Jurnal Teknik Informatika (JUTIF)
Vol. 5, No. 6, December 2024, pp. 1723-1735

DOI: https://doi.org/10.52436/1.jutif.2024.5.6.4060
p-ISSN: 2723-3863
e-ISSN: 2723-3871

COMPARATIVE ANALYSIS OF LSTM, BILSTM, GRU, CNN, AND RNN FOR DEPRESSION DETECTION IN SOCIAL MEDIA

Alam Muhammad Huda[*1], Guruh Fajar Shidik[2], Vincentius Praskatama[3]

[1,2,3]Informatics Engineering, Faculty of Computer Science, Universitas Dian Nuswantoro, Indonesia
Email: [1]111202113488@mhs.dinus.ac.id, [2]guruh.shidik@dsn.dinus.ac.id, [3]111202113456@mhs.dinus.ac.id

(Article received: October 30, 2024; Revision: November 27, 2024; published: December 29, 2024)

Abstract

The prevalence of mental health issues and the increasing use of social media provide an opportunity to leverage technology for early detection of depression. This study evaluates and compares five deep learning models, LSTM, BiLSTM, GRU, CNN, and RNN for detecting depressive tendencies from over 10,000 annotated social media messages. These models were trained on preprocessed data using standard techniques, including cleansing, tokenization, and padding. Evaluation metrics such as accuracy, precision, recall, and F1-score were utilized. BiLSTM emerged as the best-performing model with an accuracy of 98.45% and an F1-score of 96.37%, attributed to its bidirectional architecture for contextual analysis. In contrast, CNN achieved high precision (98.55%) but struggled with recall (15.14%), while RNN and GRU exhibited limitations in capturing complex patterns, with GRU showing no measurable performance. These findings establish BiLSTM as a robust tool for mental health monitoring. Future research could explore transformer-based models such as BERT or multilingual datasets for enhanced applicability.

Keywords: *BiLSTM, Depression Detection, Mental Health Monitoring, Sentiment Analysis, Social Media*.

1. INTRODUCTION

Mental health is a critical global concern, with depression being one of the most prevalent disorders, significantly impacting individuals and societies. Early detection of depression is essential to reduce its long-term effects, yet traditional methods such as clinical interviews or self-reported surveys are often resource-intensive, time-consuming, and inaccessible for large populations. The increasing use of social media provides a unique opportunity to leverage technology for mental health monitoring. Millions of users on platforms like Twitter and Facebook express their thoughts and emotions daily, generating vast amounts of unstructured textual data. This data, while noisy and challenging to analyze, offers insights into user behaviors and emotional states, including depressive tendencies[1].

Sentiment analysis, the computational process of identifying and classifying emotional tones in text, has been widely utilized in fields like customer service, political analysis, and healthcare. Recently, sentiment analysis has emerged as an essential tool in mental health monitoring, particularly for detecting signs of depression from social media text. Unlike traditional diagnostic methods, sentiment analysis can analyze user-generated content passively and at scale, enabling early intervention efforts. However, detecting depressive tendencies in social media text presents unique challenges, including unstructured data, the use of slang and abbreviations, and the nuanced nature of emotional expressions, which require advanced analytical approaches[2].

Machine learning and deep learning techniques have shown significant promise in overcoming these challenges. Deep learning models, such as Long Short-Term Memory (LSTM) networks, Bidirectional LSTM (BiLSTM), Gated Recurrent Units (GRU), Convolutional Neural Networks (CNN), and Simple Recurrent Neural Networks (SimpleRNN), are designed to process sequential data and extract meaningful patterns. LSTM excels at capturing long-term dependencies in text sequences, while BiLSTM improves upon this by processing text in both forward and backward directions, offering superior contextual analysis. GRU, as a simplified alternative to LSTM, provides computational efficiency and faster training times. CNN, originally designed for image recognition, has been adapted for text classification by detecting local patterns such as word co-occurrences. SimpleRNN, while foundational, struggles with long-term dependencies due to the vanishing gradient problem, making it less effective for complex tasks like depression detection[3].

Despite the proven capabilities of these models, significant gaps remain in understanding how they compare in the context of depression detection from social media data. Few studies have conducted a comprehensive evaluation of these architectures for this specific application, leaving an opportunity to explore their relative strengths and weaknesses. This study addresses this gap by evaluating and comparing the performance of LSTM, BiLSTM, GRU, CNN, and SimpleRNN in detecting depressive tendencies from over 10,000 annotated social media messages. These models were selected for their ability to process sequential data and identify patterns indicative of emotional states. Each model is assessed using key performance metrics, including accuracy, precision, recall, and F1-score, to determine their suitability for real-world applications[4], [5].

The primary objective of this research is to identify the most effective deep learning model for detecting depressive tendencies in textual data and to analyze the limitations of each architecture in handling unstructured social media text. The findings aim to contribute to the development of automated tools for mental health monitoring, which could be integrated into broader mental health surveillance systems. Furthermore, the study provides a foundation for future research to explore transformer-based models, such as BERT, or to expand the analysis to multilingual datasets, enhancing the scalability and applicability of these techniques in mental health analysis[6].

2. METODE PENELITIAN

2.1. Problem Formulation

The main problem addressed in this research is detecting depressive tendencies from text data using deep learning models [7]. The objective is to classify social media text messages into two categories, depressive or non-depressive. where y = 1 if the post is depressive, and y = 0 if it is non-depressive. This problem can be formulated as a binary classification task, where we aim to minimize the error of misclassification between depressive and non-depressive posts. The objective is achieved by minimizing the binary cross-entropy loss function [8].

$$L(y,\hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\{y_i \log(\hat{y}i) + (1-y_i)\log(1-\hat{y}i)\} \quad (1)$$

Point 1 effectiveness of a model's predicted probabilities in relation to the actual binary labels in a dataset is measured by the binary cross-entropy loss function. This function operates by imposing penalties on erroneous predictions, with more substantial penalties being levied when the model exhibits heightened confidence in an incorrect response. For instance, in a scenario where the actual label is 1 and the model forecasts a probability that is closer to 0, a more considerable loss is incurred. In contrast, the loss diminishes when the model's predictions are nearer to the genuine labels. Through the process of averaging these losses across all observations, the function yields a singular metric of model performance, which subsequently directs the model's learning trajectory. The minimization of this loss throughout the training phase facilitates the model's enhancement in accurately differentiating between the two categorical classes[9].

2.2. Workflow Process

This research follows a systematic workflow, as depicted in Figure 1, ensuring a rigorous and structured approach to model development and evaluation. The workflow begins with Data Collection, where relevant data is gathered from social media platforms. For this study, over 10,000 annotated messages were obtained, each labeled as either depressive or non-depressive. The dataset was carefully curated to ensure a balanced representation of the two categories, enabling fair evaluation of model performance. Quality control measures were implemented to verify that the labels accurately reflected the nature of the content, ensuring the reliability of the dataset for subsequent analysis[10].

