Early Detection of Mental Health Disorders using Twitter-STMHD

Ben Chapman-Kish
School of Engineering
bchapm02@uoguelph.ca

Parya Abadeh
School of Engineering
pabadeh@uoguelph.ca

John Quinto
School of Engineering
jquinto@uoguelph.ca

Om Bhosale
School of Engineering
obhosale@uoguelph.ca

Abstract—**This report explores the development of a deep learning model for early detection of mental health disorders using social media data. We propose a system that analyzes historical tweets from users with known mental health conditions in the Twitter-STMHD dataset (focus on depression group). By employing text pre-processing and word embedding techniques, the model aims to identify patterns and sentiment indicative of potential mental health issues. A Long Short-Term Memory (LSTM) network and Gate Recurrent Unit (GRU) will be used to analyze the sequential nature of tweet data and capture long-term dependencies within user communication. This model predicts the likelihood of a user developing a specific mental health disorder (focusing on depression), aiming to facilitate earlier intervention. The project evaluates the model's performance using metrics like accuracy and leverages the potential of social media data and deep learning for advancements in early detection of mental health issues.**

Keywords—mental disorder, LSTM, deep learning, social media, word embedding.

I. Introduction

Mental health disorders affect a significant portion of the global population, with depression being the most prevalent. Early detection and intervention are crucial for improving treatment outcomes and promoting overall well-being. Traditional methods for mental health diagnosis often rely on clinical assessments, which can be time-consuming, resource-intensive, and may not always capture early signs of an emerging issue.

This report investigates the potential of leveraging social (media data and deep learning techniques for the early detection of depression. We propose a novel approach utilizing the Twitter-STMHD dataset, focusing specifically on users diagnosed with depression. By analyzing historical tweets posted, our model aims to identify linguistic patterns and sentiment that might indicate potential signs of depression. This approach has the potential to provide a non-invasive and scalable method for early detection, allowing for timely intervention and improved treatment outcomes for a large portion of the population suffering from depression.

This work contributes to the field of mental health informatics by:

- Developing a deep learning model for early detection of depression from a *user's history of social media posts* rather than their *individual posts*: The proposed model analyzes historical user social media data to identify early signs of depression.
- Utilizing the Twitter-STMHD dataset: This rich dataset provides a valuable source of real-world user data for model development and evaluation, with a focus on depression diagnoses.
- Investigating the effectiveness of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks: We focus on LSTMs and GRUs, for their ability to capture long-term dependencies within sequential tweet data.
- Investigating the effectiveness of transformer-based BERT models: We analyze pre-trained BERT models due to their ability to bidirectionally analyze contextual relations between words in sentences up to 512 words long, which is useful for long sequences of tweets.

The remainder of this paper is organized as follows. Section 2 provides related works. Section 3 delves into the methodology, outlining the data pre-processing techniques, model architecture, and evaluation metrics. Section 4 presents the results obtained from training and evaluating the model. Section 5 discusses the implications of the findings and potential future directions. Finally, Section 6 concludes the paper by summarizing the key contributions and highlighting the potential impact on early detection of depression.

II. Related Works

Chen et al. [1] proposed a two-stream approach for detecting 7 mental health disorders. The authors compiled the social media post history of 5624 diagnosed users and 17,209 control users to produce "symptom" and "risky post" streams. They utilize BERT models to extract symptom-related features from each user's post history, then highlight the *K* posts that are most indicative of psychiatric symptoms. Their approach outperforms other similar models since it also considers domain knowledge of psychiatric disorders, including overlapping symptoms between different conditions, and the potential of comorbidities.

Singh et al. [2] employ RNN, GRU, and CNN architectures to predict depression from Twitter data, examining both character-based and word-based models alongside pre-trained and learned embeddings. Using a dataset of 13,385 manually labeled tweets, the study explores various model configurations, with the word-based GRU and CNN models showing the highest accuracy. The study acknowledges a limitation in dataset quality due to the selection criteria based solely on the presence of the keyword "depression." It suggests future improvements could include integrating CNN-LSTM architectures and adjusting learning rates, as well as enhancing the dataset with labels provided by domain experts to improve the models' utility and accuracy in clinical settings.

Rizwan et al. [3] explored the use of small pretrained language models for classifying the severity of depression from individual tweets. Given a labelled dataset of 73,368 tweets the authors attempted to categorize tweets into 3 levels of depression severity: "mild", "moderate", and "severe." They obtain these labels by applying the valence-aware dictionary and sentiment reasoner (VADER) [4] sentiment analysis model to calculate the polarity of each tweet, then applying depression diagnostic criteria from the ICD-10 [5]. Of models they analyzed, two were versions of the BERT architecture with fewer than 15 million parameters: Albert [6] and DistilBERT [7]. This study demonstrates that neural networks with fewer parameters can achieve similar, if not better performance than models with more parameters while requiring less computational resources. Smaller and more efficient models may be more valuable for mental health disorder detection due to their ability to perform real-time analysis of user tweets.

