**Multiclass Mental‑Health Sentiment Analysis: A Comparative Evaluation of Machine‑Learning to Transformers**

Kai-Kian Toh2, a), Lee-Ying Chong1 ,2, b), Siew-Chin Chong1, 2, c), Pey-Yun Goh1, 2, d)

1Centre for Advanced Analytics, COE of Artificial Intelligence

2Faculty of Information Science and Technology   
Multimedia University, Jalan Ayer Keroh Lama, 75450 Bukit Beruang, Melaka, Malaysia

b) Corresponding author: lychong@mmu.edu.my

a) kaikian1105@gmail.com

c) chong.siew.chin@mmu.edu.my

d) pygoh@mmu.edu.my

**Abstract.** Sentiment analysis of mental-health expressions in social-media posts can help identify individuals at risk and make sure they can receive the necessary support. This study performs an evaluation of nine classification architectures on a seven-label dataset comprising 53385 expert-labeled posts including Normal, Depression, Suicidal, Anxiety, Bipolar, Stress and Personality-disorder. Four traditional machine-learning techniques, logistic regression, random forest, XGboost, and extra-trees achieved accuracy ranging from 90% to 95% on the TF-IDF features, proving the efficient power of the sparse representations coupled with the ensemble classifiers. Three deep-learning networks, CNN, BiLSTM, and CNN-BiLSTM, on Word2Vec embeddings all reached the accuracy into the mid-90 percent range, confirming that semantic representations and sequence modeling outperform count-based approaches. Two transformer-based models, BERT and BERT-BiLSTM use contextual token embeddings to capture sentence-level nuance and achieve the highest results with 99.49% accuracy and 99.51% F1 score. All the models were fit on the same preprocessing steps such as text cleaning, stemming or lemmatization, and class balancing, as well as through adopting the same evaluation protocol for fair comparison. Comparative analysis of feature-extraction schemes reveals progressive gains from TF-IDF to Word2Vec to transformer embeddings and underscores the effectiveness of hybrid architectures in capturing complex multi-class mental-health sentiment.

**INTRODUCTION**

Social media platforms can be a rich resource for personal expression, offering potential signs of a shift in mental-health status. In this study, a seven-label corpus of expert-annotated posts distinguishes Normal, Depression, Suicidal, Anxiety, Bipolar, Stress and Personality-disorder sentiments. The goal is to understand which model classes are best suited for different fine-grained categories and provide a fair benchmark with same preprocessing and evaluation protocols. Nine models spanning three families are compared on this corpus. Classical machine-learning approaches employ TF-IDF vectorization with logistic regression, random forest, XGBoost, and extra-trees classifiers. Deep-learning models use CNN, bidirectional LSTM, a combination of both CNN-BiLSTM and map sequences of texts into Word2Vec embeddings. Transformer-based methods fine-tune a pre-trained BERT encoder alone or with an added BiLSTM layer. Contributions include head-to-head model comparison, analysis of feature-extraction impacts, and per-class evaluation against prior logistic regression benchmarks.

**LITERATURE REVIEW**

Earlier work involved feature engineering to convert text to numeric vectors like bag-of-words or term-frequency inverse-document-frequency. These vectors were then labeled using classification algorithms including support-vector machine, logistic regression, naive Bayes, decision tree, random forest, k-nearest, and extreme gradient boosting. For example, Khan et al. [1] extracted count-vector features and applied 6 classifiers on Bengali social media text and obtained 86.7% accuracy. Tusar and Islam [2] used the TF-IDF features from 14000 US- airline tweets, and trained support-vector machine (SVM), logistic regression, naive Bayes and random forest and the result showed that SVM and logistic regression both achieve 77% percentage. Hitesh et al. [3] substituted TF-IDF by Word2Vec embeddings and applied them to random forest on election tweets and achieved 86.9% accuracy to demonstrate context-aware embeddings bring better features.