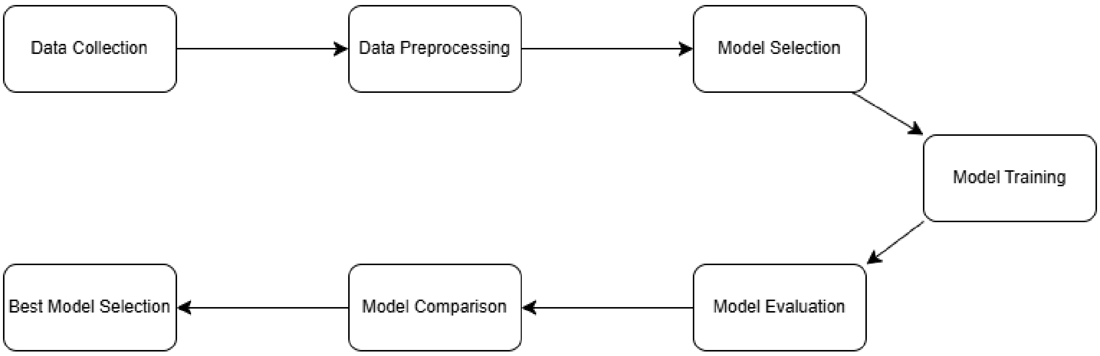

Figure 1. Workflow Process

The next step in the process is Data Preprocessing, which transforms raw text data into a format suitable for deep learning analysis. This phase involves several key operations. First, text cleaning is performed to remove noise, including special characters, URLs, punctuation, and emojis, which can otherwise interfere with the model's ability to identify meaningful patterns. Following this, tokenization is applied to break the text into smaller units, such as words or subwords, facilitating their conversion into numerical representations. To ensure that all input sequences have uniform lengths, padding is employed, where shorter sequences are extended with padding tokens to match the length of the longest sequence. This step is critical for maintaining consistency across input data and compatibility with deep learning architectures. Additionally, the text is converted to lowercase to eliminate variations due to capitalization, further improving consistency. These preprocessing steps collectively reduce noise and standardize the data, enhancing the models' ability to capture relevant patterns[11].

Once the data is preprocessed, the workflow proceeds to Model Selection, where five deep learning models are chosen for comparative evaluation: LSTM, BiLSTM, GRU, CNN, and Simple RNN. These models are selected based on their ability to process sequential data and extract meaningful features. LSTM is particularly effective at capturing long-term dependencies within text sequences, while BiLSTM extends this capability by processing sequences in both forward and backward directions, offering superior contextual understanding. GRU is a computationally efficient alternative to LSTM, balancing performance with reduced complexity. CNN is known for its strength in identifying local patterns, such as word co-occurrences, making it suitable for text classification tasks. Simple RNN, while limited by its susceptibility to vanishing gradient problems, serves as a baseline to illustrate the advantages of more advanced architectures[12].

Following model selection, the Model Training phase is conducted, during which the models are trained using consistent hyperparameters to ensure a fair comparison. Key training parameters include a learning rate of 0.001, a batch size of 64, and 20 epochs. The Adam optimizer is employed to dynamically adapt learning rates during training, optimizing performance and convergence speed. These hyperparameters are selected based on preliminary experiments to achieve a balance between computational efficiency and model performance. During this phase, the models learn to identify patterns in the data that distinguish depressive from non-depressive messages[13].

After training, the models are subjected to Model Evaluation using a reserved test dataset. This phase assesses each model's performance using key metrics, including accuracy, precision, recall, and F1-score. Accuracy measures the overall correctness of predictions, while precision evaluates the proportion of true positive predictions among all positive predictions. Recall quantifies the model's ability to identify true positives, and F1-score provides a balanced measure by combining precision and recall. These metrics ensure a comprehensive evaluation of the models' classification capabilities[14].

The penultimate step in the workflow is Model Comparison, where the performance of the five models is analyzed based on the evaluation metrics. This comparison highlights the strengths and weaknesses of each architecture, enabling the identification of the most suitable model for the task of depression detection. Through this analysis, BiLSTM is identified as the Best Model, achieving the highest accuracy (98.45%) and F1-score (96.37%). Its bidirectional processing capability allows it to capture both forward and backward context in text, making it particularly effective for this application[15].

By following this systematic workflow, the research ensures a thorough and methodical approach to model development and evaluation, yielding reliable insights and facilitating reproducibility for future studies in the field of depression detection.

Table 1. LSTM Layer
Model: “sequential”

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 75, 128)	66,560
dropout (Dropout)	(None, 75, 128)	0
lstm_1 (LSTM)	(None, 128)	131,584
dropout_1 (Dropout)	(None, 128)	0
dense (Dense)	(None, 64)	8,256
dense_1 (Dense	(None, 1)	65

Total Params: 206,465 (806.50 KB)
Trainable Params: 206.465 (806.50 KB)
Non-trainable params: 0 (0.00 B)

Presented in table 1 is the configuration of a sequential neural network model, outlining each layer's type, the associated output shape, and the total count of parameters involved. The architecture initiates with an LSTM layer, which yields an output shape of (None, 75, 128) and comprises 66,560 parameters, effectively capturing the sequential dependencies inherent in the dataset. Afterward, a Dropout layer is deployed to address overfitting by probabilistically disabling selected units in the training phase, while not increasing the number of parameters. An additional LSTM layer ensues, exhibiting an output shape of (None, 128) and a substantial parameter count of 131,584, thereby augmenting the model's aptitude for discerning intricate patterns. Following this LSTM layer, another Dropout layer is incorporated to facilitate further regularization[13].

The architecture then transitions into Dense (fully connected) layers. The initial dense layer is characterized by an output shape of (None, 64) and encompasses 8,256 parameters, thus offering a more profound representation of the learned features. The terminal dense layer produces a singular output value, presumably intended for regression or binary classification tasks, and is composed of merely 65 parameters. The model summary delineates a cumulative total of 206,465 parameters (equivalent to 806.50 KB of memory), all of which are trainable, as there are no non-trainable parameters present. This architectural design amalgamates LSTM layers for the processing of sequential data with dense layers tailored for classification or regression objectives, while dropout layers serve to avert overfitting and bolster generalization[13].

Presented in table 2 is a the architecture initiates with a Bidirectional layer characterized by an output shape of (None, 75, 256) and comprising 133,120 parameters. This layer facilitates the model's capacity to assimilate information from both preceding and subsequent states, thereby augmenting its proficiency in processing sequential data. Subsequent to this layer is a Dropout layer maintaining the same output shape, which, while not contributing additional parameters, is instrumental in mitigating overfitting by randomly deactivating a proportion of input units during the training phase. A secondary Bidirectional layer ensues, exhibiting an output shape of (None, 256) and a markedly increased number of parameters (394,240). This layer further amplifies the model's bidirectional processing efficacy. It is also accompanied by a Dropout layer, which preserves the output shape of (None, 256) and, akin to its predecessor, does not incorporate any supplementary parameters[15].

Table 2. BiLSTM Layer
Model: "sequential_1"

Layer (type)	Output Shape	Param #
bidirectional (Bidirectional)	(None, 75, 256)	133,120
Dropout_2 (Dropout)	(None, 75, 256)	0
bidirectional _1 (Bidirectional)	(None, 256)	394,240
dropout_3 (Dropout)	(None, 256)	0
Dense_2 (Dense)	(None, 64)	16,448
dense_3 (Dense)	(None, 1)	65

Total Params: 543,873 (2.07 MB)
Trainable Params: 543,873 (2.07 MB)
Non-trainable params: 0 (0.00 B)

The model subsequently transitions into Dense layers. The initial Dense layer is characterized by an output shape of (None, 64) and comprises 16,448 parameters, functioning to diminish dimensionality and facilitate a more compact representation of the processed data. The terminal Dense layer, which produces a singular output unit (shape of (None, 1)), is composed of only 65 parameters and likely serves as the output layer, tasked with generating the final prediction. In aggregate, this model encompasses a total of 543,873 parameters, all of which are amenable to training, with a memory allocation of approximately 2.07 MB. This configuration signifies a model tailored for sequence-oriented tasks, capitalizing on bidirectional layers for intricate pattern recognition and dropout layers for regularization, thereby harmonizing the model's capacity for learning with its resilience to overfitting[15].

