“Comparing recurrent neural based techniques against CNN for Mental health classification through sentiment analysis”

by Soham Das

A PROJECT REPORT
on

"Comparing recurrent neural based techniques against CNN for Mental health classification through sentiment analysis"

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN
Computer Science And Engineering

BY

Prasanna Sahoo	**2105980**
Shreyas Pande	**21051936**
Soham Das	**21051942**
Sudeepta Jena	**2105756**

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA-751024

NOVEMBER 2024

Abstract

The report presented to you is a comparison of LSTM and CNN in the context of mental health conditions. Here we are trying to classify different user-generated text by means of sentiment analysis. The two approaches of CNN and LSTM are quite different in terms of architecture and brilliant in their own ways. Time dependencies in sequential data are accurately and effectively represented by LSTM; on the other hand, CNNs recognize the local characteristics within text by the application of convolutional layers. These model workouts are rated based on the dataset of text related to mental health to catch up with anxiety, depression, and other issues. Use of CNN for textual based analysis is exciting in itself as its traditional or common use case is in computer vision. The application of deep learning tools in mental health programs will be greatly improved with continuous experiments and findings like these as they become guidelines for the development of accurate as well as scalable systems.

Keywords: Classification, Convolutional Neural Networks (CNN),Long Short-Term Memory (LSTM), Mental Health, Sentiment Analysis.

Chapter 1

Introduction

Sentiment analysis, which is one of the fundamental areas of Natural Language Processing (NLP), refers to the process of emotions, opinions, and attitudes extraction from textual data. Thus, this system entails a straightforward understanding of the subjective material included in the written content and offers knowledge about how people's evaluations of various topics can be. The domain of mental health has witnessed the rising importance of sentiment analysis which can help in the identification and deciphering of psychological disorders such as anxiety, depression, and stress in the last few or so years. Although the current models are still not accurate enough to identify, without any error, about the emotional or psychological state of a person.

Data sets for such training and testing for models based on mental health classification is itself a tough task. Some ways are by inspecting the content that is user-engineered from social media sites, online forums, and blogs, mental health practitioners and researchers can likewise obtain significant information about the emotional conditions and difficulties these individuals are encountering presently. Classification of mental health conditions in this project is largely based on deep learning methods such as Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN) which require intensive computing. Uniqueness of the LSTM network is the peculiar structure that makes it possible to decode the sequence of data even when it includes long-term dependencies. This property gives it an edge in solving problems related to time-series data, such as text analysis, where the meaning of words might depend on a previous context. However, when it comes to CNNs, they master through detecting local patterns in text, so they can precisely determine the features, like the word associations and phrases, that indicate those human emotions.

These deep learning methods make it possible for language to convey the subtle emotional shades, and hence the deepening of the knowledge of emotions, which is a tool used in the classification of mental health problems. Sentiment analysis inclusion into mental health diagnosis assists psychotherapists over the individual not only to help the person gain insights but also come up with strategies making therapy more successful.

Chapter 2

Basic Concepts

2.1 Neural Network Models

1. **Recurrent Neural Networks (RNNs):** They are the neural networks used to process sequential data, the sequence of which is sent to them. As a result, they have the ability to store the states of the past in their hidden layers.
2. **Long Short-Term Memory (LSTM):** A modification of RNNs that solves the vanishing gradient problem, which enables it to keep track of the long-term dependencies in sequences, thereby making it applicable to text data.

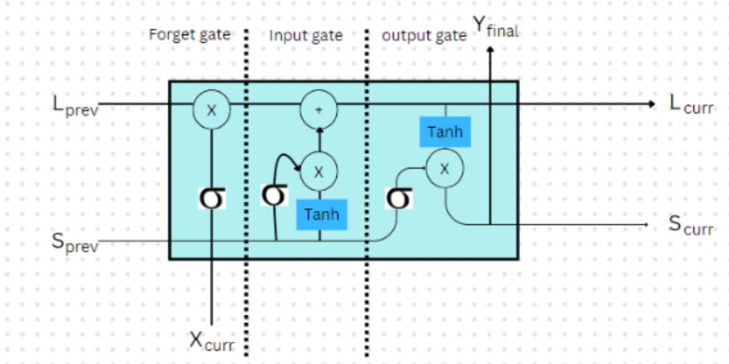


Fig1 Single cell of LSTM

Meaning of terms in Fig1:
L_{prev}: Previous state of the Long term memory
S_{prev}: Previous state of the Short term memory
X_{curr}: Current Input
L_{curr} :Current state of Long term memory
S_{curr}: Current state of Short term memory

LSTMs focusing on the cell state and gates:

A. **Cell State:** Long Term Memory is the representation of it. The network is a long way to go as it carries data through the whole network. (L_{prev})

B. **Hidden State:** It is sometimes referred to as Short Term Memory. The working memory is the temporary memory, which is shared with the next state and thus becomes the latest information. (S_{prev})

C. **Forget Gate:** This port is the one that tells what to get rid of out of the cell state. A filter is like a kind of it, which is examining information and then assigning a forgetfulness score between 0 and 1. A score of 0 explains the cell to persistently forget that subject while 1 expresses it to learn it all.

D. **Input Gate:** Input gates are used to control the amount of new information that is allowed to be included in the cell state. This process has a three-step structure:
   a. Screening the information that comes in
   b. Constructing new information candidates
   c. Harmonizing the new and the old

E. **Output Gate:** This gate determines which parts of the cell state are relevant for the current output and the next cell's hidden state. It does this through a three-step process:
   a. Preparation of the Information
   b. Pruning the output:
   c. Dispatching the filtered information

3. **Convolutional Neural Networks (CNNs):** Basic concept revolves around finding convolutions in the data and saving the important features of data to identify and process possible patterns in them. Traditionally used in image processing, it is effective in text classification as well with the convolutional layers to extract local features from text data.

