Enhancing Mental Health: Stress Level Prediction through a Machine Learning and NLP Approach

Raja Muppidi
Applied Computing
Michigan Technological University
Houghton, USA
rmuppidi@mtu.edu

Bhuvana Muriki
Applied Computing
Michigan Technological University
Houghton, USA
bmurki@mtu.edu

***Abstract*—The increasing global incidence of mental health issues calls for innovative methods for early detection and intervention. This study uses machine learning and natural language processing (NLP) to predict stress levels from textual data from online platforms. Preprocessing steps like text normalization, tokenization, and vectorization transform the raw text into a structured format for ML analysis. Various models, including Naive Bayes classifiers and more sophisticated neural network architectures like LSTM and GRU, were developed and compared. Our results revealed that the GRU model slightly outperformed the LSTM with an accuracy of 91.58%, precision of 91.65%, recall of 91.58%, and an F1-score of 91.58%, but the ensemble model exhibited the best overall performance with an accuracy of 92.46%, precision of 92.50%, recall of 92.46%, and an F1-score of 92.46%. These findings highlight the potential of advanced NLP techniques in real-time monitoring and intervention in mental health, paving the way for future applications.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Mental health disorders, particularly stress, have become a significant public health challenge globally. Traditional methods for diagnosing stress involve psychological assessments and are often subjective and time-consuming. As a result, there is a growing need for more scalable and objective approaches. Machine learning (ML) offers promising tools for analyzing complex datasets and extracting meaningful patterns, providing a compelling alternative for the automatic detection and prediction of stress levels based on behavioral data.

The motivation for our project is twofold: firstly, to harness the power of ML to offer an automated, quick, and more objective method of stress detection that can handle large-scale data effectively, and secondly, to enhance early diagnosis and personalized treatment plans that could potentially prevent the onset of severe mental health issues.

The application of ML in stress detection is not entirely new, with various studies exploring different facets of this problem with varying degrees of complexity and success. U.S. Reddy et al. [1] investigated the use of boosting and bagging ensemble methods with decision trees to predict stress among working employees, showing that ensemble methods could improve prediction accuracy compared to individual classifiers. However, their study was limited to structured data and did not explore unstructured text data, which can provide deeper insights into an individual's psychological state.

Our project builds upon these foundational studies by introducing several key innovations: unlike the studies focusing on structured data [2], our approach utilizes unstructured text data from online platforms, providing a richer dataset for analysis. We employ advanced NLP techniques, including custom embedding layers and sophisticated neural architectures like LSTM and GRU, to capture the sequential and contextual nuances of language that are critical for accurate stress prediction.

Comparative analysis: We not only develop individual models but also perform a comparative analysis across multiple models, including an ensemble approach, to identify the most effective technique in stress level prediction. Our project aims to bridge the gap between traditional mental health assessments and modern automated techniques by leveraging state-of-the-art ML and NLP methods.

II. Dataset

A. Overview

The dataset utilized in Our Project a comprises textual data extracted from various online sources where individuals express their thoughts and emotions. These textual expressions are indicative of their mental state, which we analyze to predict levels of stress. The text data offers a real-time, rich source of linguistic features that are instrumental in detecting subtle nuances related to stress.

B. Data Collection

The dataset was sourced from public online forums, social media platforms, and other digital communication channels, which is available publicly on Kaggle. These sources provide a diverse array of textual content, ranging from casual conversations to more structured posts, all of which reflect the users' psychological states at different moments. The collection methodology ensured a broad representation of demographics, including age, gender, and geographic diversity, to enhance the generalizability of the study [3].

C. Data Preparation and Prepossessing

1) Data Composition: Upon initial inspection using the df.info() function, the dataset was confirmed to have no null entries, indicating a complete dataset with no missing values. The dataset comprises two columns: text, which contains the textual data, and label, which denotes the classification of the text as stress-related or not.

The balance of the dataset was assessed by plotting the value counts of the 'label' column, revealing a near-even distribution. With the pie chart showing a nearly 50/50 split between the two classes, as Fig. 2.

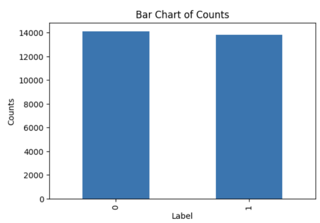

Fig. 1. Count of Distribution Between Non-Mental and Mental-Health Related Texts

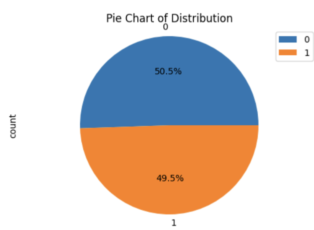

Fig. 2. Percentage of Distribution Between Non-Mental and Mental-Health Related Texts

2) Data Splitting: The dataset contains a total of 27,972 entries. It was split into training, validation, and test sets with a distribution of 70

3) Prepossessing: Data preprocessing is a crucial step in preparing text data for machine learning models. It involves transforming raw data into a standardized format that enhances model performance and accuracy. In Our Project a, the dataset underwent several preprocessing steps to ensure it was suitable for sophisticated NLP techniques.

