**Multiclass Classification of Mental Illness using NLP: Traditional Models vs Deep Learning Models**   
  
*Sahil Vermaa  
aCentre of Excellence for Data Science, Artificial Intelligence and Modelling, University of Hull, Kingston Upon Hull HU6 7RX, England, United Kingdom*

|  |  |  |
| --- | --- | --- |
| ***Keywords*** |  | ***Abstract*** |
| **NLP**  **Naive Bayes**  **Random Forest**  **Logistic Regression**  **BiLSTM**  **CNN** |  | **Mental illness is a relatively underdeveloped area in psychology and can be challenging to diagnose. Early sign of mental illness can easily be interpreted as slang or sarcasm which makes it more dangerous. In this study, we have conducted an in-depth analysis of traditional models and deep learning models on an open-source dataset with multiple categories of mental illness diagnosed based on text. We have developed a total of 5 model from scratch and used several hyperparameter tuning to achieve an accuracy of over 76.55% on an unseen test dataset.** |

1. Introduction

Mental health is a serious and highly complex problem today. “In 2019, 1 in every 8 people, or 970 million people around the world were living with mental disorder. In 2020, the number rose significantly because of COVID-19 pandemic” (*Mental disorders*, 2022). With recent developments in psychology, we now have in-depth classification of mental health issues. Most studies in this field are targeted towards detecting the presence of one mental illness.

The scope of this study is to attempt to create an NLP model which can detect a range of mental issues and categorize them based on severity so a more targeted and sophisticated support can be provided to those in need. This is an extremely important field which can easily be overlooked as most times the signs are subtle and can be easily misinterpreted.

1. Background

To understand the relevant work in this scope, several papers were referenced from reputed peer-reviewed journals and arXiv was also searched for open-access papers.

Lamichhane, (2023), conducted research where he used LLM-based gpt-3.5-turbo (ChatGPT) in 3 text-based mental health classification. He developed three classification models and observed a substantial increase in their F1 scores as compared to a baseline model which was biased towards majority class. However, the study did not focus on prompt engineering which could significantly affect the model accuracy. Additionally, there are other popular and highly versatile LLMs which were not considered.

Aggarwal et al., (2025), in their recent study, leveraged LLMs for mental health detection. They used pre-trained NLP models like DistilBERT, MentalRoBERTa. The authors developed MentaLLaMA based on LLaMA3 as foundation. By using a pre-trained model, the authors concluded that the DistilBERT generally outperformed upon fine-tuning with LLaMA3. The main challenge the authors mentioned in their paper was, the use of slangs, emojis and informal languages, and they also pointed the narrow dataset as the models were trained on data collected from only one social media platform (Reddit).

Calvo et al., (2017) conducted research where they used non-clinical text from social media platforms including Reddit, Twitter, Facebook, blogs, and others. They also trained the model to distinguish between authentic and fake suicide notes where they achieved a F1 score of 0.61 using combination of techniques. The key strength of this research is the use of non-clinical data from a wide variety of source; however, the use of slangs and other informal writing was not handled.

Cook et al., (2016) used NLP and machine learning to analyse text responses of patients who have recently been discharged from psychiatric institutions. The study involves analysing the SMS response to general mood texts and predicting suicidal ideation and mental distress. A key strength of this study involved real-time NLP based monitoring. However, predictive accuracy was limited compared to structured data models and relied heavily on the honesty and clarity.

Tewari et al., (2018) proposed a mobile application called MentalEase, which integrates mental health assessment tools and regular therapy to assist patients with mild anxiety and depression. The model achieved and accuracy of over 85% in detecting emotions with a precision and recall of 80% and 78% respectively in generating relevant responses. Used satisfaction upon survey was reported to be 75%. The key strength of this study is possible integration of MentalEase to provide comprehensive mental health support. However, authors did mention that chatbots cannot fully replicate the empathy and nuanced understanding of a human therapist.

