Adolescent Stress Level Classification Using Ensemble Machine Learning and LSTM Models

Resheetha Jeslin A[1], Dr. Thephilah Cathrine R[2], Dr. Rajathi S[3], Dr. B. Jayabharthi[4]

[1]Research Scholar, Department of Pediatric Nursing
Meenakshi Academy of Higher Education & Research (MAHER), Chennai, Tamil Nadu, India, Email: resheethajeslin@gmail.com

[2]Associate Professor, Department of Psychiatric Nursing
Meenakshi Academy of Higher Education & Research (MAHER), Chennai, Tamil Nadu, India, Email: mhn@mcon.ac.in

[3]Vice-Principal, Department of Pediatric Nursing
Hindu Mission College of Nursing, Chennai, Tamil Nadu, India
Email: rajathisakthi80@gmail.com

[4]Vice-Principal, Department of Obstetrics & Gynecological Nursing
Meenakshi College of Nursing, Chennai, Tamil Nadu, India
Email: vp@mcon.ac.in

Abstract
As stress-related disorders grow increasingly common, we need effective therapies that can be widely utilized to detect and manage them early. This research shows how to use machine learning (ML) and natural language processing (NLP) to find and sort people's stress levels. We gathered textual data that showed emotional and mental states and used typical NLP methods to clean it up. This included tokenization, removing stop words, and TF-IDF vectorization. We used many machine learning models, such as Multinomial Naive Bayes (MNB), k-Nearest Neighbors (KNN), and Logistic Regression (LR), to sort the stress levels into low, medium, and high. We also trained a Long Short-Term Memory (LSTM) model to fmd patterns in the data that happen in a certain order and have certain significance. We used accuracy, precision, recall, and F1-score to see how well each model worked. The results show that it is possible to use machine learning and textual signals to locate tension without having to touch the person. This might be useful for keeping an eye on mental health in schools and in corporate wellness initiatives. The research shows that using NLP with both shallow and deep learning models is a good way to understand how people feel.

Keywords: Academic stress anxiety; school adolescents; Natural language processing; Logistic regression
How to cite this article: Jeslin AR, Cathrine RT, Rajathi S, Jayabharthi B. Adolescent stress level classification using ensemble machine learning and LSTM models. Int J Drug Deliv Technol. 2026;16(75s): 610-620. DOI: 10.25258/ijddt.16.75s.73

1. Introduction

Everyone endures stress in their lives. A number of activities in our daily lives might be unpleasant. People can get stressed in a variety of settings, including their homes, workplaces, and societies. Being stressed mans having a lot of mental and physical problems when a person's needs aren't being addressed in a balanced manner [1].

The Stress also showed that 80% of individuals deal with stress at work daily and need to learn how to deal with it. The study found that a large number of student deaths by suicide between the ages of 15 and 29 were caused by stress. In 2023, 15,000 cases were recorded, and the goal of the study was to identify stress early on. Numerous diseases, such as hypertension and sleep deprivation, have mostly been produced by these numbers and how stress affects people [2].

Whenever stress is not managed appropriately, it can lead to serious consequences, including suicide. This is critical for identifying and controlling stress before it becomes out of control. Numerous researchers are investigating the identification of stress in a number of disciplines.

This paper will leverage data from IoT devices to improve stress diagnosis in five different scenarios. Researchers have looked at and analyzed a number of deep learning and machine learning methods. Early detection could be able to measure stress.

The World Health Organization (WHO) says that stress is any change that puts mental, physical, or emotional strain on a person [3]. Stress is what happens to a person's body when they have to stop what they're doing or focus on something else [4]. Humans experience positive as well as negative feelings, which influence their ability to interact with others, perceive the environment, and feel overall. How a person deals with difficult conditions determines their comfort and health [5].

Vegetative-somatic indicators change in response to stress. In moments of danger, the nervous system's sympathetic nervous system is activated, the adrenal glands produce more of the stress hormone adrenaline, blood pressure rises, heart rate increases, levels of sugar in the blood rise, and digestion slows [6].

This research looks at the topics and features that make people throughout the globe stressed by analyzing posts on Reddit and looking for symptoms of stress. This might be a chance to change how people communicate about psychological health and help individuals get help for depression early on. We need systems to find stress and mental diseases early on.

Everyone tested machine learning algorithms in this work to find social media postings that may be stressful. The main goal of this research is to show how frequently used artificial intelligence models for text classification and stress detection may work together with different embedding methods to make big improvements. This work trains a model to distinguish between stressful and non-stressful Reddit articles using the Dreaddit dataset, which looks at instances of social media blog postings, including mental stress that people have self-reported. After that, the model is put to the test on the Reddit site. This article takes a look at several well-known ML models that classify Reddit posts as stressful using various NLP and embedding approaches. These models include BERT. TF -IDE and Word2Vec. The third stage is to utilize the labeled corpus to teach a model how to tell the difference between stressful texts and those that aren't. After that, the machine learning model will be able to determine with excellent precision if a given post means mental stress. It looks at social media blog posts that say they are mentally stressed.

