Journal of High School Science

Data requirements for suicidal and depressive thought detection from Reddit™ with Recurrent Neural Networks

Chakka L

Submitted: September 24, 2024, Revised: version 1, February 10, 2025, version 2, February 18, 2025, version 3, April 9, 2025, version 4, July 30, 2025, version 5, August 1, 2025
Accepted: August 23, 2025

Abstract

With suicide and depression rates becoming a growing concern worldwide, timely identification of individuals at risk is important for providing support and intervention. Identifying these patterns plays a critical role in detection, however distinguishing between the two mental illnesses remains a challenging task because afflicted individuals' texts and/or messages at online discussion forums express similar verbiage of hopelessness and emotional struggle. Machine learning models offer a solution to this conundrum by utilizing labeled data to analyze language patterns and categorize text while accounting for these subtle variations between the two mental states. However, the process of collecting and managing large datasets for training these models can be resource-intensive and costly. This study's objective was to determine if there was an 'ideal' amount of data required to train models effectively, balancing optimal performance with resource efficiency. By evaluating the effect of different data sample sizes, this research provides a practical insight for developing scalable and cost-effective models. Recurrent Neural Network (RNN)-based architectures, including the Simple RNN model, Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRUs) were employed to detect suicidal thoughts in Reddit™ posts and distinguish them from depressive content. Network complexity was also examined by experimenting with different node configurations within the models. Model performance, across all models, remained stable and effective when trained on dataset sizes ranging from 36,000 (20% of the data) to 126,000 samples (70% of the data), with minimal improvements observed beyond this range, despite the full dataset containing 180,000 samples. Soberingly, none of the models could accurately identify 'normal' texts from a manually generated *post-hoc* sample which mimicked teen slang, demonstrating that – regardless of the number of posts the models trained on – they were ultimately not able to decipher the contextual and semantical vernacular that is representative of texts on social media posts from this vulnerable demographic.

Keywords

Suicide detection, Suicide risk, Depression, Reddit, Machine Learning, Dataset size, Network complexity, Recurrent Neural Network, Social media posts, Teen slang

Lakshmi Chakka, Northwood High School, 4515 Portola Pkwy, Irvine, CA, 92620, USA. cs4lakshmi@gmail.com

https://doi.org/10.64336/001c.143549

1. Introduction

Mental health has always been a fundamental aspect of overall well-being, influencing emotions, thoughts, and social interactions. It determines how individuals think, feel, and act, and because of its profound impact, it is critical to address mental health issues proactively (1). Among the various mental health issues, depression and suicide stand out as particularly pressing. Depression affects millions worldwide, often leading to persistent emotional pain and social withdrawal. In some cases, it can escalate into suicidal thoughts or behaviors. Suicide, now one of the leading causes of death among selected demographics, represents an urgent crisis that demands immediate attention. Preventing both depression and suicide requires early identification and timely support, which are essential to improving outcomes and saving lives (2).

Social media platforms have become a significant part of everyday life for many individuals. These platforms allow users to reflect on their emotions, interactions, and mental states. As a result, social media provides a source of data that can offer insights into an individual's mental health (3). Specifically, Reddit™ stands out as a valuable source because it allows users to post anonymously and participate in communities called subreddits™ focused on – among other content - mental health, where they can share their experiences and feelings much more openly compared to other platforms (4). Analyzing this data can help identify signs of suicidal/depressive thoughts, which might otherwise go unnoticed.

With the growth of machine learning, new opportunities have emerged to address mental health issues more effectively. This paper explores the use of Reddit™ data to train models that can differentiate between suicidal, depressive, and general posts.

Various model architectures, including Simple Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRUs), were studied to determine their effectiveness. The models were tested with varying data sample sizes in order to determine the amount of data necessary to achieve acceptable accuracy. By learning to recognize the distinctions between different types of Reddit™ posts, the models trained in this study aimed to more accurately detect signs of suicide and depression given a sufficient amount of data. This is particularly important because often, the amount of data is either too little to produce accurate results or too excessive, resulting in a waste of resources. By identifying the optimal range of data necessary for accurate results, this research focuses on balancing accuracy with efficiency. It provides a practical foundation for collecting enough data to train models capable of generalizing across diverse posts while also minimizing unnecessary resource use, which is especially important when working with sensitive data in resource-limited or ethically restricted areas.

2. Related work

2.1 Mental health detection using social media

Haque et al. proposed a method that combined Natural Language Processing (NLP) with Machine Learning to detect suicide from social

media posts. The authors highlighted the effectiveness of the deep learning model Bidirectional Long Short-Term Memory (BiLSTM) networks, which can collect information in both the forward and backward directions (5).

Ramirez-Cifuentez et al. explored detecting suicidal ideation among Spanish-speaking social media users by combining multimodal, relational, and behavioral data. Their research incorporated text analysis, user posting patterns, social interactions, and images. Deep learning models, particularly Convolutional Neural Networks (CNNs) were utilized, to process and analyze this data. The authors also emphasized using word embeddings to represent text as numerical vectors. By integrating diverse data types and employing various deep learning techniques, the study aimed to improve the accuracy of detecting users at risk of suicide (6).

Khoo et al. systematically reviewed the use of machine learning in detecting mental health disorders by employing multimodal data sources. Their review focused on combining different data types, such as video, audio, social media interactions, smartphone usage, and wearable device data, to improve the detection of mental health symptoms. These insights proved valuable for enhancing the accuracy and efficiency of mental health detection systems as well as choosing the most suitable data sources and methods (7).

2.2 Analysis of Machine Learning models utilizing varying data sample sizes

Rajput et al. analyzed the effect of different sample sizes on Machine Learning model performance and found that small sample sizes often resulted in overfitting and occasional high accuracy numbers because of random fluctuations. Larger samples, however, tended to provide more reproducible results because they reduced randomness and allowed for replicable patterns (8).

It is important to note that while state-of-the-art research has extensively focused on either automated mental health detection or the effects of data sample sizes in general Machine Learning applications, there remains an unexplored gap: the intersection of these two domains. Additionally, there is a paucity of research which has comprehensively analyzed how varying data sample sizes impact model performance and at what point additional data no longer contributes significantly to improving model accuracy. Understanding this threshold is crucial for determining the minimal data requirements for achieving accurate results.

3. Materials and Methods

3.1 Dataset

The dataset used for this study was ‘Suicidal Thought Detection’ from Kaggle deposited at https://www.kaggle.com/code/abhijitsingh001/suicidal-thought-detection originally published https://doi.org/10.1109/ICOSEC51865.2021.9591887. This Kaggle dataset consisted of collected different Reddit™ subreddits™ : “SuicideWatch” and “Depression”, utilizing the Pushshift API for data extraction. The “SuicideWatch” subreddit™ contained posts from December 16, 2009 to January 2, 2021, while the “Depression” subreddit™ included posts from January 1, 2009 to January 2, 2021.