1. SMART Objective  
   1. Specific

The main objective of this study is to develop and compare traditional and deep learning NLP models to classify a gradient of mental health issues. Additionally, the study used a dataset which comprises a wide range of data from various Kaggle datasets.

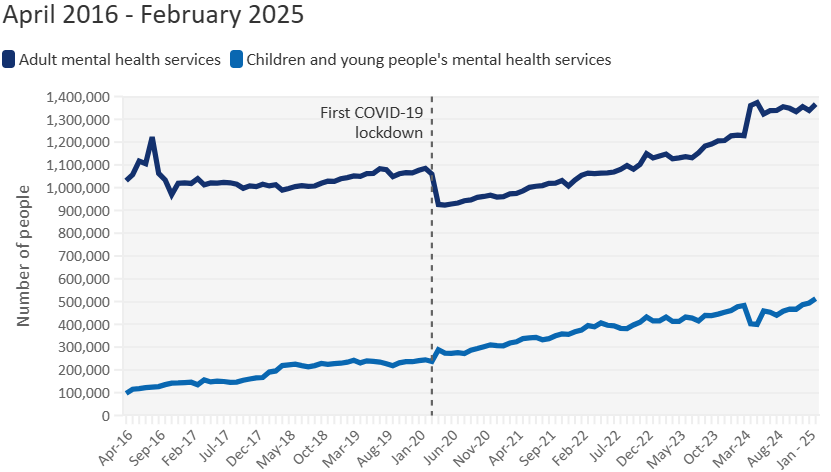
* 1. Measurable

We used several metrics to assess the models which includes, F1 score, accuracy, precision, recall, ROC-AUC and Macro-average PR-AUC. Based on the existing literature, we aim to achieve and overall precision and F1 score of over 75%.

* 1. Achievable

The analysis is backed by existing research in the domain along with the tremendous development in NLP processing machine learning models which includes both traditional model like Naïve Bayes and deep learning models such as BiLSTM.

* 1. Relevant

Give the age of social media, we have more textual context than before. Additionally, there has been a considerable increase in reported mental illness evident by the report publish by *Mental health pressures data analysis* (2025)  
  
Figure 1: Number of people in contact with mental health services (April 2016 – February 2025)  
  
Hence, it is highly relevant to develop models which specialise in detective an array of mental illness via general text over social media or other platforms.

* 1. Time Limited

The time limit of 2 weeks was imposed which included finding appropriate data, performing exploratory data analysis, develop and fine-tune traditional and deep learning models and write a detailed report.

1. Dataset [[LINK](https://www.kaggle.com/datasets/suchintikasarkar/sentiment-analysis-for-mental-health)]

For the study, we used an open-sourced dataset from Kaggle (*Sentiment Analysis for Mental Health*, 2024). The dataset comprises of a collection of mental health statuses from various statements. It was then cleaned and compiled to create the final dataset for the purpose of creating chatbots and performing multiclass classification analysis. The sources used in this dataset as listed below:

* [Depression Reddit Cleaned](https://www.kaggle.com/datasets/infamouscoder/depression-reddit-cleaned)
* [3k Conversations Dataset for Chatbot](https://www.kaggle.com/datasets/kreeshrajani/3k-conversations-dataset-for-chatbot)
* [Human Stress Prediction](https://www.kaggle.com/datasets/kreeshrajani/human-stress-prediction)
* [Predicting Anxiety in Mental Health Data](https://www.kaggle.com/datasets/michellevp/predicting-anxiety-in-mental-health-data)
* [Mental Health Dataset Bipolar](https://www.kaggle.com/datasets/michellevp/mental-health-dataset-bipolar)
* [Reddit Mental Health Data](https://www.kaggle.com/datasets/neelghoshal/reddit-mental-health-data)
* [Students Anxiety and Depression Dataset](https://www.kaggle.com/datasets/sahasourav17/students-anxiety-and-depression-dataset)
* [Suicidal Mental Health Dataset](https://www.kaggle.com/datasets/aradhakkandhari/suicidal-mental-health-dataset)
* [Suicidal Tweet Detection Dataset](https://www.kaggle.com/datasets/aunanya875/suicidal-tweet-detection-dataset)

The data contained 3 columns, unique identifier, statement and status. There are 7 possible statuses representing an array of mental disorders.