1.1 Background and Motivation

Since stress has a wide range of effects on people's mental, emotional, and physical health, it is becoming an increasingly significant public health issue. Prolonged stress may cause major health issues, including heart disease, anxiety, depression, and cognitive impairment. Conventional methods of assessing stress, such as self-reported stress surveys or clinical exams, may be tedious, arbitrary, and difficult to compare. With the growth of digital communication, MI. and NLP, it is now possible to swiftly and objectively identify stress patterns in user-generated text data, including journal entries, chat transcripts, and social media postings. Real-time, non-intrusive monitoring of mental health is made feasible by machine learning algorithms, particularly those that can analyze language and context. The urgent need for an advanced, automated method that can accurately determine someone's level of stress based just on their writing was the reason for this investigation. This study's goal is to close the gap between advanced deep learning algorithms and traditional machine learning methods in artificial intelligence and psychological assessment. Improving early stress management and mental health support networks is the ultimate goal.

1.2 Objective

To develop an NIP-driven system that preprocesses and analyzes textual data to extract linguistic features indicative of human stress levels. MNB and ICNN are two ensemble learning algorithms that may be used to test and evaluate their performance, along with Logistic Regression and LSTM, for accurate classification of stress into low, medium, or high categories. To assess how well deep learning and conventional models work together for scalability, non-invasive stress detection applicable to real-world mental health monitoring systems.

2. Related works

An anxiety that impairs reaction time or judgment under dangerous circumstances is one factor that leads to car accidents [7]. Hence, some research has tried to lower the number of road accidents by finding stress in children at a young age [8]. Physiological, facial, and automobile motion evaluations are among the indicators that have been used to identify drivers experiencing stress. Other factors that are considered include the vehicle's motion measurements, lane position, steering angle, managing movement patterns, and acceleration and braking speeds [9].

The results of such measurements could be affected by factors such as roadway circumstances, operating styles, and vehicle types. On a similar note, it is possible to monitor facial expressions like looking intently ahead. dilated pupils, yawning, blinking rate, and head movement, all while the driver is in the driver's seat. The reliability of these measures may be compromised in situations when there is insufficient light, adverse weather, nightfall, or when the driver is using eyeglasses. Physiological tests unrelated to stress are unaffected by driver conduct or illumination circumstances. An additional perk is that drivers may be able to detect stress levels thanks to physiological signals collected by body wear, which might provide valuable information about their internal health [10]. Because of the correlation between the activity of the autonomic nervous system and stress reaction, HR and GSR are often regarded as reliable stress markers. Consequently, stress detection challenges may be addressed using a variety of affordable and readily accessible physiological indicators [11]. In stress detection research, physiological signals have usually been retrieved as features in the frequency or temporal domains. Generally, data collection segments are truncated into time domain features using window sliding techniques [12], whereas low- and/or high-frequency areas are employed to get information from the frequency domain. We employed statistical variables, including skewness, kurtosis, standard deviation, and mean, to figure out

the difference between stressed and non-stressed situations based on these attributes. On top of that, other studies made use of domain-dependent traits that were defined by experts in the field of signaling types or human psychological stress [13].

2.1 Social Media Stress Analysis

This part gives a full summary of the research on utilizing language processing tools to look at anxiety on social media. Over the last decade, several studies have investigated various ways to utilize social media to identify mental health problems. Researchers have looked at mental diseases in the past by looking at how people in different fields use certain words.

According to research from Tsinghua University in China [14], several behavioral features were taken into account while compiling a dataset from an existing collection of tweets. Finding a method to recognize hybrid threats was the aim. Over the last several years, social media has become a more popular platform for self-expression. Through the use of social media data collection, Deep Learning for harmful speech detection, and even utilizing neural networks to identify users' self-reported symptoms of unhappiness, researchers are making strides in the field of NLP [15].

2.2 Methods for Word Embedding

Embeddings from Language Models (ELMo) were used to convert tokens into features. Bidirectional language model embedding representations are called ELMo [16]. A set of features suitable for training purposes was generated using the BERT Large tokenizer. Making contextual word embeddings that assist in determining whether the words in the sentence represent stress is particularly beneficial [17]. Toxic speech on social media networks has also been found using BERT and fastText embeddings [18]. After extracting textual characteristics and compiling a vocabulary of known terms, the bag-of-words method matches the important point extracted from the intended data source to a graphical word [19].

2.3 Machine Learning Models:

Using a linear support vector machine (SVM) model, we can categorize social media postings as either indicating tension or a lack thereof. The SVM's input was a feature matrix generated by the embeddings and annotated labels indicating the effects of mental stress. This result lends credence to previous studies that found SVM to be successful with fewer datasets for classification tasks [20]. A number of applications have made use of Extreme Gradient Boosting (XGBoost) algorithms. These include picture classification [21] and the detection of diseases such as renal illness [22] and heart disease [23]. One popular machine learning method, Logistic Regression (LR), shows how well a given data point fits a target distribution by using the sigmoid value function for two potential binary categorization labels. Its widespread use in classification algorithms and good accuracy led to its selection [24].

A selection of pertinent research that has previously looked at the stress detection method is included in Table 1.