In addition to the suicide and depression classes, this study also incorporated another set of posts obtained from GitHub at https://github.com/linanqiu/reddit-dataset.
These posts were collected from various general subreddits™ including "entertainment", "gaming", "humor", "learning", "lifestyle", "news", and "television", representing everyday conversations unrelated to mental health. This category of data served to test whether the model could accurately separate genuine signs of mental distress (including depression and suicide) from unrelated regular social media language, such that casual content was not incorrectly classified as as suicidal or depressive.

The final dataset consisted of 180,000 pieces of texts, which were categorized as suicidal, depressive, or general content (9). Each class contained 60,000 posts. These labels were pre-assigned during the original dataset creation process based on the specific subreddit that the data was pulled from (e.g. posts from "SuicideWatch" would be classified as suicidal).

The dataset contained short posts ranging from a few words to several sentences. Posts marked as suicidal often reflected expressions of hopelessness or direct references to self-harm. For example, one of the posts pulled from the dataset read "It ends tonight. I can't do it anymore. I quit." Non-suicidal, depressive posts, on the contrary, included personal struggles, usually related to low self-worth or isolation. For example, one of the posts from this dataset read "Recently, I've never really felt comfortable talking to people about my problems… Honestly, I'm so far beyond caring". General content was much more casual or lighthearted, often unrelated to mental health. For example, one of the general posts read "What's your favorite candy?"

To conduct this research, the Python programming language was used to build the model, utilizing a T4 GPU to enhance computational efficiency.

3.2 Data preprocessing

Data preprocessing is a critical step in preparing text data for effective training of the model. The preprocessing of the text data used in this study focused on ensuring uniformity and maintaining only relevant information to improve model performance. Each text entry was first converted to lowercase to ensure that words with different cases were treated with equal importance. Next, special characters and non-alphanumeric symbols were removed from the dataset as they do not contribute meaningfully to the analysis of the text. The text data was then tokenized, breaking down each sentence into individual words or tokens (10). Tokenization is a fundamental preprocessing step which helps in improving the interpretability of text by different models.

3.3 Model architectures

3.3.1 *Recurrent Neural Network (RNN) models*

RNNs are a type of neural network used to recognize patterns in sequences of data, such as time series, speech, text or video (11). RNNs are composed of neurons that work together to perform tasks by processing data. These neurons are grouped into three different kinds of layers: the input layer, the hidden layers, and the output layer (12). Data is first introduced to the network in the input layer. It is processed

and analyzed in the hidden layers. Finally, the output layer generates the final prediction based on the processed information.

What makes RNN models more suitable and distinctive from other models is their capability to maintain a "memory" of previous inputs (13). Essentially, this means that the output from the previous step is fed as input to the current step. This unique feature allows RNNs to process data in a way that takes into account the context and order of the inputs, making them particularly effective for tasks in which sequential data is required (14). One of the most essential aspects of RNN is its hidden state, which is fundamentally the "memory" where all the information about the sequence is obtained.

3.3.2 *Long Short-Term Memory (LSTM) models*

In a traditional RNN, the hidden state is passed from one time step to the next, making it difficult for the network to learn long-term patterns. LSTM models, however, introduce a memory cell that retains information over longer periods. These memory cells involve three gates: the input gate, the forget gate, and the output gate. The input gate controls what information enters the memory cell. The forget gate controls what information is taken out of the memory cell. The output gate determines what information is output from the memory cell (16).

3.3.3 *Gated Recurrent Unit (GRU) models*

GRU Models are similar to LSTMs, however, GRU uses two gating mechanisms: the reset gate and the update gate. The reset gate controls how much of the previous hidden state is forgotten, and the update gate controls how much of the new input should be considered to update the hidden state. This updated hidden state is what is used in order to calculate the output of the GRU (18).

3.3.4 *Benchmark*

To contextualize the performance of this study's model, accuracy from prior mental healthclassification research was examined. Reported accuracies across multiple studies ranged from 70% to 93% (20,21). By computing the mean of reported values from previous papers (93.7, 90.8, 84.7, 70, and 93), an estimated benchmark of 87% was established as a reasonable reference point for evaluating model performance for this research.

4. Results

4.1 Exploratory data analysis

Figure 1 shows the top ten most common words in texts classified as suicidal. These included "im", "dont", "like", "want", "know", "feel", "get", "life", "would", and "ive". The most frequently used was "im" with over 140,000 appearances, while words like "would" and "ive" appeared over 50,000 times. While these words can be considered to be commonly used in everyday conversations, words like "feel", "life", and "want" may have a stronger connotation with suicidal ideation. Understanding these word patterns helps reveal both what the model is learning and what warning signs suicidal text may include.

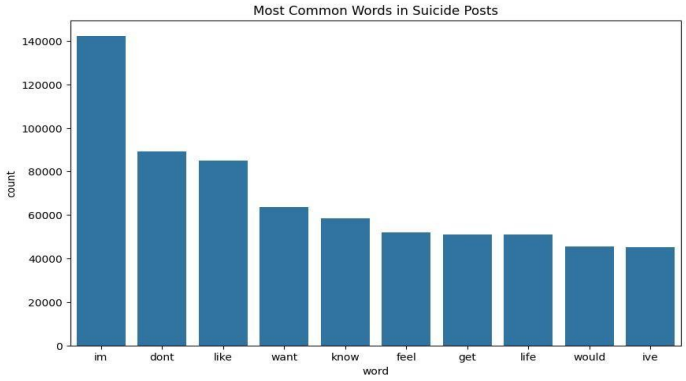

Figure 1. Common words used in text classified as suicidal

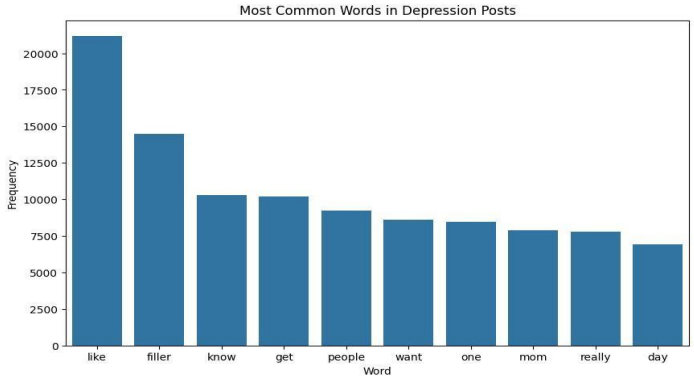

Figure 2. Common words used in text classified as depressive

Figure 2 recognizes the top ten most common words in text classified as depressive. These included “like”, “filler”, “know”, “get”, “people”, “want”, “one”, “mom”, “really”, and “day.” The most frequently used word was “like” with over 20,000 occurrences, while words like “really” and “day” appeared ~ 7,000 times. While many of these words are innocuous, some words such as “people”, “mom”, and “day” may hint at the everyday emotional challenges or relationship-based stress which appear commonly in depressive posts. Understanding these word patterns helps highlight how language in depressive content often centers around personal experiences and emotional discomfort.

