Jurnal Teknik Informatika (JUTIF) Vol. 6, No. 2, April 2025, Page. 557-574
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4206

Depression Detection in Indonesian X Social Media Text using Convolutional Neural Networks and Long Short-Term Memory with TF-IDF and FastText Methods

Karina Khairunnisa Putri*[1], Erwin Budi Setiawan[2]

[1,2]Informatics Study Program, Faculty of Informatics, Telkom University, Indonesia

Email: [1]karinakhairunnisap@telkomuniversity.ac.id

Received : Dec 11, 2024; Revised : Jan 1, 2025; Accepted : Jan 4, 2025; Published : Apr 26, 2025

Abstract

Depression is a growing mental health issue in the modern era, with social media offering a unique opportunity for automated detection through text analysis. However, challenges such as unstructured language, ambiguity, and contextual complexity in social media text hinder accurate detection. This research aims to develop and evaluate a hybrid deep learning model to detect depression in Indonesian social media text. A data set of 50523 entries was obtained and cleaned and TF-IDF was used for feature extraction while FastText was used for feature expansion. The classification was done by using Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and a combination of both CNN and LSTM models and the performance of the models was measured using the accuracy, precision and recall scores. The experimental results showed that the LSTM model gave the best result in terms of accuracy which is 83.58%, the second best was the LSTM-CNN hybrid model with an accuracy of 83.20%. The current study thus provides a new approach for identifying depression in Indonesian language data and can be said to significantly advance the fields of informatics and computer science. It also shows how AI can be utilized in improving mental health practices and in designing better social media environments. The findings of this study contribute to the growing body of research on cross-cultural mental health detection and highlight the importance of developing language-specific machine learning models.

Keywords: *CNN, Depression detection, FastText, LSTM, Social media, TF-IDF.*

1. INTRODUCTION

Depression has been recognized as a critical health challenge in the modern era, with projections of it becoming one of the leading causes of disease by 2030 [1]. Factors such as lifestyle changes and increased socio-economic pressures exacerbate depression and increase the need for early and appropriate detection and intervention. Mental well-being and associated concerns need attention at all phases of life, whether it is during childhood, teenage years, or adulthood [2]. Individuals experiencing depression often endure fleeting or enduring periods of low spirits that diminish creativity and zest for daily activities. In the last twenty years, much has been learned by researchers about the clinical course of major depression during the medium to long term, but little attention has been given to the patients' social functioning while being depressed [3]. Extended periods of low mood and constant stress can escalate, becoming severe or recurring, ultimately leading to significant health issues. [4]. However, barriers such as social stigma and lack of effective diagnosis and treatment make it difficult to effectively address this issue

Individuals affected by depression often experience signs like sleeplessness isolation, , diminished desire for food and rest, difficulty focusing on both professional and personal matters, and, in some cases, a significant risk of self-harm [5]. Therefore, there is an urgent need to improve access

to and accuracy of depression diagnosis[6], and research on automated detection methods through social media content analysis is being particularly promoted. Early detection of depression is crucial as it can prevent possible suicide and improve the quality of life of individuals and society [7].

Social media platforms such as Facebook, X, and Reddit are widely used as platforms for people to express their emotional states, particularly for those experiencing depression or having suicidal thoughts [8]. Prompt recognition of depressive signs, coupled with assessment and intervention, can markedly enhance an individual's longevity by alleviating the intensity of the foundational condition [9]. This timely recognition can also greatly mitigate the negative effects on overall wellness and health, as well as on individual, occupational, and social life [10]. One of the main reasons it is difficult to detect depression is the absence of visually visible symptoms. Depression has a huge impact on the lives of both the sufferer and those around them. However, due to a lack of knowledge in this area, people with depression are often unaware of their own condition [11]. Language use is one of the indicators that can help identify symptoms of depression in a person [12].

Research related to depression detection has been carried out, some of which have implemented a Hybrid Deep Learning approach, namely by implementing a combination of Convolutional Neural Network (CNN) classification models with Bi-directional Long Short-Term Memory (Bi-LSTM) [13]. With the application of this concept, the accuracy value is 94.28%. However, the research in the journal only uses basic embedding methods (without mentioning FastText or other feature expansion techniques) and there is no Similarity Corpus formation step. Another depression detection classification method research is using the CNN method with Word2Vec word embedding which shows an accuracy of 84.8% [14]. However, in the research in the journal, the pre-processing step uses basic steps such as tokenization, lowercase, removal of stop words, stemming, and lemmatization. Then in the research there is no similarity corpus formation step. In research [15], applying deep learning methods CNN, LSTM, and Bi-LSTM with an accuracy of 92%, 80%, and 88%, respectively. But in this study, they evaluated only the hybrid models of CNN, LSTM, and Bi-LSTM, without comparing them with non-hybrid methods such as CNN or single LSTM.

In view of this study which is grounded on depression detection through the application of hybrid deep learning models on Indonesian social media text, it is important to compare it with other similar studies from different fields that have also used hybrid deep learning models including CNN and LSTM. Research conducted in the journal [16] discusses the prediction of Remaining Useful Life (RUL) using sensor data from industry using the CNN-LSTM hybrid method. However, this research uses RMSE and R^2 metrics as the final result [16]. Unlike our approach, they do not use feature extraction techniques such as TF-IDF or embedding, instead directly processing the raw data using convolutional layers and LSTM. In addition, research conducted in the journal which focuses on detecting cyber-attacks, using the CNN-LSTM hybrid learning method only focuses on structured data from networks, making it less flexible in applications to other types of data such as text even though it has higher accuracy[17].

Furthermore, the results of the analysis that we did in the journal [18] using the same method, namely hybrid deep learning CNN-LSTM. However, in the journal, the pre-processing step uses basic techniques such as Z-score normalization and Min-Max scaling without involving semantic feature enrichment techniques such as FastText applied in this study. In addition, the journal does not discuss unstructured text processing, which is the main focus of this research. The CNN-LSTM hybrid model in the journal was used to predict QoS metrics on numeric datasets, without comparing it with alternative models for text data or enrichment methods such as TF-IDF. The metrics used in the study were MAE and RMSE [18]. There have been many studies that use CNN-LSTM hybrid deep learning. The next is research conducted in the [19] journal. The preprocessing process only includes

tokenization, padding, and removal of symbols and irrelevant words. Then for feature extraction in the journal [19] using simple Keras embedding for word representation.

In addition, in a study conducted by the journal "Logistic Regression Classification with TF-IDF and FastText for Sentiment Analysis of LinkedIn Reviews" [20], the study analyzed sentiment from LinkedIn user review datasets and used a combination of TF-IDF and FastText as feature representation. The results showed that the Logistic Regression model with the combination of TF-IDF and FastText was able to achieve an accuracy of 91.86% [20]. However, the method used is limited to Logistic Regression and does not compare with other models, such as deep learning. Looking at the results of previous studies, our research integrates TF-IDF and FastText with additional steps such as the formation of a similarity corpus to enrich the semantic context, providing a richer approach in sentiment analysis on Indonesian social media texts.

Depression detection methods have the potential to identify people suffering from, or at high risk of depression. Currently, depression is usually identified through questionnaires and face-to-face interviews with mental health professionals [21]. These methods have limitations in scope, cost, and accessibility. By utilizing today's popular social media, depression detection can be automated [22], more comprehensive, and more cost-effective. However, the biggest challenge in social media analysis is the complexity of textual data and the context that needs to be well understood.

