# Prediction of Mental Illness From Reddit Posts Using Natural Language Processing

John Michael Elbambo
College of Computing and Information Technologies
National University
Manila, Philippines
jm.elbambo27@gmail.com

Jezreel Carl Gatanela
College of Computing and Information Technologies
National University
Manila, Philippines
jezreelgatanela@gmail.com

***Abstract*—Mental disorders have become a prevalent health issue caused by the recent global pandemic, and individuals suffering from mental disorders use social media platforms more than the rest of the population. Detecting possible cases of mental illness through social media can be an efficient and a low-cost solution of early diagnosis of mental disorders. This study explores and proposes different linguistic feature extraction techniques from user-generated textual posts on the social media platform Reddit. This study takes advantage of random forest classifier which is further improved by hyperparameter optimization techniques in performing predictive tasks.**

***Keywords—machine learning, natural language processing, mental illness, reddit***

## I. Introduction

World Health Organization (WHO) characterized mental disorders as clinically significant disturbances in an individual's cognition, emotional regulation, or behavior. There are many different types of mental disorders, some of which are ADHD, anxiety, bipolar, depression, and PTSD. Mental disorders are rapidly growing across the world. In 2019, 1 in every 8 individuals, or 970 million people around the world suffered from a mental disorder. Among those cases, anxiety and depressive disorders are the most prominent [1]. Following the recent COVID-19 pandemic, the number of people living with anxiety and depressive disorders increased significantly in 2020. Initial estimates suggest a 26% and 28% increase in anxiety and depressive disorders respectively, in just one year [2].

Studies have shown that individuals suffering from mental disorders, including depression, psychotic disorders, or other severe mental illnesses, use social media platforms from about 70% up to 97% compared to the general population [3][4]. Other exploratory studies have discovered that many of these individuals with mental disorders appear to use social media to share their personal experiences, seek information about their mental health and treatment options, and give and receive support from others who are dealing with similar mental health issues [5][6].

In the medical field, the power of machine learning and its different techniques is harnessed and widely applied because of its ability to generate accurate and effective results. Furthermore, with the application of machine learning techniques, various mental health illnesses can be predicted. The machine learning pattern recognition algorithm is well-known for the prediction, diagnosis, and treatment of mental disorders [7].

## II. Related Work

A recent study revealed that self-reported speech samples from individuals with serious mental illnesses can be collected chronically in a community-based clinical context. It also demonstrates that linguistic and acoustic features extracted from those samples can be utilized to monitor an individual's mental health changes over time. Prediction models produced a strong correlation (up to 0.78) between predicted and actual clinical conditions where the latter is based on providers' global assessment ratings. Interestingly, despite a high level of predictability in individual speech features over time, there was little correlation between individuals in terms of which speech features were most associated with their clinical condition. This suggests that the pattern of word choice in relation to mental illness may be specific to individuals [8].

In another recent study, it has been demonstrated that deep learning algorithms combined with appropriate natural language processing techniques can be used to predict mental illnesses in individuals based on their posts. According to this study, detecting mental illness through social media may become a significant research field in the future. Among the six different subreddits, r/autism showed the highest accuracy (96.96%) in the CNN but had the lowest F1-score on the autism class (XGBoost: 38.31%, CNN: 48.73%), which is due to the class imbalance problem. Overall, CNN models showed higher accuracy than XGBoost models across all the subreddits. One of the most class-balanced subreddits, r/depression, showed the highest performance scores in terms of precision (89.10%), recall (71.75%), and F1-score (79.49%) for the depression class. Three other subreddits, r/Anxiety, r/bipolar, and r/BPD, also showed high accuracy with CNN models, 77.81%, 90.20%, and 90.49%, respectively, and their F1-scores in identifying mental illnesses ranged from forties to fifties (%), which are relatively lower than those with the class-balanced channels. In summary, the proposed model can accurately detect potential users who may have psychological disorders. The study's findings indicate the potential for social media platforms to play a role in providing a space for people suffering from mental disorders to interact with others [9].

## III. Methodology

The goal of this study is to develop a machine learning model for classifying social media posts that are at risk of mental illnesses using natural language processing techniques. More specifically, the task focuses on detecting 5 types of mental illnesses: ADHD, Anxiety, Bipolar, Depression, and PTSD, or none found in Reddit posts from the acquired dataset.