Analyzing common text lengths of message is important because it helps the model understand whether suicidal posts are longer or shorter than average, thus providing a feature for classification. This could possibly be because text length could potentially exhibit the overall intensity of the message. Detailed descriptions of distress might be longer while others may be shorter, such as those that call for help.

Suicidal posts are often shorter, possibly due to the urgency or intensity of the moment. In the dataset, the most common suicidal post length was ~ 41 words, with an average of 102, and a median of 63. This suggests that most messages are brief, though a few longer ones increase the average, showing how suicidal thoughts can be expressed in both short implorations for help or longer detailed accounts. Depressive posts, however, tended to be longer. The mode was ~ 93 words, the median was 121, and the mean was ~ 134. These texts often included deeper context, reflecting how depression is usually expressed as an ongoing internal battle. The length allows for more detailed expression, aiding the model in recognizing patterns specific to depressive language.

4.2 Implementation

The model was implemented using Python with the Keras library running on Tensorflow. The dataset of 180,000 samples was split into an 80 to 20 ratio using stratified splitting, ensuring equal representation of the three types of posts across the training and testing sets. As a result, the training set contained 48,000 posts from each class (suicide, depression, and normal), and the test set contained 12,000 posts from each class. To evaluate the model's consistency and generalization ability, repeated random subsampling validation was used. For each test, 50 independent runs were performed with different randomly selected, stratified subsets of the full dataset as a first approximation to find how much data was needed for good performance.

The model was trained using the sparse categorical cross-entropy loss function, which is suitable for multi-class classification problems. This loss function calculates the difference between the predicted class probabilities and the actual class labels. To optimize the model, the Stochastic Gradient Descent (SGD) optimizer was used with a learning rate of 0.1 and a momentum value of 0.09. This configuration ensured stable updates to the model weights and helped the model learn efficiently.

Training was performed for 20 epochs, since increasing the number of epochs further did not improve performance significantly. A batch size of 256 was selected to balance computational efficiency and stability during training. The ReduceLROnPlateau callback was implemented to reduce the learning rate dynamically if validation performance did not improve, allowing the model to fine-tune its weights for better results.

4.3 Model configuration

Three types of RNN-based architectures were tested: Simple RNN, LSTM, and GRU. Each architecture was configured with the different node sizes of 32, 64, and 128, to determine how network complexity affected performance. Hyperparameters were carefully chosen to optimize the training process and prevent overfitting.

When testing the models with 32 nodes, performance was generally lower since it was more difficult to capture complex patterns in the data. The RNN model achieved ~ 92.5% train accuracy and 90.5% test accuracy, indicating some degree of underfitting. The LSTM model showed a similar trend, with ~ 93.2% train accuracy and 91.8% test accuracy.

The GRU reached 94.0% train accuracy and 92.0% test accuracy.

When testing the models with 64 nodes, the RNN model achieved 94.1% train accuracy and 92.2% test accuracy, while the LSTM model reached 94.9% train accuracy and 92.8% test accuracy. The GRU model outperformed both, achieving 95.6% train accuracy and 93.8% test accuracy, showing that this configuration provided balance between model complexity and performance.

When testing the models with 128 nodes, there were instances of overfitting, resulting in a decline in test accuracy. The RNN model with 128 nodes achieved ~ 95.0% train accuracy while the test accuracy was 91.8%. Similarly, the LSTM model reached 95.8% train accuracy but test accuracy was around 92.5%. The GRU model achieved 96.1% train accuracy and 93.5% test accuracy. These results show that the model excels with train data, however, failed to be as effective when exposed to unseen data.

The GRU model exhibited the best performance among the three architectures. Input sequences were padded or shortened to a maximum length of 128 tokens to ensure consistent input sizes. This length was selected to capture nearly all messages in full, given that the median sequence lengths for the classes were 63 and 121 tokens. A single GRU layer with 64 nodes (after conducting experiments with 32 and 128 nodes) was chosen, as it provided an optimal balance between learning capacity and computational efficiency. The output of the GRU layer was processed through a GlobalMaxPooling1D layer, which extracted the most important features by selecting the maximum value for each feature dimension.

Following this, a dense layer with 256 nodes and a ReLU activation function was added to further process the pooled features. The output layer had 3 units with softmax activation (for suicidal, normal, and depressive classes), trained with sparse categorical cross-entropy. Batch normalization was applied after the GRU layer to standardize activations, stabilize training, and speed up convergence.

The model was trained for 20 epochs, as further training showed little to no improvement in performance. A batch size of 256 was used, allowing for efficient processing while keeping the model stable during updates. Training was optimized using SGD with a learning rate of 0.1 and a momentum value of 0.09. This configuration allowed the model to converge effectively without making overly large updates. Additionally, ReduceLROnPlateau adjusted the learning rate dynamically when validation performance stopped improving, enabling fine-tuning during later training stages. To prevent overfitting and save computation time, early stopping was used with a patience value of 3 epochs, stopping training if validation loss plateaued.

4.4 RNN model

Figure 3 illustrates the accuracy curve for the RNN model across multiple epochs. The blue line represents training accuracy, showing how well the model learned the training data, while the orange line represents test accuracy, indicating performance on unseen data. Both training and test accuracy increased quickly

during the first few epochs, after which the curves began to diverge slightly. Training accuracy continued to rise steadily, eventually ~97%. Test accuracy also improved but stabilized around 92% with fluctuations across later epochs. After roughly 15 epochs, the test accuracy became relatively stable and no longer showed meaningful improvement, indicating that the model had reached its generalization capacity. The growing gap between training and test performance at this stage suggested overfitting as the model continued to fit the training data more closely despite no significant improvement in unseen data.