García-Noguez et al. [8] attempt to classify depressed users on Twitter using temporal analyses of their tweet history. Using the CLPsych 2015 dataset, they propose a text preprocessing scheme that involves removing hashtags, retweets, repeated words, and stop words, then tokenizing the cleaned text using Global Vectors for Word Representation (GloVe) word embeddings. This work also proposes a one-dimensional CNN architecture for classifying depressive users, then compares those results with those from an XGBoost model. Of particular interest for our work is this paper’s tweet preprocessing methodology, and their use of temporal information to classify depressive users instead of depressive tweets.

Lastly, Agrawal et al. [9] explore the use of a 1D-convolutional LSTM model for classifying the polarity of individual tweets. Similarly to Singh et al. [1], the authors compiled a dataset of “depressed” tweets by using tags such as “depression”, “anxiety”, and “loneliness”, then compiled a corresponding dataset of “normal” tweets by using tags such as “joy”, “happy”, and “delight.” They then propose a convolutional LSTM architecture consisting of a word embedding layer, a 1D convolutional layer, then an LSTM layer. Their model outperforms Naive Bayes and Random Forest baselines. This work provides a relatively lightweight neural network architecture for tweet classification (with about 430,000 trainable parameters), that may be useful to compare with pure LSTM models.

III. Methodology

In this section, we aim to discuss the methodologies that have been implemented, shedding light on their underlying mechanisms, and providing a comprehensive understanding of their application.

A. LONG SHORT-TERM MEMORY (LSTM)

Unveiling user preferences in the ever-expanding world of music streaming is a challenge that deep learning techniques, particularly Long Short-Term Memory (LSTM) networks, are well-suited to address. Unlike traditional neural networks that process information one data point at a time, LSTMs offer a distinct advantage: their feedback loop architecture. This allows them to analyze entire sequences of data, such as a user's tweet history, and identify long-term patterns within it. This exceptional ability at handling sequential information makes LSTMs ideal for tasks like sequence prediction, a capability we leverage in this study.

Here, we utilize the power of LSTMs to analyze user tweeting patterns and predict the likelihood of a user developing depression (performing a classification task). Crucially, LSTMs excel at overcoming the "vanishing gradient" problem, a common hurdle in traditional neural networks that can hinder their ability to learn from long sequences of data. This makes LSTMs particularly well-suited for analyzing user tweet history, where understanding long-term trends is key to accurate predictions. In the next section, we'll delve deeper into the specific architecture of LSTM models and how it empowers them to unlock valuable insights from sequential data.

B. LSTM Architecture

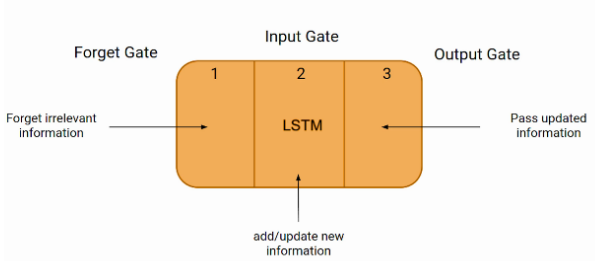
Traditional recurrent neural networks (RNNs) struggle with capturing long-term dependencies in sequential data due to the vanishing gradient problem. This limitation hinders their ability to learn from long sequences. Long Short-Term Memory (LSTM) networks address this challenge. LSTMs are a special type of RNN architecture equipped with internal mechanisms that effectively deal with the vanishing gradient problem. By understanding how LSTMs are structured, we can gain insight into how they overcome this limitation and excel at tasks involving sequential data, such as music recommendation. The LSTM network architecture consists of three parts, as shown in Figure 1, and each part performs an individual function.

Instead of passively processing information in a sequence, LSTMs excel at managing long-term dependencies through a series of "gates." These gates act like valves, controlling the flow of information within the LSTM cell.

- Forget Gate: This gate decides what information from the previous time step is no longer relevant and can be discarded. It essentially cleans up the cell's memory, focusing on the most crucial details.
- Input Gate: This gate determines what new information from the current input should be remembered. It acts like a filter, selecting valuable data to be added to the cell's internal state.
- Output Gate: Finally, the output gate controls what information from the current cell state is passed on to the next time step in the sequence. It essentially determines what the LSTM "remembers" to use for future predictions.

By working together, these gates enable LSTMs to effectively learn from long sequences and overcome the vanishing gradient problem that hinders traditional RNNs. Similarly to a layer of neurons in a traditional neural network, an LSTM unit combines these gates with a memory cell, creating a powerful tool for processing sequential data.

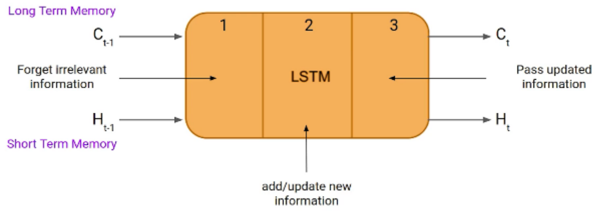
Fig. 1. LSTM network architecture



Just like a simple RNN, an LSTM also has a hidden state where H(t-1) represents the hidden state of the previous timestamp and Ht is the hidden state of the current timestamp. In addition to that, LSTM also has a cell state represented by C(t-1) and C(t) for the previous and current timestamps, respectively.

