



# Leveraging BiLSTM for Deep Learning-Based Mental Health Chatbots

Nur Afnis Agustina<sup>a</sup>, Abd. Charis Fauzan<sup>b\*</sup>, Harliana<sup>c</sup>

Departments of Computer Science, Universitas Nahdlatul Ulama Blitar, Kota Blitar, Indonesia

<sup>a</sup>nurafnisagustina30@gmail.com; <sup>b</sup>abdcharis@unublitar.ac.id\*; <sup>c</sup>harliana@unublitar.ac.id

\*Corresponding author

## Article History:

Received 31 January 2025

Reviewed 21 February 2025

Revised 21 March 2025

Accepted 30 May 2025

## ABSTRACT

The high prevalence of mental health issues and limited access to professional information and support have driven the search for innovative solutions. One promising approach is the development of chatbot systems that provide quick and accessible mental health information. This study evaluates the performance of the Bidirectional Long Short-Term Memory (BiLSTM) algorithm in identifying and classifying user inputs within a mental health chatbot system. BiLSTM is chosen for its ability to process sequential data in both directions, allowing it to capture context more effectively than unidirectional models and better understand user intent. Deep learning methods like BiLSTM have also demonstrated higher accuracy compared to traditional machine learning models. This study focuses solely on BiLSTM to evaluate its performance in this context. The mental health dataset used in this study was sourced from previous research published on the GitHub platform and contains 100 classes of mental health-related questions and statements. This dataset was used to train the BiLSTM model to recognize user intent and generate relevant responses. The model achieved 98% accuracy on the training data. For evaluation on the test set, a confusion matrix was used, yielding an accuracy of 82%. The chatbot is implemented as a web-based application using a Python framework and is designed to provide users with insights and knowledge through text-based interactions. These results highlight the potential of the BiLSTM-based chatbot system to deliver effective and efficient mental health information services.

**Keywords:** Chatbot; Mental Health; BiLSTM; Deep Learning; Classification



Lisensi: cc-by-sa

## 1. Introduction

Mental health refers to an individual's psychological state, reflecting their ability to adapt and solve problems, both internally (within oneself) and externally (in their environment) [1]. Just like physical health, mental health plays a crucial role in maintaining a person's quality of life. It encompasses emotional, psychological, and social aspects that influence how individuals interact with their surroundings. Despite its significant impact, mental health often receives insufficient attention. In many developing countries, mental health concerns are overshadowed by issues related to infectious diseases [2]. The World Health Organization (WHO) reports that over 450 million people worldwide suffer from mental health disorders, with cases increasing each year [3]. This issue is exacerbated by low awareness of available services, limited mental health literacy, high social stigma, a shortage of trained professionals, and insufficient specialized facilities. These challenges contribute to delays in diagnosis and treatment [4]. Therefore, preventive measures are essential to help individuals recognize early symptoms, reduce social stigma, prevent severe disorders, and improve access to early treatment. According to [5], increasing knowledge and awareness is a key strategy in preventing mental disorders and strengthening self-protection against potential mental health triggers.

Although mental health disorders affect over 450 million people globally, a major concern lies in the limited access to accurate information and adequate care services. Social stigma often discourages

---

individuals from seeking help, even when experiencing symptoms. Additionally, low mental health literacy makes it difficult for many to recognize early warning signs, leading to delayed diagnosis and worsening conditions. The shortage of mental health professionals and specialized facilities further exacerbates the issue, especially in underserved regions. These challenges highlight the urgent need for accessible and preventative solutions that can educate the public, promote early symptom recognition, and reduce stigma. In this context, chatbot-based digital interventions offer a promising alternative by providing fast, reliable, and 24/7 access to mental health information.