With the advent of deep-learning researchers moved to neural architecture that learn hierarchical representations from raw text. Singh et al. [4] applied multiple methods including support-vector machine (SVM), random forest, gradient boosting and XGBoost to airline tweets, and obtained 83.7% accuracy by SVM, and they also emphasized that even deep-learning frameworks often benefit from classic features. Kapali et al. [5] used Long Short-Term Memory (LSTM) and bidirectional LSTM on 4000 Bengali comments and achieved 97.3% accuracy for bidirectional LSTM. Kavitha et al. [6] applied NLTK for preprocessing and TF-IDF features with random forest for the challenging task of classifying 32000 debate tweets into 15 categories and obtaining 95% accuracy. These results verify that sequence networks outperform traditional techniques in moderate-size corpora.

Transformer-based models now lead the field by encoding full sentence context via self-attention. Fine-tuning pretrained transformers on a wide variety of benchmarks yield state-of-the-art results. Wang et al. [7] fine-tuned a Chinese BERT-based on one million COVID-related Weibo posts for three-class sentiment, achieving the highest accuracy of 75.7% when compared with SVM, naive Bayes, CNN and LSTM baselines. Haque et al. [8] utilized RoBERTa to identify suicidal ideation in posts from the SuicideWatch forum and achieved 95.2% accuracy, which outperformed BiLSTM that achieved 84.4%. Joshy and Sundar [9] evaluated BERT, DistilBERT and RoBERTa on two English Twitter data sets and they obtained BERT achieved the highest accuracy of 92.8% and 90.4%, respectively indicating the strength of transformers in cross-domain.

Narynov et al. [10] gathered depressive VKontakte posts and have tried TF-IDF and Word2Vec features to predict depressive and non-depressive posts with random forest and gradient boosting. The results show that random forest with TF-IDF achieved the highest accuracy, which is 96.3% accuracy. Raj and Banjac [11] experimented with Reddit posts annotated as depressed and not depressed and fine-tuned BERT on the annotated portion but compared it to an autoencoder on unlabeled data. Then BERT got 91.9% for the accuracy, 93% of F1-score, while autoencoder got 84.3%. Hybrid models that combine transformer embeddings with sequence architectures yield the strongest multi-class performance. Mahmud et al. [12] used GloVe embeddings with convolutional neural networks and bidirectional LSTM on a Bangla data set of 2272two texts. The results show that CNN with GloVe achieved 99.4% accuracy.

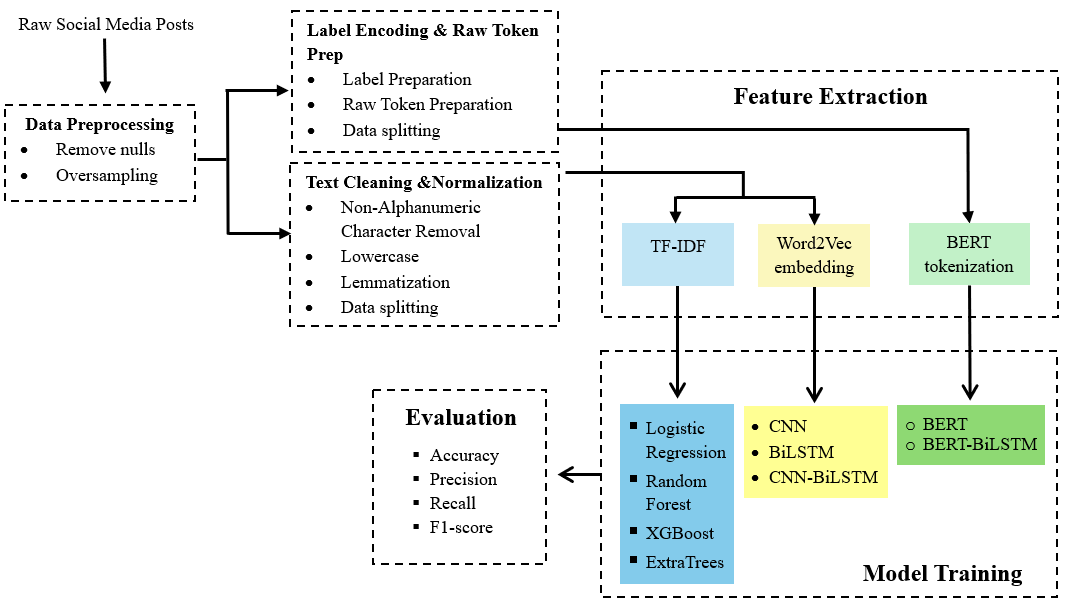
Jain and Rathour [13] applied logistic regression to the same Kaggle “Sentiment Analysis for Mental Health” dataset using TF-IDF and basic polarity features. Their preprocessing pipeline performed lowercasing, punctuation and URL removal, and stopword filtering but did not include stemming or lemmatization, and it omitted any strategy for class balancing without noting its impact on rare categories. Consequently, their model reached 87% accuracy but exhibited poor recall on minority classes such as Suicidal. This review supports the finding that enhancements in data preprocessing, class balancing and model optimization are important in improving logistic-regression performance.

