AI-Powered Therapy Analytics for Mental Health using Sentiment Analysis

Onkar Anand Yemul
CSE Department
Walchand College of Engineering
Sangli, India
onkaryemul2003@gmail.com

Omkar Rajesh Auti
CSE Department
Walchand College of Engineering
Sangli, India
omkarauti11052001@gmail.com

Akash Rakesh Metkari
CSE Department
Walchand College of Engineering
Sangli, India
ametkari13042004@gmail.com

Anurag Vijay Takalkar
CSE Department
Walchand College of Engineering
Sangli, India
anuragtakalkar@gmail.com

Abstract*—Mental health is a critical determinant of an individual's overall well-being, influencing physical health, cognitive functioning, and social interactions. According to the World Health Organization (WHO), nearly 970 million people worldwide suffer from mental disorders, with depression and anxiety the most prevalent. Studies indicate that mental health disorders contribute to 14.3% of global deaths annually, either directly or through related physical health conditions. Furthermore, untreated mental health conditions can reduce life expectancy by 10 to 20 years, emphasizing the urgent need for early detection and intervention. This research leverages Natural Language Processing (NLP) and Sentiment Analysis to assess emotional states by analyzing Cognitive Behavioral Therapy (CBT) audio recordings and textual input. Research suggests that social media behavior and speech patterns can predict depressive symptoms with up to 80% accuracy, making linguistic and acoustic analysis a valuable tool for mental health assessment. Machine learning models process CBT session transcripts and audio features to extract emotional signals, helping people gain a deeper understanding of their mental well-being. To improve accessibility and user engagement, results are visualized on an interactive dashboard, providing real-time mental health insights based on audio and text analysis. Users can track emotional trends, monitor progress, and receive data-driven feedback to support self-improvement strategies. Despite technological advancements, privacy concerns remain a key challenge. Studies show that more than 60% people are concerned about the misuse of personal health data. To address this, the proposed system integrates end-to-end encryption, federated learning models, and anonymization techniques to ensure data security while allowing effective mental health assessments. This study aims to bridge the gap between mental health awareness and early detection by combining CBT-driven AI analytics with interactive visualization tools. By integrating machine learning with behavioral data analysis, this research contributes to personalized data-driven mental health interventions, ultimately fostering improved psychological well-being and quality of life.

***Keywords*—Sentiment Analysis, Mental Health, BERT, FL-BERT+DO, Deep Learning, Text Classification, CNN, LSTM, Federated Learning, Visualization Dashboards**

I. Introduction

Mental health has become a growing concern in today's fast-paced and digitally driven world, with increasing cases of stress, anxiety, depression, and other psychological disorders being reported globally. The growing volume of user-generated textual data on digital platforms provides an opportunity to explore linguistic patterns associated with mental health conditions. Sentiment analysis and emotion detection techniques, powered by Natural Language Processing (NLP), have emerged as effective tools for understanding psychological states through text data. Recent advances in deep learning and NLP have significantly improved the ability to classify and interpret emotional and psychological content. Models such as Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and transformer-based architectures like BERT (Bidirectional Encoder Representations from Transformers) have demonstrated enhanced performance in mental health classification tasks. Furthermore, hybrid models combining CNN and LSTM and optimized variants like Fine-Tuned BERT with Dropout Optimization (FL-BERT + DO) offer greater accuracy and contextual understanding. In addition to classification, the integration of interactive dashboards, AI-powered session analysis tools, and client progress visualization platforms has further enabled the practical application of these technologies in therapeutic environments. Ethical considerations such as data privacy, model bias mitigation, and fairness are also critical to ensure the responsible deployment of such systems in real-world mental health care settings.

1.1 Motivation

- **Rising Mental Health Challenges:** In today's fast-paced and digitally connected world, mental health issues such as depression, anxiety, stress, and suicidal thoughts are becoming increasingly common. This project is motivated by the need to proactively address these growing psychological concerns using technology.
- **Lack of Early Detection Systems:** Mental health conditions often go undiagnosed due to the lack of timely detection and professional support. This project aims to fill that gap by using machine learning and NLP techniques to identify early signs of mental health disorders through textual data.
- **Need for Technology-Driven Psychological Support:** Traditional therapy approaches alone are often not scalable. Using AI-powered classification models and interactive dashboards, our goal is to enhance the support systems available to both therapists and individuals seeking help.
- **Providing Meaningful Insights from Everyday Communication:** People often express their emotions through messages, social media posts, or journals. This project uses such data to uncover hidden emotional patterns and psychological conditions, thereby offering deeper insights into mental well-being.
- **Improving Clinical Decision-Making and Therapy Outcomes:** Through visual dashboards, session analysis modules, and automated documentation tools, the project supports mental health professionals in making informed decisions, monitoring progress, and enhancing therapy outcomes.
- **Bridging the Gap Between Data and Actionable Solutions:** A major problem today is the abundance of data related to mental health without actionable insights. This project bridges that gap by transforming raw textual data into meaningful classifications and visualizations that drive real-world interventions.
- **Ensuring Ethical and Inclusive Mental Health Solutions:** The motivation also lies in creating responsible AI systems that ensure data privacy, reduce bias in model predictions, and promote inclusivity in mental health care delivery.

Our proposed solution involves developing and analyzing an AI-driven mental health assessment system that utilizes audio recordings from Cognitive Behavioral Therapy (CBT) and text-based inputs. The system integrates Natural Language Processing (NLP) and Sentiment Analysis techniques to detect the user's emotional state accurately.

It offers psychological insights in real time through an interactive and visually intuitive dashboard, allowing people to monitor their mental well-being and therapy progress effectively. Furthermore, to ensure data security and user privacy, the system adopts Federated Learning and robust encryption mechanisms, enabling secure and confidential handling of sensitive mental health data.

The model continuously improves through adaptive learning, analyzing various emotional expressions and behavioral patterns. Ultimately, this solution aims to provide accessible, ethical and technology-driven support for early detection, emotional tracking, and informed mental health interventions.

II. Literature Survey

Recent advancements in Natural Language Processing (NLP) have significantly improved the accuracy and efficiency of sentiment analysis systems. Bello *et al.* [1] proposed a BERT-based sentiment analysis framework integrated with deep learning classifiers such as CNN, RNN, and BiLSTM, achieving state-of-the-art performance on tweet datasets. Their study demonstrated that the BERT transformer model outperforms traditional methods such as Word2Vec, CNN, and RNN due to its superior contextual word representation capabilities.

