Negative Sentiment Analysis Using Twitter Comments

Tanishq Patil
Manipal University Jaipur
Rajasthan, India
tanishq.229311050@muj.maniapl.edu

Sairaj Sirsat
Manipal University Jaipur
Rajasthan, India
sairaj.229311122.@muj.maniapl.edu

Yashvardhan
Manipal University Jaipur
Rajasthan, India
yashvardhan.229310377@muj.maniapl.edu

Dr. Kanwalpreet Kour
Manipal University Jaipur
Rajasthan, India
kanwalpreet.kour@jaipur.manipal.edu

***Abstract*:** Depression is a chronic mental illness that generally goes undiagnosed until it becomes severe. With more use of online communication, individuals have more and more ways in which they share feelings through text, so it is now feasible to use sentiment analysis in identifying depressive behaviour. Various machine learning models in this study are employed to classify user comments on various websites as depressive or non-depressive. A set of user reviews was collected and pre-processed using tokenization, stop-word removal, and TF- IDF embedding. While different algorithms were experimented with, the most viable choice turned out to be Random Forest due to its accuracy vs. hardware usage trade-off. Testing through accuracy, precision, recall, and F1-score established its reliability. This research contributes to AI-based mental health surveillance and paves the way towards future integration with real-time frameworks and deep neural networks.

I. INTRODUCTION

Mental illness disorders, particularly depression, are this century's most pressing worldwide health crises [1]. The conventional diagnosis through clinical evaluation and self-report questionnaires misses early symptoms in many, and most people remain untreated [2]. Expanded access to internet forums via which people speak freely have turned user comments into a goldmine of early mental analysis [3].

This research seeks to create an affordable, scalable sentiment analysis model that can determine depressive and non-depressive comments through machine learning. It fills the gap between traditional diagnostic methods and machines by utilizing Natural Language Processing (NLP) and machine learning to process emotional expression within text. Various models such as Support Vector Machines (SVM), Naïve Bayes, and Neural Networks were experimented on to gauge efficacy [4]. However, Random Forest was found to be the most viable based on its relatively high accuracy, efficient use of available computing power, and the ease with which it can be interpreted.

The method is to collect user reviews related to depression from publicly available data sets, preprocess them through tokenization, stop-word filtering, lemmatization, and applying TFIDF for feature extraction [5]. The Random Forest model is trained and tested on the data using typical performance measures. The contribution of this work is towards the development of AI based tools for monitoring mental health, with future research directions as integration with real-time social media and improvement through deep learning techniques [1].

The following sections of this paper will give related works, literature survey, dataset preparation, research methodology, result, discussions, and future improvements. With the application of machine learning in sentiment analysis, this research is hoped to help develop AI-based tools for preventive mental health intervention and support.

II. LITERATURE REVIEW

To draft the written literatures review, an initial pool of 50 research papers was taken and viewed in order to probe into different methods in the detection of depression using artificial intelligence. From this larger bunch, 18 papers were chosen after measuring them with respect to the criteria of relevance to project scope, quality of data, and methodological diversity. These referred studies give a broad base in recognizing current trends, models, and challenges in the area of automated depression detection methodologies.

Table 1: Literature Review Table

Authors	**Algorithms Used**	**Models Used**	**Dataset**	**Accuracy**	**Findings**	**Limitations**
Vandana , et al (2023)	CNN, LSTM, Bi-LSTM	Textual CNN, Audio CNN, Hybrid LSTM/Bi-LSTM	**DAIC-WOZ Database (189 sessions)**	Textual CNN: 92 Audio CNN: 98 Bi-LSTM: 88%	Audio > Text (98% vs 92%). Bi-LSTM > LSTM. Hybrids = more robust.	Imbalanced data (4:1), Bi-LSTM slow (5+ hrs), limited to DAIC-WOZ.
Lamia Bendebane et al (2023)	Deep Learning (CNN, RNN, LSTM, GRU, BiRNN, BiLSTM, BiGRU), Grid Search	Hybrid models (e.g., CNN-BiGRU, CNN-BiLSTM)	**3.17M tweets (English)**	**93.38** (CNN-BiGRU)	Multi-class > Binary Detects depression vs. anxiety well Grid search tuned learning rate	Labeling issues Not tested on non-English tweets Needs clinical validation
Shumaila Aleem et al (2022)	SVM, RF, KNN, DT, AdaBoost, CNN, LSTM, DCNN, XGBoost	Classification, Deep Learning, Ensemble	**EEG, social media (Twitter, Reddit), clinical records (PHQ-9, BDI-II)**	76.6–98.32	SVM and RF are robust; EEG-based DL models achieve high accuracy; multimodal approaches show promise.	Small sample sizes, lack of standardized datasets, limited clinical applicability.
Faye Beatriz Turnaliuan et al (2024)	LSTM with Dropout, GRU, CNN, Naïve Bayes, Random Forest	Two-stage model (Binary + multi-class)	**86,163 tweets (English/Filipino) annotated with 13 depression categories**	Stage 1: 91 (F1: 0.90) Stage 2: 83 (F1: 0.81)	Two-stage model: Binary detection + 6 symptom types LSTM + Dropout = best performance	Errors from word associations, negation, imbalance Limited to English/Filipino; excludes regional languages
Rafael Salas-	SVM, Logistic	Word Embedding	**Twitter, Reddit,**	N/A	Twitter + SVM/embedd	Limited to studies from