### A. Dataset

The dataset used is acquired from the study of Murarka et al., *Classification of mental illnesses on social media using RoBERTa* 2021. The dataset contains the columns "ID", "title", "post", "class_name", and "class_id". The column "class_name" contains the labels "adhd", "anxiety", "bipolar", "depression" "ptsd", and none, while the column "class_id" contains the numerical representation which are from 0 to 5 accordingly.

To collect data for the study, the Reddit API was utilized to crawl 13 Reddit Subreddits for a total of 17159 posts (text and title). The content from these posts' comment threads was not collected since it tended to deviate from the subreddit's main topic. Despite the fact that there are several other mental diseases that require consideration, only 5 of them had sufficient data for the goals of the study and were chosen for the purposes of the paper. These 5 mental illnesses are bipolar, attention deficit hyperactivity disorder (adhd), anxiety, depression, and post-traumatic stress disorder (ptsd). Posts in these subreddits were given a class label that corresponded to the name of the mental disease with which they were associated. All the remaining subreddits were carefully chosen to avoid any thematic content overlap with the illness classes. The researchers selected a few subreddits from a wide range of common topics. The text from these subreddits was collected and labeled with the class None [10].

### B. Preprocessing

From the given datasets, we first stripped out unnecessary columns which won't provide any useful information for the prediction of our model. The remaining columns are the "title", "post", and "class_id". Data preprocessing techniques are applied to both "title" and "post" columns of the dataset. The texts were first tokenized or split into a list of sentences. Then, we converted each sentence to lowercase, removed non-alpha characters, and further tokenized or split into a list words. Frequently employed words known as stop words are the removed from these lists of words. To reconstruct the text, each element of this word list is then joined by a single space, and each element of the sentence list is joined by an arbitrary separator. The resulting preprocessed data is a text in lowercase form, no punctuation marks and unnecessary white spaces, and no stop words, in which sentences are separated by a vertical bar '|'. The preprocessing process is illustrated on Fig. 1.

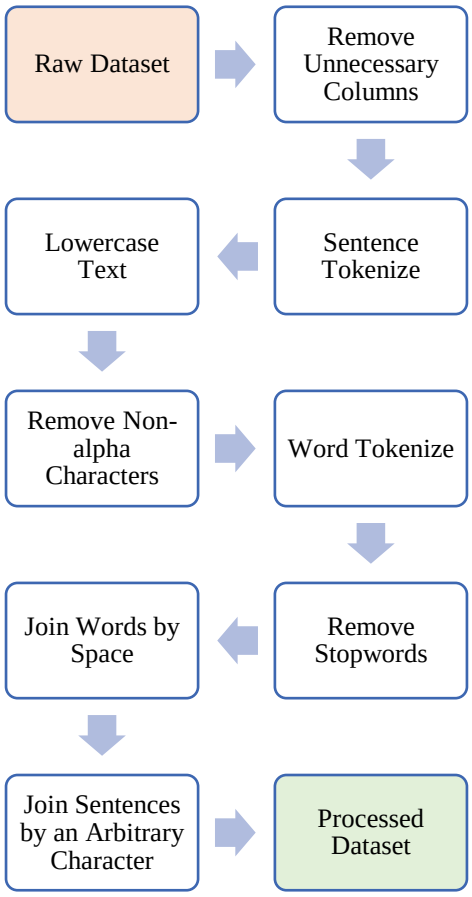


Fig. 1 Preprocessing process

### C. Feature Extraction

For feature extraction, we employed the term frequency-inverse document frequency (TF-IDF) extraction technique, which is one of the most important techniques for representing how relevant a certain word or phrase is to a document corpus in information extraction. The formula that is used to compute the tf-idf for a term *t* of a document *d* in a document set is shown on (1).

$$tfidf(t,d) = tf(t,d) * idf(t) \quad (1)$$

The idf is computed as shown on (2) where *n* is the total number of documents in the document set and df(*t*) is the document frequency of *t*; the document frequency is the number of documents in the document set that contain the term *t*. The effect of adding "1" to the idf in the equation is that terms with zero idf, i.e., terms that occur in all documents in a training set, will not be entirely ignored [11].

$$idf(t) = \log\left(\frac{n}{df(t)}\right) + 1 \quad (2)$$

For tf-idf, we only considered the top 1000 features ordered by term frequency across the corpus. This is to limit the size of our document-term matrix and to avoid overfitting the generated model.