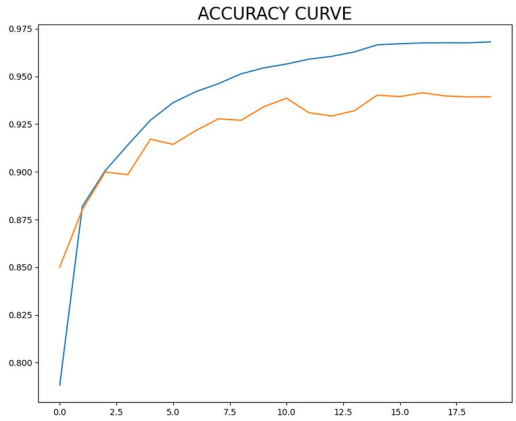

Figure 3. RNN accuracy, blue: training, orange: testing

4.5 LSTM model

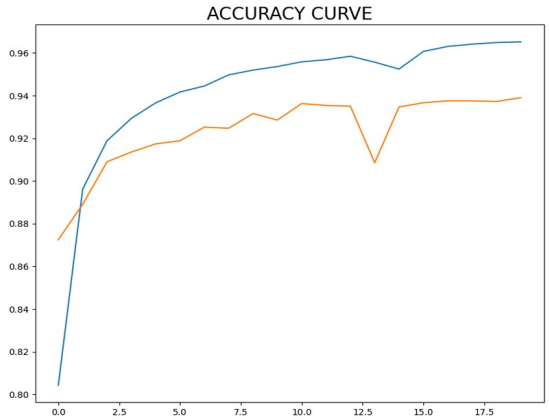

Figure 4. LSTM accuracy, blue: training, orange: testing

Figure 4 illustrates the accuracy curve for the LSTM model across multiple epochs. The blue line represents training accuracy, while the orange line represents test accuracy. Training accuracy continued to steadily increase, reaching ~96%, while test accuracy stabilized near ~93%. Around epoch 13, there was a noticeable dip in test accuracy, while training

accuracy kept improving, indicating that the model was overfitting during this period. After roughly 15 epochs, test accuracy recovered and flattened, showing that the model had reached stable generalization.

4.6 GRU model

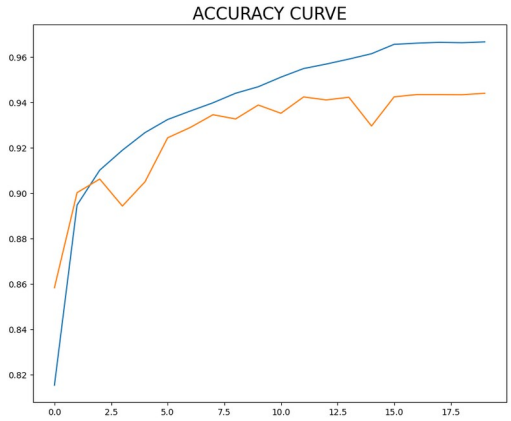

Figure 5. GRU accuracy, blue: training, orange: testing

Figure 5 illustrates the accuracy curve for a GRU model across multiple epochs. The blue line represents the training accuracy, while the orange line represents the test accuracy. Training accuracy rose consistently, eventually reaching about 97%, while test accuracy reached ~94%. In early epochs, the training and test curves were closely aligned, but as training progressed, a gap formed, indicating overfitting. This was most visible around epoch 13, where the test accuracy dipped while training accuracy continued to climb. After epoch 15, test accuracy stabilized, suggesting that the model reached a balance between learning patterns and generalization despite the signs of overfitting.

4.7 Varying dataset size for GRU

Table 1. Performance metrics of GRU model across different dataset sizes

% of Data	Accuracy	Precision	Recall	F1 Score
100%	0.9314 ± 0.0022	0.9321 ± 0.0021	0.9314 ± 0.0022	0.9310 ± 0.0022
70%	0.9315 ± 0.0014	0.9322 ± 0.0014	0.9316 ± 0.0014	0.9311 ± 0.0014
50%	0.8961 ± 0.0031	0.9003 ± 0.0029	0.8962 ± 0.0031	0.8952 ± 0.0032
20%	0.8756 ± 0.0035	0.8830 ± 0.0031	0.8755 ± 0.0035	0.8752 ± 0.0035
10%	0.8629 ± 0.0050	0.8648 ± 0.0049	0.8629 ± 0.0049	0.8619 ± 0.0050
4%	0.8305 ± 0.0082	0.8336 ± 0.0081	0.8303 ± 0.0080	0.8294 ± 0.0082
2%	0.8074 ± 0.0131	0.8101 ± 0.0132	0.8073 ± 0.0132	0.8055 ± 0.0134
1%	0.7542 ± 0.0196	0.7558 ± 0.0198	0.7544 ± 0.0197	0.7538 ± 0.0198

When analyzing test accuracy in relation to different data sample sizes, performance remained above 87%, which was the benchmark for this study, as long as at least 20% of the dataset was used. Noticeable gains ocurred as the data increased up to 70%, but beyond this point, the improvements were minor, indicating that the GRU model reached a point of stability once a sufficient amount of training data was available.

4.8 Varying dataset size for LSTM

Table 2. Performance metrics of LSTM model across different dataset sizes

% of Data	Accuracy	Precision	Recall	F1 Score
100%	0.9241 ± 0.0024	0.9250 ± 0.0023	0.9241 ± 0.0024	0.9235 ± 0.0024
70%	0.9156 ± 0.0025	0.9173 ± 0.0024	0.9156 ± 0.0025	0.9150 ± 0.0025
50%	0.8886 ± 0.0025	0.8921 ± 0.0023	0.8886 ± 0.0025	0.8879 ± 0.0025
20%	0.8763 ± 0.0042	0.8803 ± 0.0040	0.8762 ± 0.0042	0.8753 ± 0.0042
10%	0.8414 ± 0.0059	0.8462 ± 0.0057	0.8413 ± 0.0058	0.8407 ± 0.0060
4%	0.8112 ± 0.0102	0.8171 ± 0.0099	0.8110 ± 0.0101	0.8107 ± 0.0102
2%	0.7864 ± 0.0149	0.7942 ± 0.0148	0.7861 ± 0.0150	0.7855 ± 0.0153
1%	0.7205 ± 0.0237	0.7253 ± 0.0252	0.7205 ± 0.0234	0.7144 ± 0.0243

When evaluating test accuracy at different dataset sizes, the LSTM maintained performance above the 87% benchmark as long as at least 20% of the data was used. Accuracy steadily increased between 20% and 70% of the dataset, after which the improvements were minimal, suggesting that the model had already captured most of the underlying patterns. When data availability dropped below 20%, however, performance declined more sharply, with accuracy highlighting the LSTM's higher sensitivity to limited training samples compared to larger subsets.