Zárate (2022)	Regression, Neural Networks, Random Forests	, N-grams, Bag of Words, Tokenization	**Facebook, Instagram, Weibo, NHANES**		ings most used Python tools, cross-validation standard	2016-2021. Focused mainly on English-language platforms.
Yazdavar (2020)	SVM, Neural Networks	Word Embedding, LIWC, Cohen's Kappa	**Twitter (8770 users)**	N/A	Combined linguistic and behavioural features for depression detection. Used Cohen's Kappa for validation.	Complex multimodal approach may not be scalable.
Chiong (2021)	SVM, Neural Networks	N-grams, Bag of Words	**Twitter, Facebook (22,191 records)**	N/A	Compared SVM and Neural Networks for textual analysis. Found SVM to perform better.	Limited to textual features. No accuracy metrics.
Bazen Gashaw Teferra (2024)	SVM, Logistic Regression, Neural Networks, Transformers (BERT, GPT)	Sentiment Analysis, Linguistic Markers, Word Embeddings, LLMs	**DAIC-WOZ, Weibo, Twitter, Reddit**	82.3 - 91	NLP (sentiment, LLMs) = high accuracy Key issues: ethics, cultural sensitivity	Limited databases, no meta-analysis English/Chinese focus limits generalizability
Prabhu (2022)	LSTM	Word2vec	**DAIC-WOZ (189 sessions)**	82.3	High accuracy: LSTM + Word2Vec for emotion-based detection	Clinical data only; may not generalize to social media
Choudhury (2021)	SVM	LIWC (22 linguistic styles)	**Twitter (554 users)**	72.4	Identified linguistic styles associated with depression on Twitter using SVM.	Limited to Twitter; potential bias in user selection.

Nikhil Goel et al (2024)		Hybrid (SVM + Decision Trees). Neural Networks.	**1,000 subjects (text + wearable data)**	90 (Hybrid model)	Sentiment (85%) + Behaviour (r= 0.7) Hybrid model: F1= 0.89	Reliance on self-reported data (bias risk). Contextual ambiguity in sentiment analysis. Device variability affects behavioural data quality.
Dinkel (2020)	SVM	ELMo	**DAIC-WOZ (189 sessions)**	F_1-score = 84	Achieved high performance using ELMo embeddings for sparse data depression detection.	Limited to specific datasets; may not generalize.
Rutowski (2022)	Transformers	GloVe	**American English spontaneous speech (16,000 sessions)**	AUC = 0.8	Transformers used for prediction; transfer learning proved effective	Focused on English speech; may not apply to text data.
Korti (2022)	LSTM	Word Embeddings	**Twitter**	91	Achieved high accuracy with LSTM for Twitter-based depression detection.	Limited to Twitter; potential bias in data collection.
Tejaswini (2024)	FastText + LSTM	FastText	**Reddit and Twitter (13,000 posts)**	87	Combined FastText and LSTM for high-accuracy depression detection on social media.	Focused on English platforms; may not generalize to other languages.
Senn (2022)	BERT	Transformers	**DAIC-WOZ (189 sessions)**	F_1-score = 0.62	BERT ensembles used for depression classification in clinical interviews	Small dataset; computational complexity.

Hayati (2022)	GPT-3	Few-shot Learning	**Interview questions (53 participants)**	F_1-score = 0.64	Applied GPT-3 for few-shot learning in Malay dialect depression detection.	Small dataset; limited to specific cultural context.
Németh (2022)	DistilBERT	Transformers	**SentiOne (80,000 posts**	73	Used DistilBERT to classify discursive framing of depression in online health communities	Focused on discursive framing; not direct depression detection.