The raw textual data was first cleaned to remove irrelevant or noisy information, such as URLs and HTML tags. Special characters like punctuation marks and numerical figures were also removed to focus the model's learning on stress indicators. Lowercasing was done to prevent the same words from being counted as distinct tokens across the dataset.

Tokenization and stop word removal were then performed on the cleaned text entry. Tokenization breaks down sentences into individual words or tokens, while stop word removal removes common English words that typically do not carry significant meaning in the context of the analysis. Lemmatization reduces the variability of the language and consolidates words with similar meanings, aiding in the generalization of the model.

The final step in the preprocessing phase was the conversion of textual data into a machine-readable format through text vectorization and normalization. TensorFlow's 'TextVectorization' layer was used to map words to integers, standardizing the length of text entries. Input normalization was set by setting a fixed length for all text entries, ensuring uniform structure in inputs fed into the models.

In conclusion, the comprehensive preprocessing of the dataset was critical in developing robust models for stress level prediction. By systematically cleaning, tokenizing, lemmatizing, and vectorizing the text data, the dataset was primed for the application of advanced NLP techniques, contributing to the success of the machine learning models employed in Our Project a.

D. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a crucial step in data-centric research, providing initial insights into the nature and characteristics of the dataset. In the context of predicting stress levels from textual data, EDA helps in understanding the distribution of classes, the balance of the dataset, and the typical structure of the text data. Text length analysis is essential for determining the necessary padding or truncation parameters for vectorization, ensuring that models have sufficient context for making accurate predictions.

The distribution of word counts in text entries is pivotal, as it affects how data is preprocessed and configured in neural networks. A histogram of text lengths visualizes the frequency distribution of text lengths, revealing a broad range of text sizes and helping identify an appropriate sequence length for modeling. The 95th percentile of text lengths is calculated to ensure that 95

The dataset was also analyzed to understand the distribution of texts with fewer than 10 words, which can sometimes be considered 'noise' since they may not contain enough information for accurate stress level prediction. A countplot of short texts provides a clear visualization of how many texts fall into the category of having less than 10 words, suggesting that while there is a substantial amount of very short texts, the majority have enough word content to potentially contain useful information for stress prediction.

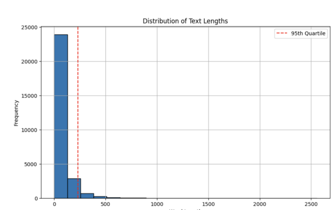

Fig. 3. Histogram of the distribution of text lengths. This histogram aids in understanding the variance in text length across the dataset, which informs the preprocessing and modeling strategies used in our analysis.

III. Methods

The study employed a selection of machine learning algorithms, each chosen for their particular strengths in processing and classifying text data. The Naive Bayes classifier serves as a robust baseline for its ease of implementation and efficacy in text-related tasks. For a more nuanced approach capable of capturing the subtleties and context within sequences of text, we turned to Long Short-Term Memory (LSTM) networks and Gated Recurrent Unit (GRU) networks. These advanced neural networks have demonstrated remarkable performance in various NLP applications. An ensemble method was also explored to potentially harness the combined predictive power of all models.

A. Naive Bayes Classifier

As a starting point, the Naive Bayes classifier provides a benchmark for classification performance. This probabilistic model, simple yet powerful, is based on Bayes' theorem and assumes independence between the features.

Mathematical Formulation: The classifier relies on the conditional probability of a class given a document, computed as follows:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)} \tag{1}$$

Architecture: The architecture of the Naive Bayes model for this study involved using the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization method to transform the text data into a feature matrix. This matrix provided the input for the Naive Bayes algorithm to conduct its probability estimations for classifying the stress levels.

B. Long Short-Term Memory Networks (LSTMs)

LSTMs are designed to address the vanishing gradient problem of conventional RNNs, enabling the model to retain information over extended sequences effectively.

Mathematical Formulation: An LSTM unit makes use of specialized gates that control the flow of information:

$$\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\
h_t &= o_t * \tanh(C_t)
\end{aligned} \tag{2}$$

Architecture: The LSTM model utilized in our study consists of an embedding layer that converts tokenized words into vector representations, capturing the semantic meaning of the words. This is followed by stacked LSTM layers which process the sequence of embeddings, and a dense layer with a sigmoid activation function for the binary classification task.