With advancements in technology, one innovative approach to improving mental health awareness is through digital information services. A promising solution is the development of chatbots that provide users with reliable answers to mental health-related queries. A chatbot is a computer program designed to simulate conversations with humans via text or voice interactions [6]. Research findings indicate that chatbots are highly efficient, cost-effective, time-saving, and available 24/7 for information retrieval [7]. In the context of mental health, several studies have shown that chatbots can serve as effective tools for improving mental health literacy. AI-based chatbots can help reduce mental health stigma by facilitating social contact and encouraging users to open up about their experiences [8]. Chatbots also have the potential to provide accessible and scalable mental health interventions, helping users manage conditions like anxiety and depression [9]. Furthermore, they can provide valuable support, especially in underserved populations, by offering timely access to information and promoting mental well-being [10]. However, a major challenge in chatbot development is ensuring that the chatbot can understand and respond to conversations naturally and contextually, mimicking human-like communication. To address this challenge, this study integrates artificial intelligence (AI) techniques, particularly deep learning, which has proven effective in solving complex problems [11].

Deep learning, a subfield of machine learning, utilizes multi-layered neural networks to process and understand complex data. Several deep learning algorithms can be applied to chatbot systems, one of which is Long Short-Term Memory (LSTM). LSTM has been shown to effectively process long and sequential data, making it well-suited for chatbot applications that require contextual understanding [12]. However, LSTM has limitations, as it processes text in only one direction (forward). To overcome this, researchers developed Bidirectional Long Short-Term Memory (BiLSTM), which allows text to be processed in both forward and backward directions, enhancing the chatbot's comprehension ability [13]. A previous study by [14] utilized a dataset containing fictional dialogues from movie scripts and applied LSTM and BiLSTM models. The results demonstrated that the BiLSTM model outperformed the standard LSTM model in terms of accuracy. Previous studies have also implemented LSTM for mental health chatbots but faced challenges such as overfitting due to limited data [15]. However, there is still limited research on the application of BiLSTM for intent classification in mental health domains. This study aims to address these gaps by using BiLSTM with a larger and more varied dataset to enhance performance and contextual understanding.

Based on the discussions above, this study employs the Bidirectional Long Short-Term Memory (BiLSTM) algorithm to develop an advanced chatbot system for mental health using text-based conversational data. By leveraging deep learning, this chatbot aims to provide more contextually aware and accurate responses, ultimately improving access to mental health information. This study utilizes a larger and more diverse dataset, applies hyperparameter tuning, and incorporates comprehensive evaluation metrics (precision, recall, F1-score) to ensure optimal performance and generalization.

## 2. Method

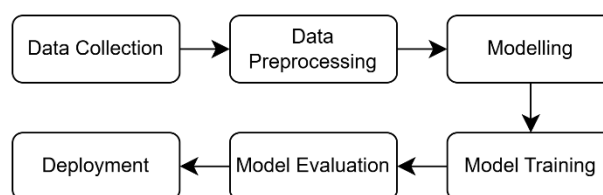


Fig. 1. Research Method

Based on Fig 1, this study begins with data collection from public sources, consisting of conversation data related to mental health. Next, the collected dataset undergoes text preprocessing to clean and prepare raw text data for processing by machine learning algorithms. This preprocessing involves case folding, removing punctuation, tokenizing, padding, and label encoding. Following this, the modeling stage is carried out using the BiLSTM architecture. Once the BiLSTM model is built, training is conducted to optimize the model's weights. After training, the evaluation phase is performed using a confusion matrix, and the model is assessed based on accuracy, precision, recall, and F1-score. Finally, the deployment phase integrates the trained model into an application that can be accessed by users.

## 2.1 Data Collection

The dataset used in this research consists of conversations related to mental health. The dataset is sourced from previous studies available on the GitHub.com platform. This dataset was selected based on its relevance to the topic of mental health issues. It is stored in JSON format, which includes three main elements: tags or labels used to categorize information, patterns that contain variations of sentences users might use to express a specific intent, and responses, which are the chatbot's replies to user inquiries. An example of the dataset as shown in Table 1.

Table 1. Example Dataset

Tag	Patterns	Responses
Greeting	hello haloo hai	Hai! Mindcare disini, ada yang bisa dibantu? ...
Definisi	Apa yang dimaksud dengan gangguan mental? Kesehatan mental itu apa? Apa sih definisi gangguan mental?	Gangguan mental atau kesehatan mental adalah kondiri yang ...

