# **Sentiment Analysis for Mental Health**

## Introduction:

Sentiment analysis in mental health examines text messages for indications of emotional discomfort, like anxiety and depression, using natural language processing (NLP). It assists in identifying those who might benefit from early mental health care by categorizing evidence as positive, negative, or neutral. This strategy is critical because it allows for tailored therapy, enhances treatment compliance, provides immediate support, and may even prevent more serious mental health problems.

Wankhade et al. (2022) says that it is the process of reading people's thoughts from online content. Making decisions is beneficial for people, companies, and governments. Although it has a wide range of uses, precisely identifying and classifying emotions is a difficulty. In this paper, various sentiment analysis techniques will be examined, their advantages and disadvantages are also compared.

## Literature Survey

Birjali et al. (2017) used machine learning and semantic sentiment analysis to predict suicide thoughts from Twitter. To make precise predictions, the study builds a vocabulary around suicide, gathers tweets, and uses machine learning techniques. However, the language-specific nature of the analysis, biases in training data, and human vocabulary construction present challenges.

How can the SS3 model be adapted to improve early depression risk detection while addressing ethical and privacy concerns? Burdisso et al. (2019) present the SS3 model, emphasizing its explainability, efficiency, and accuracy in the early diagnosis of depression risk using social media data. Although the model has shown promise in evaluations such as CLEF's eRisk2017 assignment, the study does not address privacy and ethical issues related to using social media data, and its practicality in real-world scenarios is yet unknown.

Fitri et al. (2019) study analyzes the Naïve Bayes, Decision Tree, and Random Forest algorithms to assess Twitter comments on Indonesia's anti-LGBT campaign, they discovered that Naïve Bayes achieved accuracy at 86.43%. Misspellings, slang, and the limitations of custom Indonesian dictionaries for stop word removal and stemming presented obstacles for the study's preprocessing and classification accuracy.

Babu and Kanaga (2021) emphasize the value of multi-class classification. They examine feature extraction techniques such as Word2Vec, N-Gram, and TF-IDF and stress the significance of clean data. The study concludes that although deep learning models frequently yield improved precision, they are dependent on high-quality data and demand significant processing resources.

How can custom lexicons improve the accuracy of sentiment analysis models in predicting suicidality in psychiatric clinical notes? George et al. (2021) study find clear distinctions in sentiment between cases who are suicidal and those that are not, especially in terms of thought process and content. To improve the analysis, a new vocabulary was created. The results showed that random forest and logistic regression classifiers had good accuracy. However, a tiny, unbalanced dataset limits the investigation and makes cross-validation more difficult.

Yu et al*.* (2021) study on Sina Weibo from 2014 to 2017 reveals a rise in support and mentions of social support. The study's non-representative sample and emphasis on well-liked posts may have restricted its ability to detect key issues using text analysis, which could leave out more widely held opinions.

How can the analysis of financial news sentiment contribute to understanding its impact on public mental health? Alanazi et al*.* (2022) examine how financial news on The Guardian affects people's mental health by analyzing their sentiment. They discovered that the Single Layer Convolutional Neural Network (SLCNN), with an accuracy of 93.9%, had the best performance using machine learning approaches. Although the study emphasizes the value of sentiment analysis in comprehending financial and mental health policies, its applicability to other news kinds and long-term trends may be limited due to its timing and dataset restrictions.

How can automated sentiment analysis models be refined to better incorporate context-specific vocabularies in medical settings? Huisman et al. (2024) compare automated sentiment analysis with human rater evaluations to measure its efficacy in treating eating disorders. Because it lacked context-specific terminology, automated analysis did less well even if it exhibited moderate agreement with human raters. The paper highlights shortcomings such a small sample size and possible rater bias and recommends improvements, particularly in incorporating specialized terminology.

## Smart Objectives:

SPECIFIC: In this study, sentiment analysis will be used to analyze clinical notes and social media to identify mental health problems such as depression, anxiety, and suicide thoughts. A study of the literature and an exploratory data analysis on a Kaggle dataset will come first. Then, both deep learning (LSTM, RNN) and classical machine learning (Naïve Bayes, Logistic Regression) models will be implemented.

MEASURABLE: The project will analyze a few chosen peer-reviewed articles and provide a summary of important techniques with an emphasis on mental health disorders and datasets. The standard machine learning models and deep learning models targeting maximum F1-score, precision, recall, and ROC-AUC scores.