Other linguistic features that are extracted are considered as shallow features. The number of characters, syllables, words, unique words, and sentences are extracted on the samples. With those 5 initial features, 7 more features are further extracted by feature engineering techniques. These features are average number of characters per word, average

number of characters per sentence, average number of words per sentence, average number of syllables per word, average number of syllables per sentence, average number of unique words per word, and average number of unique words per sentence.

These 1012 features are each extracted from the "title" and "post" columns. 1000 of which are from the TFIDF vectorizer while the remaining 12 features are extracted using the different feature engineering techniques. In total, there are 2024 features extracted for each sample data.

### D. Algorithm

For the prediction task, we implemented a random forest classification algorithm which consists of many decisions trees. We used the "RandomForestClassifier" class from the sklearn module provided by scikit-learn.[1] A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. In random forests, each tree in the ensemble is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set. Furthermore, when splitting each node during the construction of a tree, the best split is found either from all input features or a random subset of size *max_features* [12]. In our implementation, we used the square root of the number of features which is the value set by default. The purpose of this randomization is to minimize the variance of the forest estimator since individual decision trees have a large variation and tend to overfit. Forests introduced with randomness provide decision trees with partially decoupled prediction errors. Some inaccuracies can be minimized averaging those predictions. Random forests minimize variance by merging various trees, sometimes at the expense of a minor bias increase. In practice, the variance reduction is frequently considerable, resulting in a better overall model [12].

For hyperparameter optimization, we first conducted a simple grid search to identify a good starting parameter configuration for our random forest classifier. Grid search is simply an exhaustive searching process using an arbitrary subset of the hyperparameter values to get the best combination. We measured the performance of each combination using the resulting F1 score. The hyperparameters we used for conducting grid search are "n_estimators", "max_depth", "min_samples_split", "min_samples_leaf", and "max_leaf_nodes", where each of them is assigned the values 50, 100, and 150. After getting the best hyperparameter value combination on grid search, we conducted hyperparameter tuning to further optimize the model using the grid search results as base or initial parameter values for our random forest classifier with a tuning range of ± 40 and increments of 10. After the hyperparameter optimization processes, we now have the final model.

### E. Evaluation Metrics

We used different evaluation metrics for assessing the performance of our model. These evaluation metrics are precision, recall, F1 score, and the overall accuracy. The formula for these metric follows, where *tp*, *tn*, *fp*, and *fn* is the number of true positives, true negatives, false positives, and false negatives, respectively.

The precision evaluation metric is computed using the formula shown on (3). The precision is intuitively the ability of the classifier not to label a negative sample as positive [13].

$$precision = \frac{tp}{tp + fp} \tag{3}$$

The recall evaluation metric is computed using the formula shown on (4). The recall is intuitively the ability of the classifier to find all the positive samples [13].

$$\text{recall } = \frac{tp}{(tp \ + \ fn)} \tag{4}$$

The F1 score evaluation metric is computed using the formula shown on (5). The F1 score is defined as the harmonic mean of precision and recall. In the F1 score, we compute the average of precision and recall. They are both rates, which makes it a logical choice to use the harmonic mean [13].

$$F1\ score = 2 * \frac{precision * recall}{precision + recall} \tag{5}$$

The last evaluation metric is accuracy. The accuracy evaluation metric is computed using the formula shown on (6). Accuracy is simply the ratio of number of correct predictions to the total number of predictions made.

$$accuracy = \frac{tp + tn}{\text{tp} + \text{tn} + \text{fp} + \text{fn}} \tag{6}$$

## IV. Results and Discussion

TABLE I. below shows the results after performing grid search on our random forest classifier using the values 50, 100, and 150. The results show the best parameter value combination returned by grid search.

TABLE I. Results of Grid Search on Random Forest Classifier

| Hyperparameter | Value |
|---|---|
| n_estimators | 100 |
| max_depth | 50 |
| min_samples_split | 50 |
| min_samples_leaf | 50 |
| max_leaf_nodes | 50 |

Hyperparameter tuning is then conducted to our random forest classifier using the results of the grid search as base or initial values for the different parameters. The results of each hyperparameter tuning with respect to F1 score is visualized using line graphs on the next following figures.