**METHODOLOGY**

Figure 1 presents the proposed research methodology. Raw social media posts are collected and preprocessed by removing entries with missing values. To address class imbalance, resampling is applied. The process then splits into two pipelines: one for label encoding and token preparation, and another for text cleaning and normalization. Both lead to feature extraction. Traditional machine learning models (Logistic Regression, Random Forest, XGBoost, ExtraTree) use TF-IDF, while BiLSTM, CNN, and CNN-BiLSTM use Word2Vec. BERT and BERT-BiLSTM rely on BERT tokenization. All nine models are evaluated using accuracy, precision, recall, and F1-score.

## **Data Preprocessing**

At the beginning, any posts missing text or labels are dropped so only complete samples remain. This step prevents gaps from harming the model’s learning and makes sure the analysis stays accurate. After removing empty entries, the research checks how many items are left and notices that some sentiment labels appear much more often than others. To fix this imbalance, smaller classes are randomly oversampled until each label group matches the size of the largest one. Once all labels have equal counts, the balanced pieces are joined back together, creating a clean dataset ready for reliable training and testing.

**FIGURE 1**.Methodology pipeline of the mental-health sentiment classification

## **Text Cleaning and Normalization**

Each post is cleaned by removing anything that is not a letter, number, or space, and then converting all letters to lowercase. This removes punctuation, emojis, and odd symbols so they do not clutter the vocabulary, and it makes “Hi” and “hi” count as the same word. After cleaning, lemmatization turns each word into its base form such as “changes,” “changed,” and “changing” all become “change.” This step shrinks the vocabulary and groups related words together. Finally, the cleaned text is split into training and testing sets (80% train, 20% test) using a fixed seed for repeatable results.

## **Label Encoding and Raw Token Preparation**

In this stage, each unique sentiment label is turned into a number by sorting the labels and assigning consecutive indexes. Those numbers become one-hot vectors, where only the correct label position is marked with a 1. For text input, the cleaned sentences go straight into BERT’s tokenizer, which creates token IDs and attention masks without extra lemmatization or stopword removal. The tokens and one-hot labels form a tf.data.Dataset, which is shuffled, then split into training (80%), validation (10%), and test (10%) sets. Training and validation data are batched in groups of 32 for fast GPU use, while the test set stays unshuffled. This setup keeps each data split clear and reproducible for model building and evaluation.

## **Feature Extraction**

This analysis transforms each preprocessed statement into numeric form in one of three ways, depending on the model category. Classical machine‑learning methods build a high‑dimensional sparse vector of token weights, using a term‑frequency inverse‑document‑frequency (TF-IDF) vectorizer over unigrams and bigrams. This restriction controls memory usage and helps prevent overfitting. For deep‑learning models, this study uses the Keras tokenizer and transform a text to integer index sequences, padding each sequence to the length of 100, and then map each integer index to a 300‑dimensional Word2Vec embedding (pretrained on Google News and fine‑tuned during training time). This means a 100\*300 dense representation for each post. Transformer‑based models structure text into BERT‑base‑uncased tokenizer to split text into subword units, insert special classification tokens, and generate fixed‑length arrays of input IDs and attention masks at length 128, these serve as input to the BERT encoder for contextual embedding.