Similarly, Atandoh *et al.* [2] introduced an integrated deep learning paradigm called BMLCNN, which achieved remarkable accuracy across multiple datasets including IMDB and Amazon reviews. The model outperformed baseline architectures such as BERT and CNN, and statistical validation using the Friedman test confirmed its superior performance.

Zhang *et al.* [3] advanced this field further by combining BERT embedding with a sliced multi-head self-attention Bi-GRU model, enhancing both classification accuracy and training efficiency compared to conventional RNN-based approaches.

In another study, Aslan *et al.* [4] developed a TCNN–Bi-LSTM model using FastText embeddings to analyze sentiments of tweets related to COVID-19 vaccines. Their model significantly outperformed baseline deep learning and traditional machine learning techniques, highlighting the effectiveness of FastText over other embedding techniques like GloVe and TF-IDF in this context.

In the legal domain, Abimbola *et al.* [5] designed a CNN-LSTM hybrid model for sentiment classification in Canadian maritime case law, achieving an impressive accuracy of 98.05%. The CNN component extracted prominent features from the text, while the LSTM effectively captured temporal dependencies and contextual meaning.

A broader survey conducted by Rahman *et al.* [6] presented a state-of-the-art review on sentiment analysis, discussing various models, applications, and ongoing challenges, while emphasizing the importance of NLP in decision-making and customer behavior analysis.

Furthermore, Ahsan *et al.* [7] addressed privacy concerns in mental health sentiment analysis by proposing a FL-BERT+DO model, which integrates federated learning with data obfuscation techniques. Their framework not only provided high predictive performance but also ensured enhanced data privacy, outperforming baseline FL-DP methods in both accuracy and privacy protection.

III. Methodology

3.1 Dataset Description

3.1.1 ***Data Overview****:* The dataset is categorized into six mental health statuses:

- sadness
- anger
- love
- surprise
- fear
- happy

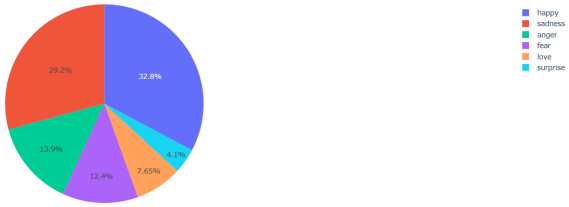

Fig. 1: Proportion of each emotion

The above figure 1 illustrates the distribution of emotion categories in the dataset. It highlights the imbalance in class proportions, with some emotions appearing more frequently than others. Such imbalance can impact model performance and may require techniques like resampling or weighted loss.

Each record in the dataset includes the following fields:

- **unique_id**: A unique identifier for each entry.
- **Text**: The textual content.
- **Emotion**: The corresponding label derived from sentiment and mental health annotations.

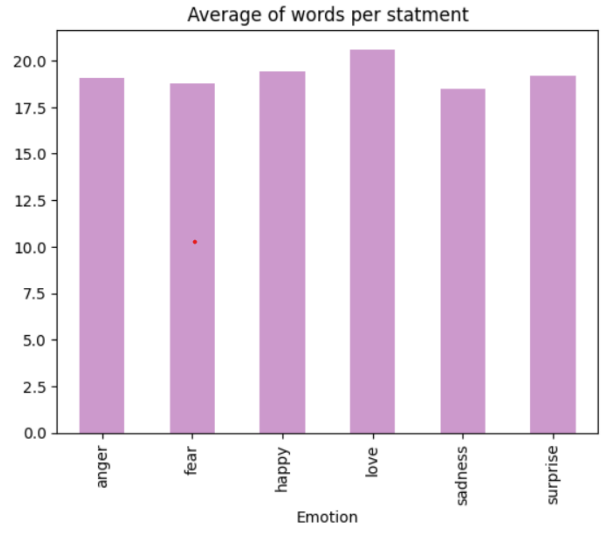

Fig. 2: Average of words per emotion

The figure 2 shows the average number of words per statement for each emotion category. It provides insight into how verbose expressions are across different emotions.

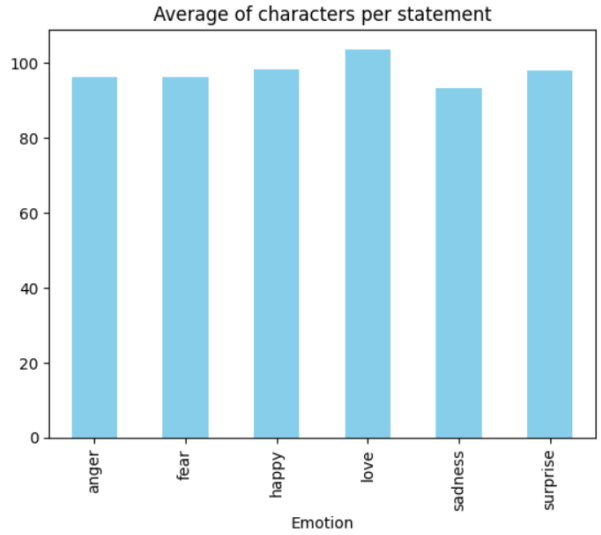

Fig. 3: Average of characters per emotion

The figure 3 shows the average number of characters per statement per emotion. This helps further analyze text length patterns, which can influence tokenization and model input size.

3.2 Data Preprocessing

To prepare the data for sentiment analysis, standard preprocessing techniques were employed using the Natural Language Toolkit (NLTK) . The following steps were implemented:

A. Text Cleaning

- **Lowercasing:** Conversion of all text to lowercase format.
- **Special Character and Punctuation Removal:** Elimination of all non-alphabetic characters.
- **Tokenization:** Segmentation of text into individual tokens or words.
- **Stopword Removal:** Removal of frequently occurring yet semantically insignificant words (e.g., “the”, “is”, “and”).

B. Text Normalization

- **Lemmatization:** Transformation of words into their base (dictionary) forms (e.g., “running” to “run”).
- **Stemming:** Reduction of words to their root forms using stemming algorithms (e.g., Porter Stemmer).

3.3 Word Frequency Analysis

A word frequency analysis was conducted post preprocessing to identify prominent terms within each mental health category using word clouds and term frequency counts. The analysis revealed the most commonly used terms across the categories:

- **Sadness:** feel, im, know, time, im feel, want, think.
- **Anger:** feel, im, time, want, im feel, peopl, think.
- **Love:** feel, love, im, support, realli, know, want.
- **Surprise:** feel, im, feel amaz, amaz, realli, time, look.

- **Fear:** feel, im, im feel, go, know, time, feel littl.
- **Happy:** feel, im, im feel, time, go, love, know.