Based on the literature review, most of the previous studies have explored the use of deep learning models such as CNN, LSTM, and their combination for depression detection from text data. Although the results obtained are promising, most of these studies were conducted on English data. Not many studies have applied similar methods to Indonesian data, especially on popular social media platforms such as X. In addition, most of the previous studies still rely on words or phrases as the main features, without optimally utilizing feature expansion techniques that can capture vocabulary mismatches and provide richer word representations.

Based on the problem description and previous research, this study contributes by not only evaluating the CNN-LSTM hybrid model, but also comparing its performance with individual models, namely CNN and LSTM. This approach provides a more comprehensive analysis of the advantages and disadvantages of each method. Then the author adds a Similarity Corpus formation step to increase the relevance of data to the context of depression in Indonesian. This is an innovation that is not found in previous research and provides added value in the classification accuracy of the model.

Therefore, this research aims to fill the gap by developing a depression detection model on Indonesian language X data using a combination of CNN and LSTM and utilizing TF-IDF feature extraction and FastText feature expansion techniques. In addition, this research seeks to answer the main question of how can the combination of these methods improve the accuracy and relevance of depression detection in the context of Indonesian text data. To the best of the authors' knowledge, this research is one of the first to apply a single model and a combination of CNN-LSTM models using TF-IDF extraction feature and FastText expansion feature on Indonesian data from platform X for depression detection.

2. METHOD

This study employs a blended framework that integrates Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) to detection of depression on X social media platform. Data is collected through crawling techniques on social media X, followed by labeling and pre-processing. Features are extracted using TF-IDF method to represent words based on their frequency weights, while the use of Fastest helps enrich the semantic context of the features. The performance of the model is evaluated using confusion matrix to measure accuracy and precision. Below is a system design scheme for Text Analysis on social media using the Convolutional Neural Networks (CNN)

and Long Short-Term Memory (LSTM) methods. Figure 1 below is the system design used for this research

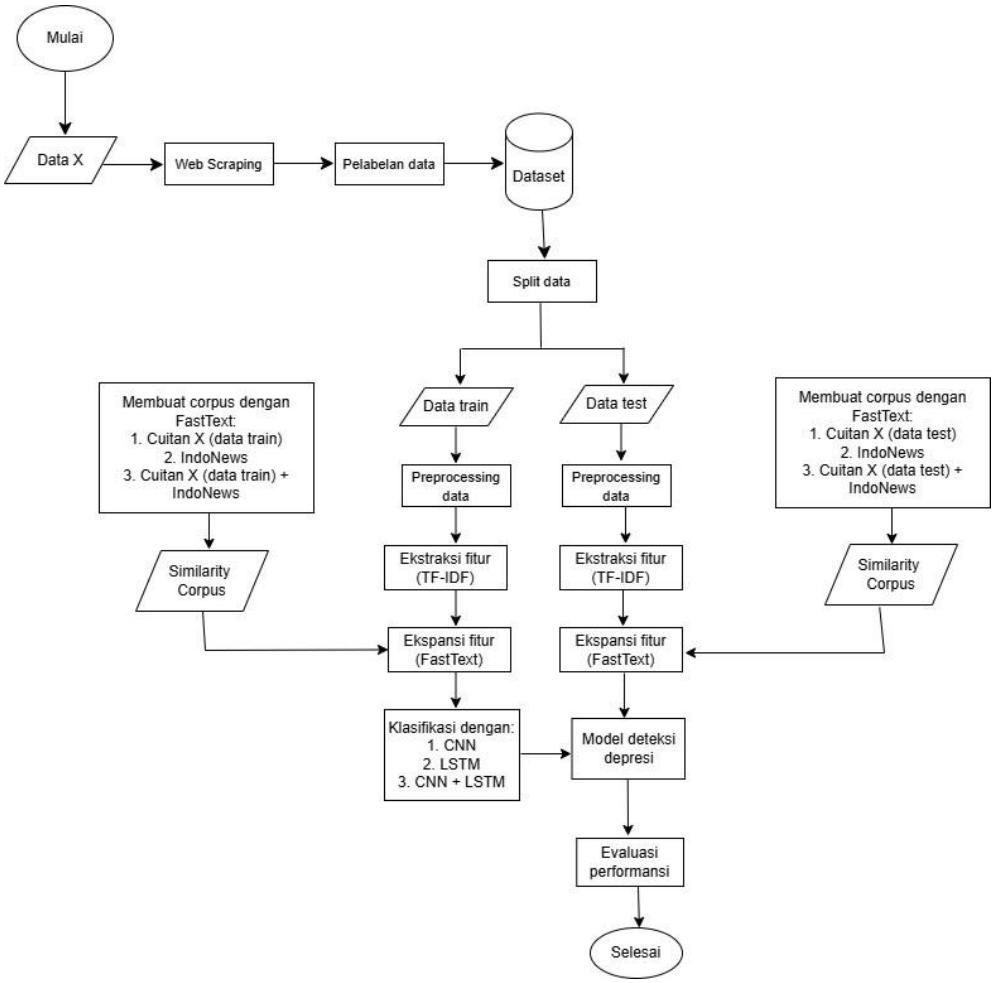

Figure 1 System design flowchart

2.1. Data Crawling

Data crawling, also known as web scraping, is a method that enables the automatic and structured collection of data from websites. A crawler is a computer program designed to automatically and systematically browse web pages [23]. Through X crawling, tweet data can be systematically collected based on specific keywords, hashtags, locations, or user accounts.

In addition, additional datasets were also taken from Marshall's GitHub repository [24] to enrich and complement the existing data collection. The addition of datasets from GitHub aims to increase the diversity and representativeness of the sample, thus providing a more comprehensive perspective in social media-based mental health analysis. The selection of keywords to identify indications of depression in this study is based on an in-depth review of various international scientific journals. in the research conducted by Vedula [25] conducted a linguistic and emotional analysis of social media that produced negative emotions: sedih, putus asa, gelisah, cemas.

This data is then indexed and stored in a repository that enables further processing and analysis. The total data collected is 50523 entries, which will be used as the main dataset in the analysis and testing of the depression detection model.

Table 1. List of keywords from X

Keywords	Value
Capek	5137
Gelisah	10023
Lelah	4180
Putus asa	4774
Sedih	1305
Sengsara	610
Stress	494
Value	26523

Table 2. Value keyword from GitHub

Keywords	Value
General depression	24000
Value	24000

2.2. Data Labelling

Before moving on to the classification stage, the process of labelling the data that has been collected is an important first step. Through this labelling, each data will be identified and categorized first, thus facilitating the subsequent analysis and categorization process. This is done so that the data can be easily classified and understood. Such binary labelling is often used in binary classification tasks, where the goal is to classify data instances into one of two classes (in this case, the depression class is denoted by 1 and non-depression is denoted by 0). Table 3 shows the group and number of labelling data

Table 3 Total data for each table

Group	Label	Value
Depression	1	25281
Non depression	0	25242
Value		**50523**

2.3. Data Pre-processing

Data pre-processing is a very important step in the machine learning process as the quality of the data and the advantages that can be derived from the unprocessed information will significantly influence the model's capacity to learn and function effectively. [26]. The tweet data that has been collected through the crawling process often contains noise [27], which is data that does not have relevant information for analysis purposes.

