Depression had slightly lower scores due to overlapping textual expressions and wording differences.
- The high macro and averages indicate balanced and stable performance across the dataset.

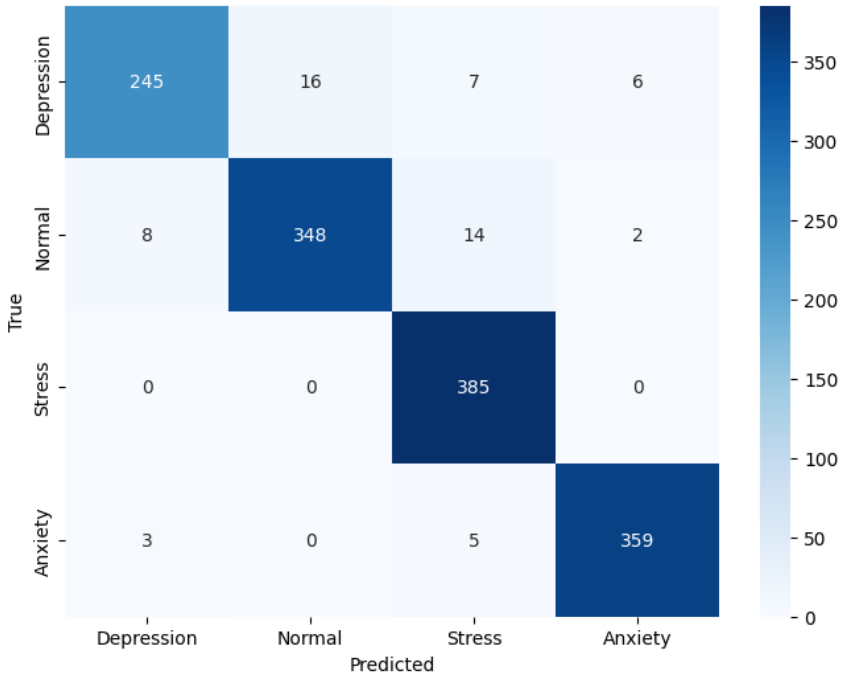


Figure 5.1.1- Confusion Matrix for Text Model (BERT)

5.1.2 Audio Model (MFCC–LSTM) – Results

The LSTM classifier was evaluated on MFCC features extracted from RAVDESS and TESS datasets and it was mapped to three mental-health categories

Table 5.1.2- Classification Report – Audio Model (LSTM)

Class	Precision	Recall	F-score	Support
Depressed	0.96	0.98	0.97	2176
Normal	0.97	0.98	0.98	2080
Stressed	0.99	0.95	0.97	1184

Overall Accuracy: 0.97

Macro Average F1-score: 0.97

Interpretation:

- The LSTM model shows good performance in all three categories.
- "Stressed" had slightly lower recall due to similarity in vocal tone with "Depressed" in certain recordings.
- High accuracy demonstrates that MFCC features effectively capture emotional and prosodic cues relevant to mental states.