ACHIEVABLE: Kaggle datasets will be used in the project, and time will be allotted for a thorough literature review. Python libraries will be used for data analysis and visualization, while Scikit-learn will be used for model training. TensorFlow will be used to create deep learning models, and GPU acceleration will be used for maximum efficiency.

RELEVANT: To conduct a focused analysis, the research focuses on mental health issues and datasets. Reviewing existing research will inform model selection, while early data analysis will refine the project scope. To improve performance measures, a baseline between deep learning and machine learning models will be established.

TIME LIMITED: The project will start with a precise definition and scope, then a thorough background check. In a few days, preliminary data analysis will be finished, resulting in a report. Deep learning models will be developed and assessed after machine learning models and are evaluated over a predetermined period. Preliminary testing, and performance assessment are included in the last stage.

## Comment on the Dataset:

The data set is taken from Kaggle: [Sentiment Analysis for Mental Health (kaggle.com)](https://www.kaggle.com/datasets/suchintikasarkar/sentiment-analysis-for-mental-health/data).

There are 53,043 text entries in the Kaggle dataset for sentiment analysis in mental health, which is divided into seven labels including "Anxiety," "Depression," and "Suicidal." It consists of two columns: "statement" (text) and "status" (labels), with 362 missing values in the "statement" column. 51,073 distinct statements total in the sample; the longest statement has 32,759 characters and 6,300 words. Figure 1 illustrates the relationship between the statement length and number of words. The dataset is appropriate for training and sentiment analysis linked to mental health because of its significant size and variety.

A black and white squares with white text

Description automatically generated

**Figure 1 : Correlation between statement length and number of words.**

The dataset has seven unique as shown in Figure 2. However, the data is imbalanced, with "Normal" having 16,343 instances, while "Personality disorder" has only 1,077 instances. Other classes vary, with "Depression" having 15,404 instances and "Anxiety" having 3,841 instances.

A graph of a number of blue bars

Description automatically generated

**Figure 2 : Distribution of status column.**

The dataset's effectiveness arises from its huge sample size and variety of mental health categories, both of which are useful for in-depth analysis. Nevertheless, it has drawbacks like missing values, data imbalance, and many unique statements that could cause overfitting.

## Exploratory data analysis:

The dataset has 362 missing values in the "statement" column, which will be dropped. The preprocessing steps include converting text to lowercase, removing punctuation, stop words, and repetitive words. Given the dataset's imbalance, where the "Normal" class significantly outnumbers others like "Personality disorder," we will first convert the text data to Term Frequency–Inverse Document Frequency (TF-IDF) features due to class imbalance. Next, SMOTE will be applied to balance the dataset before splitting it into training and test sets. The baseline will be a majority class classifier, which performs poorly with a ROC-AUC score of 0.50 and large error rates as shown in Table 1, and then Visualize the original and resampled class distribution as shown in Figure 3 and Figure 4 and a confusion matrix as shown in Figure 5.

|  |  |  |  |
| --- | --- | --- | --- |
| **Features/ Metrics** | **Precision** | **Recall** | **F1-score** |
| **Anxiety** | 0.00 | 0.00 | 0.00 |
| **Bipolar** | 0.00 | 0.00 | 0.00 |
| **Depression** | 0.00 | 0.00 | 0.00 |
| **Normal** | 0.00 | 0.00 | 0.00 |
| **Personality disorder** | 0.00 | 0.00 | 0.00 |
| **Stress** | 0.14 | 1.00 | 0.25 |
| **Suicidal** | 0.00 | 0.00 | 0.00 |

**Table 1 : Classification Report for Majority Class Classifier**

A graph of a class distribution

Description automatically generated

**Figure 3 : original class distribution.**

A graph showing a class distribution

Description automatically generated

**Figure 4 : Resampled class distribution.**

Table 1 and the confusion matrix below shows that classifier produces 0 accurate predictions for all other classes since it constantly predicts the same class for every input data. This suggests that a more complex model is required due to the significant error rate.

A graph with numbers and symbols

Description automatically generated

**Figure 5 : Confusion Matrix.**

## Machine Learning Models

This section compares several machine learning techniques and explains why Naive Bayes and Logistic Regression are frequently the best options for this task.

In domains like mental health, where comprehension of model decisions is crucial, **Logistic Regression** presents a compelling blend of interpretability and performance. Its versatility makes it applicable in more situations by enabling multi-class classification and producing probabilistic results. Its dependence on a linear decision boundary, which could miss more intricate patterns in the data, is the main drawback (Tan et al., 2023).

