*2024 3rd International Conference for Innovation in Technology (INOCON)*
*Karnataka, India. Mar 1-3, 2024*

# Psychological Distress Detection and Classification

Parth Gujarathi
*Department of Computer Engineering*
*Sardar Patel Institute of Technology*
Mumbai, India
parth.gujarathi@spit.ac.in

Kartik Menon
*Department of Computer Engineering*
*Sardar Patel Institute of Technology*
Mumbai, India
kartik.menon@spit.ac.in

Jai Patel
*Department of Computer Engineering*
*Sardar Patel Institute of Technology*
Mumbai, India
jai.patel@spit.ac.in

Sunil Ghane
*Department of Computer Science*
*Sardar Patel Institute of Technology*
Mumbai, India
sunil_ghane@spit.ac.in

2024 3rd International Conference for Innovation in Technology (INOCON) | 979-8-3503-8193-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/INOCON60754.2024.10511974

***Abstract*—This research paper introduces an innovative method for stress detection and classification by utilizing a combination of machine learning, deep learning techniques, and BERT. Today with the advancement of technology and techniques like artificial intelligence and machine learning various manual tasks can be replaced very easily. Hence in this research paper, we have proposed various machine learning algorithms, deep neural networks, and BERT pre-trained models for the detection of stress based on their social media posts and then classifying their stress levels based on their responses to the universally accepted DASS(Depression Anxiety Stress Scales) questionnaire. This study aims to develop an effective system that can accurately detect and classify stress levels based on textual data. The results reflect that our system outperforms existing methods for stress detection and classification and proposes a refined approach for the classification of stress levels. This system, therefore, has potential applications in various domains, including mental health diagnosis, social media monitoring, and personalized stress management.**

***Keywords — Stress, DASS, CNN, RNN, BERT, word2vec, glove***

## I. Introduction

Psychological distress refers to a range of symptoms that can occur due to various factors, including stress, anxiety, depression, and trauma. It can have a profound impact on an individual's mental and physical health, leading to reduced quality of life, impaired functioning, and increased risk of various health conditions. Therefore, early detection and classification of psychological distress are essential for effective management and treatment.

In recent years, social media has become a ubiquitous part of modern society, providing individuals with a platform to share their thoughts, emotions, and experiences. This vast amount of user-generated data presents an opportunity to study and understand psychological distress in a novel way. By analyzing social media data, researchers can gain insights into the mental health of individuals and populations in real time. Therefore, there is a growing interest in using social media data and machine learning techniques to detect and classify psychological distress.

Today a considerable number of messages are posted on social media handles which tend to reflect the person's mental state at that given time. Therefore social media can be an important indicator of the stress levels of a person and can be used effectively for timely detection and cure.

But sometimes these messages posted on social media may not always reflect the mental health of the person. It is hence very important to also verify the prediction based on a social media post and not completely rely on it.

This is why we in this research paper have come up with a two-tier approach where we first detect whether a user is stressed or not based on his/her social media post. Then the user answers the universally verified DASS questionnaire from which we can exactly predict the stress level of the user.

Our approach aims to develop an effective system that can accurately detect and classify stress levels based on textual data and the information provided by the user by answering the DASS questionnaire. We train and test our proposed system on a dataset of social media posts and compare its performance with existing methods for stress detection and classification. Our results show that the proposed system outperforms existing methods.

## II. Literature Review

Before starting, it was important for us to gain insight and study the existing work that had been done in this field so that we can take inspiration from that work and further enhance it. We have studied various research methods on stress detection and classification based on social media posts and some on the DASS questionnaire. These papers helped us understand how to move forward with our work and what were some of the drawbacks we could overcome.