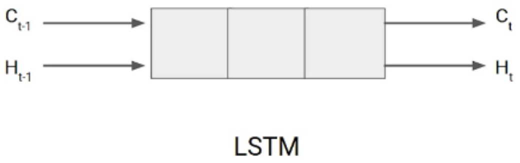
Here the hidden state is known as Short term memory, and the cell state is known as Long term memory. Refer to the following image.

Fig. 2. LSTM network architecture (hidden states)



It is interesting to note that the cell state carries the information along with all the timestamps.

Fig. 3. LSTM network architecture (overview)



C. GRU Architecture

GRUs, or Gated Recurrent Units, are a type of recurrent neural network (RNN) architecture like Long Short-Term Memory (LSTMs). Both are designed to tackle sequential data, where information unfolds over time. However, GRUs offer a simpler and more computationally efficient alternative to LSTMs.

The core distinction between GRUs and LSTMs lies in how they handle the memory state. LSTMs utilize a separate memory cell state updated by three gates: forget, input, and output. In contrast, GRUs replace the memory cell with a "candidate activation vector" and manage it using just two gates: reset and update.

- Reset Gate: This gate acts as a filter, determining how much of the previous hidden state (the network's memory) is no longer relevant and can be discarded.

- Update Gate: This gate controls how much of the newly created candidate activation vector, incorporating the current input and filtered past information, should be incorporated into the new hidden state.

By effectively managing information flow through these gates, GRUs can learn from long sequences and avoid the vanishing gradient problem that hinders traditional RNNs.

Like other RNN architectures, GRUs process data sequentially, updating their hidden state (internal memory) based on the current input and past information. At each step, the GRU computes a candidate activation vector that blends the current input with the previous hidden state. This candidate vector is then used to create a new hidden state for the next time step.

In essence, GRU cells maintain essential information throughout the network with the help of these two gates. The GRU architecture takes two inputs at each time step:

The previous hidden state (network's memory)
The current input data

These inputs are processed through the update and reset gates to derive the output for the current time step. Finally, a dense layer with SoftMax activation is applied to generate the final output and a new hidden state that is passed on to the next time step.

While like LSTMs in functionality, GRUs offer advantages in terms of simplicity and computational efficiency due to their fewer parameters. This makes them a compelling choice for tasks where resources are limited, or a simpler architecture is preferred.

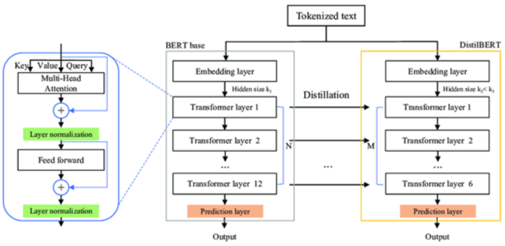
We will use both LSTM and GRU in our dataset and report the results in the results and conclusion section.

D. BERT/DISTILBERT Architecture

The BERT (Bidirectional Encoder Representations from Transformers) architecture has shown widespread success on natural language tasks, including text classification [3]. BERT, which consists of several transformer encoders, uses the self-attention mechanism to retain long-term dependencies in long sentences (up to 512 words) and represent contextual information from sentences and words [3]. This “understanding” is enabled by BERT’s ability to bidirectionally analyze word tokens during training; BERT tokens from both the left and right sides to learn contextual information from text [3]. Language context is learned by pre-training BERT’s self-attention mechanism on a large corpus of text, such as the BooksCorpus (800 million words) and English Wikipedia (2500 million words) datasets [10]. To use BERT for a downstream task like text classification, transfer learning is used to fine-tune the model’s parameters on a smaller dataset.

DistilBERT is a variant of BERT with 40% fewer parameters, 60% faster speed, and 97% of its language understanding capabilities [7]. This model compression was achieved using knowledge distillation, where a simpler student model learns to reproduce the outputs of a larger teacher model, in this case BERT [7]. The DistilBERT student architecture is similar to BERT, but with half the number of layers (see Figure 5) [7].

Fig. 4. DistilBert network architecture (overview) [7]



IV. Experimental setup and dataset

A. *Dataset*

The dataset used for this project is the Twitter–Self-Reported Temporally-Contextual Mental Health Diagnosis (T-STMHD) dataset [11]. The T-STMHD dataset comprises user data collected from Twitter to study mental health disorders. The dataset covers eight major mental health disorder classes, each corresponding to branches in the DSM-5 diagnostic manual. These classes include depressive disorders (depression, major depressive disorder, post-partum depression), trauma and stress-related disorders (post-traumatic stress disorder), neurodevelopmental disorder (attention-deficit/hyperactivity disorder), anxiety disorders, bipolar disorders, and obsessive-compulsive disorders. Additionally, a negative class of users least likely to exhibit any of these disorders is included as a control group for comparison. We only use the depression and control classes of users in our project.