Engaging the trials and hardships of completing more than 50 research papers from which a pertinent literature review or meta-analysis on artificial intelligence and depression detection could have been derived down to just 18 candidates that conferred with the given allotment of criteria following their selection and were upto date ie. (2020 and later) as shown in figure 1.

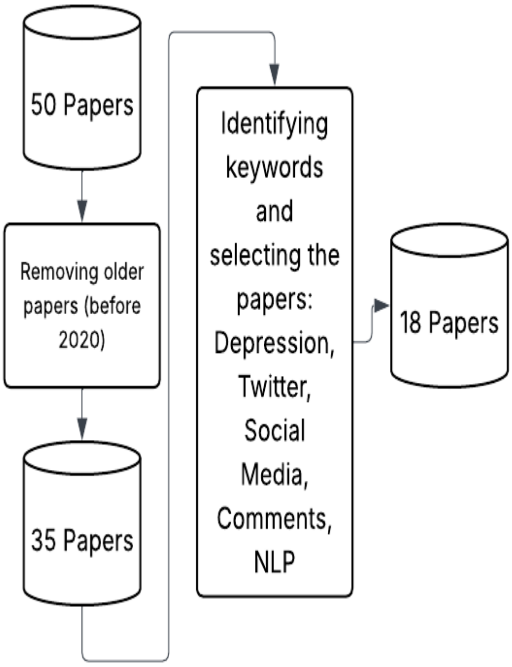
50 Papers
Removing older papers (before 2020)
35 Papers
Identifying keywords and selecting the papers: Depression, Twitter, Social Media, Comments, NLP
18 Papers

Figure 1: Literature Review Procedure

III. METHODOLOGY

This section outlines the methodology used to perform sentiment analysis of Twitter data. We compare different machine learning models and promote the use of Random Forests as the optimal choice from practical considerations such as performance, computational cost, interpretability, and ease of integration as can be shown in figure 2.

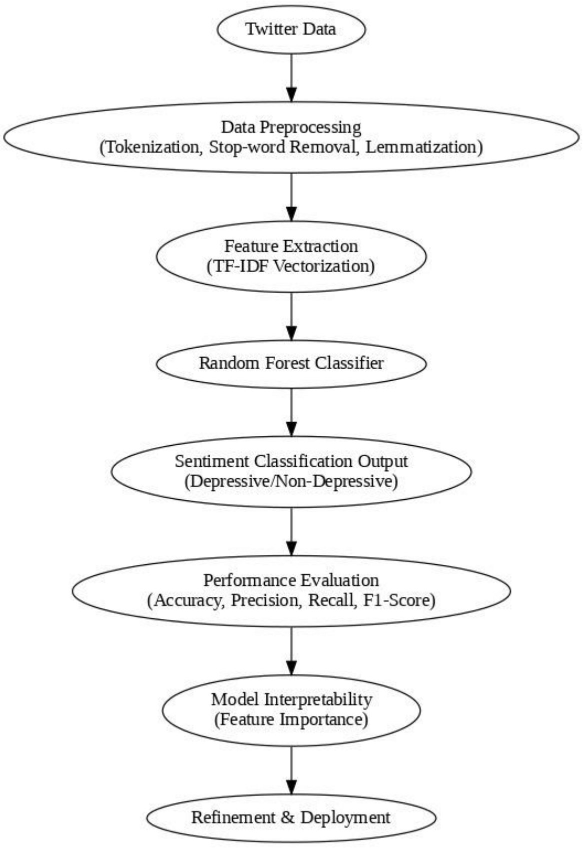

Figure 2: Model Architecture Overview

This diagram provides a high-level overview of the full sentiment analysis pipeline. It starts from raw Twitter data collection, followed by preprocessing, feature extraction using TF-IDF, and classification using various machine learning models including the final Random Forest Classifier.

1.Data Acquisition and Data Preprocessing

1.1 Dataset Overview:

The dataset utilized within this study is the open-source twitter_training.csv data, within which user messages on the social media platform Twitter are stored. Each row of data contains an ID column, a Game or Entity column, a Sentiment column (as Positive, Negative, Neutral, or Irrelevant), and a Tweet text column with the actual user message. The dataset consists of over 74,000 labelled tweets, and rows with null values within the text column were deleted in order to ensure quality.