To further understand the stress levels predicted by the DASS questionnaire the paper [3] written by A Priya, S Garg, N Tigga was studied by us. The goal of this study was to predict the stress and anxiety levels of individuals by analyzing their responses to a questionnaire. The questionnaire included questions related to symptoms of depression and

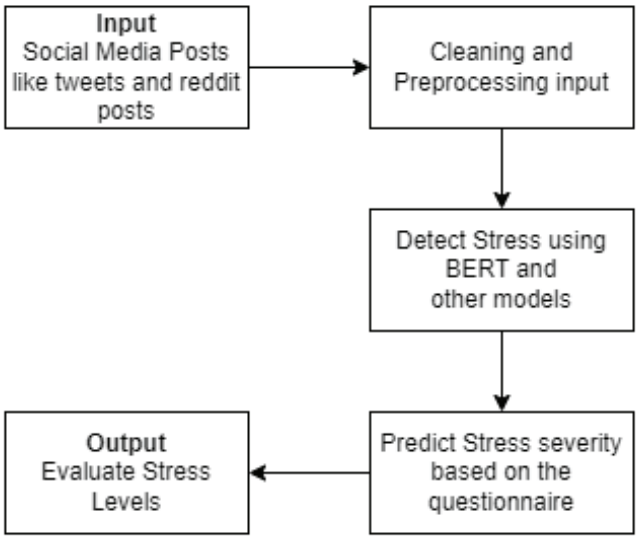


Fig. 1. General Approach of the System

stress, which were used to classify the severity of distress. Five different levels of anxiety, depression, and stress were detected in this paper. Classification algorithms were trained on the collected data to predict the level of distress. The accuracy of the different algorithms was compared, and Naive Bayes was found to have the highest level of accuracy in predicting distress severity. The accuracy reached was 97 percent.

The next paper [1] we studied was written by RR Baheti, S Kinariwala. This research paper introduced a technique to identify stress and relaxation expressions on social media platforms, particularly Twitter. The study used a dataset from Twitter and employed the TensiStrength framework, a lexicon-based method for detecting stress strength in text. The tweets were classified into stress levels ranging from -5 to +5 using this framework. It analyzed the sentiment of a user's tweet and based on that predicted the stress level. However, the accuracy achieved in this paper was fairly low i.e. 68

The paper [7] predicted the stress of a user based on a social media post. In this paper, they also created datasets based on Reddit and Twitter posts which we also used as a part of our research work. The study collected data from various subreddits and utilized transfer learning to gather and preprocess the data into four categories. The data was cleaned by removing noise and adverts. The study employed pre-trained models such as BERT to fine-tune the data. The stress detection process involved using lexicon, embedded, and PLM-based methods. The pre-trained models were leveraged to enhance the process of detecting stress. The study also utilized the lexicon-based approach, which involves using a dictionary of stress-related terms to classify text into stress categories. The embedded method involved transforming the text into a numerical representation to classify stress, while the PLM-based method used pre-trained language models to classify stress levels. One of the major drawbacks of this paper was that the model's accuracy was reduced on large-sized texts. Also, the lexical and embedded-based models failed to catch dependencies between the texts

To further understand the use of natural language processing in detection of stress the paper [8] written by T. Ananthakrishna and others provided us with deep insights into various techniques to process and analyze textual information. This research aimed at building models for the detection of user's sentiments and emotions which could then be used for stress control and then classifying the textual data into 5 emotions. The paper also used the LDA algorithm for the analysis of user tweets and to provide in-depth data visualization. The study emphasized the significance of visualizations in gaining a comprehensive understanding of the data. LDA was used to determine the number of topics in the extracted tweets and the occurrence of a word in a particular topic.

Another approach to the detection of stress which we came across was based on images posted on social media. For this, the paper [6] detected and predicted stress in microblog messages based on words, images, emoticons and social interaction. This paper classified emotions into two categories: positive and negative. The study also used w feature extraction techniques such as dullness, brightness, and clearness to identify stress in images. The features were calculated using the mean value between specified threshold values. The two major drawbacks of the paper were that 1) it classified dark images as negative which might not be the case always. 2)It used a conversation between individuals to predict stress in them. The drawback to this is these conversations are private and hence it cannot be applied to the masses and hence the scope becomes very limited.