## 2.2 Data Preprocessing

Data preprocessing is carried out to prepare the data. This preprocessing stage begins with data merging, as the data used is obtained from multiple sources. The merging is done by combining the list of intents from each file. Next, data sorting is performed manually to eliminate any identified duplicate entries. The data, initially in JSON format, is then converted into a DataFrame for further text preprocessing, which includes:

- Case folding – the process of converting and standardizing all text to lowercase to ensure that differences in uppercase and lowercase letters do not affect the model [16]. Example: “Saya SEDIH” → “saya sedih”.
- Remove punctuation – the process of removing punctuation marks such as periods (.), commas (,), question marks (?), exclamation marks (!), and other symbols that have no meaningful value [17]. Example: “saya sedih hari ini!!” → “saya sedih hari ini”.

- c. Tokenizing – the process of breaking text into smaller units called tokens [18]. Example: “*saya sedih*” → [“*saya*”, “*sedih*”].
- d. Padding – the process of adding values (usually zeros) to each sequence of numbers so that all sequences have the same length [19]. Example (if max length = 5): [“12”, “7”] → [“12”, “7”, “0”, “0”, “0”].
- e. Label encoding – the process of converting text-based labels into numerical form [20]. Example: “*Stres*” → 0, “*Depresi*” → 1, “*Cemas*” → 2.

Once data preprocessing is complete, the dataset is then split into training data and test data. The training data is used to teach the model to recognize various patterns, while the test data is used to objectively evaluate the model's performance.

### 2.3 Modeling

At this stage, the BiLSTM architecture is used to build the model. Bidirectional Long Short-Term Memory (BiLSTM) is an advanced version of the LSTM algorithm. The LSTM algorithm consists of three gates: forget gate, input gate, and output gate, which function to control the balance between the input stored in the memory cell and the previous state that needs to be discarded [21]. Equations (1), (2), and (3) are applied to each gate in the LSTM architecture.

$$i_t = \sigma(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + b_f) \quad (2)$$

$$O_t = \sigma(W_{xo} \cdot x_t + W_{ho} \cdot h_{t-1} + b_o) \quad (3)$$

BiLSTM combines two LSTM networks that operate in opposite directions. One network functions forward (forward layer), while the other functions backward (backward layer). The output of BiLSTM is a hidden representation that integrates contextual information from both processing directions [22]. In the BiLSTM architecture, the process occurring in the forward LSTM can be expressed as equation (4). Meanwhile, the process in the backward LSTM can be expressed as equation (5). The final output of BiLSTM is a combination of the forward and backward outputs, formulated as equation (6) [23].

$$\vec{h}_t = LSTM(x_t, h_{t-1}) \quad (4)$$

$$\overleftarrow{h}_t = LSTM(x_t, h_{t+1}) \quad (5)$$

$$h_t = [\vec{h}_t, \overleftarrow{h}_t] \quad (6)$$

In this research, the BiLSTM model consists of five layers as follows:

- a. Input layer, serves as the starting point of the network. It receives an integer vector with a length determined by a predefined limit. This vector is obtained from the data preprocessing stage carried out earlier.
- b. Embedding layer, Performs word embedding, which represents words or tokens as dense, high-dimensional numerical vectors [24]. These embedding vectors store semantic relationships between words, allowing the model to understand meaning and contextual connections in text data.
- c. BiLSTM layer, executes the Bidirectional Long Short-Term Memory (BiLSTM) method, which consists of both forward and backward LSTM units. This enables the model to capture context from both past and future words, improving text comprehension.

- d. Dropout layer, helps prevent overfitting by randomly deactivating a certain number of neurons, setting them to zero. This ensures the model does not become overly reliant on specific neurons.
- e. Dense layer, serves as the output layer, utilizing the softmax activation function. This function assigns probability distributions across different class labels, determining the likelihood of each input belonging to a specific class [25].