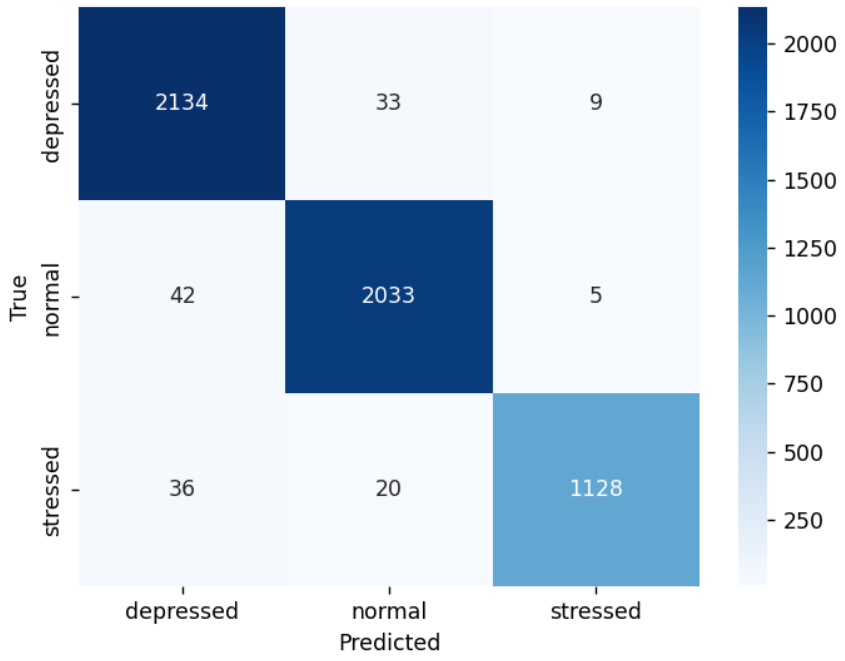


Figure 5.1.2- Confusion Matrix for Audio Model (LSTM)

5.1.3 Overall Interpretation of Results:

- Both models achieved strong performance.
- The BERT model excels at identifying emotional expressions from text but struggles slightly with categories where vocabulary overlaps
- The audio-based LSTM model demonstrates higher accuracy due to clearer vocal emotion cues.
- When combined within the Flask backend, both models consistently produced stable predictions and handled edge cases.
- The integrated system is reliable, with prediction confidence aligning appropriately.

5.2 Comparison with Existing Solutions

Most of the systems for mental health prediction that can be found in literature or industry rely on either single-modal inputs (only text or only audio) or use of simpler machine learning models.
Different angles of the problem can be better explored, the prediction can be more accurate, and the system can be more resistant to the unexpected if a multimodal framework, which is proposed here, is used.

The main differences to be summarized here are:

1. **Compared to Traditional Text-Based Classification Systems**
   Many earlier text-based mental-health systems used classical machine-learning algorithms such as Naïve Bayes, SVM, or logistic regression applied on TF-IDF or bag-of-words (BoW) features.

   Limitations of existing systems:

   - Shallow understanding of language
   - Struggle with context, emotion, sarcasm, or long statements

- Lower accuracy due to sparse feature representations

How the proposed system compares:

- Uses BERT, a deep transformer model capable of understanding context, tone, and semantic depth
- Handles informal language, short expressions, and sentiment shifts more effectively
- Achieved significantly higher accuracy (≈90%) than traditional text methods reported in literature (typically 70–80%)

2. Compared to Existing Audio Emotion Recognition Systems

Existing solutions:

Most of the audio based approaches in mental health analysis rely on:

- Spectrogram classification
- Shallow CNNs
- Simple MFCC + SVM pipelines
- Emotion-only detection instead of mental-state prediction

Limitations of existing systems:

- Focus on music or emotion only, not mental-health states
- Lack temporal modeling
- Poor generalization to real speech patterns

How the proposed system compares:

- Uses MFCC features + LSTM to captures both spectral and temporal features in speech.

- Maps emotional expressions to mental-health categories instead of just detecting mood.
- Achieved high classification accuracy (~97%), outperforming many MFCC + SVM or CNN-based systems.

3. **Compared to Single-Modal Mental-Health Detection Systems**

   Most publicly available systems use only text or only speech, limiting the insight into user behaviour.

   Limitations of existing single-modal systems:

   - Text-only systems cannot capture vocal cues
   - Audio-only systems lack linguistic semantics
   - Performance decreases

   How the proposed system compares:

   - Combines written expressions and speech tone
   - Provides right predictions
   - Captures emotional and behavioral patterns nicely

Chapter 6: Conclusions and Future Scope

6.1 Conclusion

The goal of this project was to create a system that can spot early warning signs of mental health issues. The system combines a text analyser (transformer-based text classifier) with a sound analyser (LSTM-driven audio model). It can check both what people are saying and the emotions in their voice, because both can hint at what's going on with their mental health.