## III. Proposed Methodology

### A. Part 1 - Stress Detection

*1)* ***Datasets****:* We majorly used two datasets for our research work-

a) We first used the twitter emotion dataset publicly available on kaggle for stress detection. The dataset contains two columns- the tweet of the user and the emotion of the tweet. The emotion column has 6 classes namely joy, sadness, anger, fear, surprise and love. Emotions such as joy and love are commonly termed as not stress and sadness, anger and fear are commonly linked to stress. Thus we classified joy and love as not stressed while we classified sadness, anger and fear as stressed. The final dataset roughly contained twenty thousand rows and it was fairly balanced between the two classes.

b) Another dataset on which we did our research work that was taken from another research paper we had studied. It contained tweets from Twitter and posts from subreddits and whether the tweets, and posts were stressed or not stressed in two different files. We combined both the files to make one dataset which roughly contained 5000 rows but was a bit unbalanced with stress posts being more than the non-stress ones.

*2) **Data pre-processing**:* Once the dataset is transformed into the most suitable form, the next step is to preprocess and clean the data to make it suitable for us to apply various algorithms.The steps involved are:

- Removing Hashtag, Mention, URLs
- Text to lowercase
- Stemming - Stemming is the process of deriving base or root form of words by removing their affixes. In natural language processing, stemming is a technique used to normalize words to their base form, or ”stem,” which allows for more efficient text analysis and retrieval.
- Lemmatizing - Lemmatizing is similar to stemming, but instead of just removing suffixes to obtain the base form of a word, it uses a dictionary-based approach to determine the base form of a word depending on its context in the sentence.
- Removing Punctuations
- Removing stopwords - Stopwords are common words that are often removed from the text during natural language processing to focus on the most relevant words and phrases. Examples of stopwords in English include ”the,” ”a,” ”an,” ”and,” ”or,” ”in,” ”on,” and ”of.”
- Substitution of emojis

Data preprocessing is a critical step in NLP that helps to improve data quality by removing noise and inconsistencies, such as typos, punctuation, and grammatical errors, improving algorithm performance, extracting relevant features, and standardizing data representation which makes it easier to compare and combine different datasets.

*3) **Vectorization**:* After data preprocessing, only the important words for each sentence will remain. Before applying ML models, the preprocessed text is passed through a vectorizer. This is done because traditional machine learning models require numerical inputs, while text data is typically present in the form of raw or unstructured format. Vectorization gives the numeric representation of the text data as input to ML algorithms, and reduces the dimensionality of the data, making it more manageable for the algorithm to process. The vectorizers we used were:

- **Bag of Words:** The bag of words (BoW) vectorizer converts each text document into a vector of fixed-length of word frequencies. Each element of the vector represents a word in the vocabulary, and its value shows the frequency of that word in the document.
- **TF-IDF:** TF-IDF stands for Term Frequency-Inverse Document Frequency. Term Frequency (TF) refers to the number of times a word appears in a document and Inverse Document Frequency (IDF) refers to how rare the word is across the entire text corpus. The TF-IDF score is obtained by multiplying TF and IDF and it indicates the importance of the word in the document. Higher scores imply greater importance.

*4) **Traditional Machine Learning Models**:* After vectorization, ML models were used to train on the given input. Sentiment analysis involves categorizing user-generated text into stress or not stress. Machine learning models, including traditional ones like SVM, Naive Bayes, and Random Forest, as well as advanced techniques like RNNs, CNNs, and transformers (e.g., BERT), are used for this classification task. These models aim to interpret language nuances and context to accurately predict sentiment, facilitating applications in social media analysis, customer feedback interpretation, and opinion mining by extracting insights from large volumes of user text data. The traditional machine learning models used were:

- **SVM** Once data has been vectorized, SVM seeks to find the optimal hyperplane that separates these vectors into different sentiment classes (e.g., stress or not stress) by maximizing the margin between the classes. This involves mapping the textual features to a higher-dimensional space, allowing SVM to create an effective decision boundary. The model is trained on labeled sentiment data to learn the patterns in the text that correspond to specific sentiments. During prediction, SVM uses this learned decision boundary to classify new user-generated text into sentiment categories based on its position relative to the separating hyperplane.
- **Naive Bayes:** For this research work, we have used 2 variants of Naive Bayes classifier: **Multinomial** and **Gaussian Naive Bayes**. Naive Bayes models are applied in sentiment analysis tasks by calculating the probability of a text belonging to different sentiment classes (e.g., stress or not stress) based on the occurrence of words or features in the text. It estimates the conditional probability of each sentiment given the observed words, assuming that the words are independent of each other, despite this simplification. During classification, the model selects the sentiment class with the highest probability as the predicted sentiment for the given user-generated text.
- **Random Forest:** Random Forest operates as an ensemble learning method composed of multiple decision trees. Random Forest constructs multiple decision trees by selecting random subsets of features and data instances. During training, each decision tree in the ensemble is trained on different subsets of the data and features, learning various patterns in the text related to sentiments (e.g. stress or not stress). During prediction, each tree ”votes” on the sentiment class, and the final sentiment prediction is determined by aggregating these votes. The model outputs the sentiment class that receives the most votes across all decision trees

However, the conventional classification models failed to yield accurate results due to their inability to manage complex relationships, nuances, and contextual information present within textual data. Consequently, to address this limitation, we delved deeper and opted to utilize advanced deep learning techniques such as neural networks and transformers. These methodologies were chosen specifically to effectively capture and interpret the genuine sentiment expressed by users in their textual content.