* Normal
* Depression
* Suicidal
* Stress
* Bi-Polar
* Anxiety
* Personality Disorder

|  |  |  |
| --- | --- | --- |
| ***Status*** | ***Count*** | ***Percentage*** |
| Normal | 16343 | 31.0% |
| Depression | 15404 | 29.2% |
| Suicidal | 10652 | 20.2% |
| Anxiety | 3841 | 7.3% |
| Bipolar | 2777 | 5.3% |
| Stress | 2587 | 4.9% |
| Personality Disorder | 1077 | 2.0% |

Table 1: Distribution of mental illness in the raw dataset

Since there is a large disparity in the most frequent category (Normal) and least frequent category (Personality Disorder) is vast, we have categorized the illness based on severity into 3 prominent categories.

|  |  |
| --- | --- |
| ***Concise Category*** | ***Original Status*** |
| Neutral | Normal |
| Mood Disorders | Anxiety  Depression  Stress |
| Severe Conditions | Suicidal  Bipolar  Personality Disorder |

Table 2: Categorization of status based on severity

After categorization, we had 21832, 16343 and 14506 entries for “Neutral”, “Mood Disorders” and “Sever Conditions” respectively. For data balancing, we collected 14506 sample from each of these three categories and randomized the dataset resulting in a completely balanced dataset.

1. Exploratory Data Analysis  
   1. Data Cleaning

In the dataset, we had 362 rows with missing values (NaN) within the ‘statement’ column. Given the overall size of the dataset, we can drop these missing values without affecting the balancing of the dataset.

5.2 Pre-processing

As part of pre-processing, we first removed all punctuations, special characters and made all text lower case only. We then tokenized each statement into a list of words and “*STOPWORDS*” were removed. This was also used to visualize the most frequently used term in the dataset.

A close-up of words

AI-generated content may be incorrect.

Figure 2: Word Cloud representing most frequently used terms in the dataset

Next, we lemmatized all texts to retain only noun, verbs, adverbs and adjectives. “Lemmatization is the process of finding the normalized form of a word” (Plisson et al., 2004).

Additionally, we implemented unigram, bigram and trigram processing. “Multi-word features such as these convey more specific meaning than single word features, and therefore should be more effective in targeting relevant information” (Johnson et al., 2006).

1. Traditional machine learning methods

A total of 3 traditional machine learning models were developed and compared which are Naïve Bayes, Random Forest and Logistic Regression classifiers. We also used grid search to optimize hyperparameters for each of these three models.

In addition to previous pre-processing, TF-IDF (Term frequency-inverse document frequency) was applied to capture relevance and significance of words within the text. “TF-IDF represents the intuition that some words are more informative than others” (Dethlefs, 2024).

* 1. Naïve Bayes Classifier

Naïve Bayes, a fundamental supervised learning algorithm utilizes Bayes’ Theorem for statistical classification (*Naive Bayes Classifier Tutorial: with Python Scikit-learn*, 2023). Naïve Bayes is very effective and performs very well with higher dimensionality data which makes it ideal for text classification. However, it assumes features to be independent which is unrealistic for NLP.

A chart of a number of problems

AI-generated content may be incorrect.  
Figure 3: Confusion Matrix from Naïve Bayes traditional model

Naïve Bayes classifier achieved a test accuracy of 71.58% with precision and F1 score of 72%.

* 1. Random Forest Classifier

“Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution of all trees in the forest” (Breiman, 2001). Random Forest is ensemble in nature which help with overfitting. On the other hand, it is computationally very expensive and slow to train.