Dataset Structure and Collection Period

Data collection involved scraping the entirety of public tweets on the Twitter platform and selecting users who self-reported a diagnosis for one of these disorders in a tweet – this is termed as the user's *anchor tweet*. Each user's data collection period spans four years, covering two years before and after the anchor tweet. This duration aligns with the DSM-5's suggested observation periods for the eight disorders. Furthermore, the dataset was extended to include a period during the COVID-19 pandemic to capture its impact on mental health.

Anchor Tweet Identification

Anchor tweets for were identified using a preliminary loose pattern search for a given disorder, focusing on tweets containing self-disclosure of diagnosis. The search pattern used took the format "diagnosed with <disorder name>". A combination of manual annotation and precise pattern matching was then employed to ensure accuracy, resulting in a high-precision dataset. This is one of the examples of their anchor tweet detection:

> *"I was diagnosed with depression, anxiety, ptsd at 16. I'm now 23 and still struggle with all. But I feel like theres something else going on up in my head, has been for well over a year, its different, I feel like I know it's not any of those things, but I'm not sure what."*

Control-user Class

The negative class of users in the control group could not be assembled as easily as the positive classes, as there is no equivalent to a disclosure of self-diagnosis from individuals without any (known) mental health disorders. This group was identified rather by random sampling of all Twitter users who were not matched as being part of one of the eight disorder groups, and then further pruning was performed using a set of lexicons that relate to mental health. This method was employed until the control class had as many users as the disorder class with the highest number of users, which was the depression group.

Users and User-Data

For each user selected for this dataset, the authors collected both information about the user's profile as well as the timeline tweets. Profile information includes a variety of attributes such as when the profile was created, the user's biography, and how many favourites, friends, and followers the user has. The attributes collected for the timeline tweets include the raw tweet text, the timestamp of when the tweet was posted, the source of the tweet (e.g. the web interface, a phone app), and how many likes, quotes, and replies the tweet received, among other information that is irrelevant to our project. The dataset includes a flag indicating if a tweet contains mental health discourse. Additionally, media attached to tweets is categorized, providing the dataset with multimodal capabilities.

Ethical Considerations

The dataset was collected from publicly available data on Twitter, without direct interaction with users. Care was taken to protect user privacy and ensure ethical use of the data. Discussions around ethical considerations for using Twitter data concluded that it can be ethically used for research purposes. Overall, the T-STMHD dataset provides a comprehensive resource for studying mental health disorders and their impact on social media users during both normal periods and the COVID-19 pandemic.

B. *Pre-Processing Steps*

After downloading and extracting the depression and control classes of T-STMHD, the data was pre-processed using a scheme inspired by Garcia-Noguez et al. [8]. Before any useful analysis or pattern extraction can be performed on a user's tweet timeline, there are a series of transformations that must be applied to the text of each tweet.

Text Cleaning

Some aspects of written language are not very pertinent to the author's intended message. The first pre-processing step to be performed on a tweet's text requires us to clean it up such that only English words remain. Regular expressions are utilized to achieve the following cleaning steps:

- Remove URLs, #hashtags, @mentions, retweet text, and other symbols and references to external content
- Correct misspellings and expand contractions
- Remove excess whitespace, punctuation, digits, and other special characters (leaving only Latin characters)
- Convert all remaining letters into lowercase
- Remove duplicate words and repeating characters
- Convert emoji and emoticons into a textual representation of what the pictogram represents

Word Reduction

Once a tweet's text has been cleaned in the aforementioned process, the next series of transformations are meant to extract only the words that are significant to the meaning of a tweet. There are two aspects to this stage.

The first step is to remove all *stop words* from the tweet text. Stop words are common English words filtered out from text for the purposes of natural language processing, as these words only serve grammatical functions (such as articles, pronouns, prepositions, etc.). Examples of stop words that are removed from the tweet text include “the”, “a”, “I”, “or”, and “and”.

The second step is to reduce all words to their root form via *lemmatization*. This allows words that are morphologically similar or derivationally related to be grouped together for analysis, as sentence subjects and tenses do not usually impact the meaning of the sentence. An example of this is the words “singer”, “singing”, “sings”, “sang”, and “sung”, which will all be reduced to the base lemma “sing”.

Word Vectorization

The final transformation that must be applied to tweet text before it is in a usable form is the process of *vectorizing* the text – taking as input the vector of each word still remaining after the previous transformations, and by the means of some yet-to-be-decided method, producing as output a vector of numbers that corresponds to the input.

One method that is often employed to convert textual data into numerical data is the *Bag-of-Words* approach, which simply involves counting how many times a word appears in a document, and taking the ratio of this with how many times that same word appears in an entire corpus of documents. Our project will use a more complex strategy called Word Embeddings, which is capable of capturing the context of a word in a tweet, semantic and syntactic similarity, its relation with other words, and other such nuances. This method actually relies on an underlying neural network of its own, but the design of such a network is outside of the scope of this project, so we have made use of the popular *Word2Vec* implementation provided in the Gensim library [11].