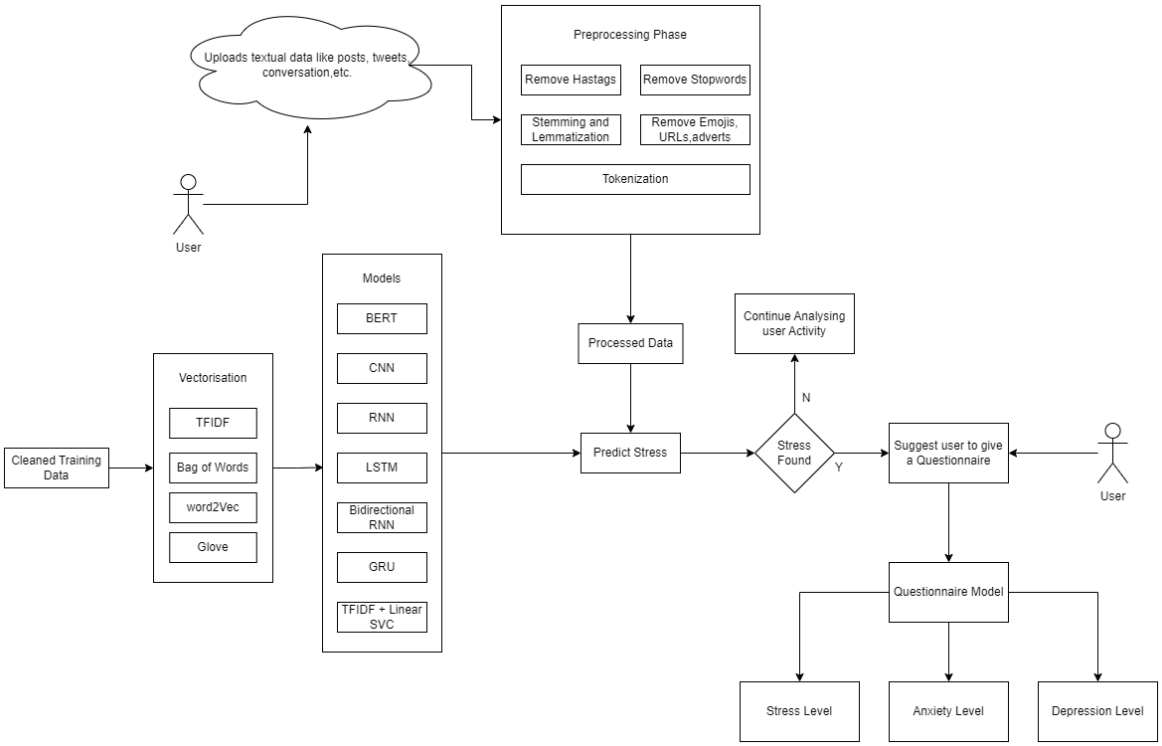


Fig. 2. System Diagram

*5) **Deep Learning based techniques**:* Due to the failure of traditional machine learning models to detect nuances in human text, advanced deep learning techniques such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and transformer models like BERT were also for this classification task. These deep learning models were employed to comprehend intricate language nuances and contextual cues, facilitating precise sentiment prediction. The deep learning based techniques used were as follows:

- **RNN:** RNNs process textual data sequentially, considering the order of words or characters in the input text. They utilize a recurrent structure that allows information to persist and flow through the network, enabling them to capture contextual dependencies and nuances present in the text. For this research work, after tokenization, we padded the sequences to a fixed length of 1500 and then built a model with an embedding layer, a SimpleRNN layer with 100 hidden units, and a dense output layer with a sigmoid activation function. The model is then compiled using binary cross-entropy loss, Adam optimizer, and accuracy metric.
- **LSTM:** LSTM, which stands for Long Short-Term Memory, is a type of RNN designed to capture long-term dependencies in sequential data. For sentiment analysis, LSTM is used for capturing contexts in case of lengthy texts. LSTM networks use a cell state that runs through a series of gates to decide what information to keep, forget, or output to the next time step. After pre-loading the GloVe word embeddings, we build an LSTM model for binary classification. It uses the pre-trained vectors for embedding words, then adds an LSTM layer with dropout regularization and 100 hidden units, and a dense output layer with sigmoid activation. The model is trained on padded training data for 5 epochs with a batch size adjusted for distributed training. This model is also compiled with binary cross-entropy loss, Adam optimizer, and accuracy metric.
- **GRU:** GRU stands for Gated Recurrent Unit, is another type of recurrent neural network (RNN) architecture identical to LSTM networks. Like LSTMs, GRUs handle the vanishing gradient problem that can occur with traditional RNNs. Similar to LSTM, it uses pre-utilized GloVe word embeddings. The model includes an Embedding layer for mapping words to vectors, a SpatialDropout1D layer for preventing overfitting with dropout rate set to 0.3(Dropout rate refers to the probability of randomly dropping out individual units in a neural network layer during training), a GRU layer with 300 units for learning long-term dependencies between words, and a Dense layer with sigmoid activation for predicting class probabilities. The model is configured for training with binary crossentropy loss and Adam optimizer.

  For **LSTM** and **GRU**, GloVe vectors were used as the embedding layer. GloVe (Global Vectors) is a type of unsupervised learning algorithm used to create word embeddings. It uses a co-occurrence matrix of words

in a corpus to capture semantic relationships between words. GloVe is similar to other word embedding algorithms, such as Word2Vec, but it takes a different approach to creating embeddings. GloVe creates a global word-to-word co-occurrence matrix and then factorizes it to obtain word embeddings. The resulting embeddings are dense vectors representing each word in a high-dimensional space, where words with similar meanings are closer. Additionally, both use binary crossentropy loss and Adam optimizer. Binary crossentropy loss is a function measuring the difference between predicted and actual probabilities in binary classification tasks (0 or 1). It is easy to interpret and efficiently optimizes models. Adam optimizer is an algorithm for updating model weights during training, addressing limitations of other optimizers like slow convergence and sensitivity to parameter initialization. It was chosen over other optimizers because of its fast convergence and lower sensitivity to parameter initialization