Ref no	Dataset	Methods	Results
25	Own dataset-based ECG and EMG	Gradient Boosting Decision tree	GBDT accuracy is 93.4% Tree
26	SAID Dataset	Artificial Neural Network	ANN accuracy is 75.82%
27	WESAD Dataset	Random Forest and Adaboost	RF accuracy is 95%, and Adaboost accuracy is 84%
28	RAVDESS Dataset	Convolution Neural Network	CNN accuracy is 94.2%

3. Methodology

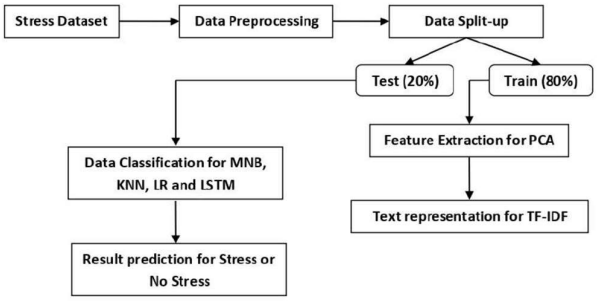

Figure 1: System Architecture

3.1 Dataset

The stress dataset is designed to identify indicators of psychological stress by analyzing text data from online forums. A distinct text item submitted by a user is represented by each row in the dataset. In order to comprehend the context or topic of the discussion. the subreddit column specifies the particular online community (such as a Reddit forum or group) where the post first appeared. Each post has a unique identification in the post_id column, which makes it simple for researchers to monitor or refer to specific postings. When a single post is divided into many sentences for in-depth analysis, the sentence_range parameter is particularly helpful since it indicates the location or index of the sentence inside the post. The user's actual statement or message, which is the main input for stress detection models, is stored in the text column. The

label is a binary indication that shucks if the text's content is deemed stressful, with 0 denoting "no stress" and 1 denoting "stress." The model's or annotator's level of confidence in the assigned label's accuracy is indicated by a numerical value in the confidence column, which is often between 0 and 1 or expressed as a percentage. Last but not least, the social timestamp provides the time and date of the post's creation, which is crucial for temporal analysis tasks like seeing patterns in stress over time or at certain occasions. When combined, these columns provide a thorough framework for identifying. classifying, and evaluating writing on social media that contains stress-related information. The dataset available is: https://www.kaggle. com/codesubhran ilmon dal 1 2/stress-prediction-complete-analysis-with-ml/notebook

3.2 Data Preprocessing

In stress detection, data preparation is essential for converting unstructured textual data into a machine-learning-friendly format. Cleaning the data which includes eliminating duplicates, null values, and unnecessary variables, is the first stage since stress detection sometimes entails examining user-generated content from forums or social media. The actual sentences in the text column are subjected to text preprocessing, which includes lowering all text to lowercase and eliminating punctuation, special characters, L-RLs, and numbers that don't add to the emotional context. After that, stopwords common words like "is." "the," and "and" are eliminated to cut down on noise, and tokenization is used to divide the text into distinct terms or phrases. Words are subsequently standardized text for better pattern recognition by being stripped down to their base or fundamental fonn via lemmatization or stemming. At the same time, there's the label column that shows how stressful the content is, examined to make sure it is formatted correctly in binary. To enhance model performance, low-confidence inputs might be eliminated or weighted lower according to the confidence score. Converting the social_timestamp data to a datetime format and extracting helpful time-based attributes are examples of optional preprocessing. To input the cleaned and processed text into machine learning models, it is finally converted into numerical features using methods such as TF-IDF. In general, this preprocessing pipeline improves stress detection systems' precision and dependability.

3.3 Data Split-up

To properly assess model performance in stress detection tasks, as a general rule, sets for both training and testing are created from the dataset. One common approach is the 80:20 split, whereby 80% of the data is used to train the model used for machine learning and 20% is reserved for testing. This guarantees that the model gets assessed on unseen instances while learning from a significant amount of the data. The test set assesses the generalizability of the model, as the training data aids in the identification of stress patterns in text. This balanced split provides an honest evaluation of the model's function and helps avoid owrfitting.

3.4 Data Extraction

For extracting data for a stress detection model, Principal Component Analysis (PCA) is a common dimensionality reduction technique to impiove the model's interpretability and performance. The feature space that results firm preprocessing and vectorizing the textual data (TF-IDF) may be sparse and high-dimensional. By reducing the original features to a smaller number of main components that capture the greatest variation in the data, PCA aids in the extraction of the most important patterns. PCA speeds up training and improves the generalization of stress detection models by reducing noise and redundancy by keeping just the best components. This step reduces computational cost while improving performance.

3.4.1 TF-IDF

TF-IDF (Term Frequency—Inverse Document Frequency) is a method that is often used in Natural Language Processing (NLP) to transform textual information into useful numerical formats. When working with a CSV file dataset for stress detection, where each row contains a text entry (such as a social media post), TF-IDF helps quantify the weight of each word in the context of the whole dataset.

The TF measures how many times a word appears in a piece of writing, while the IDF disregards the relevance of terms that are used a lot in favor of phrases that are more specific and helpful. We get TF-IDF by multiplying TF and IDF for each word within the text. This provides us with a score that shows how important each word is. The machine learning models. like LR, MNB, or KIN to translate raw text into vectors with features. When it comes to finding stress, TF-IDF helps the model focus on words that are more relevant to stress-related patterns, improving classification accuracy. TF-IDF is a key approach in text analysis that converts textual input into numerical characteristics. TF-IDF identifies which curds in a statement or post are most significant for identifying emotional or mental stress.