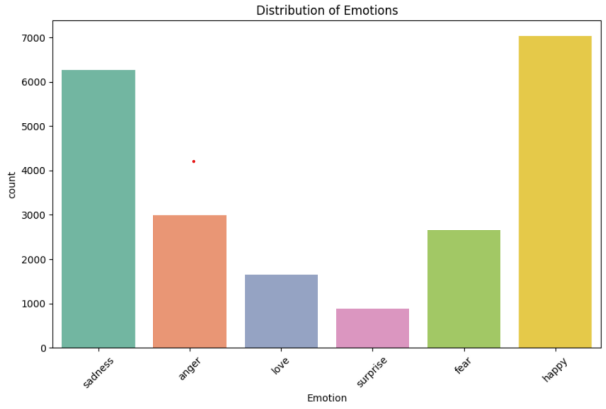

Fig. 4: Distribution of Status (Type vs Count)

This figure 4 presents the distribution of emotion types based on their occurrence count. It highlights how frequently each emotion appears in the dataset. Identifying dominant and underrepresented classes is essential for understanding class imbalance. This can inform preprocessing and model training strategies, such as class weighting or data augmentation.

Such word-level insights assist in identifying critical linguistic patterns associated with various mental health conditions.

Fig. 5: word cloud of emotions

This figure 5 displays a word cloud representing the most frequent words across emotion-labeled texts. Larger words indicate higher frequency, offering a quick visual summary of common vocabulary. It helps in understanding prominent terms associated with various emotions. Such insights can guide feature selection and preprocessing steps.

3.4 Algorithmic Methodology

1) Why Do We Require Algorithms for Sentiment Analysis?: Sentiment analysis involves the process of extracting opinions, emotions, and attitudes from textual data. However, human language is highly complex, with varying grammatical structures, contextual meanings, and emotional undertones. Manually analyzing large volumes of text is not only impractical but also prone to inaccuracies. Hence, machine learning and deep learning algorithms are employed to automate this process effectively.

Technique	Strengths	Limitations
Lexicon-Based Approach	Simple and does not need training data.	Limited to predefined words, lacks context.
Corpus-Based Approach	Expands vocabulary dynamically.	Requires large datasets for good accuracy.
Dictionary-Based Approach	Uses existing dictionaries for analysis.	Cannot handle slang or new words.
Machine Learning Approach	Learns patterns from data and adapts to different domains.	Needs large labeled datasets, computationally expensive.
Supervised Learning	High accuracy with labeled data.	Requires extensive labeled training data.
Unsupervised Learning	Works with unstructured data, no labeling needed.	Less precise, requires feature engineering.
Reinforcement Learning	Learns dynamically from interactions.	Complex and needs high computational power.
Naïve Bayes Classifier	Fast, scalable, and works well for text classification.	Assumes features are independent, which is often incorrect.
Support Vector Machine (SVM)	Handles high-dimensional data effectively.	Computationally expensive, slow for large data.
Neural Networks	Captures complex patterns, highly accurate.	Requires significant computing power and data.
Decision Tree Classifier	Easy to interpret and implement.	Prone to overfitting with noisy data.
Hybrid Approaches	Combines multiple methods for better accuracy.	Computationally expensive, difficult to tune.

TABLE I: Comparison of Sentiment Analysis Methods

This table 1 summarizes various sentiment analysis techniques along with their strengths and limitations. It includes both traditional and modern machine learning approaches. The comparison helps in understanding trade-offs between accuracy, computational cost, and data requirements. Useful for selecting an appropriate method based on dataset size, complexity, and resource constraints.

These algorithms are essential because:

- They learn hidden patterns from data automatically.
- They capture contextual sentiment in varying linguistic forms.
- They scale efficiently to massive datasets.
- They improve classification performance over time with increasing data.

Challenges in Sentiment Analysis:

- **Context Understanding:** Sentiment analysis models struggle to understand context, leading to misinterpretation.
- **Sarcasm and Irony:** Difficult for models to detect sarcasm, which can flip the meaning of a sentence.
- **Data Imbalance:** Some sentiments (e.g., positive reviews) may be more common than others, leading to

biased results.

- **Domain Dependence:** Models trained in one domain (e.g., movies) may not work well in another (e.g., finance).
- **Multilingual Sentiment Analysis:** Handling sentiment across different languages and dialects is complex.
- **Real-Time Processing:** Analyzing large-scale social media data in real time is computationally expensive.

Probable Solutions:

- **Context-Aware Models:** Use transformer-based models like BERT and GPT for better understanding of context.
- **Sarcasm Detection Models:** Incorporate advanced NLP techniques such as attention mechanisms and sarcasm-labeled datasets.
- **Data Augmentation:** Balance datasets using synthetic data generation to prevent bias.
- **Domain Adaptation:** Use transfer learning and fine-tuning to adapt models across domains.
- **Multilingual NLP Models:** Implement cross-lingual embeddings and translation models to handle multiple languages.
- **Big Data Tools:** Use Hadoop, Spark, and real-time sentiment classification algorithms for large-scale sentiment analysis.

Proposed Solution: Utilizing deep learning models such as Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), ensemble models, and Fine-Tuned BERT with Dropout Optimization (FL-BERT + DO) allows for capturing both local syntactic patterns and long-range dependencies in text while ensuring contextual awareness.

2) Algorithms and Their Mechanisms / Architecture with Sentiment Analysis:

A. Convolutional Neural Network (CNN)

A1. Why CNN is Used in Sentiment Analysis or Text Processing?: Originally designed for image recognition tasks, Convolutional Neural Networks (CNNs) have been effectively adapted for text and sentiment analysis. They are capable of extracting local features and patterns in sequential text data. CNNs identify significant phrases or n-gram features such as:

- *"Absolutely fantastic"*
- *"Not good"*
- *"Terribly boring"*

These patterns are critical in understanding sentiment even without analyzing the full sentence structure.

CNN Architecture – Flow Overview

CNNs follow a hierarchical layer-based architecture, where each layer contributes to transforming raw input into meaningful features for sentiment classification.

1) **Convolution Layer (Feature Extraction)**
   The primary function of this layer is to extract relevant features from the input text. A filter (kernel) slides over the input to identify patterns such as word combinations or phrases. For example, filters of size 2–3 can capture word patterns like *"very bad"*, *"good movie"*, etc.