In this study, the pre-processing phase will involve multiple steps, including Data Cleaning, Case Folding, Normalization, Tokenization, Stopwords Removal, and Stemming.

1. Data Cleaning
   Data cleaning is the process of cleaning or removing irrelevant attributes from input data, such as symbols, punctuation marks, numbers, URLs, and correcting missing values.
2. Case Folding
   Case folding is an operation that standardizes text by transforming every character into lowercase. The aim of this operation is to ensure that all text data processed maintains uniform and consistent formatting.
3. Normalization
   Normalization is the process of replacing non-standard words into standard words. The goal is to make the text more uniform and consistent so that further analysis and processing becomes easier.

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4206

During the normalization process, words that have different spelling or writing, or that do not meet grammatical standards can be changed or replaced with more common and standard forms [28].

4. Tokenization
   In this section, the sentence will be broken down into a list of words to make it easier for the model to understand the data and allow exploration of each word in a sentence [29].
5. Stopwords Removal
   Stopwords is the process of removing common words that are not important or relevant, which aims to reduce the amount of data entered into the classification model.
6. Stemming
   Stemming involves transforming words within a sentence into their root form by eliminating the prefixes and suffixes attached to each word in the sentence. [29].
   Table 4 shows a step-by-step example of pre-processing in this study.

Table 4. Example of pre-processing data

Step Pre-processing	Tweet	Result
Data cleaning	Takutttt gimana iniiii aku anxiety (lagi),, aku gabisa diem keringet dingin trus gelisah gitu	Takut gimana ini aku anxiety aku gabisa diem keringet dingin trus gelisah gitu
Case folding	Takut gimana ini aku anxiety aku gabisa diem keringet dingin trus gelisah gitu	takut gimana ini aku anxiety aku gabisa diem keringet dingin trus gelisah gitu
Tokenization	takut gimana ini aku anxiety aku gabisa diem keringet dingin trus gelisah gitu	[takut, gimana, ini, aku, anxiety, aku, gabisa, diem, keringet, dingin, trus, gelisah, gitu]
Normalization	[takut, gimana, ini, aku, anxiety, aku, gabisa, diem, keringet, dingin, trus, gelisah, gitu]	[takut, gimana, ini, aku, cemas, aku, tidak bisa, diam, keringat, dingin, terus, gelisah, begitu]
Stopwords removal	[takut, gimana, ini, aku, cemas, aku, tidak bisa, diam, keringat, dingin, terus, gelisah, begitu]	[takut, gimana, cemas, tidak bisa, diam, keringat, dingin, gelisah]
Stemming	[takut, gimana, cemas, tidak bisa, diam, keringat, dingin, gelisah]	[takut, gimana, cemas, tidak, bisa, diam, keringat, dingin, gelisah]

2.4. TF-IDF Feature Extraction

Term Frequency (TF) is the simplest way of weighting the terms and is, therefore, the most basic approach to assigning weights to text. Term Frequency-Inverse Document Frequency (TF-IDF) is a mathematical model used for determining the importance of a particular word in a certain document in comparison to other documents in the same set of documents. This framework is often used in determining the weight of words in information retrieval or text mining scenarios [30]. The following is the formula for calculating TF on the subject:

$$\mathrm{TF}(t,d) = \frac{Number\ of\ occurrences\ of\ t\ in\ d}{\text{Total number of words in d}} \qquad (1)$$

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4206

Where:
t : specific word
d : specific tweet

TF-IDF in is a a document measure is for one assigning factor weights that to determines frequently the occurring weight terms. of The a frequency term. of The a second term factor which will be helpful in assigning low weight to a term if it appears very frequently in many documents is IDF which stands for Inverse Document Frequency. The following IDF formula calculates how rarely a word appears in all documents:

$$\text{IDF}(t,d) = log\ (\frac{N}{|\{d \in D: t \in d\}| + 1}) \quad (2)$$

Where:
N : total number of tweets
$|\{d \in D: t \in d\}|+1$: number of tweets containing word t

The TF-IDF technique employed to determine the significance of every term within a document can be expressed using the formula:

$$TF - IDF(t,d,D) = TF(t,d) \times IDF(t,D) \quad (3)$$

Feature extraction using TF-IDF was chosen because it is a method that has been proven effective in assigning weights to words based on their frequency and uniqueness of occurrence in the data. TF-IDF has the additional capability of using the N-gram parameter. Simply put, N-gram is a statistical method in language processing that focuses on identifying specific sequences of words or items in a text. The number “N” itself indicates the length of the sequence to be analysed, allowing researchers to look at language patterns and structures in greater depth [31].

This study employs five varieties of n-grams, namely Unigram, Bigram, Trigram, Uni-Bigram, and Uni-Bi-Trigram. The aim of incorporating these diverse n-gram forms is to enhance the model's grasp of word significances and relationships, which ultimately facilitates the production of information with greater precision. Table 5 illustrates an instance of utilizing the N-gram parameter in TF-IDF for the extraction of the term "tidak bisa diam".

Table 5. Example of N-gram

N-gram	Tweet
Unigram	[tidak], [bisa], [diam]
Bigram	[tidak bisa], [bisa diam]
Trigram	[tidak bisa diam]
Uni-Bigram	[tidak], [bisa], [diam], [tidak bisa], [bisa diam]
Uni-Bi-Trigram	[tidak], [bisa], [diam], [tidak bisa], [bisa diam], [tidak bisa diam]

2.5. FastText Feature Expansion

The FastText expansion feature has the potential to improve the performance of depression detection models by generating a richer and more contextual representation of words. FastText treats text as a collection of words and converts them into a fixed-length vector representation. With this approach, FastText ignores the order of words, which makes it faster in processing. In this case, word order is no longer a major factor, which allows FastText to process text more efficiently [32].

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4206

In the area of depression detection, the FastText expansion feature can offer the following crucial benefits. First, the majority of the data set that is currently used for depression detection is collected from social media platforms and is rather unstructured and informal. This makes FastText effective in identifying signs of depression especially when applied in situations where such language is used. FastText its was capability employed of for dealing feature with expansion vocabulary because gaps and of generating more semantic features of words especially when dealing with poorly structured text data.

2.6. Corpus

This research uses FastText to explore word similarity by building three types of corpus: tweet corpus, news corpus, and mixed tweet+news corpus. The tweet corpus is generated from tweet data, while the news corpus is formed based on Indonesian news data (IndoNews). Meanwhile, the tweet+IndoNews mixed corpus was created by combining tweet data and some news data. To model the corpus, FastText Expansion was used with 10 epochs and a learning rate of 0.001. The number of words in each corpus can be seen in Table 6.

Table 6. Total number of words in each corpus

Keyword	Total
Tweet	50523
IndoNews	100594
Tweet+IndoNews	151117

Table 7 illustrates the similarity level of the word “sedih” in the tweet corpus. The research uses a ranking system that is organized by the degree of similarity of words, starting from the highest to the lowest percentage of similarity. The ranking method applied includes four different categories, namely Top 1, Top 5, Top 10, and Top 15, which allows for multilevel analysis according to the needs of the research.

Table 7. Top 10 similar words for 'sedih'

Rank	Simillar world	Value
1	kecewa	0.759
2	pedih	0.758
3	sesal	0.634
4	tangis	0.619
5	rasa	0.606
6	marah	0.599
8	bingung	0.564
9	senang	0.544
10	marah	0.541

2.7. Data Split

The acquired dataset is split into two segments: training data and validation data. The proportion of training data should be larger than that of the validation data. This is because utilizing a greater amount of data for system training yields improved outcomes in identifying depression. Consequently, the precision in categorizing depression and non-depression also rises. The ratios explored to ascertain the most effective ones are 90:10, 80:20, and 70:30.