After all of these transformations have been applied to the text of each tweet, it is encoded as well-defined numerical data that is appropriate for input to a neural network.

V. Experimental results & analysis

A. *LSTM Training and Results*

Training Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models for depression prediction requires creating sequences of each user's tweets. This process involves several steps to prepare the data for the models. First, we preprocess the text of each tweet. This cleaning step removes irrelevant information such as URLs, hashtags, and special characters. Cleaned text allows the models to focus on the core meaning conveyed by the user's words. Next, we leverage the Word2Vec library to convert the cleaned text data into numerical vectors. Word2Vec is a powerful tool for generating word embeddings, which capture the semantic meaning of words. By understanding these embeddings, the models can analyze the context within a user's tweets and identify patterns that might be indicative of depression. Following text pre-processing and embedding generation, we create individual sequences for each user. These sequences are chronologically ordered lists containing a user's tweets, starting from their earliest posts. This chronological order allows the models to capture the potential evolution of language patterns and sentiment over time, which might be crucial for depression prediction. To ensure consistency in the training data, we limit the maximum number of tweets included in each sequence to 200. This capping prevents excessively long sequences from skewing the training process. The chosen value of 200 can be further refined through experimentation or based on findings from relevant research in the field. Finally, we select a subset of users for inclusion in the training data. In this initial exploration, we limited the user pool to 100. This number serves as a starting point, and future experiments can explore the impact of including a larger or smaller user base on model performance.

Table I details the distribution of tweet counts across the 100 users chosen for training. This table provides insights into the variety of tweet volumes within the selected user sample.

TABLE I. NUMBER OF USERS AND TWEETS

	Number of users	Number of tweets	disorder
depression	103	483,618	True
control	103	379,568	False

We constructed a Long Short-Term Memory (LSTM) model using the Keras library to analyze the generated tweet sequences and predict the likelihood of depression in users. The model architecture incorporates the following elements:

- **LSTM Layers**: The core of the model consists of two consecutive LSTM layers, each containing 128 units. These LSTM layers are adept at processing sequential data like tweet sequences and capturing long-term dependencies within the text. The number of units (128) determines the model's capacity to learn complex patterns from the data.
- **Dense Layers**: To make the final prediction, we employ two fully connected (dense) layers following the LSTM layers. Dense layers allow the model to learn non-linear relationships between the extracted features and the target variable (depression prediction).
- **Binary Cross-Entropy Loss**: As we are performing a binary classification task (predicting presence or absence of depression signs), the model utilizes the binary cross-entropy loss function. This function measures the difference between the predicted probabilities and the actual labels (depressed or not depressed). Minimizing this loss function during training guides the model to learn patterns that improve prediction accuracy.

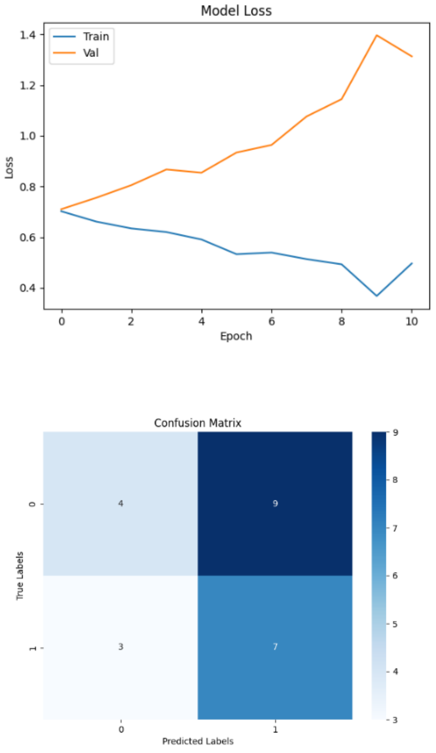
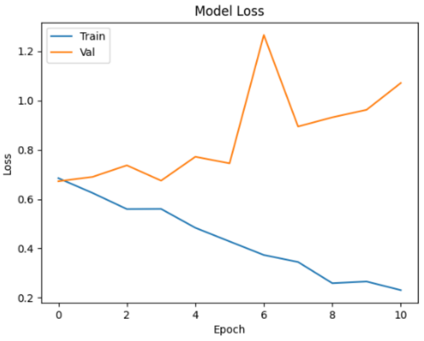


Fig. 5. LSTM results for 100 users and 200 sequences

As shown in this graph, the overfitting is occurring and the accuracy is 0.47 which is a terrible result for binary classification, there are two options for improving this accuracy: increasing number of users and increasing the maximum number of sequences. We also wanted to share the results for 400 users and 200 sequences:

TABLE II. NUMBER OF USERS AND TWEETS

	Number of users	Number of tweets	disorder
depression	408	1,466,422	True
control	408	1,752,881	False