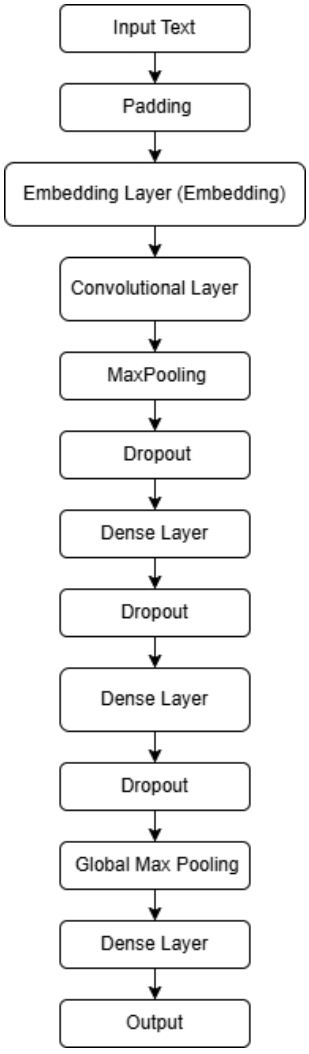

Fig. 6: The architecture of CNN model

This figure 6 illustrates the architecture of the Convolutional Neural Network (CNN) used for emotion classification. It typically includes embedding, convolutional, pooling, and fully connected layers. The model captures local features and spatial hierarchies in text data. Such architectures are effective for extracting emotion-related patterns from short texts.

2) **Stride**
   Stride determines how the filter moves across the input. A larger stride results in fewer features and faster processing but may miss fine-grained word patterns.
   Example:
   - **Stride = 1:** "very bad", "bad movie"
   - **Stride = 2:** "very bad", "movie was" (skips intermediate terms)

3) **Padding**
   Padding ensures that the size of the output feature map remains consistent with the input size. It helps in capturing edge-level words or phrases such as *"not*

good" occurring at the sentence boundaries.

4) **Pooling Layer (Downsampling)**
   Pooling reduces the dimensionality of the extracted features while retaining the most important ones.
   - **Max Pooling:** Captures the most prominent feature.
   - **Average Pooling:** Computes the average of features.

   This layer serves as a summarization step for key sentiment cues.
5) **Flatten Layer**
   This layer transforms 2D/3D feature maps into a one-dimensional vector, preparing the features for final classification.
6) **Fully Connected Layer (Classification)**
   This layer receives the flattened vector and makes sentiment predictions by connecting all learned features to the output neurons. It acts as the decision-making layer.
7) **Dropout Layer (Avoid Overfitting)**
   Randomly disables a subset of neurons during training, thereby reducing overfitting and improving model generalization.
8) **Activation Functions (Add Non-linearity)**
   These functions help the network model complex patterns by introducing non-linearity.
   - ReLU (Rectified Linear Unit): Fast and effective.
   - Sigmoid: Suitable for binary classification.
   - tanh: Similar to Sigmoid but centered at zero.
   - Softmax: Used for multi-class sentiment classification.

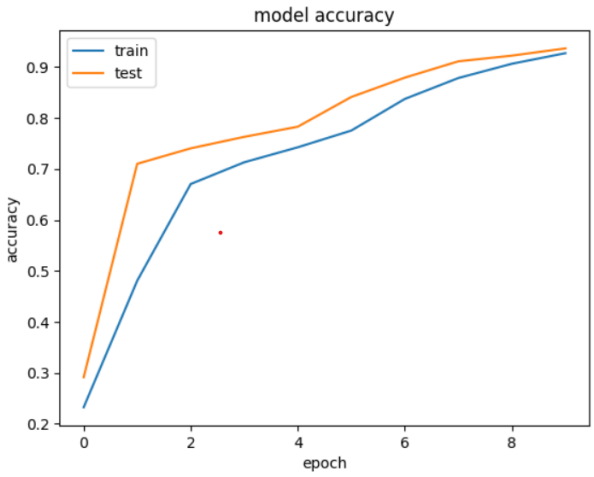

Fig. 7: CNN: Accuracy vs Epochs

This figure 7 shows the change in accuracy of the CNN model over training epochs. It helps visualize how well the model is learning from the data over time. A rising curve indicates improving performance, while plateaus or drops may signal overfitting or convergence. Useful for evaluating training progress and model effectiveness.

This figure 8 illustrates the change in loss values of the CNN model across training epochs. Decreasing loss indicates that the model is minimizing the error during training. A

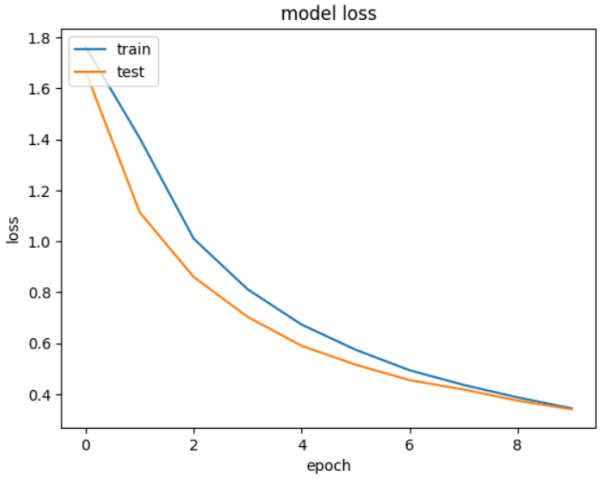

Fig. 8: CNN: Loss vs Epochs

steady decline suggests effective learning, while fluctuations may point to instability or overfitting. This plot is essential for monitoring model convergence and optimization quality.

Example Flow – Sentiment Classification Pipeline:

1) **Input Sentence:** "The movie was incredibly boring"
2) **Word Embedding:** Applied to convert words into vectors.
3) **Convolution Layer:** Identifies local features — *"incredibly boring"*.
4) **Pooling Layer:** Selects the strongest feature — *"boring"*.
5) **Flatten Layer:** Combines features into a single vector.
6) **Fully Connected Layer:** Makes final sentiment prediction — **Negative Sentiment**.