TF — IDF(t,d)= TF(t,d) IDF(t)---------- (1)*

Where equation 1 is defined as

1. TF (Term Frequency)

$$TF(t,d) = \frac{f_{t,d}}{\sum_k f_{k,d}} \text{---------- (2)}$$

Where equation 2 is defined as fry the frequency of a term t in a documentd, Ek fk,d it is the total number of terms in document d.

2. IDF (Inverse Document Frequency)

$$IDF(t) = \log\left(\frac{N}{1+DF(t)}\right)\text{----------(3)}$$

Where equation 3 is defined as N the total number of documents, DF(t) and the number of documents that contain the term t.

3.4.2 Principal Component Analysis PCA:

In machine learning. PCA is a dimensionality reduction technique used to pinpoint the salient features of high-dinrnsional data. The feature space may become sparse and extremely big in stress detection, particularly when working with text data that has been transformed into enormous numerical vectors using techniques like word embeddings or TF-IDF. By minimizing the number of features and finding and maintaining the directions (referred to as principal components) in the data that capture the greatest variation, or the most significant patterns.

PCA reduces noise and redundancy by projecting the original high-dimensional data onto a smaller collection of components. which improves model performance and makes the dataset easier to handle. In stress detection, INthere hundreds of textual elements may be present, this is very helpful. The basic information required to categorize stress levels (low, medium, or high) is still captured by the decreased feature set after PCA, but there is less chance of overfitting and less computing complexity.

z = x. w---------- (4)

Where equation 4 x is the original data matrix, w is the matrix of eigenvectors, z and is the transformed data. PCA is especially useful in stress detection. there large text-based feature sets are common, since it simplifies complex datasets, enhances processing, performance, and usually improves the accuracy of classification by removing duplicate or redundant features.

3.5 Data Classification

In this work, stress detection is carried out using a variety of approaches to machine learning, including LR, K\IN, and MNB to classify stress levels into low, medium, and high categories. These models help in identifying patterns in the textual data and classifying the stress accordingly. Additionally, a technique in deep learning that makes use of Long Short-Term Memory (LSTM), which is effective in the textual representation of sequential relationships. An ensemble learning technique is used to combine machine learning and neural networks for more robust and dependable stress categorization to increase overall performance and accuracy.

3.5.1 Multinomial Naive Bayes (MNB)

A well-liked supervised machine learning technique for text categorization is Multinomial Naive Bayes (MNB), pseudocode shown in Table 2, particularly when features indicate discrete word counts or frequencies. In stress detection, MNB uses word occurrence patterns to categorize user-generated content (such as posts or comments) into predetermined groups like low, medium, or high stress. With the naive assumption that every word in the category label, the statement is independent, the Bayes Theorem is used. It assigns the class that has the highest likelihood of receiving the supplied text after calculating the likelihood of each class using the training data.

Table 2: Pseudocode for MNB

Input:
A dataset containing text samples and corresponding stress labels (Low, Medium, High)
Output:
Predicted stress label for new/unseen text samples
Preprocessing
Clean the text data (remove punctuation, lowercase, etc.)
Tokenize the text into words
Remove stopwords and apply lemmatization or stemming
Build a vocabulary of all unique words from the training data
Training Phase
a. Calculate Prior Probability for Each Class
For each stress class (Low, Medium, High):
Count the number of training samples belonging to the class
Divide by the total number of training samples to get the prior probability
b. Calculate Likelihood of Each Word Given a Class
For each word in the vocabulary and each class:
Count how many times the word appears in the documents of that class
Apply Laplace smoothing:
Likelihood = (Word count in class + 1) / (Total word count in class + Vocabulary size)
Prediction Phase
a. For each new test text sample:
Preprocess the text as done during training
For each class (Low, Medium, High):
Initialize the total log probability with the log of the prior probability
For each word in the sample:
if the word exists in the vocabulary:
Add the log of the likelihood of the word given the class to the total log probability
The class with the highest total log probability is the predicted label
Output
Return the predicted class(stress level) with the highest score

3.5.2 K-Nearest Neighbors (KNN)

A straightforward and efficient supervised learning approach for text data classification based on similarity is K-Nearest Neighbors (KNN), pseudocode shown in Table 3. The algorithm is used in stress detection to categorize words or postings on social media into low, medium, and high stress

levels. Using techniques like TF-IDF or word embeddings, the text input is first preprocessed and transformed into numerical vectors. KNN uses a distance metric, often Euclidean or cosine distance, to determine how similar a new, unlabeled text is to all training samples. It then identifies the "K" most similar values, or neighbors, from the training set. The stress level that is most prevalent among its K closest neighbors is given to the new text. Because distances must be calculated for each new input, this approach may be sluggish with huge datasets, but it performs well when the data is clean and well distributed.

Table 3: Pseudocode for KNN
Input: Training dataset with text samples and labels (Low, Medium, High) Test sample (new text) K (number of nearest neighbors) Step 1: Preprocess the training and test data Clean the text (remove punctuation, lowercasing) Tokenize and remove stopwords Convert text to numerical feature vectors (e.g., TF-IDF) Step 2: For the test sample Calculate the distance between the test sample and all training samples using a distance metric (e.g., Euclidean or cosine distance) Step 3: Sort the distances in ascending order Step 4: Select the K nearest neighbors Step 5: Count the number of neighbors belonging to each stress class Step 6: Assign the class (Low, Medium, or High) with the highest count among the K neighbors of the test sample Output: Predicted stress level for the test sample

Preprocessing both training and test text samples, which include cleaning, tokenization, and text conversion into numerical vectors, is the first step in the KNN method. The technique uses a similarity metric to calculate the distance between a fresh text sample and every example in the training set. The K closest (most similar) neighbors are then found. The labels of these neighbors' stress levels, low, medium, and high, are examined. The forecast for the new sample is the label that shows up most often among the K neighbors. Since this method is non-parametric, it is highly dependent on the quality and structure of the training data and does not assume anything about the distribution of the underlying data.