[1] https://scikit-learn.org/stable/index.html

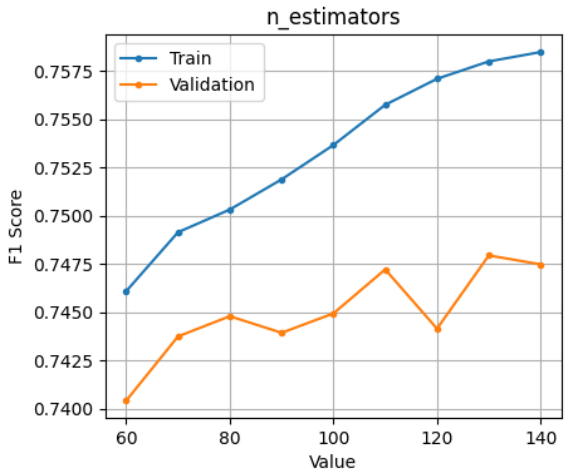


Fig. 2 Hyperparameter tuning graph for "n_estimators"

Fig. 2 illustrates the F1 score of the random forest classifier as we perform hyperparameter tuning on its parameter "n_estimators". The base value for this parameter is 100 and the tuning range is set from 60 to 140 with 10 increments. We can see on the graph that the train score increased steadily until it smooths towards the end. On the other hand, the validation score initially increased and jitters towards the end. The optimal value that we selected is 110 since higher values did not give any significant gains in the validation set and will noticeably increase the computation time. It also already provides good scores for both the train and validation set.

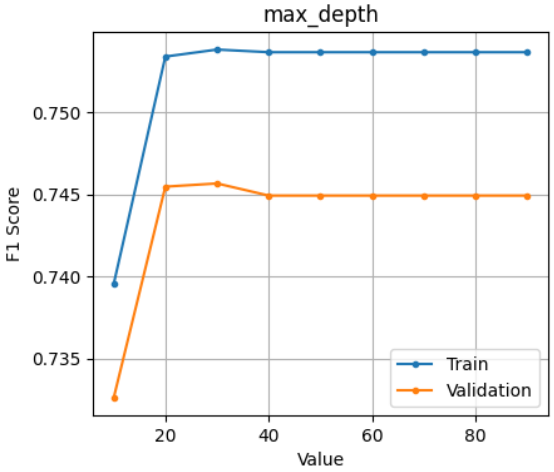


Fig. 3 Hyperparameter tuning graph for "max_depth"

Fig. 3 illustrates the F1 score of the random forest classifier as we perform hyperparameter tuning on its parameter "max_depth". The base value for this parameter is 50 and the tuning range is set from 10 to 90 with 10 increments. We can see on the graph that both the train and validation scores increased dramatically when the parameter value is changed from 10 to 20. Subsequently, the scores stiffed when applying further parameter changes. The optimal value that we selected is 20 since higher values did not virtually result in any change in score and will also increase the computation time noticeably.

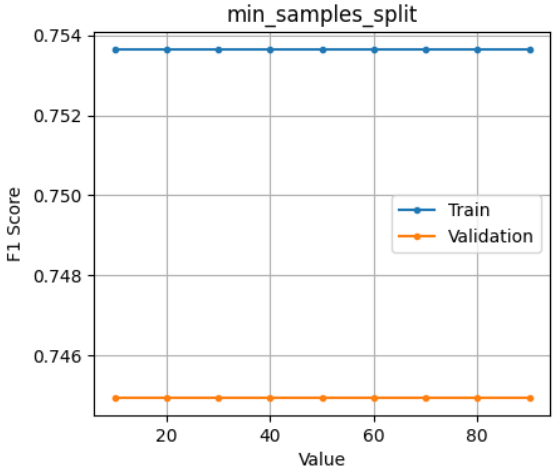


Fig. 4 Hyperparameter tuning graph for "min_samples_split"

Fig. 4 illustrates the F1 score of the random forest classifier as we perform hyperparameter tuning on its parameter "min_samples_split". The base value for this parameter is 50 and the tuning range is set from 10 to 90 with 10 increments. Interestingly, the graph illustrates constant scores across different values on both the train and validation set. This means that finetuning the hyperparameter value of "min_samples_split" does not influence the F1 score of our model. Thus, we just opted for its base value which is 50.