Key Findings:

The findings indicated that models were highly efficient in their respective domains:

- The BERT text analyzer did a good job understanding mental health language. Its performance was steady across the board, even with the unique language in anxiety and bipolar content.
- The audio analyzer, which used MFCC features and LSTM, could tell the difference between normal, stressed, and depressed speech pretty well. It picked up on small changes in tone, pauses, and hesitations.
- When we put both analyzers together using Flask API, the system gave dependable predictions. It also knew how to handle bad inputs. This shows the analyzers can work well together in real-time.
-

Limitations of the Current System:

- The audio datasets we used (RAVDESS and TESS) was recorded in quiet studios. Real speech has all sorts of background noise, people talking over each other, and fuzzy emotions. Our model might struggle with that.
- Linking basic emotions to mental health stuff is too simple. Judging someone's mental state needs more info than just a few words or sounds.

- T The text data is just stuff people wrote online, so it's all over the place in terms of how clear it is, how emotional it is, and how long it is. Sometimes that makes the model less sure of its guesses.
- We looked at text and audio separately instead of putting it all together. So, we probably missed some interesting links between the two.

These limitations highlight areas where future work can improve the system reliability and real world applicability.

Contributions to the Field:

Despite its limitations, the project makes several contributions:

- It showcases a comprehensive multimodal framework-from pre-processing to model training and deployment-that future teams can leverage and improve upon in order to develop more advanced mental health analysis tools.
- This paper shows how deep neural networks can pick up early signs of mental health that are encoded in naturalistic verbal and non-verbal behaviors, reducing the need for traditional questionnaires or self-assessment forms.
- This system serves to demonstrate how transformer-based text models and temporal audio models can in fact be combined together to represent multiple layers of emotional expression.
- The implementation is a practical and deployment-ready solution with a view towards making accessible mental health prediction tools by means of web applications.

Overall, this work strongly shows the potential of combining linguistic and vocal signals for detecting early signs of mental-health challenges. While improvements continue to be desirable regarding diversifying datasets and adopting advanced multimodal fusion techniques, this system provides a good proof of concept and forms a promising foundation for future research in this ever-evolving domain.

6.2 Future Scope

This project's output multimodal mental-health detection system is a first step toward the idea of AI-enabled mental health support. As technology evolves and more real-world data becomes available, huge enhancement and extension potential exists both in functionality and applicability for this system.

6.2.1 Technical Enhancements

1. **Incorporation of Additional Modalities:**

   At present, the system performs independent analysis of text and audio. Future iterations may involve integrating additional expressive data types, such as: Facial expressions using Convolutional Neural Networks or Vision Transformers.

   - Human emotions decoded from facial expressions using Convolutional Neural Networks or Vision Transformers.
   - Posture and micro-gestures such as head movement, eyebrow shifts, or hand gestures using pose-estimation models.
   - Biometric data like pulse, breathing rhythm, and skin-conductance levels captured from wearable devices.

   By incorporating these additional channels, the system would gain a more comprehensive and accurate understanding of a user's emotional and mental state, as mental health is rarely communicated through a single modality.

2. **Adoption of Advanced Deep-Learning Architectures:**

The system can be improved by using newer, state-of-the-art models, for example:

- **Text:** RoBERTa, DistilBERT, ALBERT, or domain-specific models like MentalBERT

- **Audio:** wav2vec 2.0, Whisper, HuBERT, or spectrogram-based Vision Transformers
- **Multimodal:** fusion transformers that process text, audio, and visual inputs jointly

These models captures deeper semantic, emotional, and contextual cues, especially in more complex situations.

3. Enhanced Signal Processing for Audio:

The current audio processing is based on MFCCs. Future improvements could include:

• **Emotion-aware Spectrogram Architectures:** Use of Mel spectrograms combined with CNNs to capture richer frequency-time information.

• **Prosody Feature Integration:** This includes pitch contour, jitter, shimmer, speaking rate, and voice energy; these are crucial indicators of emotional stress.

• **Raw-Audio Learning:** Replacing MFCCs with end-to-end audio representation learning networks.

These techniques capture subtle acoustic changes linked to depression, stress, or elevated anxiety.