4.9 Varying dataset size for RNN

Table 3. Performance metrics of RNN model across different dataset sizes

% of Data	Accuracy	Precision	Recall	F1 Score
100%	0.9204 ± 0.0025	0.9226 ± 0.0024	0.9204 ± 0.0025	0.9199 ± 0.0026
70%	0.9036 ± 0.0027	0.9052 ± 0.0026	0.9036 ± 0.0027	0.9030 ± 0.0027
50%	0.8993 ± 0.0024	0.9033 ± 0.0023	0.8993 ± 0.0024	0.8984 ± 0.0024
20%	0.8735 ± 0.0038	0.8748 ± 0.0038	0.8734 ± 0.0038	0.8726 ± 0.0038
10%	0.8524 ± 0.0057	0.8541 ± 0.0057	0.8523 ± 0.0056	0.8518 ± 0.0057
4%	0.7441 ± 0.0108	0.7970 ± 0.0090	0.7437 ± 0.0100	0.7345 ± 0.0113
2%	0.6990 ± 0.0150	0.7695 ± 0.0145	0.6985 ± 0.0134	0.6707 ± 0.0163
1%	0.7102 ± 0.0188	0.7441 ± 0.0167	0.7097 ± 0.0194	0.7090 ± 0.0193

For the RNN model, accuracy remained above the 87% benchmark when at least 20% of the dataset was used, indicating that the model could still capture the main distinctions between classes at this scale. Performance improved with larger training sizes, but gains leveled off after 50-70%, showing that additional data provided less impact once a sufficient portion of the dataset had been processed. When the available data fell below 20%, however, accuracy declined considerably, with values dropping to the 70% range rapidly, showing that being able to learn key patterns with moderate data availability, the RNN was more dependent on adequate data.

5. Discussion

5.1 Deep Learning models

This study compared three different types of Recurrent Neural Network (RNN) architectures: conventional RNNs, LSTM networks, and GRUs, to detect suicidal thoughts from Reddit™ posts. Each model's unique strengths and limitations influenced its performance.

Conventional RNNs, while capable of processing sequence data, struggled with vanishing gradients, limiting their ability to learn long-term dependencies. LSTM networks, designed to overcome the vanishing gradient problem, showed improved results by maintaining longer-term memory. The LSTM architecture, which includes forget, input, and output gates, allowed it to store, update, and retrieve information more effectively in comparison to conventional RNNs. Despite their improvement in accuracy when compared with conventional RNNs, they underperformed GRUs.

GRUs achieved the best results among the models tested. This model architecture maintained the capacity to process long-term dependencies while being more efficient due to fewer parameters, contributing to its superior performance. The GRU model consistently achieved higher accuracy with lesser training times. As shown in Table 4, The GRU model exhibited the best performance, with 0.9267 accuracy on unseen data, indicating its ability to generalize well. LSTM followed closely behind, while RNN performed the worst for this data.

Table 4. Metrics for the different models tested in the study (100% of data)

Model Name	Accuracy	Precision	Recall	F1
RNN	0.9204 ± 0.0025	0.9226 ± 0.0024	0.9204 ± 0.0025	0.9199 ± 0.0026
LSTM	0.9241 ± 0.0024	0.9250 ± 0.0023	0.9241 ± 0.0024	0.9235 ± 0.0024
GRU	0.9314 ± 0.0022	0.9321 ± 0.0021	0.9314 ± 0.0022	0.9310 ± 0.0022

Across different classification thresholds, GRU consistently exhibited comparable separation with the other models in distinguishing between suicidal and non-suicidal posts, as evidenced by the Receptor Operating Characteristic (ROC), Area Under the Curve (AUC) as well as the Precision Recall Area Under the Curve (PRAUC) scores (Table 5).

Table 5. Performance of the different models across different classification thresholds

Model Name	Macro Average ROC:AUC	Macro Average PRAUC
RNN	0.979	0.964
LSTM	0.979	0.965
GRU	0.981	0.969

5.2 Effect of different sample size

The study highlights the significant effect of data sample size on the performance of Machine Learning models, particularly on the subject of distinguishing between suicide and depression. As dataset size increased, the test accuracy of the all three models increased, consistent with previous findings that emphasized the importance of large datasets in capturing diverse patterns and in reducing noise (8). Not only does this study support findings from other studies, but it also identifies that there is a threshold beyond which additional data does not contribute to significant gains, in the context of the dataset examined.

Given the benchmark of 87%, the GRU model chosen in this study met and slightly exceeded the expected accuracy. The model achieved test accuracies between 87% and 93% when trained on dataset sizes ranging from 36,000 (20% of the original dataset size) to 126,000 (70% of the original dataset size) samples. Beyond this range, increasing the dataset size further did not yield significant improvements in classification performance. This suggests that while larger datasets are beneficial, collecting and processing excessive amounts of data may not be necessary for achieving acceptable accuracy results.

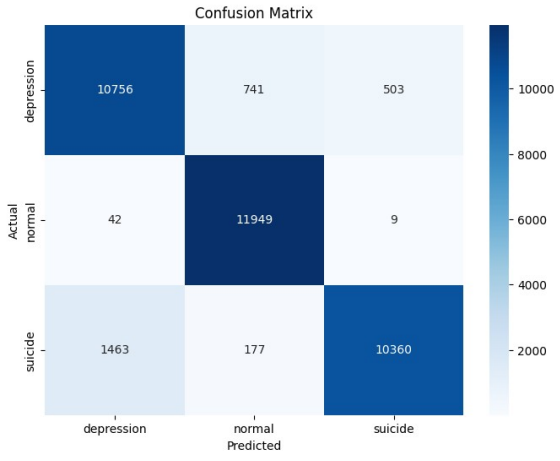

Figure 6. Confusion matrix of GRU classification accuracy when trained on the full dataset (180,000 posts) and evaluated on the 20% test set (36,000).

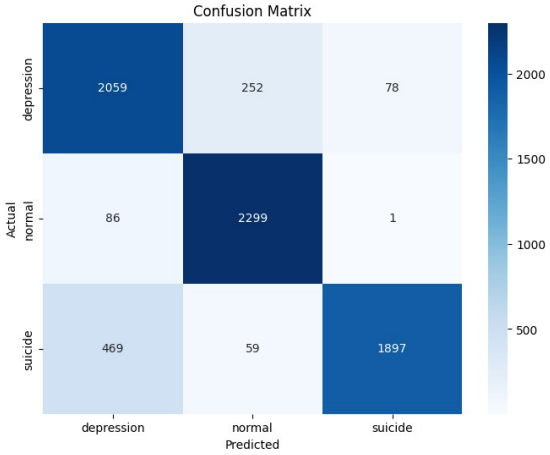

Figure 7. Confusion matrix of GRU classification accuracy when trained on a 20% subset of the data (36,000 posts) and evaluated on its 20% test split (7,200 posts)

A comparison of Figures 6 and 7 shows that the model is generally able to correctly classify the majority of posts across all three categories, regardless of whether the complete dataset is used or only 1/5th of the dataset is used. In Figure 7, the model using 20% of the data correctly identified 1,897 suicidal posts, 2,299 normal posts, and 2,059 depressive posts, showing that it could still recognize key patterns in language even with fewer training samples. Figures 6 and 7 demonstrate that the overall structure of the predictions remains consistent with a smaller dataset without a proportional loss in accuracy.