After that, the model selects the class with the highest probability as the final prediction, representing the most likely class corresponding to the input. Based on this highest-probability class, the chatbot system can retrieve the most suitable response to answer user queries.

#### 2.4 Model Training

After the BiLSTM model is built, the next step is training the model. The main objective of training is to optimize the model's weights so that it can learn patterns from the training data. Before training, the model is configured by specifying several key parameters. These include the loss function, such as categorical\_crossentropy, which calculates the error between the model's predictions and the actual labels, the optimizer such as Adam which updates the model's weights, the learning rate which controls how much the model's weights are updated at each training step, the batch size which determines how many data samples are processed before updating the model's weights, the epoch which defines how many times the model will go through the entire training dataset and the evaluation metric, such as accuracy, which assesses the model's performance during and after training.

#### 2.5 Model Evaluation

The evaluation is conducted using a confusion matrix, as shown in Fig 2. The test data used is separate from the training data to ensure that the model can generate accurate predictions on previously unseen data. The primary objective of evaluating the confusion matrix is to assess the model's performance in classifying test data by comparing the model's predictions against the actual labels. The confusion matrix provides information on the number of correct and incorrect predictions for each class. From this matrix, evaluation metrics such as accuracy, precision, recall, and F1-score can be calculated as percentages using the formulas in equations (7), (8), (9), and (10) [26].

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Fig. 2. Confusion Matrix

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

$$Precision = TP/(TP + FP) \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$F1 - score = \frac{(2*precision*recall)}{(precision+recall)} \quad (10)$$

Where, True Positive (TP) refers to positive data that is correctly predicted. True Negative (TN) refers to negative data that is correctly predicted. False Positive (FP) refers to negative data that is incorrectly predicted as positive. False Negative (FN) refers to positive data that is incorrectly predicted as negative.

## 2.6 Deployment

The trained and evaluated BiLSTM model will be saved as the foundation for the chatbot system. This stored model will be integrated into an interactive web application using the Streamlit framework, which was chosen for its ease of development and deployment of Python-based web applications. As a result, the chatbot can be easily accessed and used by users via a website, from anywhere and at any time.

## 3. Results and Discussion

### 3.1 Data Collection

The dataset used in this study consists of 100 tags, 1508 patterns, and 172 responses, all stored in JSON format. The total number of dataset entries was obtained after performing manual merging and sorting of the data. A sample of the JSON dataset is shown in Fig 3.

```

"tag": "definisi",
"patterns": [
  "Apa yang dimaksud dengan gangguan mental?",
  "Gangguan mental itu apa sih ?",
  "Gangguan mental adalah",
  "Kesehatan Mental itu apa?",
  "kesehatan mental itu apa",
  "kesehatan mental itu apa",
  "kesehatan mental adalah",
  "Mental Health ?",
  "Apa sih Mental Health itu ?",
  "Definisi Kesehatan Mental ",
  "Definisi Mental Health ?",
  "Pengertian Kesehatan Mental itu apa ?",
  "Definisi gangguan mental ?",
  "Apa sih definisi gangguan mental ?",
  "bisa jelaskan apa itu kesehatan mental ngga ?",
  "kesehatan mental",
  "penyakit kesehatan mental adalah",
  "kamu tau ngga apa itu kesehatan mental ?",
  "kamu tau ngga apa itu mental health",
  "apa itu mental health",
  "apa itu kesehatan mental",
  "kamu tau apa itu kesehatan mental ?"
],
"responses": [
  "Gangguan mental atau kesehatan mental adalah kondisi yang mempengaruhi pikiran, perasaan, dan perfi

```

Fig. 3. JSON Dataset

Next, the dataset, originally in JSON format, will be converted into a dataframe to ensure compatibility with the machine learning model. This transformation is crucial because machine learning models typically operate with structured data in a tabular format, making it easier to process, analyze, and train the model effectively. The result of converting the JSON data into a DataFrame is shown in Fig 4.