3.5.3 Logistic Regression (LR)

A well-liked supervised machine learning method for problems involving both binary and multi-class classification is logistic regression (LR). Pseudocode is shown in Table 4. LR is used in stress detection to categorize text data, such as messages or postings on social media, into low, medium, and high stress levels. Using methods like TF-IDF or word embeddings, the text is first preprocessed and transformed into numerical vectors. Then. using a logistic (sigmoid or softmax) function over LR forecasts using a linear amalgamation of the input characteristics, the likelihood that a given text falls into a certain stress class is calculated. Each class's likelihood is determined and the group of learners with the highest likelihood receives the final prediction. Following being trained on a labeled dataset, the model is improved using gradient descent to reduce the error between predicted and real stress labels.

Table 4: Pseudocode for LR
Input: Training dataset with text samples and corresponding stress labels (Low, Medium, High) Test sample (unlabeled text) Learning rate a Number of iterations (epochs) Step 1: Preprocess the text data Clean text: remove punctuation, lowercase, remove stopwords Tokenize and apply stemming/lemmatization Convert text into numerical vectors using TF-IDF or word embeddings Step 2: Initialize weights and biases to zero Step 3: For each iteration: For each training sample: Compute linear combination: z= (weight • input_vector) + bias Apply the sigmoid or softmax function to compute the probability Compute the error between the predicted and the actual label Update weights and bias using gradient descent: weight = weight -ox (Loss/ôweight) bias = bias - a x (Loss/bias) Step 4: For a test sample: Preprocess and convert to a vector Compute probability using trained weights and bias Assign a stress label with the highest probability Output :Predicted stress level (Low,Medium,Or High)

To determine if a particular phrase communicates low, medium, or high tension, logistic regression is used in stress detection. Before being vectorized into a numerical representation, the text data undergoes preprocessing. After calculating a weighted sum of input characteristics, logistic regression generates probabilities for each class by using a softmax function for classifying more than one class or a sigmoid function for classifying just two classes. The model changes the weights and bias during training using gradient descent to make predictions more accurate. The model picks the stress class with the best chance of being correct. For linearly separable data. LR is straightforward, interpretable,

and effective, which makes it appropriate for stress classification applications.

3.5.4 Ensemble Learning

Ensemble learning is a machine learning technique that combines the predictions of several models to get a more accurate and dependable result than any particular classifier could provide on its own. By combining more traditional machine learning (ML) algorithms like Multinomial Naive Bayes (MNB), Logistic Regression (LR), and k-Nearest Neighbors (ICNN) with deep learning (DL) models like Long Short-Term Memory (LSTM), ensemble learning may be used to identify stress. The strengths of each of these models vary; MNB performs well with word frequencies, LR effectively manages linear relationships, KNN captures pattern similarity, and LSTM is particularly good at comprehending the order and context of words in text. Ensemble learning uses a variety of these models to better classify stress levels (low, medium, and high) by merging them using methods like voting stacking or mixing. This hybrid method increases the overall efficacy of the system in detecting stress from text input, decreases the shortcomings of individual models. and boosts prediction dependability.

3.5.4.1 Long Short-Term Memory

The LSTM is a kind of recurrent neural network (RNN) that particularly excels at recalling dependencies that persist in sequential input, such as text, as Table 5 shows. LSTM is used in stress detection categorization to look at the order of words in a phrase or post to figure out how the user is feeling. LSTM is different from other models since it takes into account the sequence and context of words. This is important for recognizing subtle ways that people show stress. The LSTM network gets the input text after it has been preprocessed and turned into word embeddings. The model goes over each phrase one at a time and keeps an internal recall of crucial stress-related patterns. The LSTM gives a forecast at the conclusion of the sequence that shows the stress level, such as low, medium, or high. LSTM works well for stress classification from natural language data because it can grasp sequences and pick up on emotional tone and nuanced language signals.

Table 5: Pseudocode for LSTM

```
Input:
Pre-processed text data with stress labels (Low, Medium, High)
Number of LSTM units
Word embedding dimension
Epochs, batch size
Step 1: Preprocess the text data
Clean and tokenize text
Convert words into sequences of integers
Pad sequences to the same length
Map words to embeddings (e.g., Word2Vec or Embedding Layer)
Step 2: Define the LSTM model
Initialize the input layer
Add an embedding layer
Add one or more LSTM layers with desired units
Add a dropout layer (to prevent overfitting)
Add a dense (fully connected) output layer with softmax (for multi-class)
Step 3: Compile the model
Choose a loss function (e.g., categorical cross-entropy)
Choose optimizer (e.g., Adam)
Define evaluation metrics (e.g., accuracy)
Step 4: Train the model
Fit the model to the training data for several epochs and batch sizes
Validate using test data
Step 5: Predict
Input new text
Preprocess and convert it to a padded sequence
Feed into the trained model
Output predicted stress label (Low, Medium, High)
Output: Classified stress level for the input text
```

LSTM is used in stress detection because of its ability to comprehend word context and sequence, which is essential for identifying the emotional tone of phrases. Text preprocessing, which includes cleaning, tokenizing, and turning words into numerical sequences, comes first in the process. Padding is then used to guarantee that inputs are of the same length. Following that. these sequences are transformed into word embeddings, which are dense vector representations of word meaning This sequential input is processed by the LSTM layer, which retains crucial information over time. To prevent overfitting, a dropout layer might be used. The input text is categorized into stress levels by the final output layer using softmax activation. The program can estimate stress levels from unseen text with strong contextual awareness after being trained with labeled data.