## **Model Training**

The traditional machine learning methods had text data processed with TF-IDF. The random forest classification model is seeded with a predetermined random state for reproducibility and trained directly from the TF-IDF features of the training corpus. Finally, the model was used to predict the test set. Similarly, the extra-trees classifier was fitted on the same TF-IDF vectors. The model used random tree splits to mine different patterns in the data. As XGBoost model doesn’t take string label, label encoding is needed. Labels of training and testing set were converted to numeric values before being used for training. The model was set up with a random seed and a log-loss as metric for the multi-class classification. After training, the corresponding numeric predictions were inverse transformed back to their actual text form using label encoder. This made the results intuitively commensurate with the other models. Logistic regression was tuned using hyperparameters to find the best configuration for performance. A grid search was conducted over different regularization strengths using five-fold cross-validation. The best model selected based on accuracy was then trained on the whole TF-IDF training set. After training, it made predictions on the test set that were evaluated.

Deep learning models began by converting each post into a sequence of 100-word indices, each mapped to a 300‑dimensional Word2Vec vector. The convolutional network applied multiple layers of one‑dimensional filters to these sequences to learn local phrase features, while the bidirectional LSTM processed the same sequences in forward and reverse order to capture long‑range dependencies. In the combined CNN‑BiLSTM model the outputs of both branches were merged before the final classification layer to exploit both spatial and temporal information. All networks were trained with the Adam optimizer on a categorical cross‑entropy objective, using a batch size of 128. Training stopped automatically when validation loss failed to improve for three consecutive epochs, and the learning rate was reduced by half after two such plateaus. The convolutional and hybrid models ran for up to 20 epochs and the pure LSTM for 10, with the best‑performing weights saved for evaluation.

Transformer‑based models fine‑tuned the BERT‑Base‑Uncased encoder on token sequences of 128. Inputs included both token identifiers and attention masks to signal real versus padded positions. A small learning rate of 0.00005 and a batch size of 32 ensured stable updates during gradient descent. Dropout was applied before the final classification layer to guard against overfitting, and training halted when validation accuracy did not improve for two epochs. For the hybrid variant a bidirectional LSTM layer was added on top of BERT’s final hidden states, enabling additional sequence modeling. The entire network was then fine‑tuned end‑to‑end under the same stopping and checkpointing protocol.

## **Evaluation**

Model performance is evaluated using four widely accepted classification metrics, each offering distinct insights into predictive effectiveness. Accuracy, defined in Equation (1), represents the overall proportion of correctly classified samples relative to the total number of test instances, providing a general sense of model correctness. Precision, as shown in Equation (2), quantifies how many of the samples labelled as positive by the model are truly positive, which is especially critical in imbalanced or high-risk classification tasks. Recall, outlined in Equation (3), captures the proportion of actual positive cases that the model successfully identifies, reflecting its sensitivity to the positive class. Equation (4) defines the F1-score, the harmonic mean of precision and recall, which balances both metrics into a single value, particularly useful when trade-offs between false positives and false negatives must be considered. All metrics are computed on a held-out test set using a weighted averaging scheme to ensure fair representation of all sentiment classes regardless of their frequency in the dataset.

|  |  |  |
| --- | --- | --- |
|  | Accuracy | (1) |
|  | Precision | (2) |
|  | Recall | (3) |
|  | F1-score | (4) |

**RESULT AND DISCUSSION**

## **Dataset**

The analysis used the “Sentiment Analysis for Mental Health” dataset from Kaggle [14], which contains 53,385 social media posts. Each post was manually labelled into one of seven mental health sentiment categories, making the dataset suitable for training and evaluating sentiment classification models in the mental health domain.