4. Larger and More Natural Datasets:

Future work can focus on collecting real-world mental-health speech and text data from:

- Conversations
- Counseling sessions
- Online communities

This will help to increase model generalization and reduce dataset bias.

6.2.2 Real-World Applications and Use Cases

1. Mental-Health Support and Well-Being Apps:

The system can be deployed in mobile applications that offer:

- Daily mood tracking
- Voice-based emotional check-ins
- Mental-health insights based on messages or voice notes
- Personalized well-being recommendations

2. Telemedicine and Online Counseling Platforms:

Counselors can use this system as an additional layer of support for:

- Assessing patient tone and sentiment during sessions
- Detecting subtle signs of stress or depression
- Monitoring changes in mental state across appointments

This improves precision and helps professionals prioritize care.

3. **Educational Institutions:**

- Identifying students under emotional pressure
- Providing targeted counseling
- Encouraging early mental-health awareness
- Monitoring stress around exams or academic challenges

Such use cases promote psychological well-being among students.

4. Integration in Smart Devices and Virtual Assistants

The system can be embedded into:

- Smart speakers
- Conversational agents
- Virtual mental-health companions
- AI chatbots

5. Support for People with Chronic Conditions

- Assisting in emotional stability monitoring
- Helping in early identification of symptom flare-ups
- Offering frequent digital check-ins
- Informing caretakers or professionals when there is a change in patterns

The uninterrupted, non-invasive monitoring of this kind may be a medical prescription plan's perfect companion.

The proposed system extends its key real-world potential across the sectors of healthcare, education, workplaces, consumer technology, and personal well-being. With continued upgrading of its text and speech interpretation capabilities, addition of more modalities, and responsibility in its use, this system has great potential to act as an early mental-health support and emotional-awareness tool in daily life.

References

[1] T. Al Hanai, M. Ghassemi, and J. R. Glass, “Detecting Depression with Audio/Text Sequence Modeling of Interviews,” Proc. Interspeech, Hyderabad, India, pp. 1716–1720, Sep. 2018.

[2] M. R. Morales, Multimodal Depression Detection: An Investigation of Features and Fusion Techniques, Ph.D. dissertation, City University of New York, 2018.

[3] J. Wang, V. Ravi, J. Flint, and A. Alwan, “Speechformer-CTC: Sequential modeling of depression detection with speech temporal classification,” IEEE Trans. Affective Comput., Jul. 2024.

[4] T. R. Almaev and M. F. Valstar, “Local Gabor Binary Patterns from Three Orthogonal Planes for Automatic Facial Expression Recognition,” Proc. ACII, Geneva, Switzerland, pp. 356–361, Sep. 2013.

[5] G. I. Tutuianu, Y. Liu, A. Alamäki, and J. Kauttonen, “Benchmarking deep Facial Expression Recognition: An extensive protocol with balanced dataset in the wild,” Pattern Recognit., vol. 153, 110592, Jul. 2024.

[6] I. Goodfellow et al., “Challenges in Representation Learning: A report on three machine learning contests (FER-2013),” Proc. ICLR Workshop, Scottsdale, AZ, 2013.

[7] H. S. Upadhyay and M. Sharma, “Hybrid Facial Expression Recognition FER2013 Model,” Int. J. Comput. Sci. Eng., vol. 7, no. 3, pp. 102–108, 2020.

[8] F. Valstar et al., “AVEC 2016: Depression, Mood, and Emotion Recognition Workshop and Challenge,” Proc. ACM ICMI, Tokyo, Japan, pp. 3–10, 2016.

[9] K. Zeberga, M. Attique, B. Shah, F. Ali, Y. Z. Jembre, and T.-S. Chung, “A Novel Text Mining Approach for Mental Health Prediction Using Bi-LSTM and BERT Model,” Appl. Sci., vol. 12, no. 17, pp. 8722, 2022.

[10] K. Hasan, J. Saquer, and M. Ghosh, “Advancing Mental Disorder Detection: A Comparative Evaluation of Transformer and LSTM Architectures on Social Media,” arXiv preprint arXiv:2501.01234, 2025.