Once the dataset is partitioned, the training data is employed to educate the depression detection model, while the validation data is utilized to evaluate the model's efficacy in classifying novel data. By adopting this method, the training outcomes illustrate not only the model's capability to discern patterns in the training data, but also its capacity to generalize to previously unseen data.

2.8. Classification Model

2.8.1. Convolutional Neural Network

Convolutional neural network or CNN is a learning algorithm inspired by the way the human brain processes visual data. CNNs are designed to recognize patterns from input data such as images or objects [26]. CNNs can be composed based on convolution, fusion, and fully connected layers [33]. CNN is not only used in image classification but also for data analysis, pattern recognition, and computer vision vision as well as solving NLP tasks [34], CNN models can detect certain thoughts [35]. In Figure 2 is a picture of the architecture of the CNN

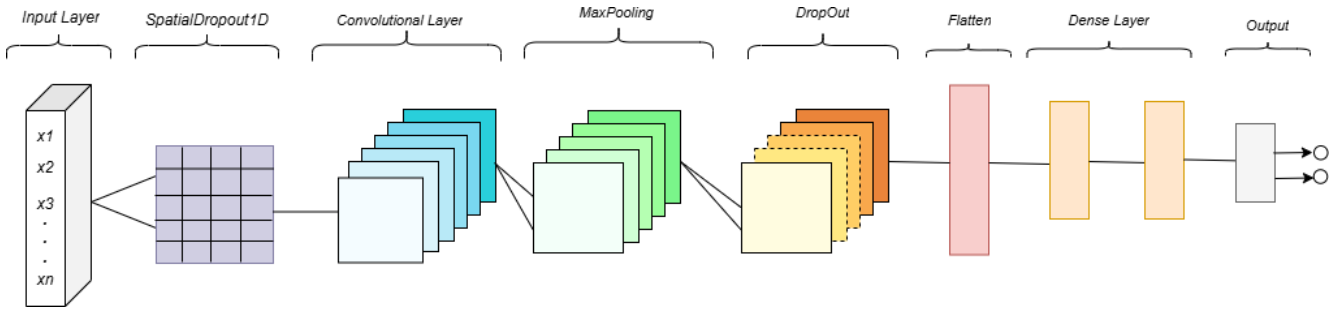

Figure 2. Architecture of CNN

This study uses a 1D-CNN (one-dimensional convolutional neural network) implemented using the TensorFlow Keras library. The model architecture consists of one one-dimensional spatial dropout layer, a Conv1D layer with 128 filters and a ReLU activation function, followed by a MaxPooling1D layer and an average layer. The model also includes two dense layers with 32 units that use the ReLU activation function, as well as one output unit with a sigmoid activation function. Optimization is performed using the Adam optimizer with a learning rate of 0.001, while the selected loss function is binary cross-entropy. In the initial stage, the convolution layer in the CNN applies a filter to identify basic features in the text, such as word strings (n-grams) or specific word sequences that frequently appear in depression-related tweets.

2.8.2. Long Short-Term Memory

Long Short-Term Memory (LSTM) artificial neural network models have surfaced as a leading-edge technique in the handling and examination of sequential data like text, audio, and temporal signals. LSTM networks represent a type of neural networks adept at capturing long-lasting dependencies and possess memory components known as cell states to retain information. [36]. When working on a series of sentences, LSTMs are most adept at recalling any information for as long as necessary [37].

The LSTM model was designed using the Tensorflow Keras library and consists of several main components, including a one-dimensional spatial dropout layer, an LSTM layer with 128 neurons equipped with a recurrent dropout of 0.2, as well as two dense layers with 32 units and a ReLU activation function. The output layer consists of one unit with a sigmoid activation function. The model is optimized using Adam optimizer with a learning rate of 0.001 and using binary cross entropy

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4206

as the loss function. In Figure 3 is a picture of the architecture of the LSTM

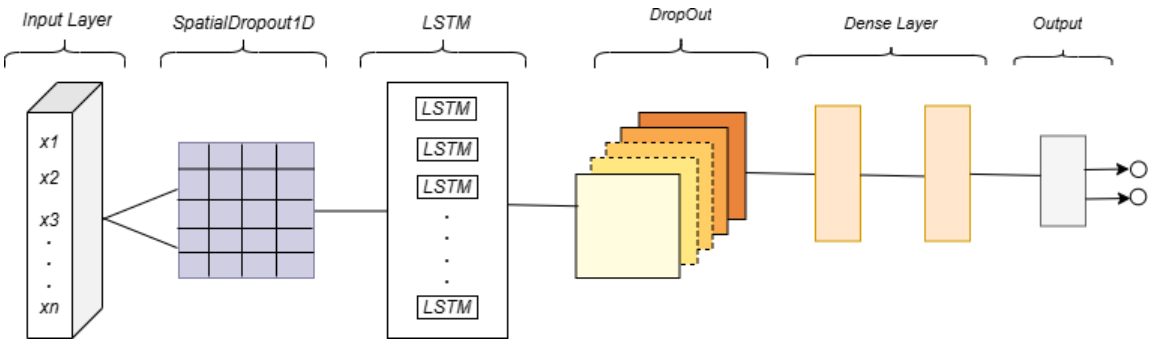

Figure 3. Architecture of LSTM

Long Short-Term Memory (LSTM) was selected for its capacity to grasp extended relationships in ordered information like text. LSTM can recall important information from previous contexts and combine it with new information, which is very useful in understanding nuances and context in text.

2.8.3. Hybrid

Using a hybrid deep learning system, this study creates a depression detection model to determine whether combining two or more classification models can increase accuracy in comparison to using just one deep learning model. Through the integration of layers from two distinct neural network architectures into a single structure, this method uses the idea of a hybrid model. This study primarily focuses on two hybrid model variations, LSTM + CNN and CNN + LSTM, which differ significantly in the layer order.

In the LSTM + CNN model, the LSTM layer is placed at the beginning to process the data first, and the results from this layer are used as input for the CNN layer which is at the next stage. In contrast, in the CNN + LSTM model, the CNN layer acts as the first layer to extract features, which are then passed on to the LSTM layer for further analysis. Figure 4 is an illustration of the hybrid model used in this study.

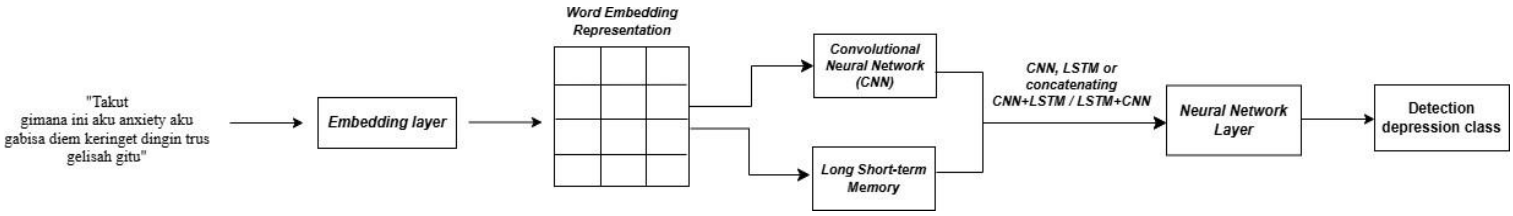

Figure 4. Illustration of hybrid model

The combination of CNN and LSTM (hybrid deep learning) was chosen because both have complementary advantages. CNN can extract local features, while LSTM can capture long-term dependencies.