- **CNN:** Convolutional Neural Network is a deep learning architecture commonly used for image recognition tasks, but it can also be applied to text analysis tasks such as text classification and sentiment analysis. CNN operates on a text input by first converting each word in the input into a vector representation (e.g., using word embeddings such as Word2Vec or GloVe). After word tokenization, the dataset is split into training and testing dataset. To input the data into the neural network, it must be converted into vectors. This has been done using Google's Word2Vec model which has been trained on a large corpus of Google News data. Word2Vec model generates dense word embeddings, which capture the semantic relationship between words by representing them as highly dimensional vectors. After creating vectors of the training data, batches of size 34 have been created to improve the efficiency of the model. After creating batches, the CNN model is trained on the training dataset for 4 epochs. Once training was complete, we tested the model on our testing dataset and calculated the accuracy of the model.
- **BERT:** BERT is a powerful pre-trained language model for NLP tasks. Unlike traditional methods, it considers the entire context, leading to high-quality word embeddings. For specific tasks like sentiment analysis, BERT fine-tunes its pre-trained model for even better performance. To use BERT for stress detection, we tokenize each text using BertTokenizer into token IDs as inputs in a BERT model should contain only tokens. Now, the dataset is split into 3 datasets: training dataset, testing dataset, and validation dataset. These 3 datasets must be converted into Tensor as input to the BERT model is fed in the form of vectors. To improve the speed and efficiency of the model, the training dataset is split into batches of size 32. Then, we trained the BERT model on the training dataset and loss is calculated for each epoch. Finally, we tested the model on our testing and validation dataset and calculated the accuracy of the model.

*6) **Output**:* The user's post is taken as the input to our model which is then given as input to our trained model which then detects whether the post was stress or not stress. We choose a model that has shown to provide the highest accuracy scores during the testing phase. The model then yields probabilities for stress and non-stress scenarios and the one with the higher value is chosen as the final decision of the model.

### *B. Part 2 - Stress Classification*

*1) **Dataset**:* Stress classification was done with the help of the DASS questionnaire which contains 42 questions based on depression, anxiety and stress. Each of the 3 categories contain 14 questions each and each question has a score from 1-5 based on the response selected. Based on the final score, we can determine the severity of depression, anxiety or stress. Scores for Depression, Anxiety and Stress are calculated by summing the scores for the relevant items:

Stress: 1, 6, 8, 11, 12, 14, 18, 22, 27, 29, 32, 33, 35, 39

Depression: 3, 5, 10, 13, 16, 17, 21, 24, 26, 31, 34, 37, 38, 42

Anxiety: 2, 4, 7, 9, 15, 19, 20, 23, 25, 28, 30, 36, 40, 41

The dataset used contains responses to all 42 questions along with the time taken to answer the question and position of the question in the survey. The dataset also contains responses to the Ten Item Personality Inventory (TIPI) which consists of 10 questions and 7 possible responses, and details about the user such as education, gender, living conditions, age, religion, race, etc. The dataset contains responses from almost 40,000 users.

*2) **Data pre-processing**:* All of the data is already in numeric form so not much preprocessing is required. As for the columns used, columns that contain the response for the 42q questionnaire, the users education, living conditions, gender, age, race, marital status and family size. Then the user's depression, anxiety and stress score is calculated based on the questions mentioned above and a severity label is given based on the scores.

*3) **ML Models used**:* Now the dataset is divided into the 3 categories i.e. depression, anxiety and stress, with each category containing their respective question responses, user details, user score and user label. For each of the 3 datasets, it is split into train and test datasets, then standardized to have mean=0 and standard deviation=1 and then SVM is applied on the dataset. Kernel used for SVM is 'RBF'.

*4) **Output**:* The user is given an option whether he wants to give a questionnaire only for Stress, Depression, Anxiety or any other combination among the three categories. The input given to the model is the user's details (education, living conditions, gender, age, race, marital status and family size), and their questionnaire responses. From this, the model is applied for the three categories: stress, depression and anxiety and as output, the severity of these categories is displayed to the user. The severity categories are: Normal, Mild, Moderate, Severe and Extremely Severe.