	tag	patterns	responses
0	greeting	[hallo, hello, helo, hai, haloo, halooo, halo,...	[Hail Mindcare disini, ada yang bisa dibantu? ...
1	Mindcare	[Apa itu Mindcare?, Apa itu Mindcare?, Siapa M...	[Mindcare adalah aplikasi chatbot yang dibuat ...
2	bantu_Mindcare	[kamu bisa bantu aku?, kamu bisa bantuin aku?...	[Apa yang bisa Mindcare bantu?, Kamu butuh ban...
3	sedang_apa	[kamu lagi ngapain Mindcare?, lagi ngapain?, s...	[Mindcare sedang ngobrol dengan kamu :], Mindc...
4	tanya	[aku mau tanya, aku mau tanya boleh?, tanya, a...	[Iya sahabat Mindcare, mau tanya apa?, Kamu ma...
...	...	...	...
95	cara_penanganan OCD	[Cara mengatasi OCD, Bagaimana mengobati gangg...	[Penanganan OCD melibatkan berbagai pendekatan...
96	stigma_dan_perasaan_bersalah	[stigma dan perasaan bersalah, stigma dan pera...	[Stigma terhadap gangguan kesehatan mental mas...
97	dukungan_gangguan_mental	[dukungan individu, dukungan individu untuk ga...	[Dukungan bagi individu yang mengalami ganggua...
98	dukungan_teman	[dukungan teman, dukungan teman untuk gangguan...	[Dukungan dari teman adalah berarti bagi indiv...
99	peran_masyarakat	[peran masyarakat, peran masyarakat dalam kese...	[Peran masyarakat sangat penting dalam menjaga...

Fig. 4. Dataframe Dataset

### 3.2 Text Preprocessing

After the data is converted into a dataframe, the next crucial step is text preprocessing, which aims to clean and prepare the data for more efficient processing and analysis. This step is particularly important for the 'pattern' column, which serves as the primary input for the model. The preprocessing process consists of three main steps. First, case folding is applied to convert all letters into lowercase, ensuring text uniformity. Second, punctuation removal is performed to eliminate irrelevant symbols and punctuation marks that do not contribute to the analysis. Third, tokenization is carried out by breaking the text into individual words or tokens, allowing the model to process them more effectively. By implementing these preprocessing techniques, the efficiency and accuracy of the BiLSTM model can be significantly improved while reducing training complexity. The cleaned and standardized data will help the model better recognize patterns and generate more accurate responses in the chatbot system. The results of these preprocessing steps are shown in Table 2.

Table 2. Result of Preprocessing

Before	Method	After
<i>Bagaimana cara mengurangi risiko bunuh diri?</i>	Case Folding	<i>bagaimana cara mengurangi risiko bunuh diri?</i>
	Remove Punctuation	<i>bagaimana cara mengurangi risiko bunuh diri</i>
	Tokenizing	Text token=[' <i>bagaimana</i> ', ' <i>cara</i> ', ' <i>menguurangi</i> ', ' <i>risiko</i> ', ' <i>bunuh</i> ', ' <i>diri</i> '] Token=[11, 6, 134, 135, 43, 33]

The next step in text preprocessing is padding, which aims to standardize the sequence length of the data. Since text data naturally varies in length, it needs to be adjusted to a uniform length to ensure efficient processing by the model. This sequence length adjustment, known as padding, is crucial for maintaining data consistency during model training. In this study, the maximum sequence length is set to 15, meaning that all sequences whether shorter or longer will be modified to have a fixed length of 15. Shorter sequences will be padded with additional elements, while longer sequences will be truncated. Padding ensures that all input data maintains the same dimensions, allowing the model to process it more efficiently and effectively. The result of the padding process is shown in Fig 5.

```
array([[ 0,  0,  0, ...,  3,  2, 74],
       [ 0,  0,  0, ...,  0, 99, 40],
       [ 0,  0,  0, ...,  2,  8, 74],
       ...,
       [ 0,  0,  0, ...,  6, 366, 92],
       [ 0,  0,  0, ..., 236,  5,  2],
       [ 0,  0,  0, ...,  95,  5, 75]], dtype=int32)
```