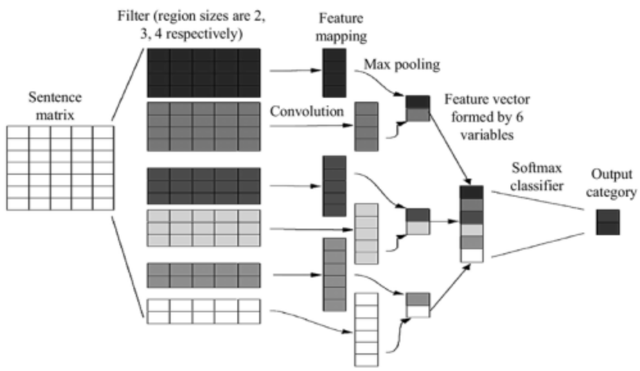


Fig.2 Convolutional Neural Networks

CNN Architecture(explained acc. to the use in text processing):

A. **Convolution layers:** In the convolution layer, a 1D kernel moves across the text's vector representation, capturing and aggregating the intricacies and complexities into further feature maps.
B. **Filters:** It is the moving kernel that is used to detect patterns in the data. It is actually a weighted-matrix that extracts features and then these are saved in the form of a matrix. Also, the amount of steps a filter takes is called strides.
C. **Pooling layer:** The pooling layer plays a key role in extracting the most vital characteristics from the convoluted matrix. This process uses various aggregation methods to significantly reduce the size of the feature map (the convoluted matrix). This reduction helps lower the memory demands during network training. Pooling is especially important because it helps prevent overfitting. Max pooling, which finds the highest value within the feature map, is used here.
D. **Activation function:** Specifically, a ReLU activation function. This function is essential, as it enables the network to learn non-linear relationships among the features. Because of this capability, the network becomes more adept at recognizing various patterns, thus enhancing its overall robustness.
E. **Fully connected layers:** These layers are situated at the final stage of a convolutional neural network and function as basic artificial neural networks (ANNs). Their inputs are derived from the flattened one-dimensional matrix created by the preceding pooling layer. ReLU activation functions are used to incorporate non-linearity, although this can sometimes cause challenges like vanishing gradients. Finally, a softmax layer is employed to produce probability scores for each possible output label. The label with the highest

probability is selected as the prediction, though this process can be affected by factors such as the model architecture and the quality of the training data.

2.2 Other Methodologies Used

4. **Natural Language Processing (NLP):** A sub-domain of AI that focuses on the relationship between computers and human language, thereby merging the two in a way that the machine can understand as well as produce text.
5. **Model Evaluation Metrics:** Evaluation mechanisms (e.g., accuracy, precision, recall, and F1-score) to estimate the effectiveness of mental health classification-type models.
6. **Adam Optimizer:** The optimizer used in the project is Adaptive Moment Estimation (Adam Optimizer). It belongs to the family of RMSprop (Mini Batch Gradient Descent). It is used in gradient descent to minimize the loss function. One of the benefits of Adam is that it computes individual learning rates for each and every single parameter in a neural network. This is especially beneficial for data with a lot of noise.

Chapter 3

Problem Statement

With the rise in mental health problems, there is an urgent need for robust tools to track and recognize psychological disorders using digital communication channels. Conventional ways of assessing is a lengthy process and even not so efficient to catch real-time feelings. The objective of this project is to build scalable and reliable sentiment analysis models using machine learning tools, mainly based on LSTM & CNN, for the automatic classification of mental health conditions from textual data in a given database. It must be kept in mind that we have built these models to provide assistance to mental health professionals in places where analyzing large amounts of data is required. The model can help them in identifying ambiguous texts that might point to psychological complications.

Chapter 4

Implementation

4.1 Data set Analysis

	Unnamed: 0	statement	status
0	0	oh my gosh	Anxiety
1	1	trouble sleeping, confused mind, restless hear...	Anxiety
2	2	All wrong, back off dear, forward doubt. Stay ...	Anxiety
3	3	I've shifted my focus to something else but I'...	Anxiety
4	4	I'm restless and restless, it's been a month n...	Anxiety

Fig 3 .Header of the Data

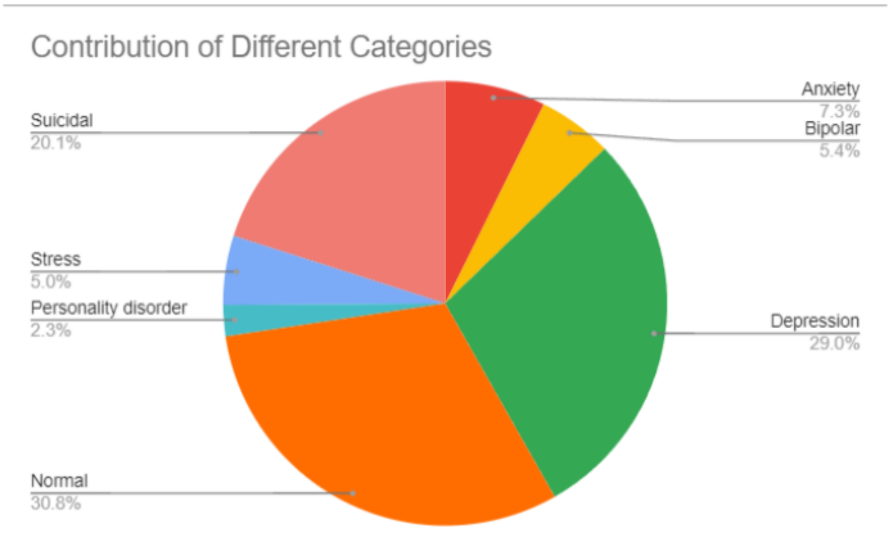


Fig.4 Distribution of classes in the data set

This dataset is obtained from Kaggle and it provides mental health sentients which are further categorizing responses into groups to indicate the future love about mental health. In the dataset, there are seven classes 'Anxiety', 'Bipolar', Depression,' Normal'al ve'Personality disorder','Stress', and'Suicidal'. They are used to identify and analyze different mental health conditions in the community.Pie chart showing breakdown of these categories in dataset 'Normal', the largest class at 30.8%, is indicative of a substantial proportion (though less than half) not showing features for any other category mentioned suppressors that might manifest due to mutations in distinct interactome zones, given prior molecular assays have shown enrichment specifically amongst C9orf72 patients [2]. This is closely followed by 'Depression' at 29%, this seems relatively high considering the percentage of depressive symptoms. A large 20.1% of respondents are described as some level prior or suicidal distress. ‘Anxiety' 7.3%, ‘Bipolar' 5.4%, 'Stress' and 'Personality disorder’ both = 5%; at the bottom, with only one-fifth in search of help format, we have Schizophrenia(Search for answers model — there is Glamour). This distribution represents a continuum of mental health that ranges from ideal mental well-being to extreme distress.