4. Results and Discussion

The stress detection system was built using an Anaconda Jupyter Notebook running Windows 10/11 64-bit, an Intel i3 CPU. and 8 GB of RAM. The project made use of popular Python data processing libraries, including machine learning, with the development of deep learning models, such as NumPy, Pandas, Scikit-learn, Keras, and TensorFlow. The dataset was imported and analyzed using Pandas, with numerical calculations performed using NumPy. Scikit-learn was used to create a number of classification algorithms, including Multinomial Naive Bayes, K-Nearest Neighbors (KNN), and Logistic Regression. Keras and TensorFlow were used to create the LSTM deep learning model. Despite a basic hardware setup, the

system performed effectively throughout model testing and training. The results show that a reliable and accurate stress detection system may be built using readily available accessible tools and little hardware resources, as LR and LSTM models performed well in recognizing stress level.

	Subreddit	Post id	Sentence Range	Text	Label	Confidence	Social Timestamp
0	PTSD	8601 tu	(15, 20)	He said he had not felt that way before, sugge...	1	0.8	1521614353
1	assistance	8lbrx9	(0, 5)	Hey there r/assistance, Not sure if this is th...	0	1.0	1527009817
2	PTSD	9ch1zh	(15, 20)	My mom then hit me with the newspaper and it s...	1	0.8	1535935605
3	relationships	7rorpp	[5, 10]	until I met my new boyfriend, he is amazing, h...	1	0.6	1516429555
4	Survivors of abuse	9p2gbc	[0, 5]	October is Domestic Violence Awar	1	0.8	1539809005
				eness Month Month a...			
5	relationships	7tx7et	(30, 35)	I think he doesn't want to put in the effort f...	1	1.0	1517274027
6	Domestic violence	Tiphly	[25, 30]	It was a big company so luckily I didn't have	0	0.8	1512854409
7	anxiety	5m3K80	(5, 10)	It cleared up and I was okay but. On Monday	1	0.8	1483582174
8	relationships	7nhy1v	(50, 55)	I actually give an assistant half my emergency..	1	0.6	1514843984
9	assistance	61eiq6	[15, 20)	I just feel like the street life has fucked my... 1		1.0	1490428087

Figure 2: Sample dataset

The dataset, as seen in Figure 2, includes information that was posted on subreddits that are associated with mentalhealth. This collection of data includes a variety of mental health issues that individuals have discussed concerningtheir lives. This dataset, fortunately, is labeled with the numbers 0 and 1, where 0 means that there is no stress and 1indicates that there is stress.

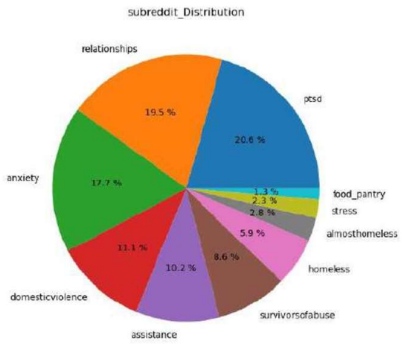

Figure 3: Subreddit Distribution

Figure 3 "subreddit_Distribution" depicts the distribution of postings from distinct subreddits in the stress detectiondataset. Each part of the figure represents a separate subreddit community, demonstrating how the data is proportionately split by subject or emotional context. PTSD accounts for the highest part (20.6%), showing a high number of entries on post-traumatic stress. This is closely followed by relationships (19.5%) and worry (17.7%),indicating that many people express stress via personal and emotional ties. Domestic violence (11.1%), help(10.2%), and survivors of abuse (8.6%) subreddits are also well represented, demonstrating that support-seeking and

Table 7: Performance Metrics: classes for 0 and 1

Algorithms	Level	F1-Score	Precision	Recall	Accuracy
KNN	0 (No Stress)	0.63	0.68	0.59	0.68
	1 (Stress)	0.72	0.68	0.76	
MNB	0 (No Stress)	0.59	0.83	0.46	0.71
	1 (Stress)	0.77	0.66	0.92	
LR	O (No Stress)	0.71	0.71	0.71	0.73
	1 (Stress)	0.75	0.75	0.76	
LSTM	0 (No Stress)	0.98	0,98	0.98	0.99
	1 (Stress)	0.99	0.98	0.97	