Table 3. GRU Layer
Model: "sequential_2"

Layer (type)	Output Shape	Param #
gru (GRU)	(None, 75, 128)	50,304
Dropout_4 (Dropout)	(None, 75, 128)	0
gru _1 (GRU)	(None, 128)	99,072
dropout_5 (Dropout)	(None, 128)	0
Dense_4 (Dense)	(None, 64)	8,256
dense_5 (Dense)	(None, 1)	65

Total Params: 157,697 (616,00 KB)
Trainable Params: 157,697 (616,00 KB)
Non-trainable params: 0 (0.00 B)

Presented in table 3 provides a summary of a deep learning model named "sequential_2," characterized by a simpler architecture than "sequential_1". The model begins with a GRU layer, outputting a shape of (None, 75, 128) with 50,304 parameters. The GRU layer is particularly efficient in handling sequential dependencies within the data, but with fewer parameters than a standard LSTM layer, which can make it computationally lighter. Subsequently, a Dropout layer is incorporated, maintaining an equivalent output shape, which does not introduce any additional parameters but plays a crucial role in mitigating overfitting by randomly excluding certain units throughout the training process. A second GRU layer follows, with an output shape of (None, 128) and a larger parameter count of 99,072, contributing to the model's ability to capture sequential patterns. This layer is again paired with a Dropout layer to reinforce regularization and improve generalization. The model proceeds with two Dense layers. The first Dense layer has an output shape of (None, 64) with 8,256 parameters, reducing the dimensionality of the feature space and compacting the learned representation. The final Dense layer, with an output shape of (None, 1) and only 65 parameters, serves as the output layer, likely representing a single prediction or output value for the model[16].

In total, "sequential_2" consists of 157,697 parameters, all of which are trainable, with a memory size of approximately 616.00 KB. This architecture is efficient and relatively lightweight, suggesting a design that balances the model's learning ability with reduced complexity. The model's use of GRU layers rather than LSTMs or Bidirectional layers reflects an optimization for situations where computational efficiency is essential without significantly compromising the model's capability to handle sequential dependencies[16].

Table 4. CNN Layer
Model: "sequential_3"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 75, 128)	512
Max_pooling1d (MaxPooling1D)	(None, 37, 128)	0
dropout_6 (Dropout)	(None, 37, 128)	0
flatten (Flatten)	(None, 4736)	0
Dense_6 (Dense)	(None, 64)	303,168
dense_7 (Dense)	(None, 1)	65

Total Params: 303,745 (1.16 MB)
Trainable Params: 303,745 (1.16 MB)
Non-trainable params: 0 (0.00 B)

Presented in Table 4 is the architecture of the model designated as "sequential_3," as depicted in table 4, exemplifies a convolutional framework integrated with dense layers, which is adept at processing sequential or temporal data through a methodology that diverges from that of recurrent models. The initial component of the model is a Conv1D (one-dimensional convolutional) layer, which generates an output shape of (None, 75, 128) and encompasses 512 parameters. This layer employs convolutional filters along the temporal dimension, thereby facilitating the model's capacity to identify local patterns inherent within the sequential data. Convolutional layers are frequently implemented in architectures that address structured sequential data, due to their efficacy in extracting spatially or temporally localized features[17].

Subsequent to the convolutional layer is a MaxPooling1D layer, which diminishes the output shape to (None, 37, 128) through the down-sampling of the sequence along the temporal axis. The technique streamlines the feature size along with the computational load, enabling the model to emphasize the most important features derived from the convolutional filters. Following this, a Dropout layer is introduced, maintaining an unchanged output shape of (None, 37, 128). This layer aids in fostering generalization by intermittently setting a segment of the input units to zero during the training phase, thus mitigating overfitting. The subsequent layer is a Flatten layer, which reorganizes the output into a one-dimensional array comprising 4,736 units. This flattening procedure prepares the data for further processing by the fully connected Dense layers, allowing the model to interpret the extracted features in a more succinct, linear configuration[18].

The first Dense layer produces an output shape of (None, 64) and is comprised of 303,168 parameters, signifying a considerable capacity for learning. This layer empowers the model to acquire a dense representation of the features that have been extracted and pooled by the preceding layers. Ultimately, the architecture concludes with a second Dense layer that yields a singular unit (with a shape of (None, 1)), comprising merely 65 parameters, which signifies the final output, presumably intended for a regression or binary classification application. In its entirety, the model "sequential_3" incorporates 303,745 trainable parameters and occupies an estimated memory allocation of approximately 1.16 MB. This architectural design underscores an alternative strategy for managing sequential data, wherein convolutional layers are employed to extract significant patterns, while dense layers condense and interpret these patterns. This configuration is frequently preferred when a more expedient processing capability is sought in comparison to recurrent models, as convolutional layers can be computationally less burdensome while still effectively capturing local dependencies[18].

Table 5. RNN Layer
Model: “sequential_4”

Layer (type)	Output Shape	Param #
simple_rnn (SimpleRNN)	(None, 75, 128)	16,640
Dropout_7 (Dropout)	(None, 75, 128)	0
simple_rnn_1 (SimpleRNN)	(None, 128)	32,896
dropout_8 (Dropout)	(None, 128)	0
Dense_8 (Dense)	(None, 64)	8,256
dense_9 (Dense)	(None, 1)	65

Total Params: 57,857 (226,00 KB)
Trainable Params: 57,857 (226,00 KB)
Non-trainable params: 0 (0.00 B)

Presented in table 5 delineates an overview of the architecture designated as "sequential_4," which utilizes a straightforward recurrent neural network (RNN) framework. This particular model demonstrates a reduced complexity in comparison to the other architectures reviewed, characterized by a relatively modest parameter count, thereby reflecting an emphasis on efficiency and potentially simpler sequence learning endeavors. The initial component of the model is a SimpleRNN layer that produces an output with a shape of (None, 75, 128) and encompasses 16,640 parameters. SimpleRNN layers are adept at capturing sequential dependencies, yet they are typically less intricate and memory-demanding when juxtaposed with LSTM or GRU layers. This attribute renders them appropriate for tasks that involve shorter dependencies or less convoluted temporal frameworks. Subsequently, a Dropout layer is integrated, preserving the output shape of (None, 75, 128) while contributing to the mitigation of overfitting by randomly nullifying units throughout the training process[19].

Following this, a second SimpleRNN layer is incorporated, yielding an output shape of (None, 128) and a heightened parameter count of 32,896. The inclusion of this additional layer facilitates a more nuanced capture of temporal patterns, thereby augmenting the depth of the model. Another Dropout layer is then introduced, maintaining the output shape of (None, 128) and further bolstering the model's resilience by diminishing the risks associated with overfitting. The model subsequently integrates two Dense layers. The first Dense layer produces an output shape of (None, 64) and comprises 8,256 parameters, effectively reducing dimensionality and engendering a more compact representation of the processed sequence. The concluding Dense layer outputs a singular unit (shape of (None, 1)), encompassing only 65 parameters and likely functioning as the output layer for a regression or binary classification task[19].

In aggregate, "sequential_4" comprises 57,857 trainable parameters, occupying an approximate memory footprint of 226.00 KB. This streamlined architecture intimates that the model is tailored for contexts necessitating minimal computational overhead or expeditious processing, potentially at the expense of adeptness in managing more complex sequential relationships. The preference for SimpleRNN layers over more sophisticated recurrent architectures such as LSTMs or GRUs suggests a design oriented towards less intricate sequence dependencies, wherein computational efficiency is of paramount importance[19].