**Support Vector Machines (SVM)** are strong against overfitting, especially when the feature space is bigger than the number of samples. They perform well in high-dimensional spaces like text data. SVMs can, however, be computationally demanding, which can slow down training and prediction, particularly when working with big datasets. Furthermore, compared to more straightforward models like logistic regression and naïve bayes, SVM models are harder to interpret (Tan et al., 2023).

**Naive Bayes** is a simple and efficient machine learning algorithm that performs well with large, high-dimensional datasets, and requires less training data, making it suitable for real-time predictions. Unseen test categories and irrelevant features can cause it to struggle, which could cause problems with probability estimations and intricate feature interactions (Tiwari et al., 2021)

To enhance overall performance and lower the risk of overfitting, Random Forests combine numerous trees, building on the advantages of Decision Trees. **Random Forests** can easily manage non-linearities and capture complicated patterns because of its ensemble approach. But compared to a single Decision Tree or more straightforward models like Logistic Regression and naïve bayes, the model can be more computationally expensive to train and has a worse interpretability (Tiwari et al., 2021).

**K-Nearest Neighbors (KNN)** is an easy-to-use algorithm that speeds up setup time because it doesn't require a training phase. However, because KNN needs to calculate the distances to every training sample, it might be computationally expensive during prediction. Additionally, the system may perform worse if features are poorly chosen or scaled due to their sensitivity to extraneous features (Tiwari et al., 2021).

**Reason why Naive Bayes and Logistic Regression are preferable**.

Tan et al. (2023) indicate that naive Bayes is a popularly used algorithm for huge datasets due to its speed and simplicity. It functions especially well with sparse data, like text data that is represented by word counts or TF-IDF characteristics, and it frequently offers a respectable starting point for text classification problems. However, the researchers also point out that one benefit of using Logistic Regression as a model is that it can be easily understood because it gives classification probabilities, which facilitates a clear understanding of the decision-making process. It works well for binary and multi-class classification issues and can manage multicollinearity, a common occurrence in text data where words can be associated as features.

## Deep Learning Models.

This section compares several deep learning techniques and explains why LSTM and RNN are frequently the best options for this task.

**Recurrent Neural Networks (RNNs)** are beneficial for applications like sentiment analysis because they can capture the order and temporal connections between words, which makes them good for processing sequential input, such as text. RNN training can also be slow and resource-intensive, particularly for lengthy text sequences (Kurniasari & Setyanto, 2020).

**Long Short-Term Memory (LSTM)** networks outperform ordinary RNNs in capturing long-term dependencies. This makes LSTMs very helpful for mental health applications such as sentiment analysis. Longer training times are a result of LSTMs' greater complexity and computational expense as compared to simpler models, particularly when dealing with huge datasets (Srinivas et al*.*, 2021).

**Convolutional Neural Networks (CNNs)** excel at extracting hierarchical features from text, which helps in identifying patterns relevant to sentiment analysis. CNNs aren't made to capture temporal dependencies, so they perform worse with sequential input. Furthermore, they usually demand input of a fixed size, which might be limiting for text sequences with varied lengths (Zhang & Wallace, 2016).

**Bidirectional Encoder Representations from Transformers (BERT)** excels in sentiment analysis by capturing deep contextual relationships through its bidirectional processing of text. BERT is adaptable to a range of applications since its pre-trained models may be adjusted with little data. BERT requires a lot of computing resources for both training and inference, though, as it is resource intensive (Devlin et al*.*, 2019).

**Gated Recurrent Units (GRUs)** offer a simplified architecture making them faster to train while still addressing the vanishing gradient problem, which is beneficial for handling sequential data in tasks like sentiment analysis. Despite their advantages, GRUs remain computationally intensive compared to non-recurrent models like CNNs and have lower interpretability than simpler models (Chung et al*.*, 2014).

**Reasons Why LSTM and RNN are Preferable**

Tan et al. (2023) indicates that sentiment analysis benefits from the processing of sequential data and the ability of Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks to capture temporal connections. As LSTMs can retain long-term information and solve the vanishing gradient problem, they are very useful for interpreting complex sentiments in extended text sequences. RNNs are a realistic option when computational resources and training time are limited because they balance simplicity and efficiency, making them less suitable for long-term dependencies but still useful for shorter sequences.