A graph of a number of problems

AI-generated content may be incorrect.   
Figure 4: Confusion Matrix from Random Forest traditional model

We were able to achieve a test accuracy of 72.30% with precision and F1 score of 72% and 71% respectively.

* 1. Logistic Regression Classifier

Effective for binary/multi-class classification tasks and often used with TF-IDF features. “In NLP, logistic regression is the baseline supervised machine learning algorithm for classification, and also has a very close relationship with neural networks” (Jurafsky & Martin, 2025). Logistic Regression is simple and effective for linearly separable text classification problems, but it struggles with non-linear and complex relationships within the data which can limit its performance as mentioned by Adusumilli et al., (2025).

A graph of a logistic regression confusion matrix

AI-generated content may be incorrect.  
Figure 5: Confusion Matrix from Logistic Regression traditional model

The model achieved 76.55% accuracy on test set with precision and F1 score of 76%.

Amongst all traditional models, logistic regression classifier outperformed the other two models.

|  |  |  |  |
| --- | --- | --- | --- |
| ***Model*** | ***Accuracy*** | ***Precision*** | ***F1 Score*** |
| Naïve Bayes | 71.58 | 72 | 72 |
| Random Forest | 72.30 | 72 | 71 |
| Logistic Regression | 76.55 | 76 | 76 |

Table 3: Comparison of traditional models along their accuracy (in %), precision (in %) and F1 score (in %)

Additionally, SVM (Support Vector Machine) and k-NN (k-Nearest Neighbour) were initially considered. Where SVM is effective in handling data of higher dimensions and performs well on small sized dataset, it is extremely computationally expensive (Editorial, 2022a). Additionally, k-NN being well suited for separating classes, this too is computationally intensive especially for large datasets due to distance commutation (Editorial, 2022b). Hence these models were not further analyses.

1. Deep learning models

In this study, we devised two deep learning models for multiclass classification in NLP. We trained models based on BiLSTM and CNN.

Necessary further pre-processing was performed to fit the padding of the text to accommodate the input layers.

* 1. LSTM (Long Short-Term Memory)

“An LSTM neural network is a type of recurrent neural network (RNN) that can learn long-term dependencies” (*Long Short-Term Memory Neural Networks*, 2025). LSTM handles long-term dependencies effectively while also overcoming the issue of vanishing gradient (Chen et al., 2024). However, it is computationally slower to train when compared to other architectures like CNN or GRU.

* 1. Bi-LSTM (Bi-directional Long Short-Term Memory)

Bi-LSTM is a word embedding technique which represents words in a vector space. A type of RNN that processes data in both forward and backward directions to capture long-term dependencies. Though this can lose focus on very long sequences is still an upgrade over LSTM.

We used the simplest model to gauge the potential of Bi-LSTM as most deep learning models are computationally expensive and time consuming. Our model was able to achieve an accuracy of 58.61% on test set with precision and recall of 59%.

*A chart of a confusion matrix

AI-generated content may be incorrect.*Figure 6: Confusion matrix for Bi-LSTM deep learning model

* 1. CNN (Convolutional Neural Network) for Text Classification

A feed-forward neural network that uses convolutional layers to extract features in form of vectors from the input data. It is widely used in text processing for its ability to capture intricate details (Luan & Lin, 2019).

The model achieved a test accuracy of 64.82% with precision and F1 score of 66% and 65% respectively.

A chart of a confusion matrix

AI-generated content may be incorrect.  
Figure 7: Confusion matrix for CNN for Text Classification deep learning model

* 1. Transformer

Transformers developed by Google in 2017, processes sequences in parallel, making them highly scalable for large datasets. However, this is extremely expensive on computation and requires significant resources for effective training (Tucudean, 2024).