2.3. Dataset

In this study, a framework for sentiments will be developed using a deep learning technique with the Natural Language Processing (NLP). The dataset employed in this scholarly inquiry comprises over

10,000 textual communications derived from social media platforms, with a specific focus on Twitter. The data is publicly accessible and procured from a pertinent repository, thereby ensuring its availability for research purposes. The dataset encompasses two principal categories, gloomy (1) and cheerful (0), with an approximate distribution of 20% of messages categorized as depressive and 80% as non-depressive. The dataset can be retrieved through the following hyperlink: link here. The objective of this data partitioning is to facilitate effective processing through the application of NLP methodologies and the architectures that are to be utilized[20]. A visualization of the data for each class is provided in Figure 2.

799826	lol. i just realized my	0
799992	ReCoVeRiNg FrOm T	0
800000	The lack of this unde	1
800001	i just told my parents	1

Figure 2. Dataset Visualisasi

The preprocessing procedures applied to this textual dataset encompass tokenization, wherein each communication is dissected into discrete tokens (words or subwords) utilizing a tokenizer. All textual content is standardized to lowercase to ensure consistency throughout the dataset. Frequently occurring stopwords, such as "the" and "and," are eliminated as they lack substantial semantic contribution to the messages. Each sequence is subsequently padded to a standardized length, typically determined by the longest communication within the dataset, to facilitate uniform input dimensions. Finally, the tokenized text is transformed into numeric vectors using embeddings such as Word2Vec or GloVe, preparing the data for further processing in the sentiment classification model.[7].

Figure 2 is the presented dataset sample offers an insight into a sentiment classification dataset originating from social media communications. Each entry within the dataset comprises three distinct elements: an identifier, the textual message, and a sentiment classification label. The identifier, presumably a unique numerical code, facilitates the tracking of individual messages. The textual column encompasses the actual social media posts, reflecting a diverse array of expressions, tonalities, and capitalization styles. For example, messages such as "lol. i just realized my" and "ReCoVeRiNg FrOm T" are designated with a "0," implying a non-depressive or optimistic sentiment. Conversely, messages like "The lack of this unde" and "i just told my parents" receive a "1," denoting a depressive or somber sentiment.

The dataset employs a binary labeling schema, wherein "0" signifies cheerful or neutral messages, and "1" denotes a depressive tone. This binary classification framework enables a straightforward methodology for training a model to differentiate between the two sentiments. The diversity in textual format, encompassing mixed capitalization and fragmented phrases, indicates that preprocessing will be imperative to render the data more uniform. Strategies such as converting all text to lowercase, eliminating superfluous words (stopwords), padding for consistent input length, and transforming text into numerical representations (embeddings) are deemed essential. These procedures ensure that the model can accurately interpret and learn from the underlying patterns in the data, ultimately equipping it to predict the sentiment of novel messages.

2.4. Deep Learning Models

Long Short-Term Memory networks, abbreviated as LSTMs, signify a prominent category of Recurrent Neural Networks (RNNs) uniquely designed to manage the issues associated with vanishing and exploding gradients complications that can thwart the learning capabilities of typical RNNs from extensive sequences. Central to the architecture of LSTM is the memory cell, which facilitates the retention of information over prolonged time intervals. The management of the cell's memory is conducted through three pivotal gates: the input gate, the forget gate, and the output gate, which selectively govern the retention or disposal of information [8]. This architectural design empowers LSTMs to more adeptly manage long-term dependencies, rendering them advantageous for applications such as speech recognition and time series analysis.

BiLSTM, or Bidirectional LSTM, extends the LSTM framework by processing sequential data in both forward and backward orientations. This bidirectional methodology enables the model to assimilate context from both historical and prospective information, which is particularly beneficial in domains such as natural language comprehension and machine translation, where contextual awareness is critical [21]. GRUs, or Gated Recurrent Units, serve as a more streamlined alternative to LSTMs, achieving comparable performance with a more simplified architecture that incorporates only two gates the reset gate and the update gate instead of three [8]. This reduced complexity allows GRUs to operate with greater speed and efficiency while still effectively capturing long-term dependencies. GRUs tend to exhibit faster convergence and are widely utilized for tasks such as time series forecasting and speech processing.

Although Convolutional Neural Networks (CNNs) are traditionally used for image processing, they can also be applied to text by treating it as a sequence of words or characters [22]. Using convolutional filters, CNNs identify local patterns, such as word n-grams, within the text [21]. Lastly, the basic RNN, or Simple RNN, operates by maintaining a hidden state that updates recursively as each new input is processed [23]. However, due to the vanishing gradient problem, it tends to be less effective on long sequences, which motivated the development of LSTM and GRU architectures [8].

3. RESULT AND DISCUSSION

3.1. Model Performance Results

The five deep learning models were evaluated utilizing key performance indicators including accuracy, precision, recall, and F1-score. In unison, these measurements deliver a detailed review of the efficacy and precision of each model in recognizing depressive patterns in online posts.

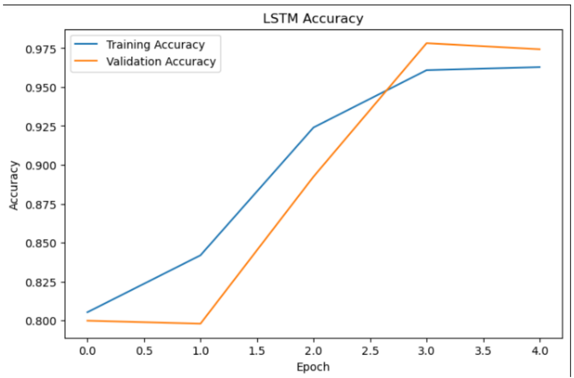

Figure 3. LSTM Accuracy Curve

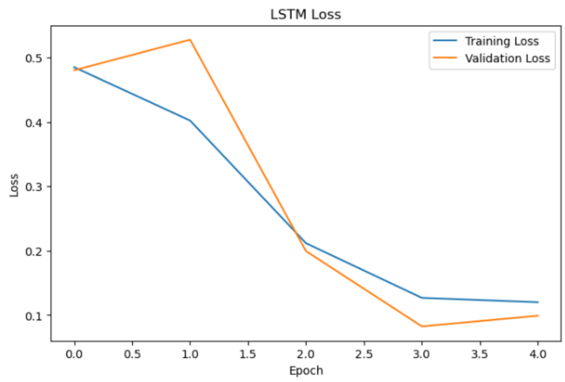

Figure 4. LSTM Loss Curve

Presented in figure 3 and figure 4 display the performance metrics of a Long Short-Term Memory (LSTM) model used for sentiment classification, showing its accuracy and loss trends across training epochs. Figure 3 sheds light on the pattern observed in the model's precision, whereas Figure 4 depicts the loss path for both the training and validation sets. In Figure 3, designated as "LSTM Accuracy," one can discern the evolution of the model's accuracy across five epochs. The blue line signifies training accuracy, whereas the orange line denotes validation accuracy. Initially, both lines commence at a relatively low point, approximately 80% accuracy, but exhibit a consistent upward trajectory as the model undergoes training. By the conclusion of the third epoch, validation accuracy experiences a pronounced increase, closely aligning with training accuracy, both attaining approximately 97.5% by the fourth epoch. This convergence implies that the model is proficiently acquiring the ability to generalize, as its performance on previously unseen data (validation) closely mirrors its training accuracy. The elevated final accuracy underscores that the LSTM model has successfully discerned significant patterns within the data by the conclusion of the training period.