## Implementation of Naïve Bayes and Logistic Regression Classifier

Training and evaluating both the classifier using the scikit-learn library. After splitting the resampled data into training and testing sets, the models are trained on the training data and used to make predictions on the test set. The model's performance is then evaluated using metrics like F1-score, precision, recall and ROC-AUC score as shown in Table 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Models / Metric** | **Max. precision achieved** | **Max. recall achieved** | **Max.**  **F1-score achieved** | **Max ROC-AUC score achieved** |
| **Naïve Bayes** | 0.96 | 0.99 | 0.86 | 0.97 |
| **Logistic Regression** | 0.97 | 0.99 | 0.98 | 0.98 |

**Table 2 : Results for Naive Bayes and Logistic Regression.**

HYPERPARAMETER TUNING - To optimize both the models, the function takes resampled data, a parameter grid, and additional settings, splitting the data into training and testing sets. It then uses GridSearchCV to identify the best hyperparameters. The optimal model is retrained on the training data, and its performance is evaluated using a F1-score, precision, recall and ROC-AUC score on the test data as shown in Table 3 and Table 4. Several sample usages demonstrate different parameter grids being tested for the best performance.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameters/ Metrics** | **Max. precision achieved** | **Max. recall achieved** | **Max.**  **F1-score achieved** | **Max ROC-AUC score achieved** |
| **Alpha = 0.1 and fit prior = False** | 0.93 | 0.99 | 0.92 | 0.98 |
| **Alpha = 0.1 and fit prior = True** | 0.93 | 0.99 | 0.92 | 0.98 |
| **Alpha = 0.01 and fit prior = False** | 0.94 | 0.99 | 0.94 | 0.98 |
| **Alpha = 0.01 and fit prior = True** | 0.94 | 0.99 | 0.94 | 0.98 |

**Table 3 : Results of Hyperparameter Tuning for Naive Bayes.**

The Alpha = 0.01 and fit prior = True and False both perform well to enhance generalization and avoid overfitting in text data, a small amount of smoothing is provided with the alpha parameter set to 0.01. While fit\_prior = False assumes equal class probabilities, which is good for balanced data or eliminating bias towards more frequent classes, fit\_prior = True corrects for class imbalances using real class distributions and the plot for confusion matrix is shown below:

A graph of mental health

Description automatically generated with medium confidence

**Figure 6 : Confusion Matrix of Navie Bayes with best hyperparameter(fir prior = False).**

A graph of mental health

Description automatically generated with medium confidence

**Figure 7 : Confusion Matrix of Navie Bayes with best hyperparameter(fir prior = True).**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameters/ Metrics** | **Max. precision achieved** | **Max. recall achieved** | **Max.**  **F1-score achieved** | **Max ROC-AUC score achieved** |
| **C=1.0, max\_iter=100, penalty=l2, solver= liblinear** | 0.96 | 0.99 | 0.98 | 0.98 |
| **C=5.0, max\_iter=150, penalty=l2, solver= liblinear** | 0.98 | 0.98 | 0.99 | 0.98 |
| **C=10.0, max\_iter=100, penalty=l2, solver= liblinear** | 0.98 | 0.99 | 0.99 | 0.98 |
| **C=10.0, max\_iter=500, penalty=l2, solver= liblinear** | 0.98 | 0.99 | 0.99 | 0.98 |

**Table 4 : Results of Hyperparameter Tuning for Logistic Regression.**

C=10.0, max\_iter=500, penalty=l2, solver= liblinear performs well because higher `C` value reduces regularization, allowing a closer fit, while `max\_iter=500` ensures sufficient iterations for complex data. The `l2` penalty helps avoid overly large coefficients, maintaining model simplicity, and the `liblinear` solver is efficient for smaller datasets and binary classification, making it well-suited for high-dimensional feature spaces and the confusion matrix plot is shown below:

A diagram of a confusion matrix

Description automatically generated with medium confidence

**Figure 8 : Confusion Matrix of Logistic Regression with best hyperparameter**

## Implementation of LSTM and RNN classifier

Both LSTM-based neural network and a SimpleRNN model involve preprocessing text data through tokenization and padding, addressing class imbalance with SMOTE, and splitting data into training and test sets. Both the models are assessed using F1-score, precision, recall, and ROC-AUC score as shown in Table 5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Models/Metrics** | **Max. precision achieved** | **Max. recall achieved** | **Max.**  **F1-score achieved** | **Max ROC-AUC score achieved** |
| **LSTM (vocab size=2000, max len=100, embedding dim = 64, lstm units = 64, epochs=5, batch size = 64)** | 0.84 | 0.91 | 0.87 | 0.87 |
| **RNN (vocab size=2000, max Len = 100,**  **Embedding dim=32, Rnn units = 64**  **Dense units = 16, Dropout rate=0.3, epochs=3, Batch size = 32)** | 0.83 | 0.89 | 0.86 | 0.82 |