* 1. GRU (Gated Recurrent Unit)

GRU’s are simplified versions of LSTM which depends on fewer parameters and trains faster while still modelling sequential dependencies. This comes with the trade-off in capturing complex, long term dependencies (CS224n: Natural Language Processing with Deep Learning Lecture Notes: Part V Language Models, RNN, GRU and LSTM, 2019).

Comparing both deep learning models, it can we stated that CNN clearly outperforms Bi-LSTM. However, it should also be noted that both deep learning models underperformed in comparison to the three traditional models. This suggests that the deep learning models could be further fine-tuned and permutation-combination of different layer with different size should be tested to further optimize the deep learning models.

|  |  |  |  |
| --- | --- | --- | --- |
| ***Model*** | ***Accuracy*** | ***Precision*** | ***F1 Score*** |
| Bi-LSTM | 58.61 | 59 | 59 |
| CNN for Text | 64.82 | 66 | 65 |

Table 4: Comparison of traditional models along their accuracy (in %), precision (in %) and F1 score (in %)

1. Implementation

We used various libraries to develop the model discussed earlier. The list of libraries are as follows:

1. NumPy
2. Pandas
3. Matplotlib
4. Gensim
5. SpaCy
6. WordCloud
7. Nltk
8. Scikit-Learn
9. Scikeras
10. Keras
11. Tensorflow

First three libraries (a, b & c) are used for data analysis and visualization. Next four libraries (d, e, f & g) as essentials for text processing and natural language processing (NLP). Next two library (h & i) are utilities for machine learning. The final two (j & k) are deep learning frameworks. There libraries were used with Python version 3.10 and CUDA 11.2.

1. Hyperparameter Tuning

The process of optimizing model parameters to improve its performance is called hyperparameter (Bhasin, 2024). During our study, we used different approaches for both traditional and deep learning models.

* 1. Traditional Models

Our initial hyperparameter optimization technique included trial-and-error approach to figure out a ballpark hyperparameter. Next, we utilized “Grid Search” approach and passed in possible contenders for each parameter. Grid Search then ran the model for each possible combination and outputs the best model based on accuracy metrics. Scikit-Learn has a built-in implementation of Grid Search which was used to find the best hyperparameters for the traditional models.

|  |  |
| --- | --- |
| **Hyperparameter** | **Best Value** |
| **Naïve Bayes Classifier** | |
| alpha | 0.1 |
| fit\_prior | False |
| **Random Forest Classifier** | |
| n\_estimator | 200 |
| max\_depth | 20 |
| min\_samples\_split | 5 |
| min\_samples\_leaf | 1 |
| max\_features | sqrt |
| **Logistic Regression Classifier** | |
| C | 1 |
| solver | 200 |
| max\_iter | lbfgs |

Table 5: Hyperparameter optimization and their best values using Grid Search methodology

* 1. Deep Learning Models

Deep learning models are a bit complex when it comes to hyperparameters tuning. Implementation of Grid Search for deep learning does not guarantee and with these models, the architecture also plays a significant role. Each layer can be customized, and this level of customization makes Grid Search implementation not as sustainable as it does for traditional models. Hence, for deep learning models, we opted to proven hyperparameter setups and values which was then tweaked and re-tested multiple times before settling on the final model. Additionally, due to system constraints, in-depth analysis into hyperparameter optimization with more layers was not conducted. However, several regulatory techniques were implored which enhanced the model performance.