Table 1 shows the distribution of various mental health statuses in the dataset. The label distribution is highly imbalanced, with the three largest classes (Normal, Depression, and Suicidal) accounting for over 80% of the data, while the remaining classes (Anxiety, Bipolar, Stress, and Personality Disorder) together comprise less than 20%. Word clouds illustrate the most frequently occurring terms in social media posts related to different mental health conditions, including Anxiety, Normal, Suicidal, Depression, Stress, Bipolar, and Personality Disorder. Larger words indicate a higher frequency of use in each category, highlighting common feelings and concerns such as "feel," "want," "know," and "time" across various emotional states, as shown in Figure 2.

**TABLE 1.** Distribution of Mental Health Statuses

|  |  |
| --- | --- |
| **Status** | **Number of Post** |
| Normal | 16 040 |
| Depression | 15 094 |
| Suicidal | 10 644 |
| Anxiety | 3 623 |
| Bipolar | 2501 |
| Stress | 2296 |
| Personality Disorder | 895 |

|  |  |  |  |
| --- | --- | --- | --- |
| A close up of words  AI-generated content may be incorrect. |  | A close-up of words  AI-generated content may be incorrect. | A close-up of words  AI-generated content may be incorrect. |
| 1. WordCloud for Normal | 1. WordCloud for Depression | 1. WordCloud for Suicidal | 1. WordCloud for Anxiety |
| A close-up of words  AI-generated content may be incorrect. | A close-up of words  AI-generated content may be incorrect. | A close-up of words  AI-generated content may be incorrect. |  |
| 1. WordCloud for Bipolar | 1. WordCloud for Stress | 1. WordCloud for Personality Disorder |  |

**FIGURE 2**.WordClouds Highlighting Key Terms for (a) Normal, (b) Depression, (c) Suicidal, (d) Anxiety, (e) Bipolar, (f) Stress, and (g) Personality Disorder

## **Experimental Setup**

All experiments use a balanced data split of 80% for training and 20% for testing. For classical models (logistic regression, random forest, XGBoost, and extra trees), hyperparameter selection is performed using 5-fold stratified cross-validation on the training set. Deep learning and transformer models are trained with early stopping on a validation subset (10% of the training data) to prevent overfitting. Neural networks use the Adam optimizer, are trained with a batch size of 32 or 128 when possible and apply learning rate reduction on plateau. All training and evaluation are conducted on a single GPU-equipped workstation to ensure precise timing control and reproducibility of results.

## **Performance Comparison among Models**

Table 2 reveals clear stratification in model effectiveness. The transformer-based approaches achieve the highest scores, with BERT reaching 99.5% accuracy and F1-score, and BERT-BiLSTM close behind at 99.3%. This represents a substantial improvement of approximately 4% over the best classical methods (Random Forest and Extra Trees at around 95%). Among non-transformer models, tree-based ensembles outperform both linear models and neural sequence models, indicating that TF-IDF features combined with ensemble voting remain highly competitive when trained on balanced data. The relatively close performance of CNN, BiLSTM, and CNN-BiLSTM models (all within the 94% to 95% range) suggests that convolutional and recurrent feature extractors using Word2Vec embeddings capture complementary patterns but still fall short of the contextual understanding provided by transformer-based models.

**TABLE 2.** Test-set performance of evaluated models

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Models** | **Feature Scheme** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-score (%)** |
| XGBoost | TF-IDF | 90.97 | 90.91 | 90.97 | 90.84 |
| Logistic Regression | 93.71 | 93.63 | 93.71 | 93.64 |
| Extra-Trees | 95.03 | 95.04 | 95.03 | 95.01 |
| Random Forest | 95.21 | 95.18 | 95.21 | 95.19 |
| CNN | Word2Vec Embeddings | 94.65 | 94.63 | 94.66 | 94.63 |
| BiLSTM | 94.24 | 94.22 | 94.24 | 94.22 |
| CNN-BiLSTM | 94.57 | 94.57 | 94.57 | 94.61 |
| BERT | BERT Tokenizer | 99.49 | 99.51 | 99.51 | 99.51 |
| BERT- BiLSTM | 99.33 | 99.45 | 99.45 | 99.45 |