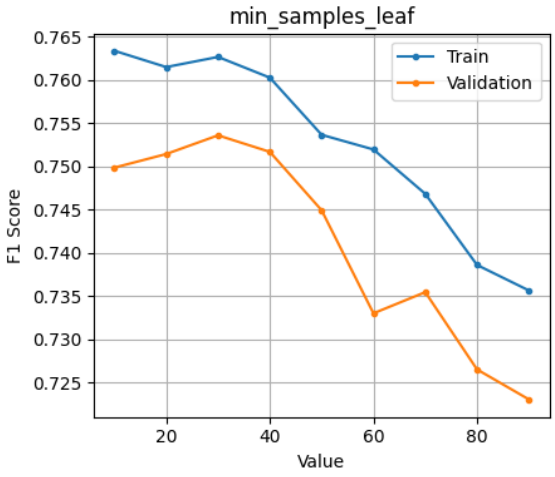


Fig. 5 Hyperparameter tuning graph for "min_samples_leaf"

Fig. 5 illustrates the F1 score of the random forest classifier as we perform hyperparameter tuning on its parameter "min_samples_leaf". The base value for this parameter is 50 and the tuning range is set from 10 to 90 with 10 increments. From the values around 10 to 30, we can observe on the graph that the train scores did not change so much while the validation scores improved a little. The next following values resulted in a steady drop of the scores of both train and validation set where the difference between the highest and lowest peaks is about 0.03. The optimal value that we selected is 30 since it noticeably provides the highest scores.

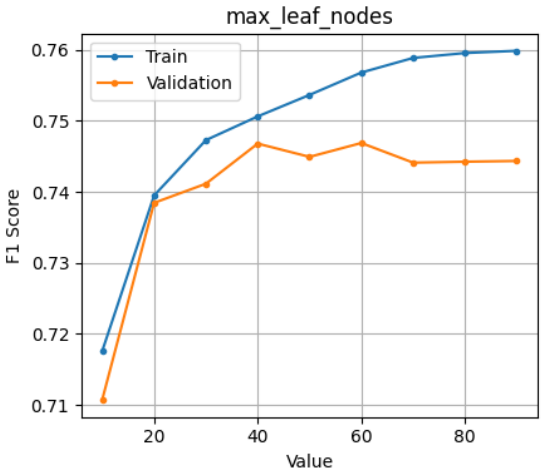


Fig. 6 Hyperparameter tuning graph for "max_leaf_nodes"

Fig. 6 illustrates the F1 score of the random forest classifier as we perform hyperparameter tuning on its parameter "max_leaf_nodes". The base value for this parameter is 50 and the tuning range is set from 10 to 90 with 10 increments. The scores for both train and validation sets increased significantly by about 0.04 from just the parameter values 10 to 40. Further adjustments to the parameter values resulted in a slowly diminishing increase of scores. The optimal value that we selected is 60 since any further increase in the parameter value does not give any more substantial gain to the validation set.

After performing hyperparameter tuning, we used the resulting hyperparameter values for our random forest classifier. The final hyperparameter values used are shown on TABLE II. .

TABLE II. FINAL HYPERPARAMETER VALUES FOR RANDOM FOREST CLASSIFIER

| Hyperparameter | Value |
|---|---|
| n_estimators | 110 |
| max_depth | 20 |
| min_samples_split | 50 |
| min_samples_leaf | 30 |
| max_leaf_nodes | 60 |

We measured the performance of our model by using the different evaluation metrics: precision, recall, F1 score, and the overall accuracy. The score for each metric is shown on TABLE II. below.

TABLE III. EVALUATION SCORES

| | Precision | Recall | F1 Score |
|---|---|---|---|
| ADHD | 0.82 | 0.71 | 0.76 |
| Anxiety | 0.77 | 0.73 | 0.75 |
| Bipolar | 0.78 | 0.63 | 0.70 |
| Depression | 0.49 | 0.80 | 0.61 |
| PTSD | 0.92 | 0.69 | 0.79 |
| None | 0.80 | 0.81 | 0.80 |

The overall accuracy of the model is measured at 0.74, meaning that the model is 74% accurate at predicting the mental illnesses on the dataset used.