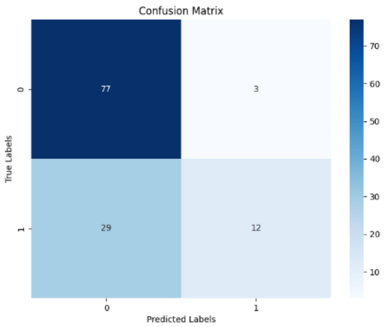


Fig. 6. LSTM results for 400 users and 200 sequences

As shown in these graphs, the accuracy of our model with 400 usesrs increased to 0.73 which is a good and acceptable accuracy. Now we want to increase the maximum number of sequences as well :

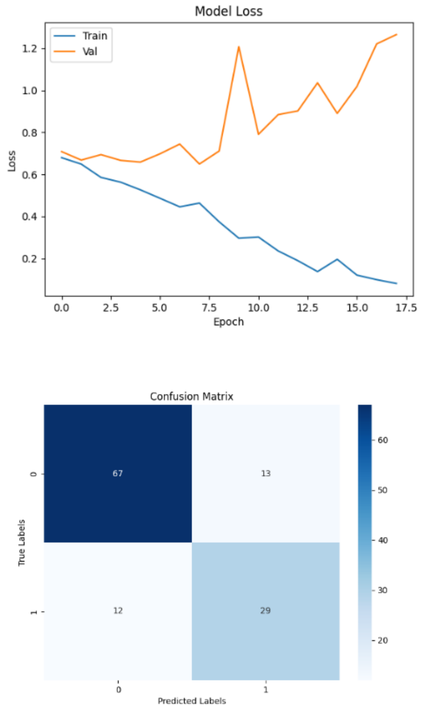


Fig. 7. LSTM results for 400 users and 300 sequences

The accuracy of this model with 400 users and 300 sequences is 0.79 and this the best accuracy that we can achieve.

B. GRU Training and Results

Transitioning from the LSTM architecture, a similar framework was adopted for the GRU (Gated Recurrent Unit) model. Sequences were generated for each user with a maximum length of 200 tokens, and padding was applied to sequences shorter than 200 tokens. The GRU model consisted of two GRU layers, each comprising 128 neurons, followed by two dense layers. Binary cross-entropy loss function was employed for training the model. With these foundational elements established, the subsequent discussion delves into the exploration of the GRU architecture.

Utilizing GRU cells offers superior performance compared to vanilla RNN cells due to their incorporation of two additional gates: the update gate and reset gate [2]. These gates provide the model with increased flexibility in processing input sequences by controlling the flow of information. Additionally, GRUs can create shortcut connections between text segments separated by arbitrary numbers of timesteps. This characteristic enables the model's memory to capture more long-term dependencies within the text [2]. Moreover, GRU cells exhibit computational efficiency when compared to their counterpart, LSTM (Long Short-Term Memory) cells [2].

The GRU model was implemented using the TensorFlow and Keras libraries to evaluate its efficacy in tweet classification. The model architecture was carefully designed to capture temporal dependencies within the tweet sequences. It comprised multiple layers, starting with a masking layer to handle variable-length input sequences effectively. The GRU layer served as the core component, enabling the model to retain important information from past states while processing each tweet. Additionally, dense layers were incorporated to facilitate nonlinear transformations and enhance the model's capacity to learn complex patterns.

During training, the model parameters were optimized using the Adam optimizer, a popular choice for its adaptive learning rate capabilities. To mitigate the risk of overfitting, early stopping was implemented as a regularization technique. This mechanism monitored the validation loss during training and halted the process if no improvement was observed over a specified number of epochs. By preventing the model from excessively fitting the training data, early stopping promoted generalization and enhanced the model's performance on unseen data.

Dataset Description: Two distinct datasets were utilized to evaluate the GRU model's performance in tweet classification. The first dataset comprised 100 users, each associated with 200 tweet sequences. This dataset provided a moderate-sized corpus for training and testing the model. In contrast, the second dataset featured a larger cohort of 400 users, with varying sequence lengths of either 200 or 300 tweets per user. This dataset presented a more extensive and diverse collection of tweets, allowing for a more comprehensive assessment of the model's capabilities across different data sizes and complexities.

By leveraging these datasets, the evaluation aimed to gauge the GRU model's ability to effectively discern between relevant and irrelevant tweets. The variation in dataset sizes and tweet sequences provided valuable insights into how the model's performance scales with the amount and structure of input data. This approach facilitated a robust evaluation framework that could inform future improvements and optimizations in tweet classification tasks.

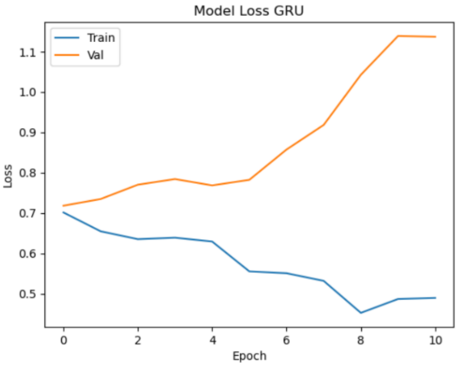


Fig. 8. GRU results for 100 users and 200 sequences

- Test Loss: 0.700
- Test Accuracy: 0.522

The GRU model achieved a relatively low accuracy of approximately 52% on the dataset comprising 100 users and 200 sequences. The test loss indicates the discrepancy between predicted and actual values, with a higher loss suggesting poorer model performance.