We tuned the number of nodes in each layers along with 50% dropout. Additionally, L2-regularization and early stopping for validation loss with patience value of 5 was used.

|  |  |  |
| --- | --- | --- |
| **Layer (type)** | **Output Shape** | **Param #** |
| Embedding | None, 100, 128 | 1280000 |
| Bidirectional | None, 128 | 98816 |
| Dropout | None, 128 | 0 |
| Dense | None, 64 | 8256 |
| Dropout | None, 64 | 0 |
| Dense | None, 3 | 195 |

Table 6: Model summary of BiLSTM deep learning model

|  |  |  |
| --- | --- | --- |
| **Layer (type)** | **Output Shape** | **Param #** |
| Embedding | None, 100, 128 | 1280000 |
| Conv1D | None, 96, 128 | 82048 |
| GlobalMaxPooling1D | None, 128 | 0 |
| Dropout | None, 128 | 0 |
| Dense | None, 64 | 8256 |
| Dropout | None, 64 | 0 |
| Dense | None, 3 | 195 |

Table 7: Model summary of CNN for text classification deep learning model

1. Evaluation

After comparing all 3 traditional and 2 deep learning models, within the scope of this study, Logistic Regression Classifier has provided the best overall accuracy on the unseen test dataset. However, it is equally important that both BiLSTM and CNN models could be further optimized, and more layers could be added which could greatly increase their performance

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 Score** | **ROC-AUC (OVA)** | **Macro Average PR-AUC** |
| Naïve Bayes | 71.58 | 72 | 72 | 72 | 87.7 | 78.4 |
| Random Forest | 72.30 | 72 | 72 | 71 | 87.8 | 78.7 |
| Logistic Regression | 76.55 | 76 | 77 | 76 | 90.9 | 83.5 |
| BiLSTM | 58.61 | 59 | 59 | 59 | 79 | 63.5 |
| CNN for Text | 64.82 | 66 | 65 | 65 | 83.9 | 72.2 |

Table 8: NLP model metrics for both traditional and deep learning models. All units are in percentage.

1. Conclusion

In this study, we compared 3 traditional and 2 deep learning models to develop and test an NLP based multiclass classification to identify mental illness based on texts from various platform and sources. We found that amongst deep learning models, CNN for text performed substantially better than BiLSTM. However, Logistic Regression classifier, a traditional model outperformed others with an accuracy of 76.55%. This model also achieved the highest ROC-AUC (90.9%) and PR-AUC (83.5%) which represents that model can distinguish between different classes clearly.

1. Future Scope of Work

As part of future scope of work, BiLSTM and CNN can be targeted for hyperparameter optimization as well as new layers can be introduced to increase the models capacity to capture hidden pattern in the sequence.

Additionally, within the scope of this study, we 7 states of mental illness into 3 categories. A new study should be performed to test these models for all 7 categories without grouping. This will provide valuable insight into multiclass text classification.

1. Reference Style

This paper has used Hull-Harvard reference style which includes author’s last name and year of publication for in-line citation. The Hull-Harvard citation style is a variant of Harvard referencing system, tailored to meet the academic requirements of the University of Hull.

1. References

Adusumilli, S.B.K., Damancharla, H. & Metta, A.R. (2025) Machine Learning Algorithm for Fraud Detection in Financial Transactions. *International Journal of Sustainable Development in Computing Science*, 5(2), p. 1–18.

Aggarwal, V., Thukral, S., Patel, K. & Chatterjee, A. (2025) Leveraging LLMs for Mental Health: Detection and Recommendations from Social Discussions. arXiv. Available online: https://doi.org/10.48550/arXiv.2503.01442.

Bhasin, H. (2024) Hyperparameter Tuning. In Bhasin, H. (ed.) *Hands-on Deep Learning: A Guide to Deep Learning with Projects and Applications*. Apress, p. 133–155. Available online: https://doi.org/10.1007/979-8-8688-1035-0\_5.

Breiman, L. (2001) Random Forests. *Machine Learning*, 45(1), p. 5–32. Available online: https://doi.org/10.1023/A:1010933404324.

Calvo, R.A., Milne, D.N., Hussain, M.S. & Christensen, H. (2017) Natural language processing in mental health applications using non-clinical texts. *Natural Language Engineering*, 23(5), p. 649–685. Available online: https://doi.org/10.1017/S1351324916000383.