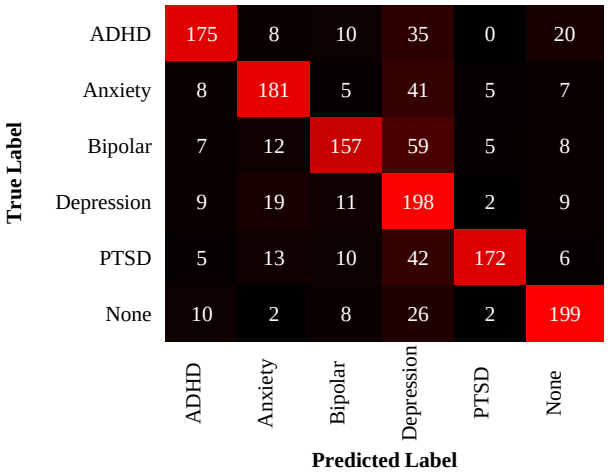


Fig. 7 Heatmap of classification result

Fig. 7 shows a visual representation of the classification results in a form of a heatmap. Interestingly, we can observe across the figure that the model tended to classify the samples as depression posts regardless of the true label of the sample. We can also observe that across all the mental illness classes, the "None" class has the least false depression prediction. This means that the model encounters some difficulty in differentiating between depression and all other mental illness classes studied.

## V. CONCLUSION

Our study explored different feature extraction techniques with the utilization of random forest classifier in predicting mental illnesses. In conclusion, we believe that our study explores an interesting field of research where the help of social media platforms and natural language processing techniques are used for early prediction of the different possible mental illnesses that an individual may have.

The evaluation of the model using multiple natural language processing techniques demonstrated the success of the proposed methodology. Conducting hyperparameter tuning shows varying results on the F1 score of the model. The model selection and hyperparameter optimization processes improved the performance of the model to a total of about 74% in predicting mental illnesses. Interestingly, the model encountered a high correlation between depression and the other mental illnesses studied. This suggests that the mental illnesses studied are highly overlapping and may require multi-label classification techniques to further improve the prediction.

## REFERENCES

[1] "Global Health Data Exchange (GHDx)," Institute for Health Metrics and Evaluation, 2019. [Online]. Available: https://vizhub.healthdata.org/gbd-results/. [Accessed: 05-Nov-2022].

[2] "Mental health and covid-19: Early evidence of the pandemic's impact," World Health Organization, 2022. [Online]. Available: https://www.who.int/publications/i/item/WHO-2019-nCoV-Sci_Brief-Mental_health-2022.1. [Accessed: 05-Nov-2022].

[3] K. A. Aschbrenner, J. A. Naslund, T. Grinley, J. C. Bienvenida, S. J. Bartels, and M. Brunette, "A survey of online and mobile technology

use at Peer Support Agencies," Psychiatric Quarterly, vol. 89, no. 3, pp. 1–10, 2018.

[4] M. L. Birnbaum, A. F. Rizvi, C. U. Correll, J. M. Kane, and J. Confino, "Role of social media and the internet in pathways to care for adolescents and young adults with psychotic disorders and non-psychotic mood disorders," Early Intervention in Psychiatry, vol. 11, no. 4, pp. 290–295, 2015.

[5] S. Bucci, M. Schwannauer, and N. Berry, "The Digital Revolution and its impact on Mental Health Care," Psychology and Psychotherapy: Theory, Research and Practice, vol. 92, no. 2, pp. 277–297, 2019.

[6] J. A. Naslund, K. A. Aschbrenner, L. A. Marsch, and S. J. Bartels, "The Future of Mental Health Care: Peer-to-peer support and social media," Epidemiology and Psychiatric Sciences, vol. 25, no. 2, pp. 113–122, 2016.

[7] M. Ernst, J. Gowin, C. Gaillard, R. Philips, and C. Grillon, "Sketching the power of machine learning to decrypt a neural systems model of behavior," Brain Sciences, vol. 9, no. 3, pp. 1–17, 2019.

[8] A. C. Arevian, D. Bone, N. Malandrakis, V. R. Martinez, K. B. Wells, D. J. Miklowitz, and S. Narayanan, "Clinical state tracking in serious mental illness through computational analysis of Speech," PLOS ONE, vol. 15, no. 1, 2020.

[9] J. Kim, J. Lee, E. Park, and J. Han, "A deep learning model for detecting mental illness from user content on social media," Scientific Reports, vol. 10, no. 1, 2020.

[10] A. Murarka, B. Radhakrishnan, and S. Ravichandran, "Classification of mental illnesses on social media using RoBERTa," Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis, pp. 59–68, Apr. 2021.

[11] "Sklearn.feature_extraction.text.TfidfTransformer," scikit. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html. [Accessed: 06-Nov-2022].

[12] "Sklearn.ensemble.randomforestclassifier," scikit. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Accessed: 06-Nov-2022].

[13] "Sklearn.metrics.precision_recall_fscore_support," scikit. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html. [Accessed: 09-Nov-2022].