Symptoms overlap among conditions, so the division is often quite thin in mental health. For example, symptoms of anxiety can coincide with depression—you may even experience both at the same time—and stress only perpetuates those states. Some people may also somewhat fit into 2

categories, or display symptoms of different classes and this overlapping nature can make it difficult to concretely determine which sentiment is what. These nuanced distinctions can blur the lines of classification leaving you with unclean data, and affect your ability to obtain accurate results.
For creating a principal predictionary model exact categories are needed as even slight overlap and or misclassification can also impair the predictive accuracy. If the symptoms overlap in such a way that has lead to individuals being wrongly categorized, then certainly, the model will be confused as well and its chance of identifying between conditions become less precise. Second, if the model does not explicitly capture comorbidity between symptoms, it may oversimplify conditions that are common in mental health and make predictions with bias or forecast them inaccurately.Ergo, the implementation of a multi-label classification or a probabilistic approach can resolve the problem and capture the actual occurrence of comorbid conditions. The above modifications can improve the model's accuracy and reflect the true nuance of the mental health states.

4.2 Data Preprocessing

- Text Normalization: All strings of a text were transformed into a lower case. This normalization step prevents the influencing of the model by different capitalization patterns and ensures the consistency of the input data.
- Stopwords Removal: The common stop words were deleted. As they frequently are the center of the human learning process, these words are pointless linguistic filler.
- Lemmatization: The last step was lemmatization. It was conducted to reduce the words to their base form. Though it decreases the vocabulary size, it allows the model to learn that all forms of a word can be applicable to a particular concept.
- Label Encoding: Each of the mental health category was assigned with a unique numerical label For classification tasks, this step is necessary because the model needs a number for each class.
- Tokenization: Sentences Tokenized to split each text input into its corresponding words or Tokens. Tokenization operates at the level of words, rather than characters; it helps in sequence modeling which is crucial to maintain dependencies between different parts.
- Data Truncation: Rows where token sizes were greater than 1,000 words long have been purged to avoid the longer sequence lengths that can pose

a challenge for training models. By pruning this way we still had most of the data, kept only 17 rows=_("rows are lost")

- Sequence Padding: To ensure that dimensions match in all inputs to the model, sequences were padded till they have equal length (padded at a maximum size along with each sequence finally. It is this same reason why the evenness that convolutional layers introduce over a batch of images are so critical to be able to batch process in neural networks.

4.3 Model Training and Evaluation

To classify mental health categories, two model architectures—LSTM and CNN—were employed. The type of variations of these models that were used in the code are:

1. Basic Long Short-Term Memory (LSTM)
2. LSTM with pre trained GloVe embeddings
3. CNN with pre trained GloVe embeddings
4. CNN with dropout + pre-trained GloVe embeddings
5. CNN with dropout + pre-trained GloVe embeddings + positional encoding

4.3.1 Description of LSTM Based Models

4.3.1.1 Description of the Models:

Basic Long Short-Term Memory (LSTM) Model: An LSTM model was initially chosen for its strength in capturing long-term dependencies, a key aspect for handling textual data that relies on contextual information spanning several sentences or passages. Also, this is the only case where Word2Vec embedding is used.

LSTM with pre trained GloVe embeddings: Building upon the basic LSTM model, GloVe embeddings were integrated to provide pre-trained word representations. Each word was mapped to a 100-dimensional vector, preloaded into an embedding matrix (E), which was defined as follows:

$$
E = \begin{bmatrix} v1 & v2 & \dots & v100 \\ \vdots & \vdots & \ddots & \vdots \\ v|V|1 & v|V|2 & \cdots & v|V|100 \end{bmatrix}
$$

Here, $|V|$ is the vocabulary size, and v_i represents the GloVe embedding for the i-th word. The embedding matrix was loaded into the model's embedding layer as non-trainable weights, providing semantically enriched starting representations. The loss function minimized during training was categorical cross-entropy, defined as:

$$
L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} log(\hat{y}_{ij})
$$

4.3.1.2 Model Architectures

Model Architecture: The LSTM model's structure included:
An Embedding layer with the pre-trained, non-trainable GloVe embeddings.Two LSTM layers, each with 100 units. The first layer returned sequences to allow further processing, while the second layer condensed this sequence data for the next layers.A Dense output layer with 7 units (one per mental health category) and a softmax activation for multi-class classification.

Optimizer and Loss: The model used the Adam optimizer for its adaptive gradient handling and convergence efficiency, combined with categorical cross-entropy loss, ideal for multi-class classification with one-hot encoded labels.

4.3.2 Description of CNN Based Models

4.3.2.1 Description of the Models:

CNN with Pre-Trained GloVe Embeddings: In response to the LSTM's overfitting issues, a CNN architecture was introduced. CNNs, known for their ability to capture localized patterns, can process short-range dependencies between words, offering improved robustness without relying on long-term memory. The model leveraged a pre-trained GloVe embedding matrix in a non-trainable embedding layer, enabling the network to start with semantically rich word representations.

CNN with Dropout + Pre-Trained GloVe Embeddings: To further combat overfitting, dropout was integrated into the CNN architecture. Dropout randomly deactivates a subset of neurons during training, encouraging the network to learn more generalized features. Combined with pre-trained GloVe embeddings, this approach balanced robust feature extraction with enhanced regularization.

CNN with Dropout + Pre-Trained GloVe Embeddings + Positional Encoding: To enhance the sequential understanding of text data, positional encoding was incorporated into the CNN. The positional encoding vector for a position pos and dimension d was computed as:

$$PE^{(pos,2i)} = sin(\frac{pos}{1000^{2i/d}}),\ PE^{(pos,2i+1)} = cos(\frac{pos}{1000^{2i/d}})$$

This encoding was added to the input embeddings, allowing the network to consider positional context alongside semantic information. Combined with dropout and pre-trained GloVe embeddings, this configuration achieved a balance between learning complex patterns and generalizing effectively to validation data.