2.9. Confusion Matrix

Confusion Matrix is a standard way to summarize the performance of classification methods [38]. The confusion matrix comprises four terms that illustrate the outcomes of performance evaluation: True Positive (TP), indicating the quantity of positive samples accurately categorized; False Positive (FP), representing the quantity of positive samples wrongly categorized; True Negative (TN), denoting the quantity of negative samples accurately categorized; and False Negative (FN),

highlighting the quantity of negative samples wrongly categorized. [39]. Table 8 is an overview of the confusion matrix.

Table 8. Confusion matrix

		Actual	
		Positive	Negative
Prediction	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

According to the confusion matrix, there are various metrics employed to evaluate the effectiveness of the classification model, specifically Precision and Accuracy. Precision represents the proportion of correct positive predictions to the overall positive predictions. Precision can be calculated using the following formula:

$$Precision = \frac{TP}{(TP+FP)} \quad (4)$$

Accuracy represents the proportion of accurate forecasts (True Positive and True Negative) relative to the entire dataset. Accuracy can be calculated using the following formula:

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \quad (5)$$

3. RESULT

This research will carry out a series of classification trials that include four different models: Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and two hybrid models, Hybrid CNN-LSTM and Hybrid LSTM-CNN. Each model will be tested to analyze its performance and classification capabilities in the context of the research conducted.

Accuracy is calculated based on the average of three program executions. Each scenario in this study uses the TF-IDF feature extraction method. A total of five scenarios have been carried out to obtain the most optimal results. The scenario can be seen in Table 9

Table 9. Scenario test sequences

Scenario	Description
1	The model evaluation process will be carried out by testing various dataset sharing ratios and identifying the model with the highest accuracy. The results of the model will be used as a reference or starting point (baseline) for the next stage of research.
2	Conducted model testing with a combination of n-grams using unigram, bigram, trigram, unigram + bigram, and unigram + bigram + trigram.
3	Testing the model by comparing maxfeatured 5000, 10000, and 15000 feature vectors.
4	Tests were conducted by utilizing the extended features generated from the similarity corpus using FastText, as well as evaluating the performance of the hybrid model by integrating CNN and LSTM (CNN + LSTM) and LSTM and CNN (LSTM + CNN). This step aims to measure the overall performance of the model.

3.1. Scenario 1

In the scenario 1, the classification model will use the TF-IDF (Term Frequency-Inverse Document Frequency) weighting method with a unigram approach. This TF-IDF technique allows to

measure the significance of each word in the document by considering its frequency of occurrence and distribution in the whole data corpus. By applying unigram, each word will be analyzed individually to provide accurate weighting in the classification process. Table 9 shows the outcomes of partitioning diverse ratio datasets of 90:10, 80:20, and 70:30, utilizing 10.000 features.

Table 10. Result of the first scenario test

Split Ratio	CNN		LSTM		LSTM+CNN		CNN+LSTM	
	Acc(%)	Prec(%)	Acc(%)	Prec(%)	Acc(%)	Prec(%)	Acc(%)	Prec(%)
90:10	82	80.53	**83.35**	**82.06**	**83.17**	**81.73**	82.32	80.56
80:20	**83.09**	**83.57**	82.51	81.21	82.59	81.73	**82.35**	**82.22**
70:30	82.47	82.47	82.79	81.88	82.35	80.56	82.10	80.82

Table 10 shows that the best data separation ratio for the LSTM and hybrid LSTM-CNN models is 90:10, with accuracies of 83.35% and 83.20%, respectively. Meanwhile, for CNN and hybrid CNN+LSTM models, the best separation ratio is found at 80:20, with accuracies of 83.09% and 82.35%. This separation ratio becomes the reference for testing in the next scenario. In the second scenario, TF-IDF was tested with Bigram, Trigram, and Unigram+Bigram and Unigram+Bigram+Trigram features.

3.2. Scenario 2

In the second case, multiple N-Gram parameters are explored within TF-IDF feature extraction to identify the N-Gram configuration that yields the most favorable outcomes for the base model. In this case, evaluations are carried out by contrasting the accuracy produced from various N-Gram combinations against the baseline model. Some of the N-Gram pairings examined include Unigram, Bigram, Trigram, Uni-Bigram, and Uni-Trigram. [40].

Table 11. Accuracy Value in Scenario 2

N-Gram	CNN		LSTM		LSTM+CNN		CNN+LSTM	
	Acc(%)	Prec(%)	Acc(%)	Prec(%)	Acc(%)	Prec(%)	Acc(%)	Prec(%)
Unigram (baseline)	**83.09**	**83.57**	**83.35**	**81.21**	**83.17**	**81.73**	**82.35**	**82.22**
Bigram	77.55	80.22	78.34	78.34	78.79	77.71	81.23	80.44
Trigram	71.78	83.56	72.84	72.84	73.28	73.00	60.95	66.72
Uni+Bi	81.95	79.90	82.53	80.55	82.75	81.26	81.95	82.20
Uni+Bi+Tri	82.63	81.91	82.58	80.61	82.38	80.87	82.30	80.19

However, the accuracy results show that the baseline unigram is superior to other n-grams. Longer n-grams (bigram, trigram) often have lower frequency scores because the distribution of those words tends to be less frequent. Table 11 Outlines the accuracy of the model in the second evaluation along with the earlier test outcomes utilizing the TF-IDF Unigram technique.

3.3. Scenario 3

In scenario 3, model testing is done by comparing the performance of feature vectors that have a maximum size of 5000, 10000, and 15000. In this research, the variation in the number of features used aims to evaluate how much influence the number of features has on the accuracy and effectiveness of the model.

Table 12. Accuracy Value in Scenario 3

Max-	CNN	LSTM	LSTM+CNN	CNN+LSTM

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4206

features	Acc(%)	Prec(%)	Acc(%)	Prec(%)	Acc(%)	Prec(%)	Acc(%)	Prec(%)
10000	**83.09**	**83.57**	**83.35**	**81.21**	**83.17**	**81.73**	82.35	82.22
5000	82.57	82.40	83.23	81.20	82.86	81.64	**83.17**	**82.08**
15000	81.72	82.79	83.31	81.79	82.92	81.50	82.53	80.89

Based on Table 12 accuracy testing above, the CNN + LSTM hybrid model experienced an increase in accuracy at max feature 5000, the increase in accuracy is 0.82%. The smaller number of features allows the CNN+LSTM hybrid model to learn patterns more efficiently, while LSTM utilizes larger features to capture complex contexts. For CNN, LSTM and hybrid LSTM+CNN models it is still better to use max featured 10000

3.4. Scenario 4

In the next scenario, model testing is performed by utilizing expansion features derived from the similarity corpus generated by FastText. In this process, expansion features serve to improve the representation of the text by capturing deeper semantic relationships between words.

We also leveraged the highest few 'N' rankings of the most comparable terms in the corpus to ascertain the ideal similarity metric that would yield the greatest precision. The corpus utilized in this evaluation encompasses datasets from tweets, Indonews, along with a fusion of both (tweets+Indonews). The leading rankings examined in this analysis include Top 1, Top 5, Top 10, and Top 15.