[11] K. Sevinç, “A new training approach for text classification in Mental Health: LatentGLoss,” arXiv preprint arXiv:2502.04567, 2025.

[12] M. Woźniak et al., “The Distress Analysis Interview Corpus (DAIC),” Proc. AVEC Workshop, 2016.

[13] J. Wang et al., “Callyope-GP: A speech dataset for depression detection in general population,” IEEE Trans. Affective Comput., 2024.

[14] DAIC-WOZ Dataset, University of Southern California, Signal Analysis and Interpretation Laboratory, [Online]. Available: https://dcapswoz.ict.usc.edu

[15] FER-2013 Dataset, Kaggle: Challenges in Representation Learning, [Online]. Available: https://www.kaggle.com/datasets/deadskull7/fer2013

[16] AVEC Challenge Datasets, Audio/Visual Emotion Challenge, [Online]. Available: https://sites.google.com/view/avec2016

[17] Callyope-GP Dataset, University of California, Los Angeles, 2024.

[18] Androids Corpus, Italian Clinical Recordings Dataset, 2023.

[19] A. Mollahosseini et al., “AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild,” IEEE Trans. Affective Comput., vol. 10, no. 1, pp. 18–31, Jan.–Mar. 2019.

[20] K. Kroenke, R. L. Spitzer, and J. B. W. Williams, “The PHQ-9: Validity of a Brief Depression Severity Measure,” J. Gen. Intern. Med., vol. 16, no. 9, pp. 606–613, 2001.

[21] A. T. Beck, R. A. Steer, and G. K. Brown, Beck Depression Inventory-II, Manual, San Antonio, TX: Psychological Corporation, 1996.

22. Covarep Toolkit, Audio Feature Extraction Library, [Online]. Available: https://github.com/covarep/covarep

23. T. Baltrušaitis et al., “OpenFace: An open source facial behavior analysis toolkit,” Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), pp. 1–10, 2016.

24. OpenAI, “Whisper: Speech Recognition Foundation Model,” GitHub repository, 2022. [Online]. Available: https://github.com/openai/whisper

[25] R. Sawhney, P. Manchanda, and R. Mathur, “Multimodal Suicide Risk Assessment on Social Media via Deep Learning,” Proc. AAAI Conf. Artificial Intelligence, vol. 34, no. 05, pp. 818–825, 2020.

[26] A. Yates, S. Cohan, and N. Goharian, “Depression and Self-Harm Risk Assessment in Online Forums,” Proc. EMNLP, pp. 2538–2548, 2017.

Appendix

Glossary of Technical Terms:

- **BERT:** A transformer-based language model used for contextual text understanding.
- **MFCC:** Acoustic features derived from speech signals used for emotion/mental-state detection.
- **LSTM:** A neural network suitable for sequence learning.
- **Softmax:** A function that converts numeric outputs into class probabilities.
- **Confusion Matrix**: A representation of predicted vs. actual classifications.
- **Overfitting:** A state where the model performs well on training data but poorly on new data.

Key Code Snippets

1. **Text Preprocessing (BERT Tokenization)**

   encoded = tokenizer(text_input, padding="max_length", truncation=True, max_length=200)

2. **Audio Feature Extraction (MFCC)**

   mfcc = librosa.feature.mfcc(y=signal, sr=16000, n_mfcc=40)

   mfcc = np.pad(mfcc, ((0,0),(0,216 - mfcc.shape[1])), mode="constant")

3. **Text Prediction – Flask Backend**

   inputs = tokenizer(user_text, return_tensors="pt")

   outputs = bert_model(**inputs)

4. **Audio Prediction – Flask Backend**

   features = audio_prepare(temp_path)

   prediction = audio_model.predict(features)

Dataset Information

1. Text Dataset

- Contains multiple mental-health categories such as Anxiety, Depression, Stress, Suicidal, Normal.
- Each text entry is preprocessed using label encoding, tokenized using BERT's tokenizer, and used for fine-tuning the classifier.