The LSTM model achieved accuracies ranging from 87% to 92% when trained on dataset sizes between 20% (36,000 samples) and 70% (126,000 samples) of the full dataset. At the full dataset size, performance reached 92%, though improvements beyond 70% were minimal. Similar to the GRU, these results show that while larger datasets improved performance, the LSTM stabilized once 70% of the data was used, with additional data providing little added benefit.

Figures 8 and 9 show that the LSTM model was able to classify posts across all three categories accurately, whether trained on the full dataset or a reduced subset. With the complete dataset (Figure 8), the model correctly classified 10,639 depressive posts, 11,950 normal posts, and 10,167 suicidal posts, across all categories. Using only 20% of the data (Figure 9), the model correctly identified 1,960 depressive posts, 2,337 normal posts, and 1,936 suicidal posts. Although the total posts are smaller because fewer samples were available, the balance across categories remained similar, showing that the LSTM was still able to separate the three classes effectively even with limited data.

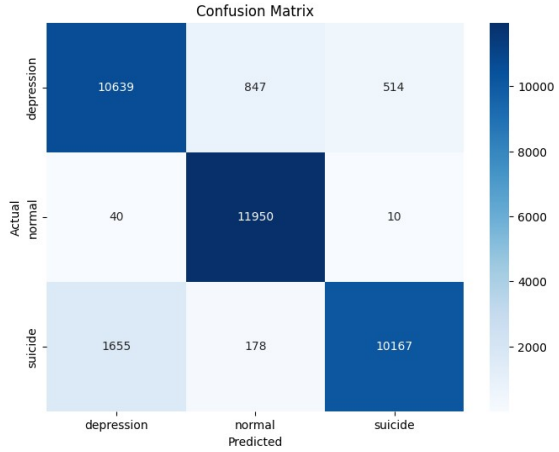

Figure 8. Confusion matrix of LSTM classification accuracy when trained on the full dataset (180,000 posts) and evaluated on the 20% test set (36,000).

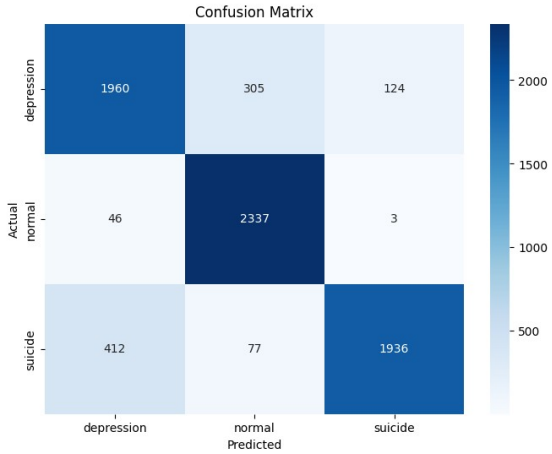

Figure 9. Confusion matrix of LSTM classification accuracy when trained on a 20% subset of the data (36,000 posts) and evaluated on its 20% test split (7,200 posts).

Figures 10 and 11 show the RNN model's performance when trained on the full dataset and on a reduced subset. With the complete dataset (Figure 10), the model correctly predicted 10,785 depressive posts, 11,949 normal posts, and 9,842 suicidal posts, demonstrating that it could distinguish between the categories accurately. When trained on only 20% of the data (Figure 11), it correctly identified 1,925 depressive posts, 2,299 normal posts, and 2,011 suicidal posts.

When comparing the three architectures, all three models performed well when trained on the full dataset, each achieving an accuracy > 92%. With only 20% of the data, performance remained above the 87% benchmark for all three, showing that smaller subsets were still

sufficient to capture the key language patterns distinguishing suicidal, depressive, and normal posts.

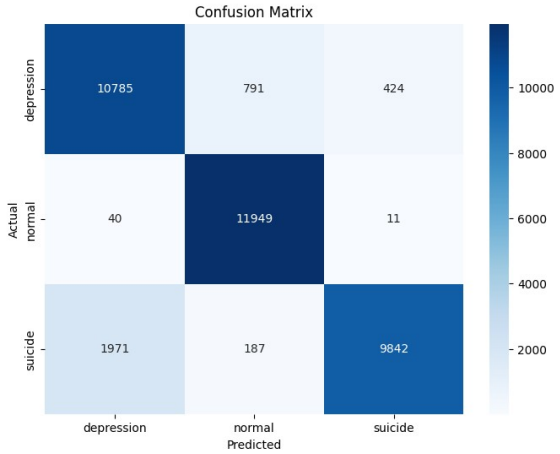

Figure 10. Confusion matrix of RNN classification accuracy when trained on the full dataset (180,000 posts) and evaluated on the 20% test set (36,000 posts)

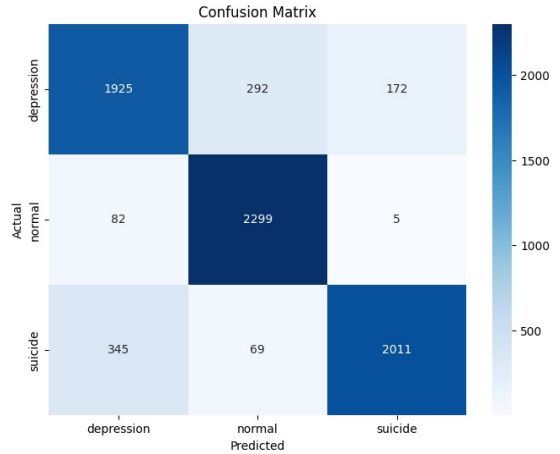

Figure 11. Confusion matrix of RNN classification accuracy when trained on a 20% subset of the data (36,000 posts) and evaluated on its 20% test split (7,200)

At this reduced data size of 20% of the full dataset, the results of GRU, LSTM, and RNN were relatively similar, suggesting that the most distinctive features of the text could be learned effectively even without the entire dataset. The advantage of GRU and LSTM becomes clearer at smaller fractions of data (Tables 1 and 2). It is possible that that the gating mechanisms of these models preserve relevant contextual information and prevent loss of signal over longer sequences. The RNN, while still comparable at 20%, may be slightly more dependent on larger sample sizes to maintain stability, and its performance tended to decline more sharply when even less data was available.