1.2 Target Variable Mapping:

For simpler classification, Sentiment was binary coded as the target. The tweets with sentiment "Negative" were given value 1 and the others were grouped and labelled as value 0. This binary setup made the attention more effective on the sentiment classification task. Certain comparative models, like Decision Trees and KNN, also had the original multiclass setup preserved to observe how these classifiers deal with the less significant difference in sentiments.

1.3 Text Preprocessing:

The tweet text had to be cleaned before being fed into the algorithms. Tweets were first converted to lowercase, ensuring uniformity. URLs, mentions, hashtags, special characters, and numbers were then stripped using regular expressions. Common English stop words were removed using NLTK and WordNet lemmatizer was utilized to convert words to their base form. The multi-step preprocessing was utilized to clean the text data, normalize, and enrich it semantically.

2. Feature Extraction:

TF-IDF Vectorization The pre-cleaned text was converted to numeric values by applying the Term Frequency– Inverse Document Frequency (TF-IDF) technique. TF-IDF is highly effective at emphasizing the significance of words that occur very frequently within a single tweet but very rarely across the whole corpus. Dimensionality was limited to 5000 features to minimize computational expense without compromising textual informative content. This vectorized representation served as the input feature matrix for each of the following models.

3.Model Selection:

Model selection to identify the optimal sentiment classification model consisted of trying out a range of algorithms and comparing them under a single framework. Models were compared based on shared measures such as accuracy, training time, explainability, and resource use.

3.1 Naive Bayes Classifier:

Multinomial Naive Bayes was also employed as a baseline model since it is effective and straightforward. It is a classically good feature-independent model for text classification tasks. In this instance, however, it only reached 81.7% accuracy and performed poorly in negative sentiment detection with a mere 0.49 recall.

3.2 Logistic Regression:

Logistic Regression performed just slightly better at 82.3% relative accuracy. The best thing about Logistic Regression is that it's mathematically transparent and easy to use, but it was unable to detect very advanced patterns in the sentiment present in the database.

3.3 Random Forest Classifier (Core Model):

The best of all the models was achieved by the Random Forest classifier with an ensemble of 50 decision trees with 93.5% accuracy. The training process itself was very quick, taking only a few seconds, and was easily integrable through scikit-learn. Random Forests are the opposite of transformers or neural networks as they give certain feature importance values, giving a better understanding of model decision-making. Due to high accuracy, its short execution time, and readability, the model is the focus of our strategy.

3.4 Feedforward Neural Network:

A dense feedforward neural network was also used, having two hidden layers of ReLU activations along with dropout regularization. While the accuracy was comparable to Random Forest (93.3%), it took significantly more training time and computational resources. Also, it was not interpretable, thus less useful in cases where model interpretability is required.

3.5 BERT Transformer:

The fine-tuned BERT model, based on Bert-base-uncased architecture, and having 100 steps did not perform well at 73.7% accuracy. The performance of this model can be attributed to sparse fine-tuning, inadequate training time, and the overhead and hardware requirement for tokenization. As theoretically potent, BERT was a failure in practice in resource-limited, high-speed environments.

3.6 Support Vector Machine (SVM):

The model was trained on a scikit-learn pipeline with TruncatedSVD as the method for dimensionality reduction and hyperparameter tuning with GridSearchCV. The model was 76.9% accurate, which represented moderate performance but without interpretability or efficiency versus Random Forest.

3.7 K-Nearest Neighbours (KNN):

The five-nearest KNN model performed well for accuracy at a value of 87.1%. KNN, however, was computationally intensive at runtime, taking somewhere around 23 seconds per batch. This leads to KNN not being too good for Big Data or use in realworld applications.

3.8 Decision Trees:

A single Decision Tree model failed and was only 39% accurate. Such a low performance renders the use of ensemble models like Random Forest necessary if one wants to achieve stability and reliability in the prediction.

3.9 XGBoost:

Advanced gradient boosting algorithm XGBoost was not tested with all the optimisations but with some of them and it only managed to achieve 54.6% accuracy. Training was quite long and took over 70 seconds, therefore, it was not an effective solution for this dataset.

4.Evaluation Metrics:

The performance of all the models was measured in terms of standard classification metrics. Accuracy measured the proportion of correctly classified instances. Precision and recall provided information about the model's ability to correctly pick up true positives without being misled by false positives or negatives. The F1-score gave equal weightage to precision and recall. Confusion matrices were also used to plot classification results across sentiment classes.