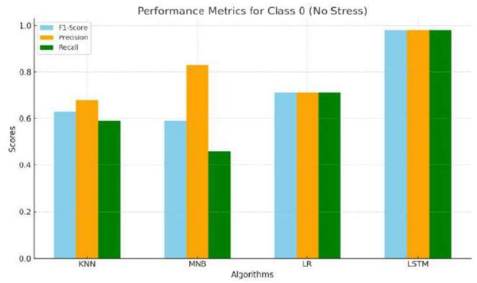

Figure 5: Performance Metrics for 0 (No Stress)

Table 7 and Figure 5 illustrate that KNN shows moderate precision (0.68) but lower recall (0.59), indicating it identifies some stress-free individuals correctly but misses others. MNB has wry high precision (0.83) but low recall (0.46), which implies it is yew confident when it predicts "no stress," but often fails to identify all actual "no stress" cases. LR maintains a balanced performance across all three metrics (-0.71), reflecting its stable classification capability. LSTM performs exceptionally well with all values at 0.98, showing it is highly effective in recognizing non-stress situations.

Table 5 and Figure 6 illustrate that KNN demonstrates better performance here than in Class 0, with Fl -Score 0.72 and Recall 0.76, indicating it's more sensitive to detecting stress. MNB in recall (0.92), showing its strong ability to identify stressed individuals, though precision drops (0.66), meaning more false positives. LR continues its balanced performance, with an Fl-score of 0.75. reflecting decent predictive ability. LSTM again dominates across all metrics (F 1-Score = 0.99), proving its robustness and deep contextual understanding for stress detection.

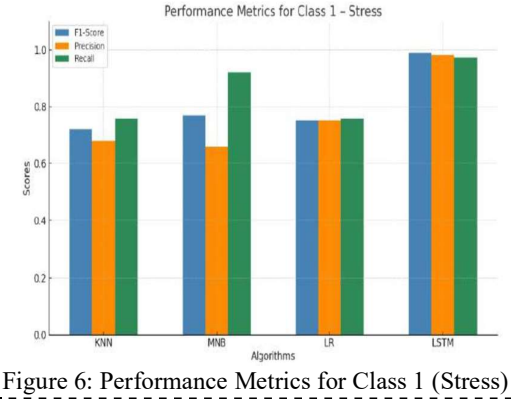

Figure 6: Performance Metrics for Class 1 (Stress)

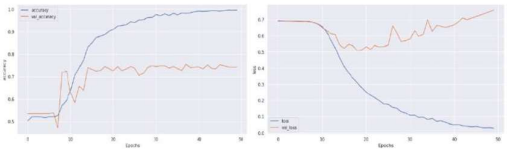

Figure 7: LSTM Accuracy and Loss

The training performance of an LSTM (Long Short-Term Memory) model employed for stress detection across 50 epochs is shown in Figure 7. The accuracy graph on the left shows a consistent increase in training accuracy over time, finally approaching 100%. This shows that the machine learning algorithm is picking up new information and fits the training data rather well. On the other hand, the precision of the validation (orange line) rises at first before varying and leveling out at 0.73. The model may be overfitting if it performs well on training data but has trouble generalizing to new data, as shown by this discrepancy between training and validation accuracy.

We can see the loss curves in the graph on the right. The model is reducing the error on the initial training set, as seen by the loss of training information (blue line) steadily declining. But after around 15 epochs, the loss of validation (orange line) starts to increase after first declining, supporting the overfitting findings. The algorithm learns from noise or certain patterns in the initial training data that cannot be generalized well if the validation loss is increasing mhile the training loss is consistent or declining.

In summary, while the LSTM model achieves high training accuracy, its validation metrics indicate overfitting. To improve generalization, techniques such as regularization, dropout, or early stopping could be beneficial.

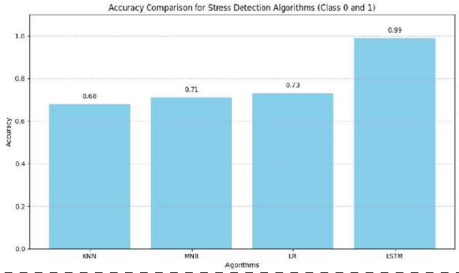

Figure 8: Accuracy Comparison

Figure 8 presents a comparison of accuracy scores for different algorithms used in stress detection, specifically for classifying two levels: Class 0 (No Stress) and Class 1 (Stress). Four algorithms are evaluated: K-Nearest Neighbors (KIN), Multinomial Naive Bayes (MNB), Logistic Regression (LR), and Long Short-Tenn Memory (LSTM). Compared to the other methods, LSTM's 0.99 accuracy shows that it has almost perfect classification abilities on the dataset. In contrast, KNN, MNB. and LR show moderate performance, with accuracies of 0.68, 0.71, and 0.73, respectively. This suggests that while traditional machine learning models perform reasonably well, they are not as effective as deep learning models like LSTM for this task. It follows that the LSTM model is superior at detecting the temporal or sequential patterns in the data that are relevant for detecting stress levels. To ensure that the projected result is not overfitting and generalizes appropriately, validation tests should be conducted concurrently.

5. Conclusion

In determining stress levels from textual data, the stress detection study using ensemble learning models shows encouraging outcomes. With the greatest F 1 -scores (0.98 for No Stress and 0.99 for Stress), precision, recall, and overall accuracy (0.99), the LSTM model substantially beat the other models among the KNN, MNB, LR, and LSTM models that were assessed. This demonstrates how well LSTM captures intricate language patterns and context in text data about stress. Traditional models, such as KNN and MNB, on the other hand, performed consistently but at a lower level, making them appropriate for baseline comparisons. The independent assessment for the stress and no-stress classes further demonstrated that LSTM is a dependable model for stress classification tasks, as it consistently performs well in differentiating between the two conditions. All things considered, deep learning methods, especially LSTM, offer a potent method for identifying stress, which may be useful in early intervention instruments and mental health monitoring systems.