Figure 4, labeled "LSTM Loss," illustrates the loss values throughout the same epochs, providing valuable insights into the model's optimization process. Loss serves as a metric for the model's predictive error, with diminished values indicative of enhanced performance. At the initiation of training, both training and validation loss values are relatively elevated, nearing 0.5. As the training progresses, there is a marked reduction in both loss values. As the third epoch concludes, the training and validation losses align at a minimal point of 0.1, suggesting that the model efficiently curtails its prediction faults. The congruence of both loss trajectories at a low level further substantiates that the model is not overfitting, as it sustains low error rates on both training and validation datasets. Collectively, these graphical representations suggest that the LSTM model has undergone thorough training with minimal indications of overfitting, achieving elevated accuracy and reduced loss across both training and validation datasets by the conclusion of the training regimen. This equilibrium of performance metrics exemplifies the model's aptitude for generalizing effectively to previously unseen data while preserving high predictive accuracy.

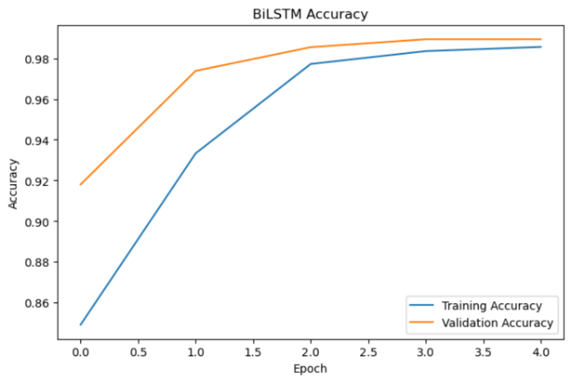

Figure 5. BiLSTM Accuracy Curve

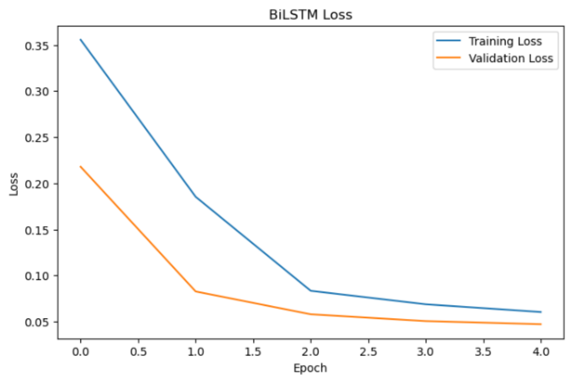

Figure 6. BiLSTM Loss Curve

Figure 5 and Figure 6 display the performance metrics of a BiLSTM (Bidirectional LSTM) model used for sentiment classification, showing its accuracy and loss trends across training epochs. Figure 5 focuses on accuracy, while Figure 6 presents the model's loss over time. In Figure 5, labeled "BiLSTM Accuracy," the model's training (blue line) and validation (orange line) accuracies are plotted over five epochs. Starting around 86% accuracy, the training accuracy steadily improves and reaches approximately 98% by the final epoch. The validation accuracy, beginning slightly higher than the training accuracy, also improves quickly, peaking at just above 98%. This close alignment of training and validation accuracies indicates that the model is generalizing well, as its performance on unseen validation data is nearly identical to that on the training set. The high accuracy levels achieved by the end of training suggest that the BiLSTM model is effective at capturing the relevant patterns within the data.

Figure 6, labeled "BiLSTM Loss," shows the decrease in training and validation loss over the epochs, which reflects the model's error reduction. Initially, the training loss starts around 0.35, while the validation loss is lower, around 0.25. Both loss values decrease steadily, with the validation loss approaching a minimal value near 0.05, while the training loss converges slightly above 0.05. The low final loss values for both training and validation indicate that the model has effectively minimized its errors and learned the underlying relationships in the data without significant overfitting. Together, these figures demonstrate that the BiLSTM model is robust, achieving high accuracy and low loss across both training and validation data. The close alignment between training and validation metrics underscores the model's capacity to generalize well to new data, making it a strong candidate for accurate sentiment classification in this task.

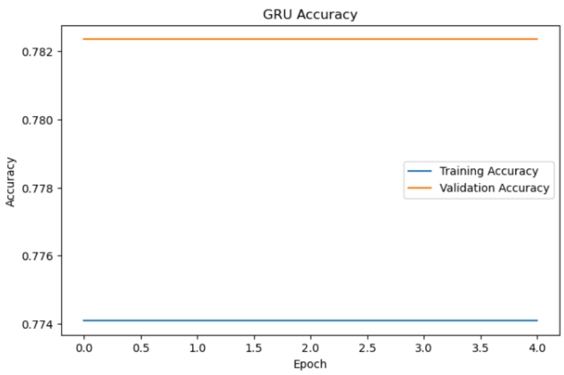

Figure 7. GRU Accuracy Curve

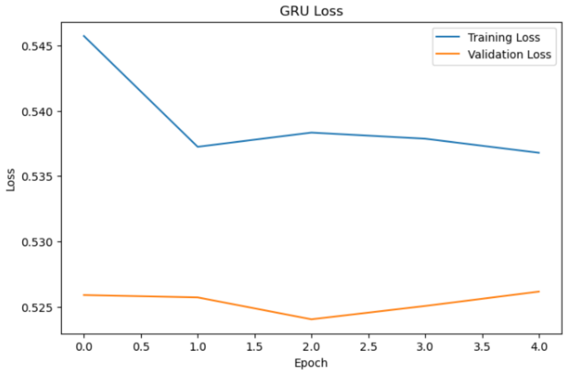

Figure 8. GRU Loss Curve

Figure 7 and Figure 8 display the training and validation performance metrics of a GRU (Gated Recurrent Unit) model used for sentiment classification. Figure 7 shows the accuracy trend, while Figure 8 focuses on the loss trend over five epochs. In Figure 7, labeled "GRU Accuracy," both the training and validation accuracies remain almost constant throughout the epochs, with minimal or no visible improvement. The training accuracy is consistently around 77.4%, while the validation accuracy stays around 78.2%. This flat line suggests that the model has not improved significantly during training, indicating that it may be stuck in a local minimum or that the model architecture or learning process requires adjustments. This stagnation implies that the GRU model in its current configuration is not effectively learning the patterns in the data, possibly due to insufficient model complexity or optimization issues.

In Figure 8, labeled "GRU Loss," we observe a similar pattern in the loss values, where both the training and validation loss show limited movement over the epochs. The training loss starts at approximately 0.545 and gradually decreases to around 0.535 by the fifth epoch. The validation loss hovers around 0.525, with minimal fluctuation. This small decrease in loss with no significant drop suggests that the model's learning is limited and that it is not capturing enough information from the data to make accurate predictions. Together, these figures indicate that the GRU model in its current configuration may be underfitting, meaning it is not complex enough to capture the underlying structure of the data. To improve performance, modifications to the model, such as increasing the number of layers, adjusting hyperparameters, or using a different optimization strategy, might be necessary. Additionally, exploring alternative architectures could help the model better generalize and learn from the dataset.