Chen, J., Gunawan, D., Taylor, P.H., Chen, Y., Milne, I.A. & Zhao, W. (2024) An Attention-Based Deep Learning Model for Phase-Resolved Wave Prediction. *Journal of Offshore Mechanics and Arctic Engineering*, 147(022002). Available online: https://doi.org/10.1115/1.4065969.

Cook, B.L., Progovac, A.M., Chen, P., Mullin, B., Hou, S. & Baca-Garcia, E. (2016) Novel Use of Natural Language Processing (NLP) to Predict Suicidal Ideation and Psychiatric Symptoms in a Text-Based Mental Health Intervention in Madrid. *Computational and Mathematical Methods in Medicine*, 2016(1), p. 8708434. Available online: https://doi.org/10.1155/2016/8708434.

CS224n: Natural Language Processing with Deep Learning Lecture Notes: Part V Language Models, RNN, GRU and LSTM (2019).

Dethlefs, N. (2024) *LanguageModels\_W8\_2024T1.pdf: Applied Artificial Intelligence (771767\_A24\_T2)*. https://canvas.hull.ac.uk/courses/74998/files/5860372?module\_item\_id=1279316 [Accessed 30 Apr 2025].

Editorial (2022a) *Pros and cons of Support Vector Machine (SVM)*. *RoboticsBiz*. https://roboticsbiz.com/pros-and-cons-of-support-vector-machine-svm/ [Accessed 30 Apr 2025].

Editorial (2022b) *Pros and cons of the K-Nearest Neighbors (KNN) algorithm*. *RoboticsBiz*. https://roboticsbiz.com/pros-and-cons-of-the-k-nearest-neighbors-knn-algorithm/ [Accessed 30 Apr 2025].

Johnson, D., Malhotra, V. & Vamplew, P. (2006) More Effective Web Search Using Bigrams and Trigrams. https://figshare.utas.edu.au/articles/journal\_contribution/More\_Effective\_Web\_Search\_Using\_Bigrams\_and\_Trigrams/23202449/1 [Accessed 30 Apr 2025].

Jurafsky, D. & Martin, J.H. (2025) Logistic Regression. In *Speech and Lanuage Processing*. https://web.stanford.edu/~jurafsky/slp3/5.pdf [Accessed 30 Apr 2025].

Lamichhane, B. (2023) Evaluation of ChatGPT for NLP-based Mental Health Applications. arXiv. Available online: https://doi.org/10.48550/arXiv.2303.15727.

*Long Short-Term Memory Neural Networks* (2025). https://uk.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html [Accessed 30 Apr 2025].

Luan, Y. & Lin, S. (2019) Research on Text Classification Based on CNN and LSTM. In *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*. *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, p. 352–355. Available online: https://doi.org/10.1109/ICAICA.2019.8873454.

*Mental disorders* (2022). https://www.who.int/news-room/fact-sheets/detail/mental-disorders [Accessed 30 Apr 2025].

*Mental health pressures data analysis* (2025) *The British Medical Association Is the Trade Union and Professional Body for Doctors in the UK.* https://www.bma.org.uk/advice-and-support/nhs-delivery-and-workforce/pressures/mental-health-pressures-data-analysis [Accessed 30 Apr 2025].

*Naive Bayes Classifier Tutorial: with Python Scikit-learn* (2023). https://www.datacamp.com/tutorial/naive-bayes-scikit-learn [Accessed 30 Apr 2025].

Plisson, J., Lavrac, N. & Mladenic, D. (2004) A Rule based Approach to Word Lemmatization. *Proceedings of IS*, 3, p. 83–86.

*Sentiment Analysis for Mental Health* (2024). https://www.kaggle.com/datasets/suchintikasarkar/sentiment-analysis-for-mental-health [Accessed 30 Apr 2025].

Tewari, A., Khalsa, A.S. & Kanal, H. (2018) A Survey of Mental Health Chatbots using NLP.

Tucudean, G. (2024) Natural language processing with transformers: a review.