5.5 Limitations

Despite the benefits, this study does have some limitations. The reliance on text-based data might not fully capture the complexity of an individual's mental state. Research using multimodal data should be considered as it could significantly enhance the accuracy of suicidal detection models. Combining text analysis with other data types, such as images, videos, and audio, can provide much better context than purely text-based models; obviously these may not simultaneously be available since a person posting on the web as an anonymous user would not be expected to post photographs, video or audio of themselves. Research into data sample size, alongside multimodal data approaches, could further establish guidelines for constructing datasets.

Additionally, future work should consider significantly expanding the variety of linguistic data used during training. This type of data should include evolving linguistic patterns and syntax; especially among teenagers, who are the most at risk of suicidal ideation or depression. A considerable amount of posts/data related to suicidal ideation or depression may be unavailable in the public domain since social media platforms may restrict or prevent availability. In such a scenario, data may need to be manufactured or synthesized to closely match verbiage and conversational usage among the at-risk demographic (as has been done in the Appendix).

6. Conclusion

This study investigated whether dataset size for detecting suicidal thoughts could be reduced while retaining predicted classification accuracy using various RNN architectures, including LSTM networks and GRUs, applied to Reddit™ posts. Reducing the original dataset containing 180000 posts to 1/5th its size reduced the accuracy from 92% to 87% with a GRU model. Among the test architectures, the GRU model demonstrated the highest accuracy and efficiency, surpassing conventional RNNs and performing slightly better than LSTMs.

These findings not only validate the effectiveness of the GRU model but also highlight the importance of selecting the optimum dataset size in suicide classification research. While increasing dataset size initially improves performance, it eventually reaches a point where further improvement becomes minimal, emphasizing the need to balance accuracy with computational efficiency.

While this research primarily focuses on detecting suicidal thoughts from Reddit™ posts, its implications extend far beyond this specific platform. The broader objective is to develop a machine learning model capable of generalizing to the differentiation of suicidal and non-suicidal content across many different data sources. Reddit™ data only serves as a starting point and provides a foundation for understanding language patterns related to mental health. This becomes all the more evident when examining the manually synthesized posts in the Appendix; where none of the models discussed in this study proved accurate.

Acknowledgement

I would like to express my gratitude to my mentor, Pramit Saha, for his guidance.

7. References

1. Prince, M., Patel, V., Saxena, S., Maj, M., Maselko, J., Phillips, M. R., Rahman, A. (2007). No health without mental health. Lancet, 370(9590), 859-877. https://doi.org/10.1016/s0140-6736(07)61238-0

2. Knox, K. L., Conwell, Y., Caine, E. D. (2004). If suicide is a public health problem, what are we doing to prevent it?. Am. J Public Health, 94(1), 37-45. https://doi.org/10.2105/ajph.94.1.37

3. Chancellor, S., De Choudhury, M. (2020). Methods in predictive techniques for mental health status on Social Media: A critical review. Npj Digit. Med., 3, 43. https://doi.org/10.1038/s41746-020-0233-7

4. Jamnik, M. R., Lane, D. J. (2017). The use of reddit as an inexpensive source for high-quality data. Practical Assessment, Research & Evaluation, 22(1), 5. https://doi.org/10.7275/j18t-c009

5. Haque, A., Reddi, V., Giallanza, T. (2021). Deep Learning for Suicide and Depression Identification with Unsupervised Label Correction. In: Farkaš, I., Masulli, P., Otte, S., Wermter, S. (eds) Artificial Neural Networks and Machine Learning – ICANN 2021. ICANN 2021. Lecture Notes in Computer Science,12895. Springer. https://doi.org/10.1007/978-3-030-86383-8_35

6. Ramírez-Cifuentes, D., Freire, A., Baeza-Yates, R., Puntí, J., Medina-Bravo, P., Velazquez, D. A., Gonfaus , J. M., Gonzàlez, J. (2020). Detection of Suicidal Ideation on Social Media: Multimodal, Relational, and Behavioral analysis. J Med. Internet Res., 22(7), e17758. https://doi.org/10.2196/17758

7. Khoo, L. S., Lim, M. K., Chong, C. Y., McNaney, R. (2024). Machine Learning for Multimodal Mental Health Detection: A Systematic Review of Passive Sensing Approaches. Sensors, 24(2), 348. https://doi.org/10.3390/s24020348

8. Rajput, D., Wang, W.-J., & Chen, C.-C. (2023). Evaluation of a Decided Sample Size in Machine Learning Applications. BMC Bioinformatics, 24, article no. 48. https://doi.org/10.1186/s12859-023-05156-9

9. Singh, A. (2021). Suicidal Thought Detection. Kaggle. https://www.kaggle.com/code/abhijitsingh001/suicidal-thought-detection

10. Fares, M., Oepen, S., Zhang, Y. (2013). Machine Learning for High-Quality Tokenization Replicating Variable Tokenization Schemes. In: Gelbukh, A. (eds) Computational Linguistics

and Intelligent Text Processing. CICLing 2013. Lecture Notes in Computer Science, vol 7816. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-37247-6_19

11. Schmidt, R. M. (2024). Recurrent neural networks (RNNs): A gentle introduction and overview. https://doi.org/10.48550/arXiv.1912.05911

12. Koutnik, J., Greff, K., Gomez, F. & Schmidhuber, J. (2014). A Clockwork RNN. Proceedings of the 31st International Conference on Machine Learning, in Proceedings of Machine Learning Research, 32(2),1863-1871. https://proceedings.mlr.press/v32/koutnik14.html

13. Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 404, 132306. https://doi.org/10.1016/j.physd.2019.132306

14. Ravichandar, H. C., Kumar, A., Dani, A. P., Pattipati, K. R. (2016). Learning and Predicting Sequential Tasks Using Recurrent Neural Networks and Multiple Model Filtering. The 2016 AAAI Fall Symposium Series: Shared Autonomy in Research and Practice Technical Report FS-16-05. https://cdn.aaai.org/ocs/14105/14105-62076-1-PB.pdf

15. GeeksforGeeks. (2024). Introduction to recurrent neural network. https://www.geeksforgeeks.org/machine-learning/introduction-to-recurrent-neural-network/

16. Gers, F. A., Schraudolph, N. N., Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. J Machine Learning Res., 3(1), 115-143. http://dx.doi.org/10.1162/153244303768966139

17. O'Shea, T. J., Hitefield, S., Corgan, J. (2016). End-to-End Radio Traffic Sequence Recognition with Deep Recurrent Neural Networks. http://dx.doi.org/10.48550/arXiv.1610.00564