## **Comparison of Feature-Extraction Schemes** **on Random Forest**

A comparison of random forest performance across three feature extraction approaches is presented in Table 3. Random forest was selected due to its higher accuracy compared to other classical models and its fast-training time. When using TF-IDF vectors, the model achieved an accuracy of 95.21% and an F1-score of 95.19%. Switching to Word2Vec embeddings reduced the accuracy to 93.97% and the F1-score to 93.93%, indicating that while static dense embeddings capture semantic information, they lack the discriminative term weighting provided by tree-based ensembles. Using BERT’s tokenizer alone resulted in an accuracy of 93.55% and an F1-score of 93.49%, suggesting that subword-level features require contextual encoding to achieve optimal performance with non-neural classifiers. his does not imply that BERT tokenization is inferior; rather, it highlights the need for contextual encoding to fully leverage its capabilities.

**TABLE 3.** Comparison of feature-extraction schemes on random forest

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature Scheme** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-score (%)** |
| TF-IDF | 95.21 | 95.18 | 95.21 | 95.19 |
| Word2Vec Embeddings | 93.97 | 93.91 | 93.97 | 93.93 |
| BERT Tokenizer | 93.55 | 93.46 | 93.55 | 93.49 |

## **Comparison Performance with Prior Logistic Regression Work**

Table 4 presents a comparison between the results of logistic regression from the reference study [13] and the improved logistic regression applied in this study. The revised model shows noticeable performance improvements across most sentiment categories compared to the reference study [13].

**TABLE 4.** Comparison against previous logistic regression study

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Method** | **Performance Metric** | **Anxiety** | **Bipolar** | **Depression** | **Normal** | **Personality Disorder** | **Stress** | **Suicidal** | **Overall Accuracy (%)** |
| Logistic Regression [13] | Precision (%） | 92.00 | 92.00 | 83.00 | 93.00 | 85.00 | 88.00 | 78.00 |  |
| Recall（%） | 91.00 | 90.00 | 83.00 | 96.00 | 81.00 | 86.00 | 77.00 | 87.00 |
| F1-Score（%） | 92.00 | 91.00 | 83.00 | 94.00 | 83.00 | 87.00 | 78.00 |  |
| Logistic Regression  (the proposed method) | Precision (%） | 95.30 | 98.88 | 97.93 | 96.89 | 85.31 | 95.56 | 86.01 | 93.71 |
| Recall（%） | 90.77 | 100.00 | 99.47 | 99.04 | 86.92 | 99.72 | 80.56 |
| F1-Score (%） | 92.98 | 99.44 | 98.69 | 97.95 | 86.10 | 97.60 | 83.19 |

Based on Table 4, the Normal, Depression, Bipolar, and Stress classes show substantial gains in both precision and recall, resulting in significantly higher F1-scores and improved classification reliability. The Anxiety and Personality Disorder classes also exhibit F1-score improvements, indicating better handling of these moderately represented categories. For the Suicidal class, both precision and recall increased compared to the previous work, leading to a higher F1-score. Overall, the logistic regression model in this study achieved an accuracy of 93.71%, compared to 87% in the earlier study. This demonstrates that enhancements in data preprocessing, class balancing, and model optimization have significantly improved logistic regression performance.

**CONCLUSION**

This study demonstrates that transformer-based models offer the best performance for seven-class mental-health sentiment analysis with accuracy and F1-scores near 100 %. Deep-learning models on Word2Vec embeddings provide strong mid-range results while classical TF-IDF ensembles establish reliable baselines. Enhanced data preprocessing, class balancing, and model optimization substantially improve logistic regression accuracy from 87 % to 93.71 % and yield consistent per-class gains over prior work. These findings set a new reference for multi-class mental-health sentiment detection and highlight the value of contextual and hybrid modelling for early intervention applications. Future work can be extended to include data augmentation, multi-modal inputs or real-time deployment to improve the early detection of mental-health issues on social media.