4.3.2.2 Model Architectures

Model Architecture: Key components of the CNN architecture included: An Embedding layer initialized with the GloVe embeddings, set as non-trainable.

Multiple Conv1D layers with 128 and 256 filters, activated with ReLU, which detected n-gram patterns in text sequences.
MaxPooling layers positioned between convolutional layers, reducing dimensionality and focusing on essential features. A Flatten layer, transforming the feature maps into a single vector for further processing. Lastly, the dense layers for learning from the single vectors, leading to the final classification.

Regularization: Dropout layers, with progressively higher rates (from 0.4 to 0.5) in deeper layers, combated overfitting.

4.4 Testing

Testing looks at how well models work on new data to check if they can handle different cases. We use precision, recall, and the F1 score to see how good they are at sorting data.

Precision: This is about how many of the model's yes guesses are right. It shows if the model can spot right things in each group without saying too many wrong things are right. A high precision means not many mistakes are made, showing the model is good at picking out real yeses from the noes.

Recall: Recall checks how many of the real yes cases the model can find. It shows if the model can catch all the important cases in each group. A high recall means it does a good job at finding the yeses and doesn't miss many.

F1 Score: The F1 score mixes precision and recall. It's useful when the data aren't balanced, stopping one from seeming too good due to low numbers of another. A high F1 means the model is well-balanced in not making too many yes or no mistakes.

These metrics collectively form the **F1 Matrix**, which is used to evaluate the model's overall classification accuracy and robustness across all classes.

Accuracy: Accuracy is the proportion of correctly classified instances relative to the total number of instances. It is the magnitude of the overall effectiveness of the model in correctly classifying or predicting outcomes across all classes. For a neural network, high accuracy typically indicates that the model has learned meaningful patterns in the data, but it may not always reflect performance on imbalanced datasets.

Macro Average (Macro Avg): The macro average computes the mean of a metric, like precision, recall, or F1 score, across all classes, giving equal weight to each class regardless of its size. This approach is particularly useful for assessing model performance when the dataset is imbalanced, as it ensures that smaller classes are not overshadowed by larger ones.

Weighted Average (Weighted Avg): The weighted average, unlike the macro average, takes the class sizes into account when calculating metrics like precision, recall, or F1 score. It provides a more balanced view of model performance by assigning greater weight to classes with a larger number of samples, making it ideal for datasets with class imbalances.

By examining these metrics, we can determine how accurately the model distinguishes between different categories and how well it handles overlapping classes, which is crucial for classification tasks with subtle boundaries between categories.

Chapter 5

5. Standards Adopted

In order to have a structured view and working we have defined certain features to make sure the project goes on track.

5.1 Code Standards

Certain coding standards are maintained throughout the project to ensure readability and maintainability. We have tried to use a consistent use of appropriate naming conventions throughout the code. We used segmentation or block format of execution where we have divided the code into different cells for better readability. Proper indentation and documentation have been done for better understanding. Code reusability for decreasing the number of lines of code can be seen while we try to test the prediction.

5.2 Testing Methods

The project follows industry-standard testing practices to ensure the reliability and quality of the predictive models. Testing standards include adherence to ISO and IEEE guidelines for quality assurance.

Chapter 6

Results

In the end, firstly let's start with seeing the results for the models for each class through a classification report.

Train Data

Classification Report for Basic LSTM:

	precision	recall	f1-score	support
Anxiety	0.97	0.95	0.96	2892
Bipolar	1.00	0.82	0.90	1989
Depression	0.88	0.82	0.84	12084
Normal	0.96	0.99	0.97	12802
Personality disorder	0.99	0.55	0.71	716
Stress	0.89	0.79	0.84	1829
Suicidal	0.79	0.92	0.85	8548
accuracy			0.90	40860
macro avg	0.93	0.83	0.87	40860
weighted avg	0.90	0.90	0.89	40860

Classification Report for Basic LSTM

Validation Data

Classification Report for Basic LSTM:

	precision	recall	f1-score	support
Anxiety	0.84	0.81	0.82	731
Bipolar	0.95	0.69	0.80	511
Depression	0.68	0.64	0.66	3001
Normal	0.89	0.92	0.91	3238
Personality disorder	0.95	0.39	0.55	178
Stress	0.58	0.42	0.49	467
Suicidal	0.59	0.72	0.65	2090
accuracy			0.75	10216
macro avg	0.78	0.66	0.70	10216
weighted avg	0.75	0.75	0.74	10216

Classification Report for Basic LSTM

Classification Report for LSTM with pre trained GloVe embeddings:

	precision	recall	f1-score	support
Anxiety	0.86	0.69	0.77	2892
Bipolar	0.83	0.67	0.74	1989
Depression	0.66	0.72	0.69	12084
Normal	0.87	0.95	0.91	12802
Personality disorder	0.42	0.17	0.24	716
Stress	0.50	0.18	0.26	1829
Suicidal	0.65	0.67	0.66	8548
accuracy			0.74	40860
macro avg	0.68	0.58	0.61	40860
weighted avg	0.73	0.74	0.73	40860

Classification Report for LSTM with pre trained GloVe embeddings

Classification Report for LSTM with pre trained GloVe embeddings:

	precision	recall	f1-score	support
Anxiety	0.80	0.67	0.73	731
Bipolar	0.80	0.66	0.72	511
Depression	0.64	0.67	0.66	3001
Normal	0.86	0.94	0.90	3238
Personality disorder	0.41	0.14	0.21	178
Stress	0.37	0.12	0.18	467
Suicidal	0.61	0.65	0.63	2090
accuracy			0.72	10216
macro avg	0.64	0.55	0.57	10216
weighted avg	0.70	0.72	0.71	10216