Fig. 5. Result of Padding

The final step in text preprocessing is label encoding, which is specifically applied to the 'tag' column. This process aims to convert labels or tags, which are typically in the form of words or phrases, into numerical representations. This transformation is essential because machine learning models, such as BiLSTM, generally operate with numerical data. Label encoding assigns a unique numerical value to each unique label, allowing the model to process and interpret categorical information more effectively. The results of the label encoding process can be seen in Fig 6, where each label in the 'tag' column has been converted into its corresponding numeric representation. With this transformation, both text data and labels have been structured into a format that is ready for training the BiLSTM model.

```
array([85, 41, 85, ..., 12, 86, 59])
```

Fig. 6. Result of Encoding

After completing the text preprocessing stage, the dataset is divided into two parts: training data and testing data. The split is conducted with a ratio of 80% for training data and 20% for testing data. Based on this proportion, the dataset consists of 1,207 samples for training and 302 samples for testing. This division ensures that the model has sufficient data for learning while also providing a separate dataset to evaluate its performance and generalization ability.

### 3.3 Modeling

The modeling results using the BiLSTM algorithm are shown in Fig 7. Below is an explanation of each layer in the model:

- Input layer, receives input with dimensions (None, 15), where None represents a flexible batch size, and 15 is the sequence length or the number of input tokens.
- Embedding layer, maps tokens from the input layer into a 100-dimensional vector representation, resulting in an output of (None, 15, 100).
- BiLSTM layer, takes the output from the embedding layer as input, processes it in both directions to capture contextual meaning, and combines the information into a single vector with a dimension of 64.
- Dropout layer, takes the BiLSTM output as input and randomly deactivates 50% of the elements during training. This prevents overfitting and encourages the model to generalize better. Since dropout only affects training and not the data structure, the output remains a 64-dimensional vector, but with some elements set to zero.
- Dense layer, serves as the output layer of the model. It receives the vector representation from the previous layer and applies the softmax activation function, producing an output of (None, 100), representing the probability distribution over 100 classes (tags). The probability values indicate the model's confidence that the user's input belongs to each class. The class with the highest probability is selected as the model's final prediction.

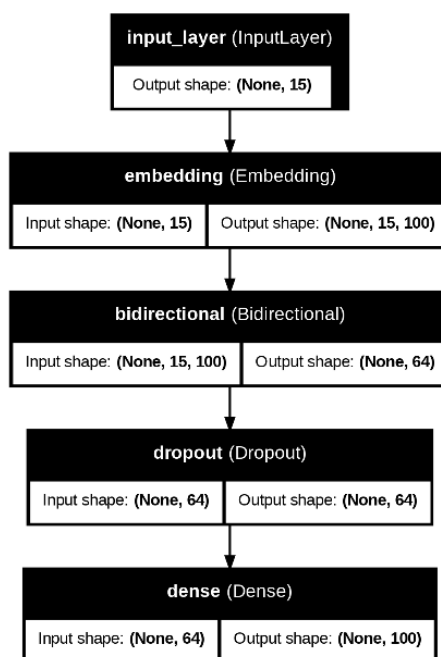


Fig. 7. Result of Modeling

### 3.4 Model Training

After the modeling stage is completed, the next step is configuring the BiLSTM model parameters for training. Prior to finalizing the configuration, several experiments were conducted to determine the most optimal combination of hyperparameters. Based on the evaluation results, the best performance was achieved using the following settings: the loss function Sparse Categorical Crossentropy to measure the model's error, the evaluation metric accuracy to assess performance during training, and the Adam optimizer for updating the model's weights. A learning rate of 0.001 was selected to control the weight update step size, batch size was set to 16 indicating that training updates occur every 16 samples, and the model was trained for 100 epochs, meaning it processed the entire dataset 100 times. The training process results are stored in the history variable and are visualized in Fig 8.

```

38/38 ----- 1s 15ms/step - accuracy: 0.9932 - loss: 0.0536
Epoch 98/100
38/38 ----- 1s 16ms/step - accuracy: 0.9888 - loss: 0.0666
Epoch 99/100
38/38 ----- 1s 15ms/step - accuracy: 0.9900 - loss: 0.0728
Epoch 100/100
38/38 ----- 1s 15ms/step - accuracy: 0.9887 - loss: 0.0733
CPU times: user 1min 41s, sys: 2.65 s, total: 1min 44s
Wall time: 1min 32s
  