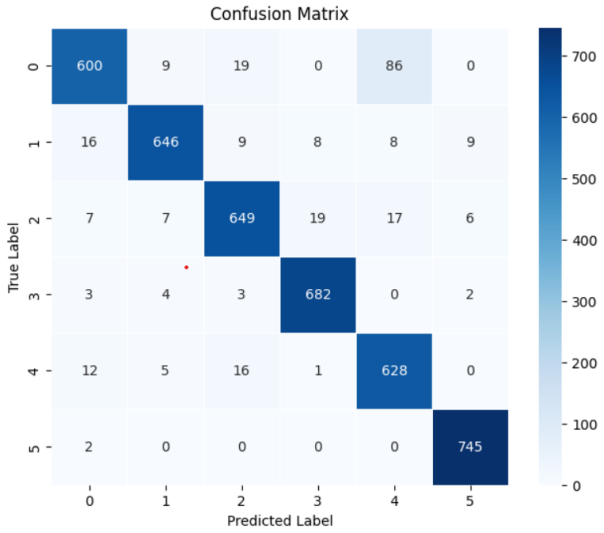

Fig. 9: CNN - Confusion Matrix

This figure 9 presents the confusion matrix for the CNN model, showing the true vs. predicted classifications. It helps

assess the performance of the model by revealing misclassifications across different emotion categories. Diagonal values represent correct predictions, while off-diagonal values indicate errors. The matrix provides valuable insights into specific class-wise performance and areas needing improvement.

A2. Benefits and Drawbacks of CNN in Sentiment Analysis:
Benefits:

- Captures local word patterns (n-grams).
- Faster and more efficient than RNNs.
- Simple for short sentence tasks.

Drawbacks:

- Poor at capturing long-range dependencies.
- Performs better when combined with RNN/LSTM/Attention.

B. Long Short-Term Memory (LSTM)

B1. Why LSTM is Required in Sentiment Analysis?: Sentiment in a sentence often depends on context and the sequential order of words, which traditional models (e.g., Bag-of-Words or simple classifiers) fail to capture. LSTM (Long Short-Term Memory), a special kind of Recurrent Neural Network (RNN), is designed to handle such sequences effectively. It understands the meaning of words in their context, enabling better sentiment prediction.

Examples:

- *"The movie was not good."* $\rightarrow$ Negative Sentiment
- *"The movie was really good."* $\rightarrow$ Positive Sentiment

LSTM networks retain relevant past information and forget irrelevant parts, making them ideal for textual data analysis where word order is crucial.

LSTM Architecture – Flow Overview

LSTM processes input sequentially, maintaining context information across time steps. It learns dependencies between words, even those far apart in the sentence.

1) **Input Layer:**
   The input sentence is first preprocessed and converted into word embeddings (using Word2Vec, GloVe, etc.).
2) **LSTM Layer:**
   Each word is passed sequentially through the LSTM units. The network maintains a memory cell that remembers important context from previous time steps.
3) **Gating Mechanisms:**
   - **Forget Gate:** Decides which information from the past should be discarded.
   - **Input Gate:** Decides what new information should be added to the memory.
   - **Output Gate:** Determines what information from the memory will be passed to the next layer.

   This figure 10 illustrates the architecture of the Long Short-Term Memory (LSTM) model used for emotion classification. LSTM networks are well-suited for sequential data, as they capture long-term dependencies and temporal patterns in text. The model typically consists of input, LSTM layers, and fully connected layers for classification. LSTMs are particularly effective in capturing context in sequential data, making them ideal for text-based tasks.
4) **Memory Cell:**
   Stores useful context and preserves long-term dependencies required for accurate sentiment analysis.
5) **Fully Connected (Dense) Layer:**
   After the LSTM processing, the final output from the sequence is passed through a dense layer to predict the sentiment label.
6) **Activation Functions:**
   - **Sigmoid:** Often used in binary sentiment classification.
   - **Softmax:** Used in multi-class classification (e.g., Positive, Negative, Neutral).

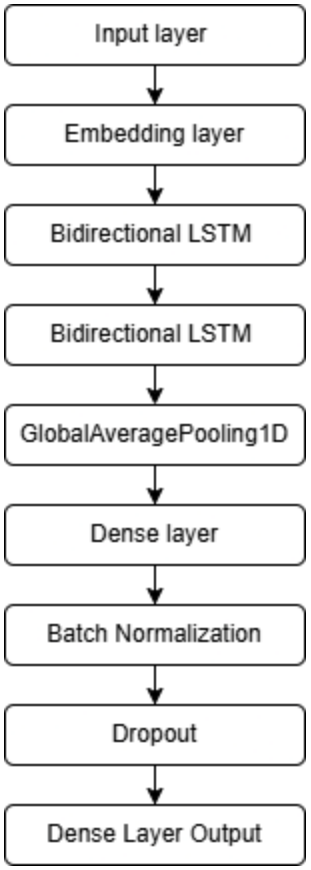

Fig. 10: The-architecture-of-LSTM-model

This figure 11 shows the loss values of the LSTM model over training epochs. A decreasing loss curve indicates that the model is improving its predictions over time. Steady decline in loss suggests good optimization, while fluctuations or plateaus could point to potential issues like overfitting. This plot helps assess the convergence and stability of the LSTM model during training.

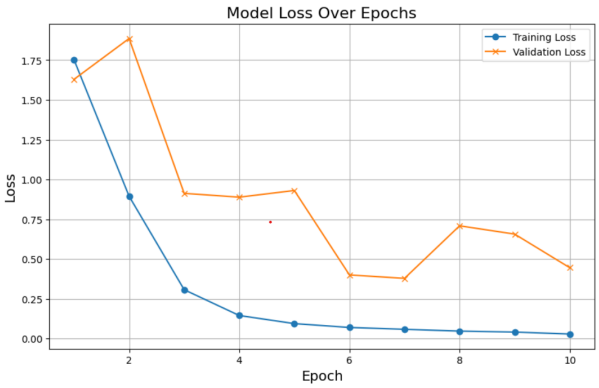

Fig. 11: LSTM LOSS vs Epochs

Example Flow – Sentiment Classification Pipeline:

1) **Input Sentence:** "The food was absolutely amazing"
2) **Text Preprocessing:** Cleaning and tokenizing the sentence.
3) **Word Embedding:** Words converted into numerical vectors.
4) **LSTM Layer:** Sequential word processing with memory tracking.
5) **Dense Layer:** Generates the final sentiment prediction — **Positive Sentiment**.