Classification Report for LSTM with pre trained GloVe embeddings

Fig 5. For LSTM Models

Train Data

Classification Report for CNN with pre trained GloVe embeddings:

	precision	recall	f1-score	support
Anxiety	1.00	0.99	0.99	2892
Bipolar	0.99	1.00	0.99	1989
Depression	0.99	0.99	0.99	12084
Normal	1.00	1.00	1.00	12802
Personality disorder	0.99	0.99	0.99	716
Stress	0.99	1.00	0.99	1829
Suicidal	0.99	0.99	0.99	8548
accuracy			0.99	40860
macro avg	0.99	0.99	0.99	40860
weighted avg	0.99	0.99	0.99	40860

Classification Report for CNN with pre trained GloVe embeddings

Classification Report for CNN with dropout + pre trained GloVe embeddings:

	precision	recall	f1-score	support
Anxiety	0.96	0.97	0.96	2892
Bipolar	0.98	0.94	0.96	1989
Depression	0.92	0.88	0.90	12084
Normal	0.97	0.99	0.98	12802
Personality disorder	0.89	0.79	0.84	716
Stress	0.94	0.86	0.90	1829
Suicidal	0.86	0.91	0.89	8548
accuracy			0.93	40860
macro avg	0.93	0.91	0.92	40860
weighted avg	0.93	0.93	0.93	40860

Classification Report for CNN with dropout + pre trained GloVe embeddings:

Classification Report for CNN with dropout + pre trained GloVe embedd

	precision	recall	f1-score	support
Anxiety	0.97	0.95	0.96	2892
Bipolar	1.00	0.82	0.90	1989
Depression	0.88	0.82	0.84	12084
Normal	0.96	0.99	0.97	12802
Personality disorder	0.99	0.55	0.71	716
Stress	0.89	0.79	0.84	1829
Suicidal	0.79	0.92	0.85	8548
accuracy			0.90	40860
macro avg	0.93	0.83	0.87	40860
weighted avg	0.90	0.90	0.89	40860

Classification Report for CNN with dropout + pre trained GloVe embeddings + positional encoding:

Validation Data

Classification Report for CNN with pre trained GloVe embeddings:

	precision	recall	f1-score	support
Anxiety	0.76	0.70	0.72	731
Bipolar	0.65	0.71	0.68	511
Depression	0.65	0.68	0.67	3001
Normal	0.89	0.88	0.89	3238
Personality disorder	0.35	0.28	0.31	178
Stress	0.43	0.42	0.42	467
Suicidal	0.59	0.59	0.59	2090
accuracy			0.71	10216
macro avg	0.62	0.61	0.61	10216
weighted avg	0.71	0.71	0.71	10216

Classification Report for CNN with pre trained GloVe embeddings

Classification Report for CNN with dropout + pre trained GloVe embeddings:

	precision	recall	f1-score	support
Anxiety	0.79	0.75	0.77	731
Bipolar	0.77	0.71	0.74	511
Depression	0.67	0.65	0.66	3001
Normal	0.87	0.93	0.90	3238
Personality disorder	0.44	0.32	0.37	178
Stress	0.53	0.43	0.48	467
Suicidal	0.62	0.65	0.64	2090
accuracy			0.73	10216
macro avg	0.67	0.63	0.65	10216
weighted avg	0.73	0.73	0.73	10216

Classification Report for CNN with dropout + pre trained GloVe embeddings:

Classification Report for CNN with dropout + pre trained GloVe embed

	precision	recall	f1-score	support
Anxiety	0.84	0.81	0.82	731
Bipolar	0.95	0.69	0.80	511
Depression	0.68	0.64	0.66	3001
Normal	0.89	0.92	0.91	3238
Personality disorder	0.95	0.39	0.55	178
Stress	0.58	0.42	0.49	467
Suicidal	0.59	0.72	0.65	2090
accuracy			0.75	10216
macro avg	0.78	0.66	0.70	10216
weighted avg	0.75	0.75	0.74	10216

Classification Report for CNN with dropout + pre trained GloVe embeddings + positional encoding:

Fig 6. For CNN Models

The above results can be summarized in this way, specifics of each models:

1. The Basic LSTM model achieves 75% validation accuracy but struggles with minority classes like "Stress" (recall: 0.42) and "Personality Disorder" (recall: 0.39), indicating limited generalization for underrepresented categories.
2. The LSTM with Pre-trained GloVe Embeddings improves class understanding but reduces overall accuracy to 72%, with poor recall for "Stress" (0.12), showing insufficient handling of imbalanced data.
3. The CNN with Pre-trained GloVe Embeddings exhibits significant overfitting, achieving 71% validation accuracy but performing poorly on minority classes such as "Personality Disorder" and "Stress."
4. The CNN with Dropout and Pre-trained GloVe Embeddings balances generalization better, with 73% validation accuracy and improved recall for "Stress" and "Suicidal" categories.

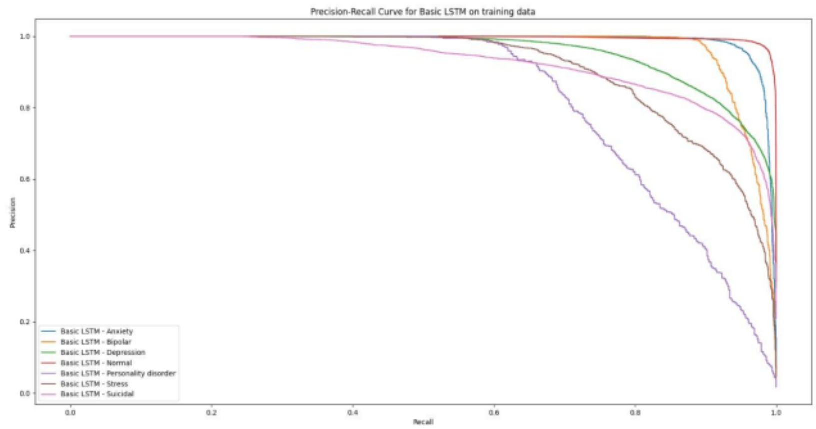
5. The CNN with Dropout, Pre-trained GloVe Embeddings, and Positional Encoding achieves 75% validation accuracy, showing enhanced generalization and slight recall improvements for minority classes like "Personality Disorder" and "Stress."