**REFERENCES**

1. Md.R.H. Khan, U.S. Afroz, A.K.M. Masum, S. Abujar, and S.A. Hossain, “Sentiment Analysis from Bengali Depression Dataset using Machine Learning,” in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, (IEEE, Kharagpur, India, 2020), pp. 1–5.
2. M.T.H.K. Tusar, and Md.T. Islam, “A Comparative Study of Sentiment Analysis Using NLP and Different Machine Learning Techniques on US Airline Twitter Data,” in *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*, (IEEE, Khulna, Bangladesh, 2021), pp. 1–4.
3. M. Hitesh, V. Vaibhav, Y.J.A. Kalki, S.H. Kamtam, and S. Kumari, “Real-Time Sentiment Analysis of 2019 Election Tweets using Word2vec and Random Forest Model,” in *2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)*, (IEEE, Jaipur, India, 2019), pp. 146–151.
4. S. Singh, K. Kumar and B. Kumar, "Sentiment Analysis of Twitter Data Using TF-IDF and Machine Learning Techniques," in *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, (IEEE, Faridabad, India, 2022), pp. 252–255.
5. N. Kapali, T. Tuhin, A. Pramanik, Md.S. Rahman, and S.R.H. Noori, “Sentiment Analysis of Facebook and YouTube Bengali Comments Using LSTM and Bi-LSTM,” in *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, (IEEE, Kharagpur, India, 2022), pp. 1–6.
6. M. Kavitha, B.B. Naib, B. Mallikarjuna, R. Kavitha, and R. Srinivasan, “Sentiment Analysis using NLP and Machine Learning Techniques on Social Media Data,” in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, (IEEE, Greater Noida, India, 2022), pp. 112–115.
7. T. Wang, K. Lu, K.P. Chow, and Q. Zhu, “COVID-19 Sensing: Negative Sentiment Analysis on Social Media in China via BERT Model,” IEEE Access **8**, 138162–138169 (2020).
8. F. Haque, R.U. Nur, S.A. Jahan, Z. Mahmud, and F.M. Shah, “A Transformer Based Approach To Detect Suicidal Ideation Using Pre-Trained Language Models,” in *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, (IEEE, DHAKA, Bangladesh, 2020), pp. 1–5.
9. A. Joshy, and S. Sundar, “Analyzing the Performance of Sentiment Analysis using BERT, DistilBERT, and RoBERTa,” in *2022 IEEE International Power and Renewable Energy Conference (IPRECON)*, (IEEE, Kollam, India, 2022), pp. 1–6.
10. S. Narynov, D. Mukhtarkhanuly, B. Omarov, K. Kozhakhmet, and B. Omarov, “Machine Learning Approach to Identifying Depression Related Posts on Social Media,” in *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, (IEEE, Busan, Korea (South), 2020), pp. 6–11.
11. A. Raj, Z. Ali, S. Chaudhary, K.K. Bali, and A. Sharma, “Depression Detection Using BERT on Social Media Platforms,” in *2024 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, (IEEE, Kota Kinabalu, Malaysia, 2024), pp. 228–233.
12. Md.S. Mahmud, Md.T. Islam, A.J. Bonny, R.K. Shorna, J.H. Omi, and Md.S. Rahman, “Deep Learning Based Sentiment Analysis from Bangla Text Using Glove Word Embedding along with Convolutional Neural Network,” in *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, (IEEE, Kharagpur, India, 2022), pp. 1–6.
13. E. Jain, and A. Rathour, “Understanding Mental Health through Sentiment Analysis: using Machine Learning to Analyze Social Media for Emotional Insights,” in *2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI)*, (IEEE, Tirunelveli, India, 2024), pp. 782–786.
14. *Sentiment Analysis for Mental Health*. https://www.kaggle.com/datasets/suchintikasarkar/sentiment-analysis-for-mental-health. Accessed 15 June 2025.