Figure: LSTM Architecture for Sentiment Analysis

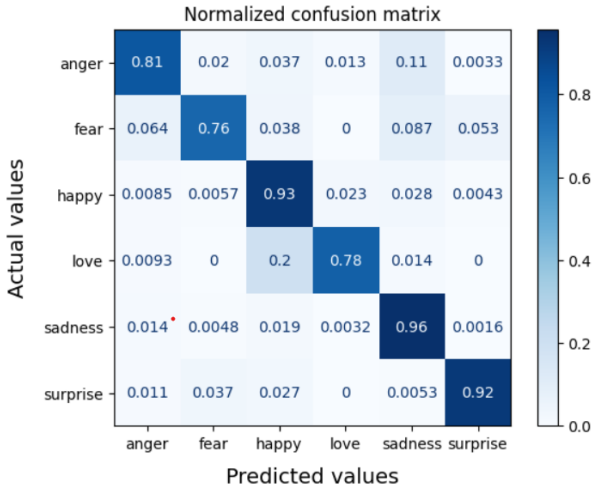

Fig. 12: LSTM Confusion Matrix

This figure 12 presents the confusion matrix for the LSTM model, showing the true vs. predicted classifications. The matrix helps assess the model's performance by visualizing errors across emotion categories. Diagonal elements represent correct predictions, while off-diagonal elements show misclassifications. It provides a clear view of how well the LSTM model is distinguishing between different emotions.

B2. Benefits and Drawbacks of LSTM in Sentiment Analysis:

Benefits of LSTM in Sentiment Analysis:

- Captures long-term dependencies and word order.
- Retains contextual meaning of words across time steps.
- Suitable for analyzing complex or subjective content.

Drawbacks:

- Higher training time compared to simpler models like CNNs.
- Requires more training data for optimal performance.
- May struggle with extremely long-range dependencies.

C. Ensemble of CNN and LSTM

C1. Why Combine CNN with LSTM?:

- CNN is great at finding local patterns or features (like n-grams).
- LSTM is great at capturing long-term dependencies or context.
- Together, they combine local and sequential understanding, making the model stronger for tasks like text sentiment analysis, emotion detection, or sequence analysis. Fig. 13 shows the architectural diagram of LSTM and CNN.

Architecture Flow:

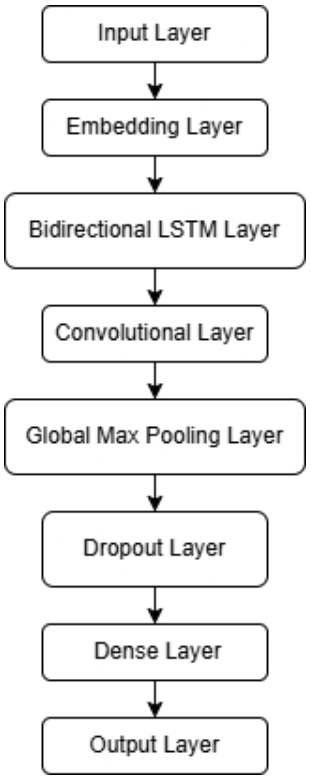

Fig. 13: Architectural Diagram of LSTM and CNN for Sentiment Analysis

This figure 13 presents a combined CNN and LSTM model architecture, showcasing how both convolutional and recurrent

layers are integrated. CNN layers capture local features from text, while LSTM layers capture long-term dependencies, making it effective for sequential data. This hybrid architecture benefits from both spatial and temporal feature extraction, offering a robust model for emotion classification.

1) **Input Sentence:**
   "The movie was surprisingly good"
2) **Word Embedding Layer:**
   Each word is converted into a vector (e.g., using Word2Vec, GloVe, or an Embedding Layer).
3) **CNN Layer (Feature Extractor):**
   - CNN detects key phrases/word patterns like "surprisingly good", "movie was".
   - Extracts spatial or local-level features using filters.
4) **Pooling Layer:**
   - Highlights the strongest features and reduces dimensionality.
5) **LSTM Layer (Context Understanding):**
   - CNN-extracted features are passed to LSTM.
   - LSTM captures sequential meaning and context (e.g., "surprisingly" affects "good").
6) **Flatten Layer:**
   Flattens the features to prepare for output processing.
7) **Fully Connected Layer:**
   Combines all extracted features and forwards them to the output layer.
8) **Activation Layer (SoftMax/Sigmoid):**
   Final prediction — Positive / Negative / Neutral sentiment.

Why It Works Well:

- **CNN:** Local feature detector.
- **LSTM:** Sequential context handler.
- **Together:** Effective for real-world sentences where both local phrases and sentence flow matter.

Example:

- Input Sentence: *"The movie wasn't that bad"*
- CNN detects: *"wasn't that"* → signals negation.
- LSTM interprets: *"bad" is negated* → Predicts **Positive**.

C2. Benefits and Drawbacks of Combining CNN and LSTM:

Benefits of Combining CNN and LSTM:

- Performs better than using CNN or LSTM alone.
- Captures both phrase-level (local) and sentence-level (sequential) features.
- Highly effective in sentiment analysis, emotion detection, and review classification tasks.

This figure 14 shows the loss values for the ensemble model combining CNN and LSTM over training epochs. The plot indicates how the hybrid model's loss decreases during training, reflecting its learning progress. A consistent reduction in loss suggests that the ensemble model benefits from both

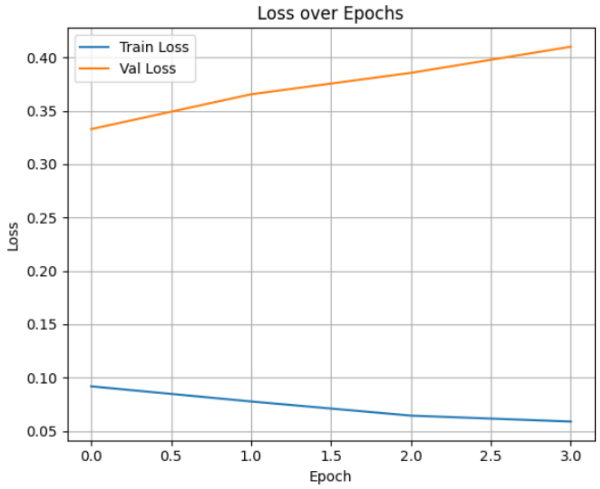

Fig. 14: Ensemble CNN + LSTM (LOSS vs Epoch)

CNN's spatial features and LSTM's temporal dependencies. This figure provides insights into the optimization and convergence behavior of the combined architecture.