Here, the significant drop in accuracy from training to validation (e.g., 99% to 71%) in CNN-based models with pre-trained GloVe embeddings strongly indicates overfitting. While CNNs excel at capturing complex patterns in training data, they often fail to generalize effectively to unseen data, particularly in the context of imbalanced datasets. This overfitting issue remains a central challenge when applying CNNs in real-world applications that demand robust generalization across diverse data distributions.

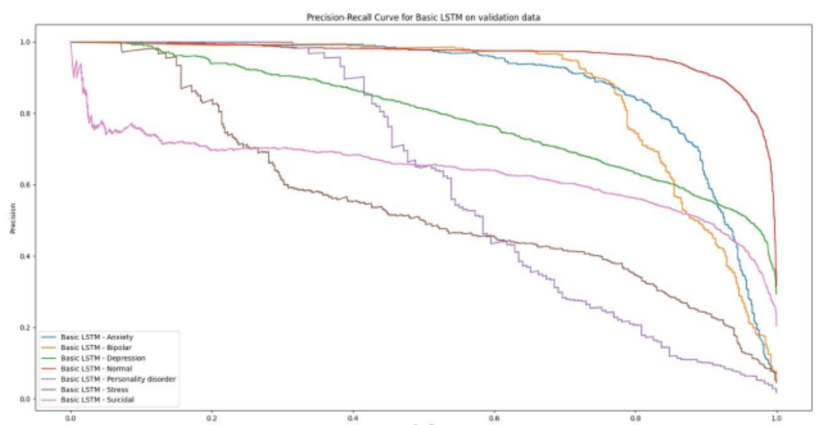
LSTM models, in comparison, exhibit more stable performance on the validation set. Despite achieving lower accuracy overall, they show better consistency and handle minority classes like "Stress" and "Personality Disorder" slightly more effectively. This balance makes LSTMs better suited for real-world scenarios where generalization to unseen data is essential. Their ability to address minority classes, which often represent critical but underrepresented conditions, adds to their practical utility.

The addition of dropout and positional encoding to CNNs does improve generalization to some extent. However, the observed drop in validation accuracy (e.g., 90% to 75%) continues to highlight the architectural limitations of CNNs when working with this dataset. Even with advanced techniques, CNNs still struggle with data imbalance and overfitting, suggesting that further architectural innovation is necessary.

Precision-Recall(PR) graph for Basic Long Short-Term Memory (LSTM)