Table 13. Accuracy Value of CNN Model in Scenario 4

Rank	**CNN**							
	Baseline		**Tweet**		**Indonews**		**Tweet+Indonews**	
	Acc(%)	Prec(%)	Acc(%)	Prec(%)	Acc(%)	Prec(%)	Acc(%)	Prec(%)
Top 1	**83.09**	**83.57**	81.68 (-1.41)	80.89 (-2.68)	82.13 (-0.96)	81.55 (-2.02)	82.83 (0.26)	82.23 (-1.34)
Top 5			80.93 (-2.16)	81.02 (-2.55)	82.46 (-0.63)	81.74 (-1.83)	82.08 (-1.01)	82.39 (-1.18)
Top 10			80.71 (-2.38)	80.80 (-2.77)	81.81 (-1.28)	83.31 (-0.26)	81.57 (-1.52)	82.70 (0.87)
Top 15			79.22 (-3.87)	83.07 (0.50)	79.03 (-4.06)	81 (-2.57)	80.81 (-2.28)	78.72 (-4.85)

Table 14. Accuracy Value of LSTM Model in Scenario 4

Rank	**LSTM**							
	Baseline		**Tweet**		**Indonews**		**Tweet+Indonews**	
	Acc(%)	Prec(%)	Acc(%)	Prec(%)	Acc(%)	Prec(%)	Acc(%)	Prec(%)
Top 1	83.12	81.21	83.07 (0.05)	81.85 (+0.64)	83.34 (-0.22)	82.20 (+0.99)	83.22 (+0.1)	82.20 (+0.99)
Top 5			82.46 (-0.66)	82.46 (+1.25)	**83.58 (+0.46)**	**81.88 (+0.67)**	83.50 (+0.38)	81.64 (+0.43)
Top 10			79.22 (-3.9)	76.63 (-4.58)	80.82 (-2.3)	78.49 (-2.72)	80.65 (-2.47)	77.58 (-3.63)
Top 15			77.63 (-5.49)	76.45 (-4.76)	80.18 (-2.94)	77.07 (-4.14)	73.38 (-9.74)	73.33 (-7.88)

Table 15. Accuracy Value of LSTM+CNN Model in Scenario 4

Rank	**LSTM+CNN**							
	Baseline		**Tweet**		**Indonews**		**Tweet+Indonews**	
	Acc(%)	Prec(%)	Acc(%)	Prec(%)	Acc(%)	Prec(%)	Acc(%)	Prec(%)

Top 1	83.17	81.73	82.73 (0.44)	81.54 (-0.19)	83.09 (-0.08)	81.73 (0)	82.89 (-0.28)	80.89 (-0.84)
Top 5			81.85 (-1.32)	81.08 (-0.65)	83.01 (-0.16)	81.16 (-0.57)	**83.20 (+0.03)**	**81.40 (-0.33)**
Top 10			78.83 (-4.34)	76.36 (-5.37)	80.16 (-3.01)	78.29 (-3.44)	79.31 (-3.86)	76.08 (-5.65)
Top 15			79.06 (-4.11)	78.97 (-2.76)	79.84 (-3.33)	77 (-4.73)	79.31 (-3.86)	76.08 (-5.65)

Table 16. Accuracy Value of CNN+LSTM Model in Scenario 4

Rank	**CNN+LSTM**							
	Baseline		**Tweet**		**Indonews**		**Tweet+Indonews**	
	Acc(%)	Prec(%)	Acc(%)	Prec(%)	Acc(%)	Prec(%)	Acc(%)	Prec(%)
Top 1	**83.17**	**82.08**	80.04 (-3.13)	84.77 (+2.69)	81.63 (-1.54)	82.23 (+0.23)	82.04 (-1.13)	82.44 (0.36)
Top 5			78.13 (-5.04)	83.09 (+1.01)	81.68 (-1.49)	81.05 (-1.03)	81.57 (-1.6)	81.05 (-1.03)
Top 10			75.58 (-7.59)	80.07 (-2.01)	81.29 (-1.88)	82 (-0.08)	69.46 (-13.71)	70.46 (-11.62)
Top 15			79.34 (-3.83)	82.66 (+0.58)	81.98 (-1.19)	81.53 (-0.55)	81.50 (-1.67)	80.19 (-1.89)

Based on Table 13 and Table 14, testing a single model shows that the LSTM model has an accuracy that increases as the curation goes up to 0.46%. FastText successfully improves semantic representation, especially in the LSTM model, which relies on understanding the word context. While CNN model has no accuracy increase. Based on Table 15 and Table 16, testing the hybrid model shows that the LSTM+CNN model has an increased accuracy of 0.03%. While the CNN+LSTM model has no increase in accuracy

4. DISCUSSIONS

In this study, a set of test scenarios has been carried out to identify the optimal model. In the figure 5, we can see the comparison of the accuracy of CNN, LSTM, CNN+LSTM, and LSTM+CNN models in the four scenarios tested. Scenario 1 serves as the baseline, with the initial accuracy of each model being 83.09% for CNN, 83.35% for LSTM, 82.35% for CNN+LSTM, and 83.17% for LSTM+CNN. In the second scenario, there was no significant improvement for all models compared to the baseline, with CNN remaining at 83.09%, LSTM remaining at 83.35%, and LSTM+CNN and CNN+LSTM also experiencing no improvement in accuracy. This shows that the addition of bigrams and trigrams in this scenario has not had a significant impact on model performance. The third scenario showed a significant improvement in the CNN+LSTM model, which achieved an accuracy of 83.17% from the baseline of 82.35%. While other models such as CNN, LSTM, and LSTM+CNN still maintain the same accuracy as the baseline. In the fourth scenario, the LSTM model experienced the highest increase in accuracy, reaching 83.58% compared to the baseline of 83.35%. The LSTM+CNN and CNN+LSTM models remained stable with no significant change, while CNN remained at 83.09%. This indicates that the additional characteristics employed are more successful in enhancing the LSTM model's effectiveness compared to the alternative approaches.

Jurnal Teknik Informatika (JUTIF) Vol. 6, No. 2, April 2025, Page. 557-574
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4206

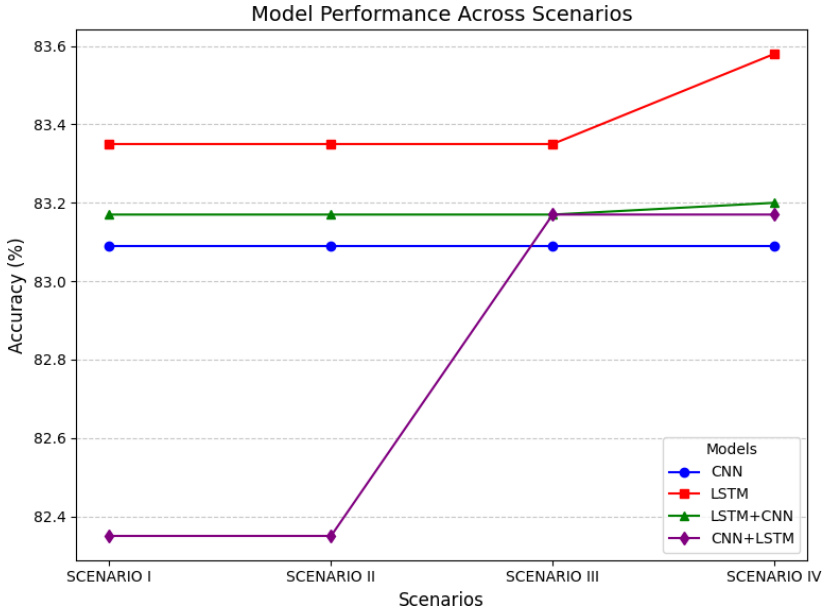

Figure 5 Graphic accuracy

In the CNN model, when viewed from scenarios one to four, the CNN model may have achieved optimal performance in the base scenario, so the addition of expansion features cannot significantly improve accuracy. This situation can occur if the CNN model is too focused on certain patterns in the training data, which makes it difficult to improve generalization ability. These results also show that using FastText as a feature expansion method successfully improves the semantic representation of the data, especially in the LSTM model. However, its effect on the CNN model is not significant, indicating that CNN relies more on local patterns rather than broader semantic relationships.