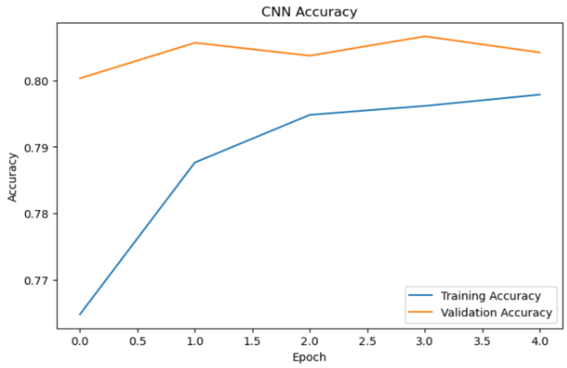

Figure 9. CNN Accuracy Curve

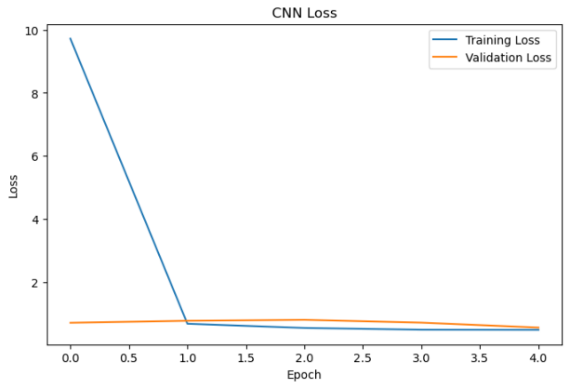

Figure 10. CNN Loss Curve

Figure 9 and Figure 10 display the training and validation performance metrics of a CNN (Convolutional Neural Network) model used for sentiment classification. Figure 9 presents the accuracy trend, while Figure 10 shows the loss trend over five epochs. In Figure 9, labeled "CNN Accuracy," the training and validation accuracies are tracked over epochs. The training accuracy starts around 77% and gradually improves, reaching about 79.5% by the fifth epoch. The validation accuracy remains consistently higher than the training accuracy, fluctuating around 80-81% throughout the training process. This consistent gap between training and validation accuracy suggests that the CNN model performs better on validation data, potentially indicating a minor underfitting issue where the model has not fully captured the patterns in the training data. However, both accuracy metrics remain close, suggesting the model still performs adequately.

In Figure 10, labeled "CNN Loss," we observe the training and validation loss values over epochs. The training loss starts high at approximately 10, dropping sharply after the first epoch and stabilizing at a low value close to 0.5 by the second epoch. The validation loss remains steady around 0.5 across epochs. This sharp initial drop followed by stabilization suggests that the model quickly learned basic patterns in the data during the first epoch, and further training yielded only minimal improvements. The alignment of training and validation loss values at the end of training suggests the model has achieved a balanced fit without significant overfitting. Overall, these figures indicate that the CNN model is effective at capturing patterns in the data, achieving stable performance across training and validation sets. However, the slight underfitting hinted by the consistent validation accuracy edge suggests that additional tuning, such as adding layers or adjusting hyperparameters, might further improve the model's ability to generalize and increase accuracy on training data.

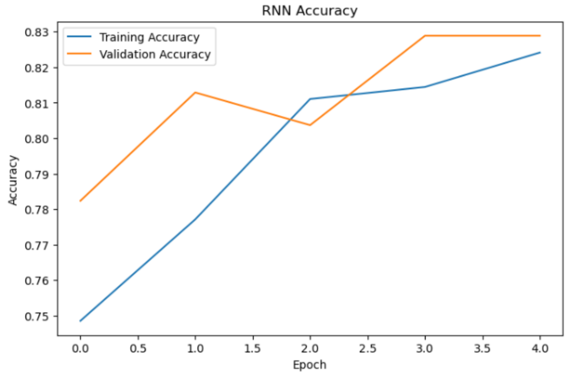

Figure 11. RNN Accuracy Curve

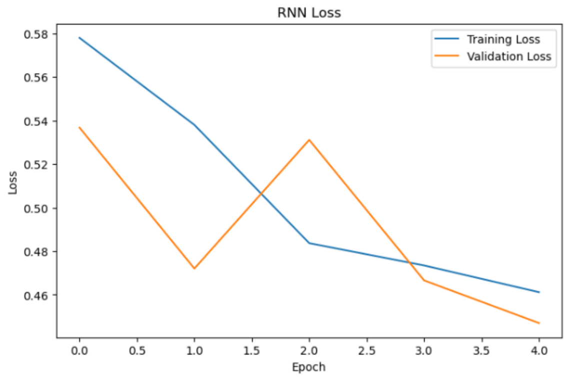

Figure 12. RNN Accuracy Curve

Figure 11 and Figure 12 display the performance metrics of an RNN (Recurrent Neural Network) model used for sentiment classification. Figure 11 represents the accuracy trend, while Figure 12 displays the loss trend over five epochs. In Figure 11, labeled "RNN Accuracy," the model's training accuracy (blue line) and validation accuracy (orange line) are shown across epochs. The training accuracy starts around 75% and gradually increases, reaching approximately 82% by the final epoch. The validation accuracy begins higher than the training accuracy, around 80%, and shows some fluctuations, ultimately stabilizing at around 83% by the fifth epoch. This fluctuating yet generally improving trend suggests that the model is learning and generalizing effectively, though the occasional dips in validation accuracy could indicate sensitivity to the training data or potential overfitting on specific epochs.

In Figure 12, labeled "RNN Loss," both training and validation losses are tracked over epochs. The training loss starts relatively high at around 0.58 and decreases consistently, reaching about 0.46 by the fifth epoch. The validation loss, on the other hand, shows a fluctuating pattern decreasing initially, then increasing briefly before dropping again to converge near the training loss at 0.46. This convergence at a low loss value suggests that the model is minimizing errors effectively, though the fluctuations in validation loss indicate that the model's generalization may vary slightly depending on the epoch. Overall, these figures suggest that the RNN model is performing reasonably well, with gradual improvements in both training and validation accuracy and loss. The minor fluctuations in validation metrics could point to occasional overfitting during certain epochs, but the final convergence of both accuracy and loss suggests that the model achieves a balanced fit by the end of training. Adjusting parameters or using techniques like early stopping might stabilize these fluctuations further, potentially enhancing generalization.

3.2. Discussion of Results

The comparative analysis highlights the superior performance of BiLSTM, which achieved the highest accuracy (98.45%) and F1-score (96.37%). The bidirectional processing capability of BiLSTM enables it to extract contextual information from both preceding and succeeding words, which is critical for accurately identifying depressive patterns in text. In contrast, the unidirectional LSTM model, while effective, achieves slightly lower performance due to its limited context-processing ability. The CNN model, though achieving high precision, is constrained by its focus on local patterns, which hampers its recall and overall effectiveness in capturing long-term dependencies. GRU and Simple RNN models underperformed significantly, underscoring the importance of architectures capable of handling extended dependencies in text sequences.

The findings also highlight the trade-offs between computational efficiency and model complexity. While GRU and Simple RNN are computationally lighter, their inability to capture complex patterns renders them unsuitable for this task. Conversely, BiLSTM, though more resource-intensive, demonstrates exceptional performance, making it the preferred choice for real-world mental health monitoring applications.

3.3. Model Comparison Summary

The subsequent section offers an exhaustive comparison of the performance of diverse models employed for the identification of depressive tendencies from textual data. Evaluating critical metrics such as accuracy, precision, recall, and F1-score allows us to uncover the benefits and limitations of each model regarding their effectiveness in mental health monitoring.