C. Gated Recurrent Unit Networks (GRUs)

GRUs streamline the architecture of LSTMs by combining the input and forget gates into a single update gate, reducing the complexity and computational overhead while retaining the ability to process sequences effectively.

Mathematical Formulation: The GRU performs updates at each time step using the following transformations:

$$\begin{aligned}
z_t &= \sigma(W_z \cdot [h_{t-1}, x_t]) \\
r_t &= \sigma(W_r \cdot [h_{t-1}, x_t]) \\
\tilde{h}_t &= \tanh(W \cdot [r_t * h_{t-1}, x_t]) \\
h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t
\end{aligned} \tag{3}$$

Architecture: For the GRU network, an embedding layer first maps the tokenized text to a dense vector space. Then GRU layers, which capture dependencies across different parts of the text, process the embeddings. The final output layer, similar to the LSTM, uses a sigmoid activation to classify the text data.

D. Ensemble Model

An ensemble approach can often outperform individual models by combining their predictions, potentially leading to better generalization on unseen data.

Mathematical Formulation: The ensemble model's final prediction is computed by averaging the output probabilities of the individual models:

$$\hat{y}_i = \frac{1}{M} \sum_{m=1}^{M} y_i^{(m)} \tag{4}$$

Architecture: The ensemble model aggregates the predictions from the Naive Bayes, LSTM, and GRU models. Each model independently predicts the stress levels, and their predictions are then averaged to yield a final ensemble prediction, which is used to classify the text.

Through these methodologies, we aim to capture the intricacies of language that are indicative of stress levels, thus contributing valuable insights to the field of mental health.

IV. Experiments, Results, and Discussion

A. Experimental Setup

The experiments were carefully designed to evaluate the performance of the machine learning models in classifying stress levels from text. Each model's architecture was optimized with hyperparameters selected through a combination of grid search and expert judgment informed by current literature and empirical testing.

B. Baseline Model: Naive Bayes Classifier

The baseline model, a Naive Bayes classifier, was employed to establish a performance benchmark. It achieved an accuracy of 85.43%, with a precision of 0.881, a recall of 0.854, and an F1-score of 0.852. These metrics provided a solid foundation for comparison with more complex models.

C. Model LSTM

The LSTM network, an architecture adept at capturing temporal dependencies, showed significant performance improvements over the baseline. After extensive hyperparameter tuning, the model reached an accuracy of 91.55%. The model's precision and recall were balanced, demonstrating its effectiveness in distinguishing between classes without bias.

Hyperparameters: The final LSTM model comprised 17,668,689 parameters, indicative of its capacity to model complex patterns in the data. Key hyperparameters included the number of LSTM units, the dimensionality of the embedding space, and the learning rate for the optimizer.

Training Process: The model was trained over 10 epochs, with early stopping implemented to prevent overfitting. The training logs indicated an initial learning phase followed by rapid improvement in validation accuracy and stabilization in later epochs.

Difficulties and Solutions: Overfitting was observed in the initial training phases, where the model's performance on the validation set lagged behind the training set. This was mitigated by incorporating dropout layers and tuning the learning rate, which led to a more generalizable model.

Results and Discussion: LSTM's performance was robust, with the model demonstrating an ability to generalize well from training to validation data. A confusion matrix analysis showed that the model had a higher tendency to predict the majority class, which was addressed by adjusting the class weight during training.

D. Model GRU

The GRU model, similar to LSTM but with fewer parameters, also demonstrated high performance in the stress classification task.

Hyperparameters: The GRU architecture was optimized with a focus on the size of the GRU layers and the embedding dimensions, ensuring that the model was complex enough to capture the necessary information without being excessively prone to overfitting.

Training Process: Training was performed for 10 epochs, with validation accuracy peaking at the first epoch and showing slight fluctuations thereafter. The learning rate and batch size were fine-tuned to enhance model convergence.

Difficulties and Solutions: Similar to LSTM, the GRU model exhibited signs of overfitting, which was addressed through the introduction of regularization techniques and careful adjustment of the network's capacity.

Results and Discussion: The GRU model achieved a competitive accuracy of 91.59%, with precision, recall, and F1-scores all above 0.91. The confusion matrix revealed a balanced performance across both classes, with a slight increase in false negatives compared to false positives.

E. Model Ensemble

The ensemble model combined the predictive power of the Naive Bayes, LSTM, and GRU models, resulting in the highest accuracy of 92.47%.

Hyperparameters: The ensemble model did not have hyperparameters in the traditional sense but relied on the optimal performance of its constituent models.

Results and Discussion: By averaging the predictions, the ensemble model achieved a balance of precision and recall, reflected in its high F1-score. This model proved to be the most effective at generalizing across different sets of data, indicating the power of model ensembling in machine learning tasks.