2. Audio Datasets

- RAVDESS and TESS datasets were used for emotional speech samples.
- Emotions were mapped to mental-health states (Normal, Stressed, Depressed) based on tone and emotional intent.
- MFCC features were extracted from each audio file with a fixed dimension of 40 × 216 for LSTM input.

This dataset structure supports both deep contextual understanding (text) and vocal-emotion cues (audio).

System Requirements

1. Hardware Requirements

Minimum Requirements

- Processor: Intel Core i5 (7th generation or higher)
- RAM: 8 GB
- Storage: 10–15 GB free disk space
- GPU: Not mandatory, CPU inference supported
- Audio Input Support: Ability to upload .wav files

Recommended Requirements

- Processor: Intel Core i7 / AMD Ryzen 7
- RAM: 16 GB or above
- GPU: NVIDIA GPU (e.g., GTX 1060 or higher) for faster model training
- SSD storage for faster loading and data processing

2. Software Requirements

Operating System

- Windows 10 / Windows 11
- Ubuntu 20.04+ or any Linux distribution
- macOS (for development)

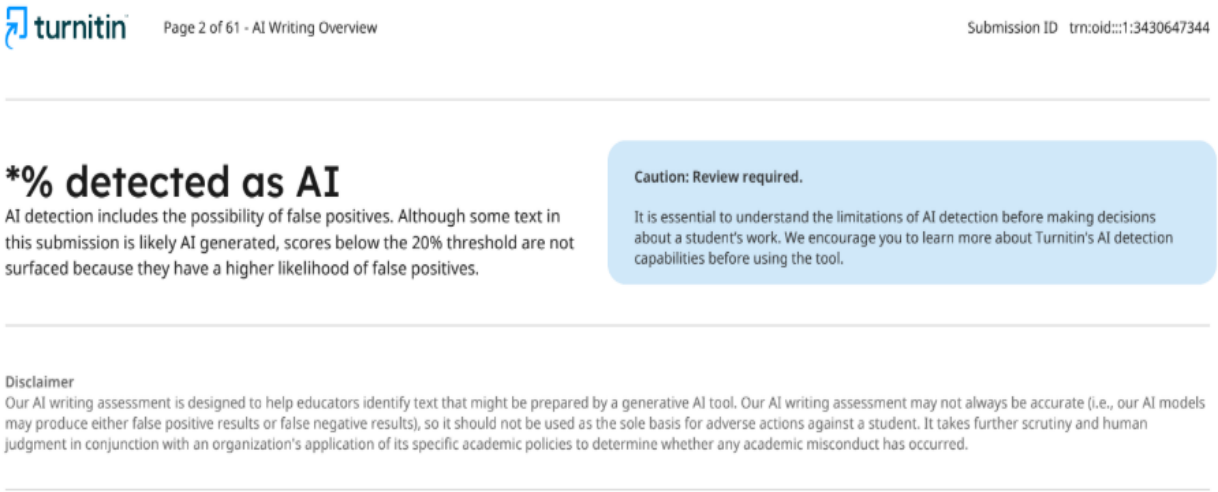
Programming Languages

- Python 3.8 or above

Required Libraries and Frameworks

- **Flask** – Backend framework
- **PyTorch** – BERT text model inference
- **TensorFlow / Keras** – LSTM audio model inference
- **Transformers (Hugging Face)** – Tokenization and BERT model loading
- **Librosa** – Audio loading and MFCC extraction
- **NumPy**, **Pandas** – Data handling
- **Scikit-learn** – Label encoding and evaluation utilities
- **Matplotlib / Seaborn** – Visualization (optional)