In this research, the discussion will be limited to the results and comparison with other journals that focus on the same topic and objective, such as articles that use deep learning techniques for the detection of depression. Out of the eight comparison journals highlighted in the Introduction section, this section will pick three journals that are particularly relevant to the topic of depression detection as the primary points of comparison in order to establish similarities, differences, and the contributions of this research. The methodology carried out in the journal [13] has similarities with our journal in using CNN to detect depression from social media texts, with pre-processing steps such as tokenization and removal of stopwords and evaluation using accuracy. However, the comparison journal uses Bi-LSTM while we use LSTM. However, our research uses the TF-IDF weighting technique and FastText feature expansion to enrich the data representation, and adds the step of establishing "Corpus Similarity," which is not present in the [13] journal. Despite the lower accuracy of our study, this approach offers an important contribution in local data analysis and enriches the feature representation for more contextualized depression detection.

Then the research conducted in the journal "Depression Detection from Social Media Text Analysis using Natural Language Processing Techniques and Hybrid Deep Learning Model" they proposed a hybrid model called FCL (FastText + CNN + LSTM) [14], where FastText is used for embedding, while CNN captures global features, and LSTM captures local dependencies in the data. the results of the study were more complex for English data, showing higher accuracy but requiring more computational resources. In our work, we utilize a combination of TF-IDF and FastText to produce a rich representation of the data, capturing local and semantic features, while Tejaswini et al.[14] focus on using FastText embedding without including TF-IDF.

In the research conducted in the paper "A Hybrid Model for Depression Detection Using Deep Learning", the hybrid LSTM model used to detect depression from audio and text data achieved an accuracy of 80%, while the CNN model using audio features alone achieved an accuracy of 92%. This research uses the DAIC-WOZ dataset with a combination of audio and text features to optimize

depression detection [15]. In my research, the LSTM model used managed to achieve an accuracy of 83.20%, which is higher than the study, while the CNN model in my research achieved an accuracy of 83.09%, which is lower than the CNN results in the paper. The results of this study not only contribute to the development of a more effective depression detection model, but also expand the application of deep learning techniques in the analysis of Indonesian texts, which is a significant challenge in informatics and computer science.

5. CONCLUSION

Based on research on depression detection that has been conducted on 50523 combined Indonesian X data and translated data from GitHub using CNN and LSTM models, 4 test scenarios were obtained. In this study, a significant increase in accuracy was seen in the LSTM and hybrid LSTM-CNN models when applied to scenario 4, which used extended features with a similarity corpus from FastText. The LSTM model showed the highest accuracy improvement of up to 83.58%, while the LSTM-CNN hybrid also experienced a consistent increase compared to the previous scenario. Meanwhile, the CNN-LSTM hybrid recorded a more striking increase in accuracy in scenario 3, where tests were conducted comparing datasets of 10.000, 5.000, and 15.000. This suggests that each model has different strengths based on the characteristics of the scenario used, with scenario 4 being more effective at capturing temporal patterns in the LSTM-based model and scenario 3 providing an advantage to the CNN-LSTM hybrid architecture. This research significantly contributes to the field of informatics and computer science, particularly in mental health and social media analysis, by leveraging Indonesian-language data and combining TF-IDF with FastText. The study demonstrates the effectiveness of hybrid deep learning models in handling unstructured text, enabling automated systems to assist mental health professionals in detecting early signs of depression and improving accessibility and efficiency in mental health interventions. For future research, the development of hybrid models can be done by incorporating other architectures or implementing more sophisticated feature expansion methods, such as context-based word embedding or transfer learning techniques to improve the model's performance in depression detection.

REFERENCES

[1] C. D. Mathers and D. Loncar, "New projections of global mortality and burden of disease from 2002 to 2030 Protocol S1. Technical Appendix Major cause regressions," *PloS Med*, vol. 3, no. 1, pp. 1–23, 2006, [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/17132052/#:~:text=The proportion of deaths due,drugs reaches 80%25 by 2012.

[2] Y. Zhang, C. Shaojun, T. Y. Akintunde, E. F. Okagbue, S. O. Isangha, and T. H. Musa, "Life course and mental health: a thematic and systematic review," *Front. Psychol.*, vol. 15, no. September, 2024, doi: 10.3389/fpsyg.2024.1329079.

[3] A. Kupferberg and G. Hasler, "The social cost of depression: Investigating the impact of impaired social emotion regulation, social cognition, and interpersonal behavior on social functioning," *J. Affect. Disord. Reports*, vol. 14, no. July, 2023, doi: 10.1016/j.jadr.2023.100631.

[4] A. Zafar and S. Chitnis, "Survey of depression detection using social networking sites via data mining," *Proc. Conflu. 2020 - 10th Int. Conf. Cloud Comput. Data Sci. Eng.*, pp. 88–93, 2020, doi: 10.1109/Confluence47617.2020.9058189.

[5] W. H. Organization, "Suicide worldwide in 2019," 2019. [Online]. Available: https://www.who.int/teams/mental-health-and-substance-use/data-research/suicide-data.

[6] J. I. Hlynsson and P. Carlbring, "Diagnostic accuracy and clinical utility of the PHQ-2 and GAD-2: a comparison with long-format measures for depression and anxiety," *Front. Psychol.*, vol. 15, no. May, pp. 1–12, 2024, doi: 10.3389/fpsyg.2024.1259997.

[7] J. Robinson *et al.*, "What Works in Youth Suicide Prevention? A Systematic Review and Meta-Analysis," *EClinicalMedicine*, vol. 4–5, pp. 52–91, 2018, doi: 10.1016/j.eclinm.2018.10.004.

[8] M. I. Mobin, A. F. M. Suaib Akhter, M. F. Mridha, S. M. Hasan Mahmud, and Z. Aung, "Social Media as a Mirror: Reflecting Mental Health through Computational Linguistics," *IEEE Access*, vol. 12, no. September, pp. 130143–130164, 2024, doi:

10.1109/ACCESS.2024.3454292.

[9] C. G. Davey and P. D. McGorry, "Early intervention for depression in young people: a blind spot in mental health care," *The Lancet Psychiatry*, vol. 6, no. 3, pp. 267–272, 2019, doi: 10.1016/S2215-0366(18)30292-X.

[10] Y. L. Chen *et al.*, "Effectiveness of health checkup with depression screening on depression treatment and outcomes in middle-aged and older adults: a target trial emulation study," *Lancet Reg. Heal. - West. Pacific*, vol. 43, p. 100978, 2024, doi: 10.1016/j.lanwpc.2023.100978.