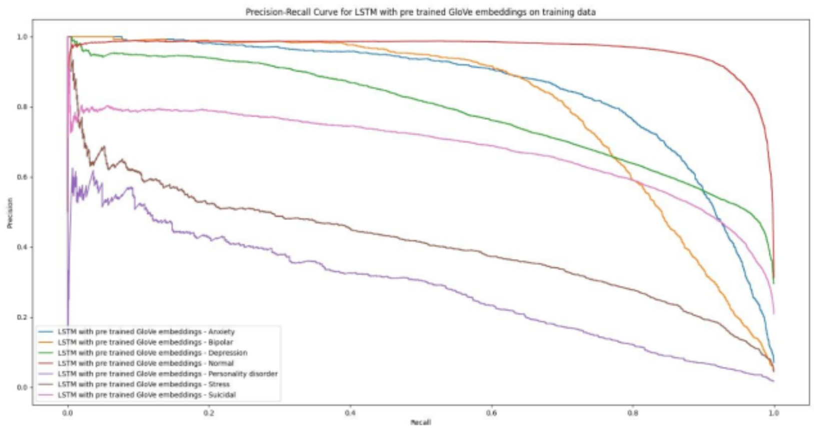


Training Data

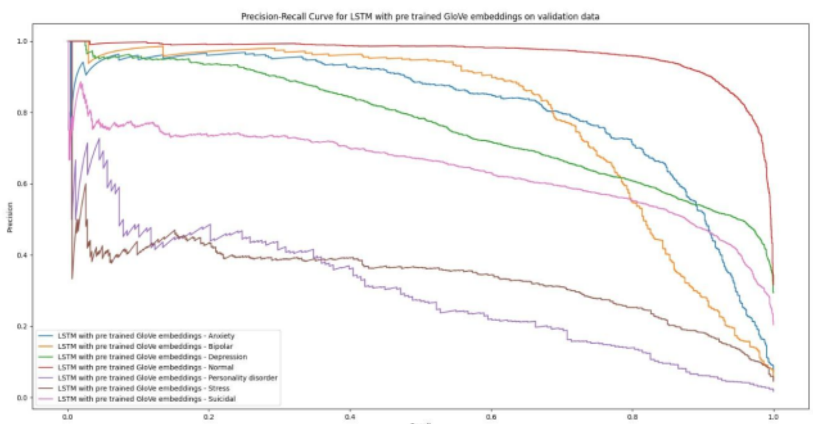


Testing Data

PR graph for LSTM with pre trained GloVe embeddings

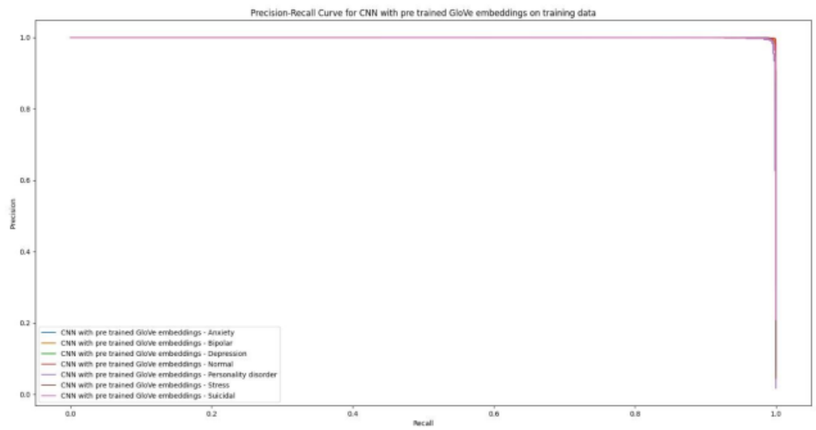


Training Data

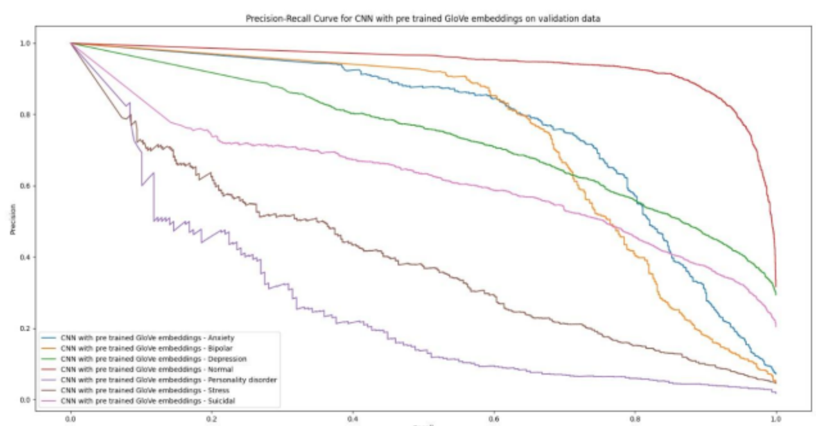


Testing Data

PR graph for CNN with pre trained GloVe embeddings

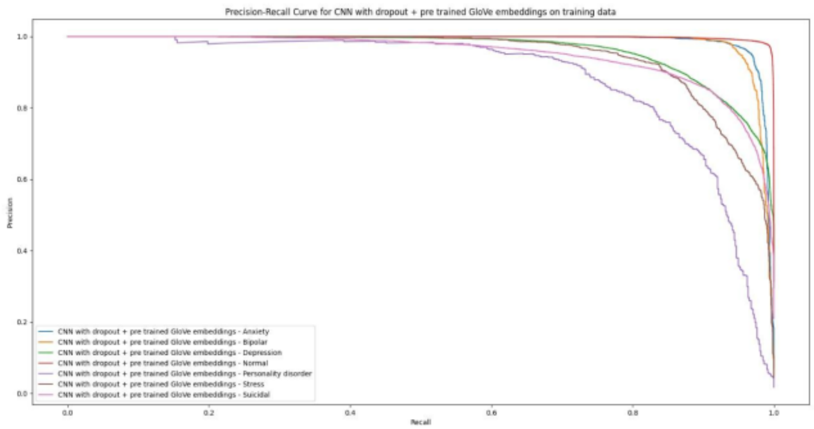


Training Data

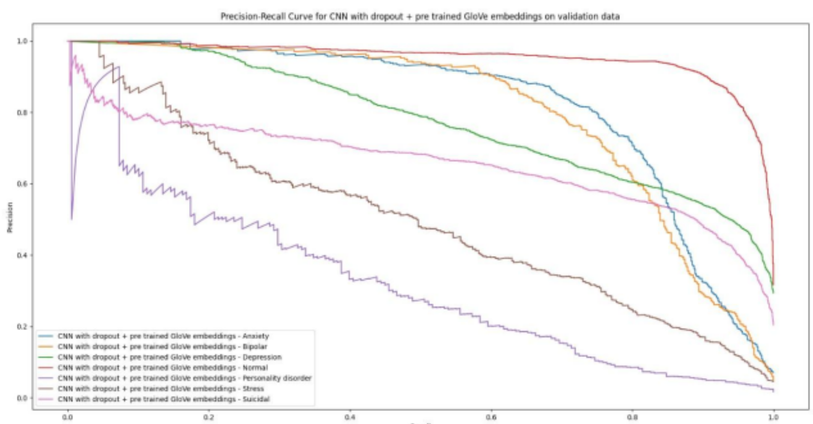


Testing Data

PR graph for CNN with dropout + pre-trained GloVe embeddings

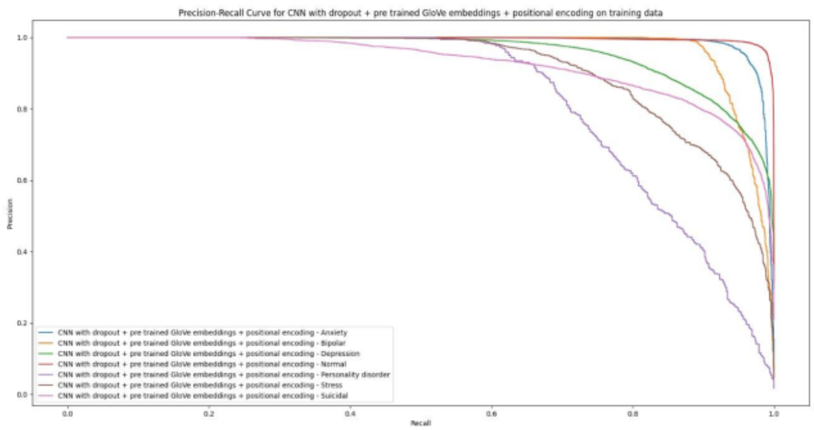


Training Data

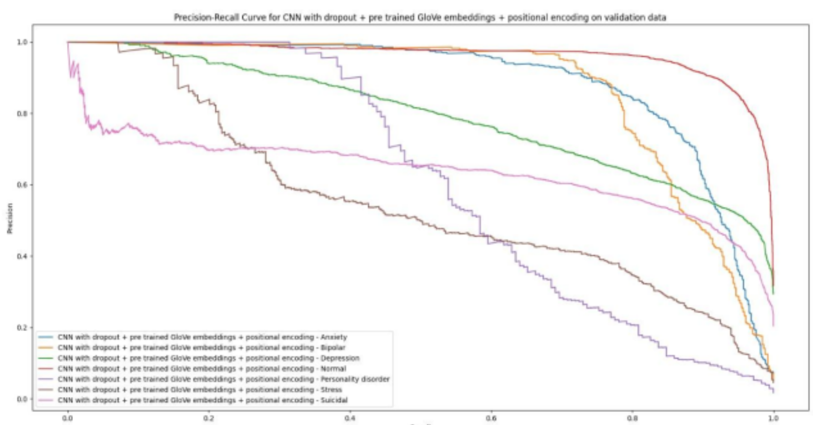


Testing Data

PR graph for CNN with dropout + pre-trained GloVe embeddings + positional encoding



Training Data



Testing Data

Now, these are the Precision and Recall Graphs(PR Graph). They play a crucial role in assessing the performance of classification models, particularly in scenarios with imbalanced datasets. It graphs precision (the ratio of true positive predictions to all positive predictions) against recall (the ratio of true positive predictions to all actual positives) across different thresholds. It offers an understanding of how effectively a model manages the trade-off between precision and recall. This is essential in scenarios where the impact of false positives and false negatives differs significantly. A well-performing model exhibits curves that approach the top-right corner of the graph, indicating high precision and high recall. By analyzing the shape and spread of these curves across classes, we can identify areas where the model struggles, such as underrepresented classes, and evaluate its ability to generalize to unseen data.