**Table 5 : Results for LSTM and RNN.**

HYPERPARAMETER TUNING – This includes functions for preprocessing text data, building, and evaluating both LSTM and Simple RNN models for classification. Both models use embedding, recurrent layers, dropout, and dense layers, and are trained with different hyperparameter settings. Performance is evaluated using F1-score, precision, recall and ROC-AUC score as shown in Table 6 and Table 7.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameters/ Metrics** | **Max. precision achieved** | **Max. recall achieved** | **Max.**  **F1-score achieved** | **Max ROC-AUC score achieved** |
| **LSTM (vocab size=2000, max len=200, embedding dim = 64,** **lstm units = 64, dropout rate=0.3, epochs=5, batch size = 64)** | 0.82 | 0.92 | 0.87 | 0.86 |
| **LSTM (vocab size=2000, max Len=200, embedding dim = 100, lstm units = 128, dropout rate=0.3, epochs=5, batch size = 64)** | 0.86 | 0.88 | 0.87 | 0.87 |

**Table 6 : Results of Hyperparameter Tuning for LSTM.**

Lstm units = 64 performs well because setting a maximum sequence length of 200 captures enough context without overloading the model, while keeping the vocabulary size limited to 2000 concentrates on important aspects. Word connections and temporal patterns are efficiently captured by an embedding dimension and LSTM units of 64. Training for 5 epochs with a batch size of 64 guarantees effective learning and model generalization, while a dropout rate of 0.3 prevents overfitting and the loss curve is shown below:

A graph of a number of people

Description automatically generated with medium confidence

**Figure 9 : Training and validation loss of LSTM with best hyperparameter.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameters/ Metrics** | **Max. precision achieved** | **Max. recall achieved** | **Max.**  **F1-score achieved** | **Max ROC-AUC score achieved** |
| **RNN (vocab size=2000, max Len = 200,**  **Embedding dim=16, Rnn units = 32,**  **Dense units = 8, Dropout rate=0.3, epochs=3, Batch size = 16)** | 0.78 | 0.90 | 0.83 | 0.81 |
| **RNN (vocab size=3000, max Len = 150,**  **Embedding dim=16, Rnn units = 32,**  **Dense units = 8, Dropout rate=0.4, epochs=3, Batch size = 16)** | 0.79 | 0.89 | 0.84 | 0.80 |

**Table 7 : Results of Hyperparameter Tuning for RNN.**

Embedding dimensions of 16 and 32 RNN units strikes an appropriate balance for temporal learning, while a vocabulary size of 2000 and a maximum sequence length of 200 control complexity and context. Utilizing 8 dense units, 3 epochs, a dropout rate of 0.3, and a batch size of 16, the model attains efficient learning and stability. and the loss curve is shown below:

A graph of a person with a blue line

Description automatically generated with medium confidence

**Figure 10 : Training and validation loss of RNN with best hyperparameter.**

## Comparative analysis of all the models with base model:

The below Table 8 the best performance of all the models in comparison with the baseline model.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameters/ Metrics** | **Model** | **Max. precision achieved** | **Max. recall achieved** | **Max.**  **F1-score achieved** | **Max ROC-AUC score achieved** |
| **N/A** | Majority Class Classifier | 0.14 | 1.00 | 0.25 | 0.25 |
| **Alpha = 0.01 and fit prior = True and Alpha = 0.01 and fit prior = False** | Naïve Bayes | 0.94 | 0.99 | 0.94 | 0.98 |
| **C=10.0, max\_iter=500, penalty=l2, solver= liblinear** | Logistic Regression | 0.98 | 0.99 | 0.99 | 0.98 |
| **vocab size=2000, max Len=200, embedding dim = 64, lstm units = 64, dropout rate=0.3, epochs=5, batch size = 64** | LSTM | 0.82 | 0.92 | 0.87 | 0.86 |
| **RNN (vocab size=2000, max Len = 200,**  **Embedding dim=16, Rnn units = 32,**  **Dense units = 8, Dropout rate=0.3, epochs=3, Batch size = 16)** | RNN | 0.78 | 0.90 | 0.83 | 0.81 |

**Table 8 : Comparative analysis of majority class classifier, naive bayes, logistic regression, LSTM and RNN.**

## References

Link of the dataset: [Sentiment Analysis for Mental Health (kaggle.com)](https://www.kaggle.com/datasets/suchintikasarkar/sentiment-analysis-for-mental-health/data)

Alanazi, S.A. *et al.* (2022) ‘Public’s Mental Health Monitoring via Sentimental Analysis of Financial Text Using Machine Learning Techniques’, *International Journal of Environmental Research and Public Health*, 19(15), p. 9695. Available at: https://doi.org/10.3390/ijerph19159695.