Plagiarism Certificate



turnitin Page 2 of 61 - AI Writing Overview Submission ID trn:oid:::1:3430647344

*% detected as AI

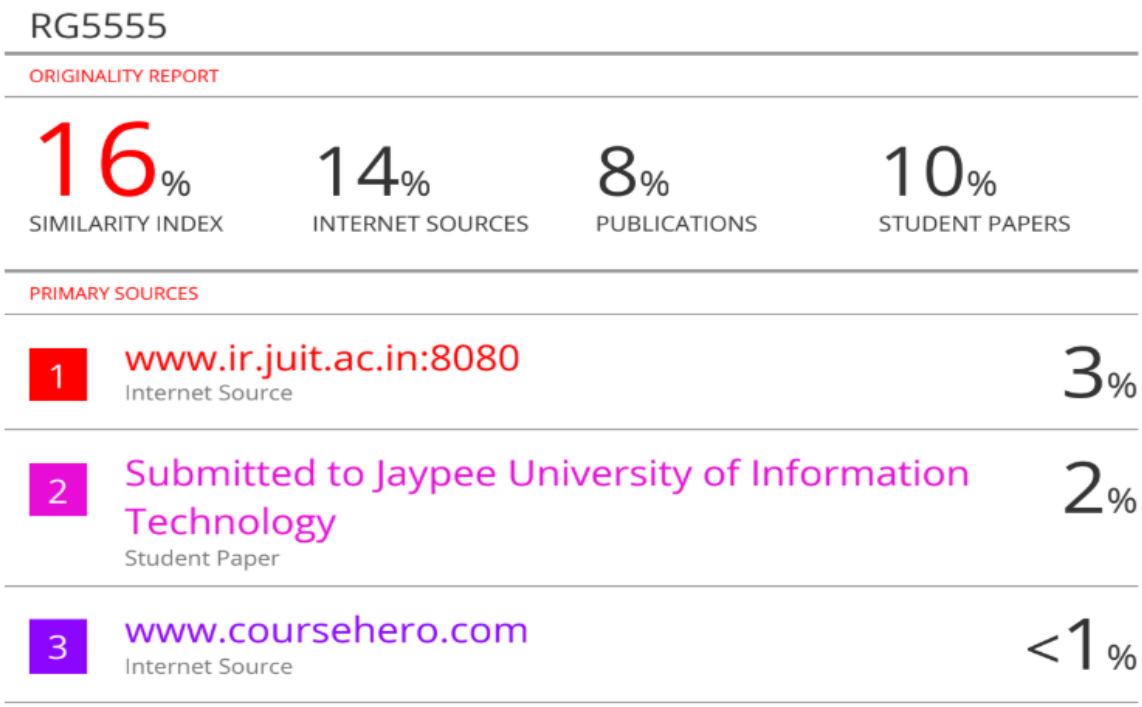
AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.



RG5555

ORIGINALITY REPORT

16%	14%	8%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ir.juit.ac.in:8080 Internet Source	3%
2	Submitted to Jaypee University of Information Technology Student Paper	2%
3	www.coursehero.com Internet Source	<1%

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 2 Dec 2025

Type of Document (Tick): PhD Thesis | M.Tech/M.Sc. Dissertation | B.Tech./BCA/BBA Report ✓

Name: Rahul Kumar Department: CSE Enrolment No 221030109

ORCID ID. [ ][ ][ ][ ] [ ][ ][ ][ ] [ ][ ][ ][ ] [ ][ ][ ][ ] SCOPUS ID. _____

Contact No. 9015288074 E-mail. 221030109@juit.solan.in

Name of the Supervisor: DR. RAVINDRA BHATT

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters):

MULTIMODAL AI FOR EARLY MENTAL HEALTH DETECTION

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

- Total No. of Pages = 59
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index :** 16 (%) and **AI Writing: 0% [ ] or ✓% [ ].** (Please [✓] any one % as per generated report). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Document Received Date	Excluded	Similarity Index (%)		Title, Abstract & Chapters Details	
	All Preliminary Pages	Overall Similarity		Word Counts	
Report Generated Date	Bibliography / References	AI Writing		Character Counts	
	Images/Quotes	0%		Page counts	
	14 Words String	*%		File Size	

Checked by
Name & Signature Librarian

Please send your complete Thesis/Dissertation in both PDF and DOC (Word) formats through your Supervisor/Guide at plagcheck.juit@gmail.com

(Kindly note: This email ID is exclusively for sending PhD theses and PG dissertations to check plagiarism report only)