18. Wang, X., Xu, J., Shi, W., Liu, J. (2019). OGRU, an optimized Gated Recurrent Unit Neural Network. J Phys.: Conf. Ser., 1325, 012089. https://doi.org/10.1088/1742-6596/1325/1/012089

19. Riaz, A., Nabeel, M., Khan, M., Jamil, H. (2020). SBAG: A Hybrid Deep Learning Model for Large Scale Traffic Speed Prediction. Int. J. Adv. Comp. Sci. Appl., 11(1), 287-291. http://dx.doi.org/10.14569/IJACSA.2020.0110135

20. Belsher, B. E., Smolenski, D. J., Pruitt, L. D., Bush, N. E., Beech, E. H., Workman, D.E., et al. (2019). Prediction models for suicide attempts and deaths: A systematic review and

Simulation. JAMA psychiatry, 76(6), 642-651. https://doi.org/10.1001/jamapsychiatry.2019.0174

21. Ehtemam, H., Esfahlani, S. S., Sanaei, A., Ghaemi, M. M., Hajesmaeel-Gohari, S., Rahimisadegh, R., et al. (2024). Role of machine learning algorithms in suicide risk prediction: A systematic review-meta analysis of Clinical Studies. BMC Med. Inform. Decis. Mak., 24(1), 138. https://doi.org/10.1186/s12911-024-02524-0

Appendix (*Post hoc* assessment)

A1. List of thirty sentences that do NOT express suicidal or depressive ideation
1. Im like, don't you know you can't get this mad at your friends?
2. Iam like, I don't even want to know dude!
3. Iam like, get a life dude!
4. Ive lived long enough to know that you can't get everything you want.
5. I hate myself for not having the sense to apply for social security benefits at age 62.
6. A cul-de-sac translates into a dead end.
7. Man; that hurt like hell, but you know what they say – pain is weakness leaving the body.
8. Trying to solve this equation is killing me. I am at the end of my rope.
9. Was there a movie titled " I am nobody " ?
10. Ironically, one of my most successful books was titled " I am planning on committing suicide at 18".
11. The premise of the movie "Die another day" was unreal – to say the least.
12. I probably won't be able to take the punishment that Bruce Willis takes in Die Hard.
13. My existence has no intrinsic value apart from the value I chose to imbibe into it.
14. I think my fish are dying.
15. I am killing it. My investment returns are insane.
16. I dreamed that I was dead after being drowned in my bathtub. It turns out it was the dog licking me in my sleep.
17. Dying may come naturally to me since I can hold my breath for 4 minutes straight.
18. Coming from a successful person like me, would you believe it if I said that my life was pathetic, lonely and depressing and I wanted to end it ?
19. If this experiment does not succeed, it is my neck on the line.
20. I am just killing time, until it actually kills me.
21. In order to keep myself from disappointing people, I am going to kill them.
22. I have nothing to look forward to so killing myself would be a logical – yet ludicrous – choice.
23. I am finding comfort in committing suicide when I stare at this abstract art painting long enough.

24. After defeating my opponent, I feel no joy; only sorrow, despair and a sense of wanting to end all future games of chess.
25. I am sick of this shit and I would rather kill myself than watch you catch more fish than I, day after day.
26. Sorry, I have had enough and I am getting out of here.
27. I am so bad at this video game that I can't even stay alive for 2 seconds.
28. I can't live anymore with this pain so I have decided to commit to a clinical trial for a new antidepressant.
29. When I am gone and buried, I want my estate planning to be impeccable.
30. I can't go on like this. I need a break from this insanity.

A2. *Post hoc* analysis

To further assess the model's ability to generalize beyond the more obvious Social media post examples, a *post-hoc* evaluation was performed using 30 manually constructed 'normal' sentences (not suicidal or depressive) that were not included in the original training or test datasets (see A1). These sentences were *not* intentionally chosen to be more contextually complex, but, rather, represented the syntactical evolution of vernacular that has become the norm among teenagers; which are – by some accounts – the most vulnerable demographic because of their propensity to depression and their ideation of suicide.

Out of these 30 examples, only 11 were correctly classified as normal by the best performing GRU model. The other 19 were misclassified as either depressive or suicidal. This suggests that the model has difficulty identifying current and evolving language styles even though they do not contain any clear indicators of mental health concerns. For example, while the model was able to correctly classify a sentence like "Was there a movie titled 'I am nobody'?" or "Sorry, I have had enough and I am getting out of here" as normal, it failed to do so for sentences like "I am so bad at this video game that I can't even stay alive for 2 seconds", "Trying to solve this equation is killing me. I am at the end of my rope", or "Ironically, one of my most successful books was titled 'I am planning on committing suicide at 18,'" showing how the model can be thrown off by certain impactful keywords, current ubiquitous slang, cultural diffusion, contextual indistinctiveness or veiled language.

The LSTM model correctly identified 12 out of the 30 sentences as normal, which is slightly better than the GRU's performance. However, the remaining 18 sentences were still misclassified as either depressive or suicidal. This indicates that the LSTM faced similar challenges in handling figurative expressions, slang, or ambiguous phrasing that did not actually convey mental health distress. For instance, while it could correctly classify "I am killing it. My investment returns are insane" as a normal sentence, it misinterpreted examples like "I am just killing time, until it actually kills me," suggesting that reliance on key terms or root words such as "kill" appears to bias the model toward non-normal categories.

The RNN model struggled the most with this *post hoc* evaluation, correctly classifying only 6 out of the 30 sentences as normal. This suggests that the simpler architecture of the RNN was more easily misled by the presence of emotionally charged words or phrases, regardless of the contextual meaning. Many sarcastic or figurative expressions were misclassifed such as "Man; that hurt like hell, but you know what they say – pain is weakness leaving the body" and "I dreamed that I was dead after being drowned in my bathtub. It turns out it was the dog licking me in my sleep." More straightforward sentences such as "Sorry, I have had enough and I am getting out of here" were correctly classified. The RNN's weaker performance compared to both the GRU and LSTM highlights how limited memory and context handling makes it more prone to false positives in identifying suicidal or depressive ideation, reinforcing the importance of using more advanced architectures and including sentences with varying language in the dataset.

The *post hoc* analysis hence serves as a note of caution when interpreting claims of high accuracy, precision or recall for classification tasks performed by Machine Learning algorithms based on Artifical Intelligence (AI). AI is only as good as the breadth of the data it is trained on. The good news is that this study has demonstrated that it required ~ a fifth of the volume of data for accuracy comparable to the complete dataset. This implies that if the breadth of that $1/5^{th}$ of data volume were increased to encompass the most variety that existed in the suicide/depression social media/web discussion space, timely help may reach all suicide ideators, regardless of their demographic and/or vernacular.