Babu, N.V. and Kanaga, E.G.M. (2021) ‘Sentiment Analysis in Social Media Data for Depression Detection Using Artificial Intelligence: A Review’, *SN Computer Science*, 3(1), p. 74. Available at: https://doi.org/10.1007/s42979-021-00958-1.

Birjali, M., Beni-Hssane, A. and Erritali, M. (2017) ‘Machine Learning and Semantic Sentiment Analysis based Algorithms for Suicide Sentiment Prediction in Social Networks’, *Procedia Computer Science*, 113, pp. 65–72. Available at: https://doi.org/10.1016/j.procs.2017.08.290.

Burdisso, S.G., Errecalde, M. and Montes-y-Gómez, M. (2019) ‘A text classification framework for simple and effective early depression detection over social media streams’, *Expert Systems with Applications*, 133, pp. 182–197. Available at: https://doi.org/10.1016/j.eswa.2019.05.023.

Chung, J. *et al.* (2014) ‘Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling’. arXiv. Available at: https://doi.org/10.48550/arXiv.1412.3555.

Devlin, J. *et al.* (2019) ‘BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding’. arXiv. Available at: https://doi.org/10.48550/arXiv.1810.04805.

Fitri, V.A., Andreswari, R. and Hasibuan, M.A. (2019) ‘Sentiment Analysis of Social Media Twitter with Case of Anti-LGBT Campaign in Indonesia using Naïve Bayes, Decision Tree, and Random Forest Algorithm’, *Procedia Computer Science*, 161, pp. 765–772. Available at: https://doi.org/10.1016/j.procs.2019.11.181.

George, A. *et al.* (2021) ‘Applications of Aspect-based Sentiment Analysis on Psychiatric Clinical Notes to Study Suicide in Youth’, *AMIA Summits on Translational Science Proceedings*, 2021, pp. 229–237.

Huisman, S.M., Kraiss, J.T. and de Vos, J.A. (2024) ‘Examining a sentiment algorithm on session patient records in an eating disorder treatment setting: a preliminary study’, *Frontiers in Psychiatry*, 15. Available at: https://doi.org/10.3389/fpsyt.2024.1275236.

Kurniasari, L. and Setyanto, A. (2020) ‘Sentiment Analysis using Recurrent Neural Network’, *Journal of Physics: Conference Series*, 1471(1), p. 012018. Available at: https://doi.org/10.1088/1742-6596/1471/1/012018.

Srinivas, A.C.M.V. *et al.* (2021) ‘Sentiment Analysis using Neural Network and LSTM’, *IOP Conference Series: Materials Science and Engineering*, 1074(1), p. 012007. Available at: https://doi.org/10.1088/1757-899X/1074/1/012007.

Tan, K.L., Lee, C.P. and Lim, K.M. (2023) ‘A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research’, *Applied Sciences*, 13(7), p. 4550. Available at: https://doi.org/10.3390/app13074550.

Tiwari, P.K. *et al.* (2021) ‘A Study on Sentiment Analysis of Mental Illness Using Machine Learning Techniques’, *IOP Conference Series: Materials Science and Engineering*, 1099(1), p. 012043. Available at: https://doi.org/10.1088/1757-899X/1099/1/012043.

Wankhade, M., Rao, A.C.S. and Kulkarni, C. (2022) ‘A survey on sentiment analysis methods, applications, and challenges’, *Artificial Intelligence Review*, 55(7), pp. 5731–5780. Available at: https://doi.org/10.1007/s10462-022-10144-1.

Yu, L. *et al.* (2021) ‘Detecting changes in attitudes toward depression on Chinese social media: A text analysis’, *Journal of Affective Disorders*, 280, pp. 354–363. Available at: https://doi.org/10.1016/j.jad.2020.11.040.

Zhang, Y. and Wallace, B. (2016) ‘A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification’. arXiv. Available at: https://doi.org/10.48550/arXiv.1510.03820.