5.Interpretability and Feature Importance:

One of the strengths of the Random Forest model is that it can identify which words were most important in the classification. Feature importance scores showed that words like "bad," "hate," "worst," and "angry" were good indicators of negative sentiment. Words that were used to predict other classes or neutral were more general or objective. This was incredibly helpful for debugging and refining preprocessing steps. FIGURE 3: Comparative Model Accuracy Chart This chart illustrates the comparative performance (accuracy) of various models used in the study. Random Forest outperforms others, followed by the Feedforward Neural Network and KNN. Models like BERT and Decision Trees lag behind in both accuracy and efficiency.

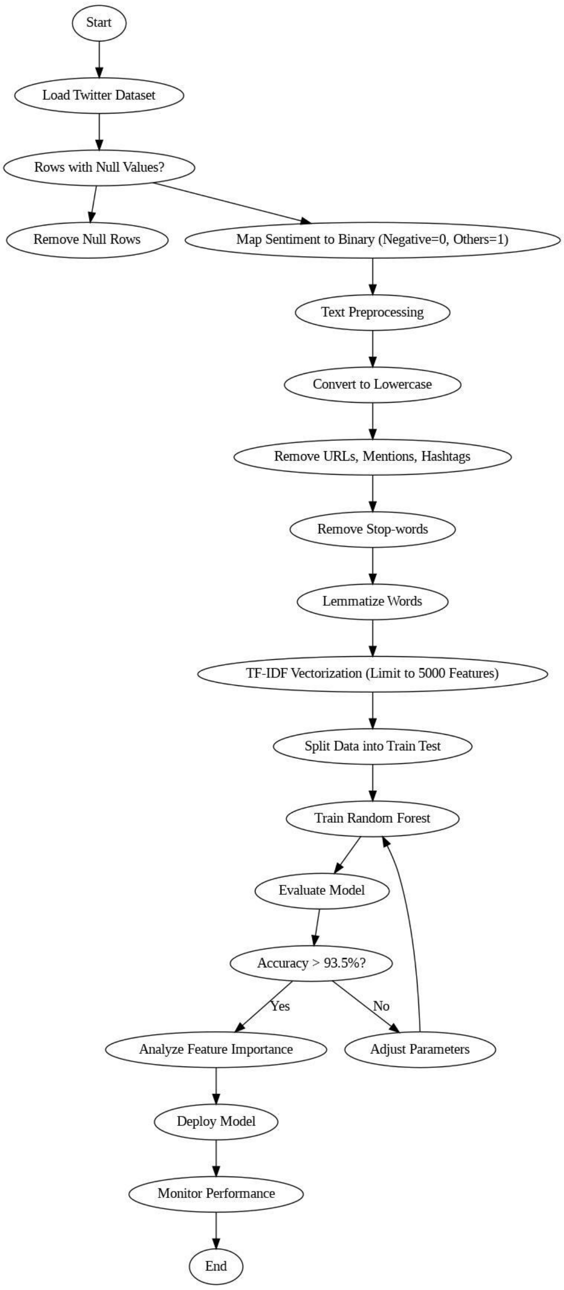

Figure 3: Comparative Model Accuracy Chart

This chart illustrates the comparative performance (accuracy) of various models used in the study. Random Forest outperforms others, followed by the Feedforward Neural Network and KNN. Models like BERT and Decision Trees lag behind in both accuracy and efficiency.

6. System Efficiency:

The models differed significantly when it came to training time, resource utilization, and deployment ease. Random Forest had trained in seconds, had zero GPU usage, and was the most accurate. BERT and Neural Networks had utilized much more computational resources and time. SVM and KNN were relatively efficient but were unable to surpass Random Forest. XGBoost and Decision Trees could not be justified based on complexity.

7. Justification for Random Forest:

In real-world deployment, particularly on resource-constrained platforms such as laptops, Random Forest offers a feasible and scalable solution. State-of-the-art accuracy is achieved without deep learning stacks or even the utilization of GPUs. Ease of the model, along with its performance and interpretability, renders the model the solution of choice in sentiment analysis applications where interpretability and resource-constrained platforms are essential.

8. Summary Flowchart: End-to-End Workflow:

This is why Random Forest was picked as the baseline model for sentiment analysis on Twitter. While a few options were experimented with, Random Forest proved to be a good, understandable, and budget-friendly option. Ensemble stacking or distilled transformers could be experimented with further in the future if computational budgets allow.