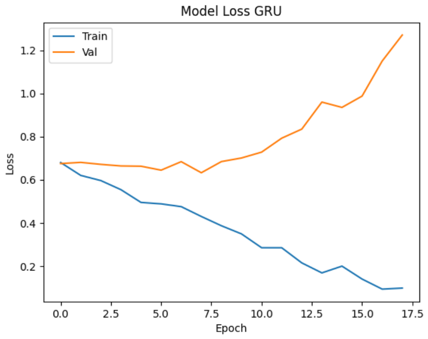


Fig. 9. GRU results for 400 users and 200 sequences

- Test Loss: 0.573
- Test Accuracy: 0.777

For the dataset with 400 users and 200 sequences, the GRU model exhibited significant improvement, achieving an accuracy of approximately 78%. This suggests that the model's performance is influenced by the size and complexity of the dataset, with larger datasets potentially providing more meaningful insights.

The GRU model's accuracy on the dataset with 400 users and 300 sequences was approximately 73%. Although the test loss metric was not provided, the accuracy indicates the model's capability to correctly classify tweets, with higher accuracy reflecting better performance.

C. Comparative Results Between LSTM and GRU

TABLE III. COMPARATIVE RESULTS

	100 users	400 users (200 - 300 sequences)
LSTM	0.47	0.73 - **0.79**
GRU	0.52	0.77 - 0.73

Table III presents a comparison of the performance achieved by the LSTM and GRU models on the depression prediction task. As the table suggests, the LSTM model generally yielded better results, particularly when trained with larger datasets containing more user sequences. This observation can be attributed, in part, to the more complex architecture of LSTMs compared to GRUs. LSTMs possess an additional gate (the forget gate) that allows them to control information flow more precisely within the network. This additional complexity may grant LSTMs an advantage in capturing intricate patterns within the user tweet sequences, potentially leading to more accurate depression predictions, especially with larger datasets.

D. *DistilBERT TRAINING AND RESULTS*

We fine-tuned a pretrained DistilBERT model from HuggingFace [13] on our 400-user dataset. Rather than taking 300 sequences, however, we concatenated the tweets for each user to a single body of text. By default, the token embedding dimension of DistilBERT is fixed to 512. Due to hardware constraints of an NVIDIA V100 GPU with 16 GB of RAM, we did not increase this token embedding dimension, and truncated any additional tokens. We also modified our tweet preprocessing scheme based on previous work showing BERT models have lower predictive power when stop words are removed and lemmatization is performed [14]. We compare DistilBERT test performance using our original text preprocessing scheme and our modified preprocessing scheme in Table IV. We trained our DistilBERT model for 5 epochs using the AdamW optimizer with a base learning rate of 8e-5, and weight decay of 0.01. For training and inference, we use a batch size of 32.

TABLE IV. COMPARATIVE RESULTS FOR DISTILBERT

Preprocessing Scheme	Test Accuracy (%)	Test Loss
Original	68.75	0.618
No stop word removal or lemmatization	72.50	0.639

On the test set, our DistilBERT model achieves an accuracy of 72.50%, and loss of 0.618 (see Figures 11 and 12). This test accuracy is worse than those of the GRU and LSTM models. While DistilBERT is a compressed version of BERT, it still has 66 million parameters [7], and likely requires a larger dataset of several thousand users to take advantage of contextual dependencies within each user's post history. In other words, given our small dataset of 400 users, DistilBERT may be overparameterized and thus more prone to overfitting than our more lightweight GRU and LSTM models. This discrepancy can also be explained by DistilBERT's token limit of 512, which limits the number of tweets that can be included per user, compared to the 200- and 300-long tweet sequences used for our GRU and LSTM models.