The observed Precision-Recall curve, particularly for the CNN model with pre-trained GloVe embeddings, demonstrates a **steep curve** structure. This steepness is characterized by an abrupt rise and a sharp alignment along the top-right corner of the graph. This can be seen specifically in the training part of the data. Such curves reflect extremely high precision and recall on the training data, indicating that the model has likely overfit to the training set. In this case, the model performs exceptionally well in recognizing patterns it has already seen but struggles to generalize effectively to new, unseen data.

A similar trend is noted in other CNN models, albeit with slightly broader curves. While the broader structure suggests marginal improvements in generalization, the pattern still points to a high reliance on training data. When compared to validation curves, which often appear more irregular and less steep, the disparity becomes evident, further highlighting the model's overfitting tendencies. This consistent trend underscores the need for better regularization techniques or a shift toward more robust architectures to achieve better real-world performance.

Also, it is evident that in the Basic LSTM model, the trends observed are somewhat similar to the Precision-Recall graph for CNN with dropout, pre-trained GloVe embeddings, and positional encoding. Both exhibit disparities between the training and validation performances, with the validation curves reflecting irregularities that hint at challenges in generalization. On the other hand, the Precision-Recall graph for the LSTM model with pre-trained GloVe embeddings shows a much closer alignment between the training and validation curves. This consistency suggests that the model has achieved better generalization, effectively learning the patterns in the data without overfitting to the training set.

Finally, it is important to note that no model has performed poorly overall. Each model captures different aspects of the data effectively, although none achieves an optimal balance across all metrics. These observations just highlight the importance of evaluating not just the numerical results but also the structural behavior of the models across metrics to understand their real-world applicability. This underscores the importance of further fine-tuning or exploring alternative architectures, such as transformers or ensemble models, to achieve a better compromise between training and validation performance.

Chapter 7

Future Scope

Firstly, in the scope of the current project, a lot can be done in terms of classification. We can change the classes that were being used to a more precise and clearer differentiation. We can include description of physical conditions along with textual statements to make the classifications more accurate. There can also be minor modification in terms of pooling layers, number of filters and embedding layers in order to observe and experiment.

In terms of new additions, we think transformers can be a good area where we explore and get some really good results. Transformers are highly effective at capturing long-term dependencies and contextual nuances, making them well-suited for analyzing complex patterns in textual data, such as those found in mental health contexts. Their ability to handle class imbalances through attention mechanisms can improve performance for underrepresented categories.Additionally, other transfer learning methods, such as fine-tuning pre-trained language models like BERT, can leverage domain knowledge from vast datasets to enhance contextual understanding and classification. They also improve performance on minority classes by transferring generalized representations to specific tasks.

Conclusion

Putting feelings check into mental health care and help marks a big shift for both study and day-to-day work. As methods to read human language get better, the power to look at and get what people feel from a lot of text helps mental health experts to know better how people feel inside. Deep learning tools like LSTMs and CNNs are good at finding out mood trends, while transformer models add more by being very smart about context and fine points.

With these steps forward, feeling check can spot mental health issues and also guess changes in how people feel over time, making ways for early finding and help. By using these tools in real-time, like keeping an eye on social media and chat help for mental health, pros can give quick, right help to those who need it. Also, putting a feeling check into mental health care systems can help keep an eye on things all the time and give personal tips, letting users follow their mood and get help when they need it.

This growing mix between AI-led feeling check and mental health care shows big hope for making mental health help more easy to get to, quick, and fit for each person. In the end, these tech steps could change mental health services, making them more ahead-looking, led by data, and better at both study and real use, and can contribute to advancements in mental health support and intervention for many.

"Comparing recurrent neural based techniques against CNN for Mental health classification through sentiment analysis"

ORIGINALITY REPORT

5%	4%	3%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	R. N. V. Jagan Mohan, Vasamsetty Chandra Sekhar, V. M. N. S. S. V. K. R. Gupta. "Algorithms in Advanced Artificial Intelligence", CRC Press, 2024 Publication	1%
2	Submitted to KIIT University Student Paper	1%
3	arxiv.org Internet Source	<1%
4	ijritcc.org Internet Source	<1%
5	Submitted to Taylor's Education Group Student Paper	<1%
6	linnk.ai Internet Source	<1%
7	Helena F. Deus, Corey Harper, Darin McBeath, Ron Daniel. "Combining pattern matching with word embeddings for the extraction of experimental variables from scientific	<1%

literature", 2017 IEEE International Conference on Big Data (Big Data), 2017
Publication

8	www.econstor.eu Internet Source	<1 %
9	fastercapital.com Internet Source	<1 %
10	ir.vanderbilt.edu Internet Source	<1 %
11	www.tdcommons.org Internet Source	<1 %
12	Bernhard Kratzwald, Suzana Ilić, Mathias Kraus, Stefan Feuerriegel, Helmut Prendinger. "Deep learning for affective computing: Text-based emotion recognition in decision support", Decision Support Systems, 2018 Publication	<1 %
13	clinical-practice-and-epidemiology-in-mental-health.com Internet Source	<1 %
14	poleval.pl Internet Source	<1 %
15	Mengqiao Liu, Tianjun Sun, Feng Guo, Hanyi Min. "Interpretable Natural Language Processing Models for Organizational Research", PsyArXiv, 2024 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		