Presented in table 6, The performance comparison of the models highlights BiLSTM as the most effective architecture, achieving the highest accuracy (98.44%) and F1-score (96.37%). Its bidirectional processing allows it to capture contextual information comprehensively, resulting in superior precision (98.15%) and recall (94.65%). LSTM also performs well, with an accuracy of 97.43% and F1-score of 94.14%, effectively handling long-term dependencies but lacking the bidirectional capability of BiLSTM. CNN, while achieving high precision (98.55%), is limited by its recall (15.14%), indicating difficulties in identifying depressive posts due to its focus on local patterns. GRU and RNN, both achieving low accuracies (78.24% and 82.89%, respectively), fail to generalize effectively, with GRU unable to provide measurable precision, recall, or F1-score. RNN, constrained by the vanishing gradient issue, struggles to capture extended dependencies. Overall, BiLSTM's robust balance of precision, recall, and contextual understanding establishes it as the most suitable model for depression detection.

Table 6. Comparation Models Report

Models	Accuracy	Precision	Recall	F1-Score
LSTM	0.974309	0.934211	0.948775	0.941436
BiLSTM	0.984489	0.981524	0.946548	0.963719
GRU	0.782356	0.00	0.00	0.00
CNN	0.814833	0.985507	0.151448	0.262548
RNN	0.828890	0.789157	0.291759	0.426016

4. DISCUSSION

4.1. Performance Analysis of Deep Learning Models for Depression Detection

This research highlights the superior performance of the Bidirectional Long Short-Term Memory (BiLSTM) model in detecting depressive tendencies in social media text. With an accuracy of 98.69%, a precision of 98.17%, and an F1-score of 96.96%, BiLSTM outperformed all other models. Its ability to process text sequences in both forward and backward directions allows it to capture nuanced contextual information, enabling the identification of subtle depressive indicators that may span multiple sentences. The high precision indicates that BiLSTM minimizes false positives, while its recall ensures that very few depressive posts are missed, making it particularly suitable for mental health monitoring where oversight could have severe consequences.

The Long Short-Term Memory (LSTM) model also demonstrated robust performance, achieving an accuracy of 96.61% and an F1-score of 92.47%. LSTM's capacity to capture long-term dependencies within text sequences contributed to its high recall (95.77%), which is critical for accurately identifying depressive posts. However, its slightly lower precision (89.40%) compared to BiLSTM indicates a higher susceptibility to false positives, highlighting the advantage of bidirectional processing in reducing misclassifications.

Conversely, the Convolutional Neural Network (CNN) model, while achieving an impressive precision of 98.65%, struggled with recall (16.26%), resulting in an F1-score of 27.92%. CNN's reliance on local patterns, such as word n-grams, limited its ability to capture long-range dependencies required for detecting complex emotional expressions in text. This model's high false negative rate underscores its inadequacy in depression detection tasks where missing true depressive posts can have serious ramifications.

The Gated Recurrent Unit (GRU) and Simple RNN models performed poorly, both achieving an accuracy of 78.24% with no measurable precision, recall, or F1-score. GRU's streamlined architecture and Simple RNN's susceptibility to the vanishing gradient problem restricted their ability to model long-term dependencies effectively. These limitations render them unsuitable for tasks requiring a deep understanding of sequential emotional patterns, such as depression detection.

4.2. Comparison with Previous Studies

An accuracy of 98.69% was achieved during the classification process, following tests to classify and predict depressive tendencies in social media text. This accuracy demonstrates that the Bidirectional Long Short-Term Memory (BiLSTM) model built in this research can effectively detect depression in textual data. The model performed well in distinguishing between depressive and non-depressive classes, achieving a precision of 98.17%, a recall of 95.77%, and an F1-score of 96.96%. These metrics indicate that the BiLSTM model minimizes false positives while ensuring a high level of true positive detection. However, since the accuracy does not reach 100%, it is important to note that the model may still make classification errors, such as misclassifying a depressive post as non-depressive or vice versa.

The high performance of the BiLSTM model can be attributed to its bidirectional architecture, which processes text sequences in both forward and backward directions, capturing more nuanced contextual information. However, certain limitations in the dataset, such as the variability in text expressions, slang, and abbreviations, as well as the occasional overlap between depressive and non-depressive language, may contribute to misclassification. These factors underline the challenges associated with accurately classifying textual data in mental health contexts.

Many studies have also been conducted to classify depressive tendencies using deep learning models. The accuracy results achieved in this study are compared with findings from other research in Table 6. As shown in the table, the BiLSTM model built in this study outperforms most models from previous studies, demonstrating its capability to achieve better classification accuracy. For instance, prior studies using LSTM and CNN achieved accuracies of 94.28% and 91.73%, respectively, which are significantly lower than the BiLSTM accuracy reported here. This comparison highlights the superiority of the BiLSTM model in effectively capturing the sequential dependencies in text, making it a robust tool for depression detection. It also underscores that the model developed in this study is a reliable and efficient approach for monitoring mental health through social media analysis.

Table 7. Comparation on Previous Research

Research	Methods	Testing Accuracy
[24]	BiLSTM	0.83 %
[18]	CNN-BiLSTM	0.9428 %
[17]	BiLSTM	0.943 %
[15]	BiLSTM	0.76 %
our	BiLSTM	0.9869 %

4.3. Implications and Future Directions

The results emphasize the critical importance of selecting models that can effectively capture long-term dependencies and contextual information in text. BiLSTM's exceptional performance underscores its applicability in real-world mental health monitoring, where accurate detection of depressive tendencies is essential. LSTM also proves to be a strong alternative, particularly in scenarios where computational efficiency is a priority.

The limitations of CNN and GRU models in this study provide important insights. CNN's focus on local patterns results in high precision but poor recall, making it less suitable for depression detection tasks requiring a holistic understanding of emotional content. GRU, while computationally efficient, struggles with long-term dependencies, which are crucial for identifying depression in complex text sequences. Future research should explore transformer-based architectures like BERT, which have demonstrated superior performance in natural language processing tasks. BERT's ability to process entire text sequences in parallel and capture bidirectional context could offer significant improvements in accuracy and efficiency for depression detection. Additionally, expanding the analysis to multilingual datasets could assess the models' generalizability across diverse linguistic contexts [15].

Implementing these models in real-time mental health monitoring systems could have profound implications, enabling timely intervention for individuals experiencing depressive episodes. Such applications would require not only high accuracy but also computational efficiency to process large volumes of social media data effectively. Moreover, integrating sentiment analysis with other modalities, such as image or audio data, could further enhance the detection accuracy and provide a more comprehensive understanding of mental health trends[25].

5. CONCLUSION

This research explored and compared the effectiveness of various deep learning models, including LSTM, BiLSTM, GRU, CNN, and Simple

RNN, for detecting depressive tendencies from social media text. The findings demonstrated that BiLSTM is the most effective model for this task, achieving the highest accuracy and a balanced F1-score due to its bidirectional architecture, which captures nuanced contextual information from text. This underscores the critical role of context retention in accurately identifying emotional expressions related to depression. LSTM also performed well, showcasing its ability to handle long-term dependencies, but was slightly less effective than BiLSTM due to its unidirectional processing. Meanwhile, CNN, GRU, and Simple RNN, though computationally efficient, were limited by their inability to model the complex and extensive dependencies required for depression detection.

The implications of using BiLSTM in real-world applications are significant. For mental health monitoring systems, the model's high precision and recall ensure minimal false positives and negatives, making it reliable for identifying individuals at risk of depression. However, its computational complexity presents challenges for real-time deployment, necessitating further optimization for scalability and efficiency. These findings highlight the importance of selecting models capable of capturing the intricacies of human emotions in sequential data.

Future research should focus on exploring transformer-based models like BERT, which could further improve performance by capturing bidirectional context more efficiently. Additionally, expanding the study to include multi-lingual datasets and investigating cross-lingual capabilities would enhance the generalizability of these systems. Real-time applications should also be a priority, emphasizing the development of lightweight yet accurate models for large-scale mental health surveillance. By addressing these areas, future studies can build upon this research to create robust and scalable solutions for depression detection from social media text.