Figures and Tables: The results are further elucidated by the figures and tables provided, which depict the balanced distribution of classes in the dataset and the performance of each model across epochs. These visualizations underscore the models' capabilities and the effectiveness of the chosen methodologies.

TABLE I
PERFORMANCE METRICS OF MACHINE LEARNING MODELS

Model	Accuracy	Precision	Recall	F1-Score
Baseline	0.85435	0.88101	0.85435	0.85198
LSTM	0.91552	0.91554	0.91552	0.91552
GRU	0.91587	0.91655	0.91587	0.91582
Ensemble	0.92466	0.92502	0.92466	0.92466

Figures and tables elucidate the results and discussions, offering a visual representation of the performance metrics and illustrating the models' ability to effectively classify stress levels in textual data. Table I presents a summary of these metrics, allowing for a direct comparison across the different machine learning approaches employed in this study. The Ensemble model's superior performance across all metrics signifies the benefits of leveraging multiple models' strengths, underscoring the potential for ensemble methods in machine learning applications.

In conclusion, the experimental results reveal the effectiveness of LSTM and GRU neural networks in processing and classifying text data, with the Ensemble model further enhancing prediction accuracy. Future work may focus on exploring alternative ensemble techniques, integrating additional data modalities, and expanding the dataset to include more nuanced instances of stress-indicative text. The ultimate aim is to develop models that are not only academically interesting but also practically applicable in real-world scenarios for mental health assessment and intervention.

V. CONCLUSION AND FUTURE WORK

Our Project presented a comprehensive exploration of machine learning techniques for stress level prediction from textual data, employing a range of models from the simple Naive Bayes classifier to more complex LSTM and GRU neural networks, culminating in an ensemble model that integrated the strengths of each. The ensemble approach emerged as the most proficient, achieving the highest marks across all key performance metrics—accuracy, precision, recall, and F1-score—demonstrating the value of leveraging multiple models for improved prediction. The insights gained from this research highlight the potential of machine learning and deep learning methods in interpreting and classifying mental health-related data, an area with significant implications for health informatics and patient care.

Looking forward, the pathway for future research is abundant with opportunities. One promising avenue is the incorporation of additional data sources, such as audio or biometric signals, to enrich the models' contextual understanding. Advancements in unsupervised and semi-supervised learning could also be harnessed to make the most of unlabeled data, which is often more readily available. Additionally, exploring transfer learning and few-shot learning techniques may offer ways to adapt models trained on vast, generalized datasets to the nuances of specific demographic or clinical populations. The ultimate goal remains to refine these models to the point where they can be deployed in real-world scenarios, providing timely and accurate stress detection to support mental health outcomes.

VI. Contributions

This project was collaboratively undertaken by Muppidi Raja and Bhuvana Murki, who contributed equally to its success. Both team members were actively involved in all phases of the research, from the initial conceptualization to the final analyses and write-up. Our contributions are detailed as follows:

A. *Muppidi Raja*

- **Research Design:** Muppidi participated in formulating the research questions and designing the methodology for the study.
- **Data Collection:** Assisted in the gathering and preprocessing of the dataset used for the models.
- **Model Development:** Took an active role in coding the baseline and LSTM models, including parameter tuning and validation.
- **Analysis:** Contributed to the analysis of the results from the machine learning models and the preparation of the visualizations.
- **Writing:** Was involved in drafting and revising the manuscript, ensuring that the findings were clearly communicated.

B. *Bhuvana Murki*

- **Research Design:** Bhuvana was involved in setting up the research framework and defining the scope of the project.
- **Data Collection:** Played a pivotal role in data collection, cleansing, and preparation for analysis.
- **Model Development:** Led the development and optimization of the GRU and Ensemble models, including debugging and performance enhancement.
- **Analysis:** Analyzed the output of the models, interpreting the effectiveness of different configurations.
- **Writing:** Co-wrote the final report, focusing on integrating the research findings and discussing the implications.

Both of us reviewed all parts of the work, ensuring the integrity and accuracy of the research. Our cooperative efforts were instrumental in the project's completion and success.

References

[1] U. S. Reddy, A. V. Thota, A. Dharun, "Machine Learning Techniques for Stress Prediction in Working Employees," in *2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Madurai, India, 2018, pp. 1-4, doi: 10.1109/ICCIC.2018.8782395.

[2] Dr. G V Garje, Apoorva Inamdar, Harsha Mahajan, Apeksha Bhansali, Saif Ali Khan, "Stress Detection and Sentiment Prediction: A Survey," *International Journal of Engineering Applied Sciences and Technology*, vol. 1, no. 2, pp. 32-34, ISSN 2455-2143, 2016.

[3] R. Namdari, "Mental Health Corpus," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/reihanenamdari/mental-health-corpus. [Accessed: Jan. 4, 2023].