To further increase accuracy and contextual awareness, the model may be improved for future work by using sophisticated deep learning architectures like Transformers or Bidirectional LSTM. Furthermore, using transfer learning methods and adding real-time social media or conversational data might strengthen and adjust the system.

References

1) Palmer S 1989. The Health and Safety Practitioner, S. 16-18. IOP Publishing. doi:10.1088/1742- 6596/1997/1/012019
2) Ahuja R and Banga A 2019 International Conference on Pervasive Computing Advances and Applications PerCAA 2019 125 349-353
3) Marmot, M.; Friel, S.; Bell, R.; Houweling, T.A.; Taylor, S. Closing the gap in a generation: Health equity through action on the social determinants of health. Lancet 2008, 372, 1661-1669.
4) Yaribeygi, H.; Panahi, Y.; Sahraei, H.; Johnston, T.P.; Sahebkar, A. The impact of stress on body fimction: A review. EXCLI J. 2017, 16, 1057.
5) Hammen, C. Stress and depression. Annu. Rev.. Clio. Psychol. 2005, 1, 293-319.
6) Tsigos, C.; Kyrou, L: Kassi, E.: Chrousos, G.P. Stress: Endocrine physiology and pathophysiology. In Endotext [Internet];

MDText.com, Inc.: South Dartmouth, MA, USA, 2020.

7) Flaskerud, J.H. Stress in the Age of COVID-19. Issues Ment. Health Nurs. 2020, 42, 99-102.
8) Rastgoo, M.N.; Nakisa, B.; Rakotonirainy, A.; Chandran, V.; Tjondronegoro, D. A critical review of proactive detection of driver stress levels based on multimodal measurements. ACM Comput. Surv. (CSUR) 2018, 51, 1-35.
9) Lim, S.; Yang J.H. Driver state estimation by convolutional neural network using miltimodal sensor data. Electron. Len. 2016, 52, 1495-1497.
10) Elgendi, M.; Menon, C. Machine learning ranks ECG as an optimal wearable biosignal for assessing driving stress. IEEE Access 2020, 8, 34362-34374.
11) Mishra, V.; Pope, G.; Lord, S.; Lewia, S.; Lowens, B.; Caine, K.; Sen, S.; Halter, R.; Kotz, D. Continuous detection of physiological stress with commodity hardware. ACM Trans. Comput. Healthc. 2020, 1, 1-30.
12) S' alkevicius, J.; Damakvi'dius, R.; Maskeliunas, R.; Laukieno, L Anxiety level recognition for virtual reality therapy system using physiological signals. Electronics 2019, 8, 1039.
13) Dash, D.P.; Kolekar, M.H.; Chakraborty, C.; Khosravi, M.R. Review of Machine and Deep Learning Techniques in Epileptic Seizure Detection using Physiological Signals and Sentiment Analysis. Trans. Asian -Low-Resour. Lang. Inf. Process. 2022.
14) Kim J, Lee J, Pads E, et al. A deep learning model for detecting mental illness from user content on social media. Sci Rep. 2020;10:11846. Ittps://doiorg110.10381s41598-020-68764-y.
15) Pirina, LL., coltekin, g. (2018). Identifying Depression on Reddit: The Effect of Training Data EMNLP
16) Anthony J. Myles; Robert N. Feudale; Yang Liu; Nathaniel A. Woody, Steven D. Brown (2004). An introduction to decision tree modeling.18 (6), 275-285. https://doi.org/10.1002/cem.873
17) Yong Z, Youwen L, Shixiong X An improved KNN Text classification algorithm based on clustering. Journal of Computers. 2009. https://doi.org/10.4304/j cp.4.3 230-237.
18) Meersman, Robert; Tari, Zahir; Schmidt, Douglas C. (2003). [Lecture Notes in Computer Science] On the Move to Meaningful Internet Systems 2003: CoopLS, DOA, and ODBASE Volume 2888 II KNN Model-Based Approach in Classification. https:// doi. org/10.1007/978-3-540-39964-3_62
19) Indra T, Wikarsa L, Turang R. Using logistic regression method to classify tweets into the selected topics. Intern Confer Adv Com Sci Inform Syst (ICACSLS). 2016;385-90. https:// doi. org/ 10.1109/ICACSIS.2016.7872727.
20) Toucan, E., Muresan, S., & McKeown, K (2021, June). Emotion Infused Models for Explainable Psychological Stress Detection. Proceedings of the 2021 conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies https:// doi. Org/ 10. 18653/ v1/ 2021. naacl- main. 230
21) Pay, T., Lucci, S., Cox, J.L. (2019). An Ensemble of Automatic Keyword Extractors: TextRank, RAKE, and TAKE. Computacion y Sistemas 23.
22) Rose, S., Engel, D., Cramer, N., Cowley, W. (03, 2010). Automatic Keyword Extraction from Individual Documents. https://doi.org/10.1002/9780470689646.chl