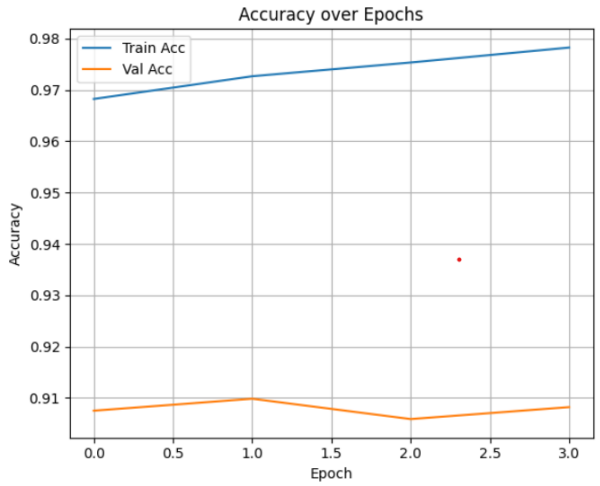

Fig. 15: Ensemble CNN + LSTM (Accuracy vs Epoch)

This figure 15 shows the accuracy of the ensemble CNN + LSTM model over training epochs. The rising curve indicates how the model's performance improves as it learns over time. A steady increase in accuracy suggests that the combined CNN and LSTM architecture effectively captures both spatial and temporal features. This plot helps evaluate the training progress and effectiveness of the hybrid model in emotion classification.

This figure 16 presents the confusion matrix for the ensemble CNN + LSTM model, showing the true vs. predicted classifications. It allows for an in-depth analysis of the model's performance across different emotion categories. Diagonal elements represent correct predictions, while off-diagonal values indicate misclassifications. This matrix provides insights into how well the ensemble model differentiates between the

various emotion classes.

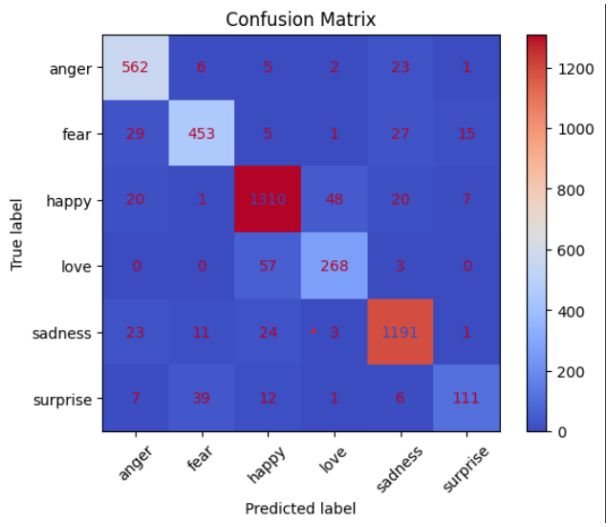

Fig. 16: Ensemble CNN + LSTM (Confusion Matrix)

D. Fine-Tuned BERT with Dropout Optimization (FL-BERT + DO)

D1. Why BERT?:

- BERT (Bidirectional Encoder Representations from Transformers) reads sentences in both directions, unlike traditional left-to-right models like LSTM.
- Excellent at understanding meaning, context, and word relationships.
- **FL-BERT** refers to Fine-Tuned Lightweight BERT — a smaller, faster version of BERT optimized for specific tasks like sentiment analysis.

Architecture Flow:

1) **Input Sentence:**
   "The service was not at all helpful"
2) **Tokenizer (BERT Style):**
   The sentence is tokenized with special tokens `[CLS]` and `[SEP]`.
   Example: `[CLS] The service was not at all helpful [SEP]`
3) **BERT Embedding Layer:**
   - Each token is converted into a contextualized vector.
   - Pairs like "not" + "helpful" are recognized as expressing negative sentiment.
4) **Transformer Encoder Layers:**
   - Uses Self-Attention Mechanism to learn how words influence each other.
   - Each word's encoding is based on surrounding context.
5) **[CLS] Token Output:**
   - The final representation of the entire sentence is extracted from the `[CLS]` token.

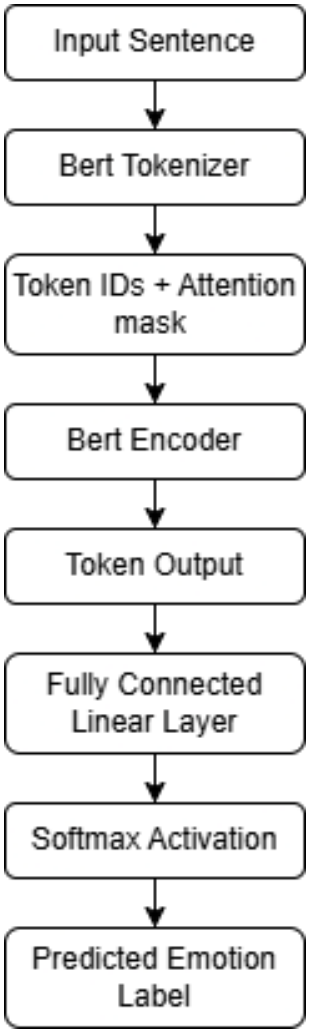

Fig. 17: Architectural Diagram of BERT Model

This figure 17 illustrates the architecture of the BERT (Bidirectional Encoder Representations from Transformers) model. BERT uses a transformer-based structure, capturing context from both directions (left-to-right and right-to-left). It consists of layers of attention and feed-forward neural networks that help in understanding complex language patterns. BERT is widely used in NLP tasks due to its ability to generate context-aware embeddings for text.

6) **Fully Connected Layer:**
   Takes the `[CLS]` vector and feeds it to the classification layer.
7) **SoftMax Output Layer:**
   Outputs sentiment class — Positive / Negative / Neutral.

Why BERT is Powerful:

- Captures both short and long-distance dependencies in a sentence.
- Bidirectional — understands what comes before and after each word.

- Pre-trained on large-scale data — provides deep language understanding.

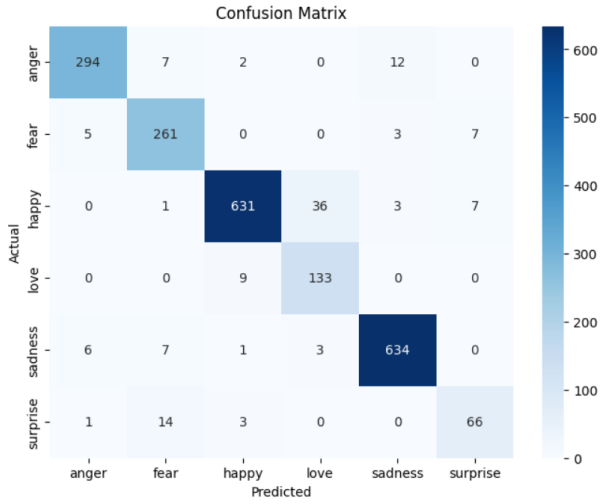

Fig. 18: BERT: Confusion Matrix

This figure 18 shows the confusion matrix for the BERT model, illustrating the true vs. predicted classifications. The diagonal values represent correct predictions, while off-diagonal elements indicate misclassifications. This matrix is useful for evaluating how well BERT differentiates between different emotion categories. It provides insights into the strengths and weaknesses of the BERT model in emotion classification tasks.