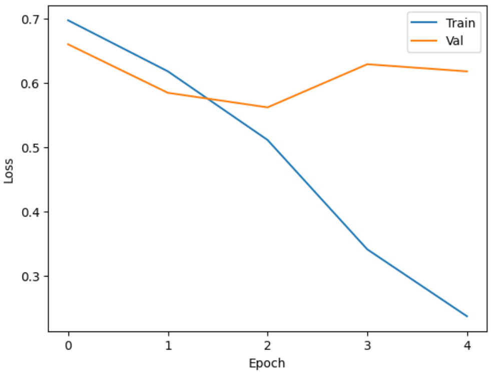


Fig. 10. DistilBERT results for 400 users

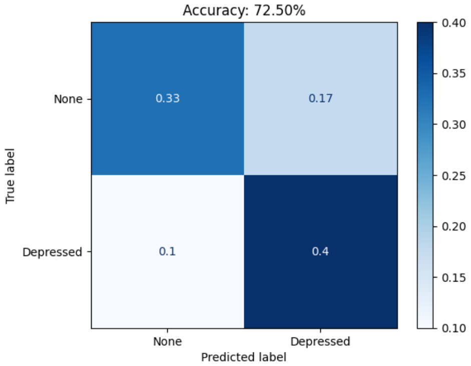


Fig. 11. DistilBERT results for 400 users

VI. CONCLUSION AND FUTURE WORKS

Using our LSTM model with 400 users and 300 sequences, we were able to predict depressive users with nearly 80% accuracy, even with our limited sample size. As evidenced by our GRU and LSTM results, reducing the dimensionality of the inputs by cleaning and lemmatizing the tweet text both reduced the training time and enabled the models to make more accurate associations between language use and mental disorders. However the opposite is true for our DistilBERT model, which required as little tweet preprocessing as possible to capture bidirectional dependencies between word tokens. While the T-STMHD has more features available for each tweet/user from which to learn patterns behind mental disorders, this woula hybrid network as the data is not appropriate for use with RNNs or BERT models. LSTMs were the most accurate model for predicting depressive users, outperforming than GRUs and the DistilBERT model on our limited data size, but a larger experiment would be needed to confirm this finding. Our theory to explain this is that LSTMs have more information gates and thus more trainable parameters to handle the complex dependencies between sentences as well as between entire tweets per user. On the other hand, DistilBERT has too many parameters or is limited by its token embedding dimension of 512, which prevents it from capturing as long of a tweet history as the other models. Ultimately, the ability of our models to consider temporal or contextual information is useful for classifying depressive users, as the timeline of when a user posted tweets can be indicative of depression symptoms as well.

VII. ACKNOWLEDGEMENTS

We would like to express our heartfelt gratitude to our instructor, Wenjing Zhang, at the University of Guelph, Canada. Her guidance, encouragement, and insightful feedback played a pivotal role in shaping the course of our research. Our sincere thanks also go to the University of Guelph for providing us with an enriching academic environment and valuable resources that facilitated our study. We extend appreciation to our fellow students and colleagues who contributed to discussions and shared insights that enhanced the quality of our work. Additionally, we want to acknowledge the support from the University of Guelph, which includes any relevant departments, for their role in fostering an environment conducive to research and learning.

VIII. REFERENCES

[1] S. Chen, Z. Zhang, M. Wu, and K. Zhu, "Detection of Multiple Mental Disorders from Social Media with Two-Stream Psychiatric Experts," in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore: Association for Computational Linguistics, 2023, pp. 9071–9084. doi: 10.18653/v1/2023.emnlp-main.562.

[2] Diveesh Singh and A. Wang, Detecting depression through Tweets - Stanford University, https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6879557.pdf (accessed Apr. 3, 2024).

[3] "Depression Classification From Tweets Using Small Deep Transfer Learning Language Models | IEEE Journals & Magazine | IEEE Xplore." Accessed: Apr. 02, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/9954391

[4] C. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text," Proceedings of the International AAAI Conference on Web and Social Media, vol. 8, no. 1, Art. no. 1, May 2014, doi: 10.1609/icwsm.v8i1.14550.

[5] S. H. Pedersen, K. B. Stage, A. Bertelsen, P. Grinsted, P. Kragh-Sørensen, and T. Sørensen, ''ICD-10 criteria for depression in general practice,'' J. Affect. Disorders, vol. 65, no. 2, pp. 191–194, Jul. 2001

[6] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations." arXiv, Feb. 08, 2020. doi: 10.48550/arXiv.1909.11942.

[7] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." arXiv, Feb. 29, 2020. doi: 10.48550/arXiv.1910.01108.

[8] L. R. García-Noguez, S. Tovar-Arriaga, W. J. Paredes-García, J. M. Ramos-Arreguín, and M. A. Aceves-Fernandez, "Automatic classification of depressive users on Twitter including temporal analysis," Netw Model Anal Health Inform Bioinforma, vol. 12, no. 1, pp. 1–13, Dec. 2023, doi: 10.1007/s13721-023-00434-1.

[9] "Depressive and Non-depressive Tweets Classification using a Sequential Deep Learning Model | IEEE Conference Publication | IEEE Xplore." Accessed: Apr. 02, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10083981

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv, May 24, 2019. Accessed: Apr. 15, 2024. [Online]. Available: http://arxiv.org/abs/1810.04805

[11] Suhavi, Singh, A., Singh, A. K., Shrivastava, S., Arora, U., Shah, R. R., & Kumaraguru, P. (2022). Twitter-STMHD: An Extensive User-Level Database of Multiple Mental Health Disorders. https://doi.org/10.5281/zenodo.6409736

[12] Řehůřek, R. Word2Vec Model. 2009. https://radimrehurek.com/gensim/auto_examples/tutorials/run_word2vec.html

[13] "DistilBERT." Accessed: Apr. 15, 2024. [Online]. Available: https://huggingface.co/docs/transformers/en/model_doc/distilbert

[14] C. R. Bass, B. Benefield, D. Horn, and R. Morones, "Increasing Robustness in Long Text Classifications Using Background Corpus Knowledge for Token Selection.," vol. 2, no. 3, 2019.