[11] 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, 2023, doi: 10.1109/TCSS.2023.3263128.

[12] L. Ansari, S. Ji, Q. Chen, and E. Cambria, "Ensemble Hybrid Learning Methods for Automated Depression Detection," *IEEE Trans. Comput. Soc. Syst.*, vol. 10, no. 1, pp. 211–219, 2023, doi: 10.1109/TCSS.2022.3154442.

[13] 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*, vol. 81, no. 17. Multimedia Tools and Applications, 2022.

[14] V. Tejaswini, K. S. Babu, and B. Sahoo, "Depression Detection from Social Media Text Analysis using Natural Language Processing Techniques and Hybrid Deep Learning Model," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 23, no. 1, 2024, doi: 10.1145/3569580.

[15] Vandana, N. Marriwala, and D. Chaudhary, "A hybrid model for depression detection using deep learning," *Meas. Sensors*, vol. 25, no. November 2022, p. 100587, 2023, doi: 10.1016/j.measen.2022.100587.

[16] G. Muthukumar and J. Philip, "CNN-LSTM Hybrid Deep Learning Model for Remaining Useful Life Estimation," pp. 38–53, 2024.

[17] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System," *IEEE Access*, vol. 10, no. September, pp. 99837–99849, 2022, doi: 10.1109/ACCESS.2022.3206425.

[18] A. Pandey, P. K. Mannepalli, M. Gupta, and S. Denmark, "A Deep Learning-based Hybrid CNN-LSTM Model for Location-Aware Web Service Recommendation," *Neural Process. Lett.*, vol. 56, no. 5, pp. 1–25, 2024, doi: 10.1007/s11063-024-11687-w.

[19] P. K. Jain, V. Saravanan, and R. Pamula, "A Hybrid CNN-LSTM: A Deep Learning Approach for Consumer Sentiment Analysis Using Qualitative User-Generated Contents," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 20, no. 5, 2021, doi: 10.1145/3457206.

[20] I. Kartika, F. H. Barmawi, and N. Yuningsih, "VISA : Journal of Visions and Ideas Kepemimpinan Ideal di Era Milenial VISA : Journal of Visions and Ideas," *Visa*, vol. 4, no. 1, pp. 104–113, 2024.

[21] A. Ranjith Kumar, K. Aditya, S. Antony Joseph Raj, and V. Nandhakumar, "Depression Detection Using Optical Characteristic Recognition and Natural Language Processing in SNS," *4th Int. Conf. Comput. Commun. Signal Process. ICCCSP 2020*, 2020, doi: 10.1109/ICCCSP49186.2020.9315254.

[22] 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," *JMIR Ment. Heal.*, vol. 9, no. 3, pp. 1–18, 2022, doi: 10.2196/27244.

[23] S. M. Cheok, L. M. Hoi, S. K. Tang, and R. Tse, "Crawling Parallel Data for Bilingual Corpus Using Hybrid Crawling Architecture," *Procedia Comput. Sci.*, vol. 198, no. 2021, pp. 122–127, 2021, doi: 10.1016/j.procs.2021.12.218.

[24] Marshall1632, "Mental Health and Suicide Ideation Assement with Social Media," 2022. https://github.com/marshall1632/Mental-health-and-suicide-ideation-assement-with-social-media-/tree/master/Dataset/data (accessed Nov. 02, 2024).

[25] N. Vedula and S. Parthasarathy, "Emotional and linguistic cues of depression from social media," *ACM Int. Conf. Proceeding Ser.*, vol. Part F1286, pp. 127–136, 2017, doi: 10.1145/3079452.3079465.

[26] L. Nahar, F. Alam, and K. S. Fairose, "Depression Level Detection Using CNN Approach from Tweet Data Depression level detection using CNN approach from Tweet data," 2020.
[27] V. Çetin and O. Yıldız, "A comprehensive review on data preprocessing techniques in data analysis," *Pamukkale Univ. J. Eng. Sci.*, vol. 28, no. 2, pp. 299–312, 2022, doi: 10.5505/pajes.2021.62687.
[28] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Glob. Transitions Proc.*, vol. 3, no. 1, pp. 91–99, 2022, doi: 10.1016/j.gltp.2022.04.020.
[29] R. Lourdusamy and S. Abraham, "A Survey on Text Pre-processing Techniques and Tools," *Int. J. Comput. Sci. Eng.*, vol. 06, no. 03, pp. 148–157, 2018, doi: 10.26438/ijcse/v6si3.148157.
[30] H. U. Khan, S. Nasir, K. Nasim, D. Shabbir, and A. Mahmood, "Twitter trends: A ranking algorithm analysis on real time data," *Expert Syst. Appl.*, vol. 164, no. August 2019, p. 113990, 2021, doi: 10.1016/j.eswa.2020.113990.
[31] E. F. Unsvåg and B. Gambäck, "The Effects of User Features on Twitter Hate Speech Detection," *2nd Work. Abus. Lang. Online - Proc. Work. co-located with EMNLP 2018*, no. 2012, pp. 75–85, 2018, doi: 10.18653/v1/w18-5110.
[32] B. Edizel, A. Piktus, P. Bojanowski, R. Ferreira, E. Grave, and F. Silvestri, "Misspelling oblivious word embeddings," *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, pp. 3226–3234, 2019, doi: 10.18653/v1/n19-1326.
[33] H. V. Manalu and A. P. Rifai, "Detection of human emotions through facial expressions using hybrid convolutional neural network-recurrent neural network algorithm," *Intell. Syst. with Appl.*, vol. 21, no. January, p. 200339, 2024, doi: 10.1016/j.iswa.2024.200339.
[34] Z. Wang, L. Chen, L. Wang, and G. Diao, "Recognition of Audio Depression Based on Convolutional Neural Network and Generative Antagonism Network Model," *IEEE Access*, vol. 8, pp. 101181–101191, 2020, doi: 10.1109/ACCESS.2020.2998532.
[35] Y. Zheng and Y. Ye, "Prediction of Cognitive-Behavioral Therapy using Deep Learning for the Treatment of Adolescent Social Anxiety and Mental Health Conditions," *Sci. Program.*, vol. 2022, 2022, doi: 10.1155/2022/3187403.
[36] J. Singh, M. Wazid, D. P. Singh, and S. Pundir, "An embedded LSTM based scheme for depression detection and analysis," *Procedia Computer Science*, vol. 215. pp. 166–175, 2022, doi: 10.1016/j.procs.2022.12.019.
[37] B. Fatima, M. Amina, R. Nachida, and H. Hamza, "A mixed deep learning based model to early detection of depression," *J. Web Eng.*, vol. 19, no. 3–4, pp. 429–456, 2020, doi: 10.13052/jwe1540-9589.19344.
[38] I. Barranco-Chamorro and R. M. Carrillo-García, "Techniques to deal with off-diagonal elements in confusion matrices," *Mathematics*, vol. 9, no. 24, 2021, doi: 10.3390/math9243233.
[39] F. Rahmad, Y. Suryanto, and K. Ramli, "Performance Comparison of Anti-Spam Technology Using Confusion Matrix Classification," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 879, no. 1, 2020, doi: 10.1088/1757-899X/879/1/012076.
[40] J. H. M. D. Jurafsky, "N-Gram Language Models N-Gram Language Models," 2020.