## IV. Results

### A. Part 1-Stress Detection

After applying the various above mentioned techniques to the datasets the BERT model best performed for stress detection of social media posts. Here is a comparative study of all the algorithms applied to both the datasets showing each of their accuracies.

| Algorithm | Emotion | Reddit |
|---|---|---|
| TFIDF + Linear SVC | 92% | 92% |
| RNN | 93% | 94% |
| LSTM | 93.3% | 97% |
| Bi-directional RNN | 94.5% | 97% |
| GRU | 96% | 96% |
| CNN | 97% | 93% |
| BERT | 98% | 95.7 |

### B. Part 2- Stress Classification

The accuracy of the SVM model is:

- For depression model: 99.1%
- For anxiety model: 99.4%
- For stress model: 99.3%

For the DASS based questionnaire model a accuracy of as high as 99.95 percent was achieved by carefully preprocessing and cleaning the data and applying the support vector machine algorithm.

## V. Conclusion

Based on the results of this research paper, it can be concluded that social media posts can be a valuable source of information for predicting and classifying stress levels. The study successfully utilized a combination of deep learning models including BERT, CNN, RNN, and LSTM to accurately classify stress with up to 98% accuracy. In addition, the questionnaire model integrated into the research allowed for the prediction of stress severity levels. This approach provides a more holistic understanding of stress levels by combining both self-reported measures and automatic analysis of social media posts. Overall, this study shows that social media can be used as a valuable resource for detecting and predicting stress levels. This information can be valuable for mental health professionals and researchers who are interested in developing targeted interventions and providing support to individuals experiencing stress. The deep learning models employed in this study can serve as a foundation for future research and may be further improved with larger and more diverse datasets.

## VI. Future Work

Based on the findings of this study, there are various directions in which we can further our study. One possible avenue is to explore the transferability of the models used in this study to different populations and languages. The current study was conducted on English language social media posts, and it would be valuable to investigate the generalizability of these models to other languages and cultures.

Another area for future work is to incorporate additional features into the models to further improve their performance. For example, social media metadata such as timestamps, geolocation, and user demographics could be used as additional features to enhance the accuracy of stress classification.

In addition, future research could explore the use of these models for real-time stress monitoring and intervention. By analyzing social media posts in real-time, it may be possible to provide immediate support and resources to individuals who are experiencing high levels of stress.

Finally, it would be valuable to explore the ethical implications of using social media data for stress analysis and prediction. Future research should address questions of privacy, data security, and informed consent to ensure that the use of social media data is transparent and respectful of individuals' rights and autonomy.

## References


[1] Baheti, R.R. and Kinariwala, S. Detection and Analysis of Stress using Machine Learning Techniques, International Journal of Engineering and Advanced Technology (IJEAT), Volume-9, 2019

[2] Thilagavathi P, Pushkala P, Suresh Kumar. A and Yamini P, Detecting Stress based on Social interactions in social network, International Journal of Engineering, Research and Technology(IJERT), 2018

[3] Priya A, Garg S, Tigga N.P, Predicting Anxiety Depression and Stress in Modern Life using Machine Learning Algorithms, Procedia Computer Science, Volume 167, 2020

[4] [4] Dham V, Rai K, Soni U, Mental Stress Detection using Artificial Intelligence Models, International Conference on Mechatronics and Artificial Intelligence, Volume 1970, 2021

[5] M. Pacula, T. Meltzer, M. Crystal, A. Srivastava and B. Marx, "Automatic detection of psychological distress indicators and severity assessment in crisis hotline conversations," 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014

[6] Ali M.M, Hajera S, Psychological Stress Detection from Social Data using a Novel Hybrid Model, International Journal of Intelligent Systems and Applications in Engineering(IJISAE), 2018

[7] Rastogi A, Liu Q, Cambria E, Stress Detection from Social Media Articles, Department of Electrical Engineering, Indian Institute of Technology Indore, India, School of Computer Science and Engineering, Nanyang Technological University, Singapore

[8] Nijhawan, T., Attigeri, G. Ananthakrishna, T. Stress detection using natural language processing and machine learning over social interactions. J Big Data 9, 33 (2022).

[9] Guntuku S.C., Buffone A, Jaidka K, Eichstaedt J, Ungar L.H, Understanding and Measuring Psychological Stress Using Social Media, University of Pennsylvania, Nanyang Technological University, 2019