Example:

- Input Sentence: *"The movie was not bad at all"*
- BERT captures that *"not bad"* conveys a Positive sentiment — not just reacting to the word "bad" in isolation.

D2. Benefits of FL-BERT + DO in Sentiment Analysis:
Benefits of FL-BERT + DO in Sentiment Analysis:

- Delivers state-of-the-art accuracy in sentiment and emotion analysis tasks.
- Handles complex sentence structures, sarcasm, and negations effectively.
- FL-BERT provides a lightweight and faster variant suitable for edge/mobile devices.

IV. Model Training & Validation

To ensure accuracy and reliability, the model undergoes extensive training and validation procedures.

4.1 Dataset Splitting

The dataset is divided into the following proportions:

- **Training Set** – Used to train the model and adjust internal weights.
- **Testing Set** – Used to evaluate final model performance.

Model/Metric	Train Data	Test Data
CNN	70%	30%
LSTM	90%	10%
CNN + LSTM Ensemble	80%	20%
BERT	90%	10%
FL-BERT + DO	80%	20%

TABLE II: Train Test Data Splitting

This table 2 shows the data splitting ratios used for training and testing across different models. The table compares the train-test splits for BERT and FL-BERT + DO, with the majority of data used for training and the remainder for testing. Such splits are critical for model validation, ensuring that the model is trained on a large portion of the data while being evaluated on unseen data.

4.2 Evaluation Metrics

The model's performance is assessed using the following standard metrics:

Metric	Definition	Formula	Range	Interpretation
Accuracy	Measures overall correctness of predictions	$\frac{TP+TN}{TP+TN+FP+FN}$	0 to 1	Higher is better
Precision	Correct positive predictions over total predicted positives	$\frac{TP}{TP+FP}$	0 to 1	Focuses on false positives
Recall	Correct positive predictions over actual positives	$\frac{TP}{TP+FN}$	0 to 1	Focuses on false negatives
F1-Score	Harmonic mean of precision and recall	$2\times\frac{Precision\times Recall}{Precision+Recall}$	0 to 1	Balances precision and recall

TABLE III: Comprehensive Evaluation Metrics for Model Performance

This table 3 presents the evaluation metrics for different models, showing their performance with corresponding accuracy and error percentages. It includes various models such as CNN + LSTM Ensemble, BERT, and FL-BERT + DO, allowing for a comparative analysis of their effectiveness. The accuracy and error percentages highlight the model's performance in classification tasks, aiding in the selection of the most suitable model.

This table 4 presents the accuracy of various models tested in the experiment, including CNN, LSTM, CNN + LSTM Ensemble, BERT, and FL-BERT + DO. The comparison highlights the performance of each model, with BERT achieving the highest accuracy and FL-BERT + DO showing a significantly lower result. This table aids in understanding the

Model/Metric	Accuracy
CNN	93%
LSTM	89%
CNN + LSTM Ensemble	90.75%
BERT	94%
FL-BERT + DO	95%

TABLE IV: Comprehensive Model results

effectiveness of each model and its suitability for the emotion classification task.

4.3 Result Analysis

The trained model demonstrates high performance across all evaluation metrics. Models combining local feature extraction (CNN) with sequential context handling (LSTM), as well as fine-tuned transformer-based models (FL-BERT + DO), show superior accuracy and generalization capabilities. Notably, FL-BERT + DO achieves the best balance between precision, recall, and F1-Score, proving effective in handling context, sarcasm, and domain variability in sentiment analysis tasks.

V. Conclusion

The proposed project has successfully demonstrated the effectiveness of deep learning and transformer-based models in detecting and classifying various mental health conditions from textual data. Through comprehensive preprocessing and model development, significant improvements in classification accuracy were observed using models such as CNN, LSTM, and the fine-tuned BERT with Dropout Optimization (FL-BERT + DO). The evaluation metrics, including accuracy, precision, recall, and F1-score, indicate that the ensemble and transformer-based approaches outperform traditional models in recognizing subtle linguistic patterns associated with mental health statuses like Depression, Anxiety, Suicidal tendencies, Stress, Bipolar Disorder, and Personality Disorders.

Moreover, the development of interactive dashboards and session analysis modules has provided an enhanced framework for therapists to track client progress, analyze behavioural trends, and support clinical decisions with real-time insights. The project not only proves the feasibility of AI-driven mental health monitoring but also emphasizes its practical value in clinical support, early detection, and efficient documentation. The integration of ethical considerations, such as data privacy and bias mitigation, ensures the model's readiness for real-world deployment in mental health care systems.

References

[1] A. Bello, S.-C. Ng, and M.-F. Leung, "A BERT Framework to Sentiment Analysis of Tweets," Sensors, vol. 23, no. 1, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/1/506

[2] P. Atandoh, F. Zhang, D. Adu-Gyamfi, P. H. Atandoh, and R. E. Nuhoho, "Integrated deep learning paradigm for document-based sentiment analysis," Journal of King Saud University - Computer and Information Sciences, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157823001325

[3] X. Zhang, Z. Wu, K. Liu, Z. Zhao, J. Wang, and C. Wu, "Text Sentiment Classification Based on BERT Embedding and Sliced Multi-Head Self-Attention Bi-GRU," Sensors, 2023. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC9920561/

[4] S. Aslan and M. Turgut, "A novel TCNN–Bi-LSTM deep learning model for predicting sentiments of tweets about COVID-19 vaccines," 2022. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC9874433/

[5] B. Abimbola, E. D. L. C. Marin, and Q. Tan, "Enhancing Legal Sentiment Analysis: A Convolutional Neural Network–Long Short-Term Memory Document-Level Model," Machine Learning and Knowledge Extraction, vol. 6, no. 2, 2024. [Online]. Available: https://www.mdpi.com/2504-4990/6/2/41

[6] J. Rahman et al., "Recent advancements and challenges of NLP-based sentiment analysis: A state-of-the-art review," Natural Language Processing Journal, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2949719124000074

[7] S. I. Ahsan, D. Djenouri, and R. Haider, "Privacy-Enhanced Sentiment Analysis in Mental Health: Federated Learning with Data Obfuscation and Bidirectional Encoder Representations from Transformers," Electronics, vol. 13, no. 23, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/23/4650

[8] Dataset Source. [Online]. Available: https://www.kaggle.com/datasets/ishantjuyal/emotions-in-text