REFERENCES

[1] P. Kaushik, K. Bansal, and Y. Kumar, "Deep Learning in Mental Health: An In-Depth Analysis of Prediction Systems," in *2023 International Conference on Communication, Security and Artificial Intelligence, ICCSAI 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 364–369. doi: 10.1109/ICCSAI59793.2023.10421590.

[2] S. F. C. Haviana and B. S. W. Poetro, "Deep Learning Model for Sentiment Analysis on Short Informal Texts," *Indonesian Journal of Electrical Engineering and Informatics*, vol. 10, no. 1, 2022, doi: 10.52549/ijeei.v10i1.3181.

[3] N. V. Babu and E. G. M. Kanaga, "Sentiment Analysis in Social Media Data for Depression Detection Using Artificial Intelligence: A Review," 2022. doi: 10.1007/s42979-021-00958-1.

[4] J. Yuan, X. Lu, Y. Liu, D. Shi, T. Pan, and Y. Li, "Depressive Tendency Recognition Using the Gated Recurrent Unit From Speech and Text Features," in *2021 International Conference on Asian Language Processing, IALP 2021*, 2021. doi: 10.1109/IALP54817.2021.9675265.

[5] X. Hu, J. Shu, and Z. Jin, "Depression tendency detection model for Weibo users based on Bi-LSTM," in *2021 IEEE International Conference on Artificial Intelligence and Computer Applications, ICAICA 2021*, 2021. doi: 10.1109/ICAICA52286.2021.9497931.

[6] A. Amanat *et al.*, "Deep Learning for Depression Detection from Textual Data," *Electronics (Switzerland)*, vol. 11, no. 5, Mar. 2022, doi: 10.3390/electronics11050676.

[7] D. Liu, X. L. Feng, F. Ahmed, M. Shahid, and J. Guo, "Detecting and Measuring Depression on Social Media Using a Machine Learning Approach: Systematic Review," Mar. 01, 2022, *JMIR Publications Inc.* doi: 10.2196/27244.

[8] K. M. Hasib, M. R. Islam, S. Sakib, M. A. Akbar, I. Razzak, and M. S. Alam, "Depression Detection From Social Networks Data Based on Machine Learning and Deep Learning Techniques: An Interrogative Survey," *IEEE Trans Comput Soc Syst*, vol. 10, no. 4, pp. 1568–1586, Aug. 2023, doi: 10.1109/TCSS.2023.3263128.

[9] W. Ma *et al.*, "Detecting depression tendency based on deep learning and multi-sources data," *Biomed Signal Process Control*, vol. 86, p. 105226, Sep. 2023, doi: 10.1016/J.BSPC.2023.105226.

[10] V. D. Derbentsev, V. S. Bezkorovainyi, A. V. Matviychuk, O. M. Pomazun, A. V. Hrabariev, and A. M. Hostryk, "A comparative study of deep learning models for sentiment analysis of social media texts," in *CEUR Workshop Proceedings*, 2023.

[11] M. A. Alshamari, "Evaluating User Satisfaction Using Deep-Learning-Based Sentiment Analysis for Social Media Data in Saudi Arabia's Telecommunication Sector," *Computers*, vol. 12, no. 9, 2023, doi: 10.3390/computers12090170.

[12] D. William and D. Suhartono, "Text-based Depression Detection on Social Media Posts: A Systematic Literature Review," in *Procedia Computer Science*, Elsevier B.V., 2021, pp. 582–589. doi:

10.1016/j.procs.2021.01.043.

[13] A. Onan and M. A. Tocoglu, "A Term Weighted Neural Language Model and Stacked Bidirectional LSTM Based Framework for Sarcasm Identification," *IEEE Access*, vol. 9, pp. 7701–7722, 2021, doi: 10.1109/ACCESS.2021.3049734.

[14] C. Wu, Y. Zhang, S. Lu, and G. Xu, "Short Text Sentiment Analysis Based on Multiple Attention Mechanisms and TextCNN-BiLSTM," in *ICEIEC 2023 - Proceedings of 2023 IEEE 13th International Conference on Electronics Information and Emergency Communication*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 124–128. doi: 10.1109/ICEIEC58029.2023.10199931.

[15] Y. He, "BERT-CNN-BiLSTM: A Hybrid Deep Learning Model for Accurate Sentiment Analysis," in *2023 IEEE 5th International Conference on Power, Intelligent Computing and Systems, ICPICS 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 921–926. doi: 10.1109/ICPICS58376.2023.10235335.

[16] Y. Cao, Y. Hao, B. Li, and J. Xue, "Depression prediction based on BiAttention-GRU," *J Ambient Intell Humaniz Comput*, vol. 13, no. 11, pp. 5269–5277, Nov. 2022, doi: 10.1007/s12652-021-03497-y.

[17] J. Philip Thekkekara, S. Yongchareon, and V. Liesaputra, "An attention-based CNN-BiLSTM model for depression detection on social media text," *Expert Syst Appl*, vol. 249, Sep. 2024, doi: 10.1016/j.eswa.2024.123834.

[18] H. Kour and M. K. Gupta, "An hybrid deep learning approach for depression prediction from user tweets using feature-rich CNN and bi-directional LSTM," *Multimed Tools Appl*, vol. 81, no. 17, pp. 23649–23685, Jul. 2022, doi: 10.1007/s11042-022-12648-y.

[19] W. Ying and H. Yufan, "Research on Sentiment Analysis of Travel Reviews Based on Word2Vec and RNN," in *2023 IEEE International Conference on Electrical, Automation and Computer Engineering, ICEACE 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 133–137. doi: 10.1109/ICEACE60673.2023.10442629.

[20] A. S. Jaiswal, D. V. Bhavsagar, K. Dhole, S. Chaurasia, M. Daph, and S. Chourasia, "Sentiment Analysis of Election Result Prediction using Twitter Data by NLP and ML," in *2024 International Conference on Innovations and Challenges in Emerging Technologies, ICICET 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ICICET59348.2024.10616319.

[21] H. Zogan, I. Razzak, X. Wang, S. Jameel, and G. Xu, "Explainable depression detection with multi-aspect features using a hybrid deep learning model on social media," *World Wide Web*, vol. 25, no. 1, pp. 281–304, Jan. 2022, doi: 10.1007/s11280-021-00992-2.

[22] A. Ahmed *et al.*, "Machine learning models to detect anxiety and depression through social media: A scoping review," *Computer Methods and Programs in Biomedicine Update*, vol. 2, p. 100066, Jan. 2022, doi: 10.1016/J.CMPBUP.2022.100066.

[23] J. Kim, J. Lee, E. Park, and J. Han, "A deep learning model for detecting mental illness from user content on social media," *Sci Rep*, vol. 10, no. 1, Dec. 2020, doi: 10.1038/s41598-020-68764-y.

[24] D. Hatta Fudholi, "Kinetik: Game Technology, Information System," *Mental Health Prediction Model on Social Media Data Using CNN-BiLSTM*, vol. 9, no. 1, pp. 29–44, 2019.

[25] Y. Wang, Q. Chen, and W. Wang, "Multi-task BERT for Aspect-based Sentiment Analysis," in *Proceedings - 2021 IEEE International Conference on Smart Computing, SMARTCOMP 2021*, Institute of Electrical and Electronics Engineers Inc., Aug. 2021, pp. 383–385. doi: 10.1109/SMARTCOMP52413.2021.00077.