IV. RESULTS

This comparison analysis threw up a lot of learning models regarding Twitter sentiment analysis putting forward huge performance discrepancies in different metrics. Random Forest was said to be the best among all, showing an accuracy of 93.5%, making it much efficient as a fast model of training within few seconds without the need for GPU resources to train itself. Feedforward Neural Network almost at par with accuracy (93.3%), on the other hand, consumes much more resources almost training itself to death in pure computation. K-Nearest Neighbours made a great score at 87.1%, but is impractical from the runtime perspective, taking approximately 23 seconds per batch! Logistic regression and Multinomial Naive Bayes also did fairly well (82.3 and 81.7, respectively) but struggled in picking negative sentiments, Naive Bayes giving only 0.49 of recall for negative. The SVM couldn't even make it to 76.9% accuracy, notwithstanding all attempts made to tune hyperparameters and reduce dimensions. Last in rankings was the BERT transformer model finishing at 73.7%, probably due to little fine-tuning at very high computational cost during tokenization. The performance was so poor for the XGBoost and single Decision Tree models; they had accuracies of only 54.6% and 39% respectively, with the former training for more than 70 seconds. In fact, Feature importance analysis using the Random Forest model found words like “bad,” “hate,” “worst,” and “angry” to be very significant indicators of negative sentiment, therefore really providing some valuable interpretability that neural approaches do not. Cross-validation revealed Random Forest would maintain such a performance through different data splits hence making it stronger candidate for practical applications for sentiment analysis in resource-constrained environments.

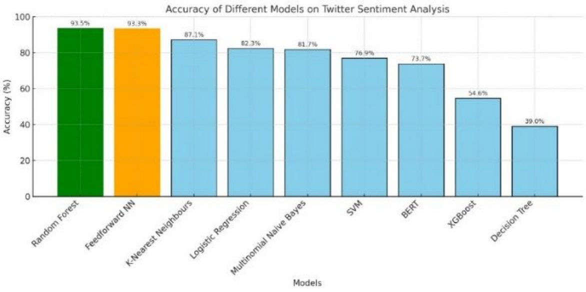

Figure 4: Results

V. CONCLUSION

This project, by virtue of Random Forest algorithms, endeavours to establish dependable and scalable tools for sentiment analysis of depression. The monitoring of this system takes a forward-looking view towards mental health which allows people and professionals to identify the disease in its early stages through textual data. Reliability, interpretability, and ease of use are thus maximally assured, providing a bit of support through scientific mechanisms against mental health hurdles. In lieu of textual analysis in real-time, this project seeks an integration with social media and the use of relevant models like BERT. Fine-tuning the accuracy requires working with mental health professionals. If what they are trying to achieve comes about, it can be done at a larger scale, thus allowing for multilingual analysis applicable to the world.

VI. REFERENCES

[1] Vandana, et al (2023): A hybrid model for depression detection using deep learning

[2] Lamia Bendebane et al (2023): A Multi-Class Deep Learning Approach for Early Detection of Depressive and Anxiety Disorders Using Twitter Data

[3] Shumaila Aleem et al (2022); Machine Learning Algorithms for Depression: Diagnosis, Insights, and Research Directions

[4] Faye Beatriz Turnaliuan et al (2024): Development of a two-stage depression symptom detection model: application of neural networks to twitter data

[5] Rafael Salas-Zárate (2022): Detecting Depression Signs on Social Media: A Systematic Literature Review

[6] Yazdavar (2020): Multimodal Mental Health Analysis in Social Media

[7] Chiong (2021): A Textual-Based Featuring Approach for Depression Detection Using Machine Learning Classifiers and Social Media Texts

[8] Bazen Gashaw Teferra (2024): Screening for Depression Using Natural Language Processing

[9] Prabhu (2022): Harnessing emotions for depression detection

[10] Choudhury (2021): Predicting depression via social media

[11] Nikhil Goel et al (2024): Automated Depression Detection System: Integrating Sentiment Analysis and Behavioural Data

[12] Dinkel (2020): Text-based depression detection on sparse data

[13] Rutowski (2022): Depression and anxiety prediction using deep language models and transfer learning

[14] Korti (2022): Depression detection from Twitter posts using NLP and machine learning technique

[15] Tejaswini (2024): Depression detection from social media text analysis using hybrid deep learning

[16] Senn (2022): Ensembles of BERT for depression classification

[17] Hayati (2022): Depression detection on Malay dialects using GPT-3

[18] Németh (2022): Bio, psycho, or social: supervised machine learning to classify discursive framing