```

Fig. 8. Result of Model Training

Based on Fig 8, at the 100 epoch, the model achieved a very high training accuracy of 0.9887 with a significantly low loss value of 0.0733. This indicates that the model has successfully learned patterns and relationships within the training data very well. To visualize the training process, accuracy and loss graphs were generated over 100 epochs, as shown in Fig 9 and Fig 10. These graphs provide a clear representation of the accuracy improvement and loss reduction as the number of epochs increases, demonstrating an effective learning process.

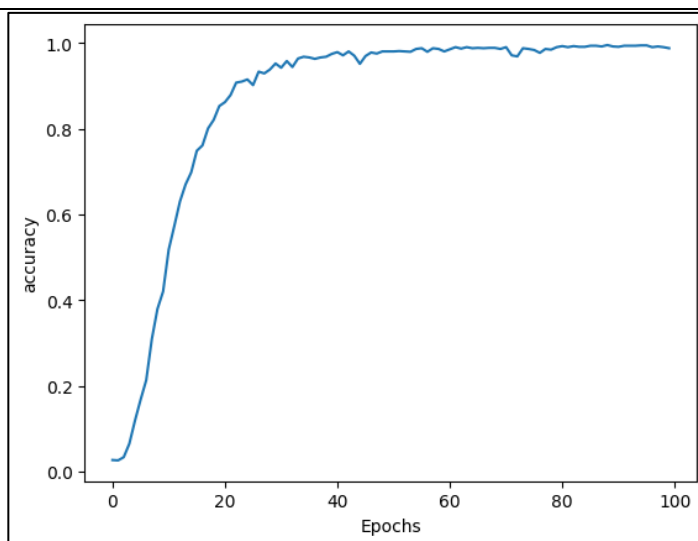


Fig. 9. Accuracy Graph

The results of the model training indicate that the model has been well-trained and exhibits high performance on the training data. The model has effectively learned patterns and features within the data, achieving high accuracy and low loss. However, performance on training data does not always reflect the model's performance on unseen data.

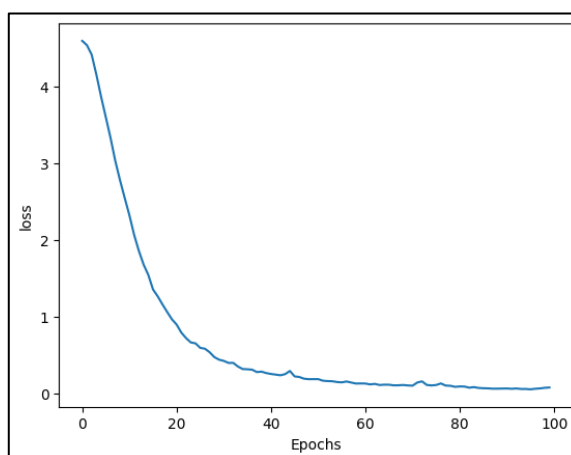


Fig. 10. Loss Graph

### 3.5 Model Evaluation

After completing the model training, the next step is model evaluation using a confusion matrix, with the results shown in Fig 11. Additionally, the model's performance is evaluated on a separate test data that the model has not seen before. The evaluation metrics, including accuracy, precision, recall, and F1-score, are presented in Table 3. By comparing the performance on the test data with the training results, we can assess the model's ability to generalize to new, unseen data. The model's performance on the test data confirms the consistency and reliability of the training outcomes.

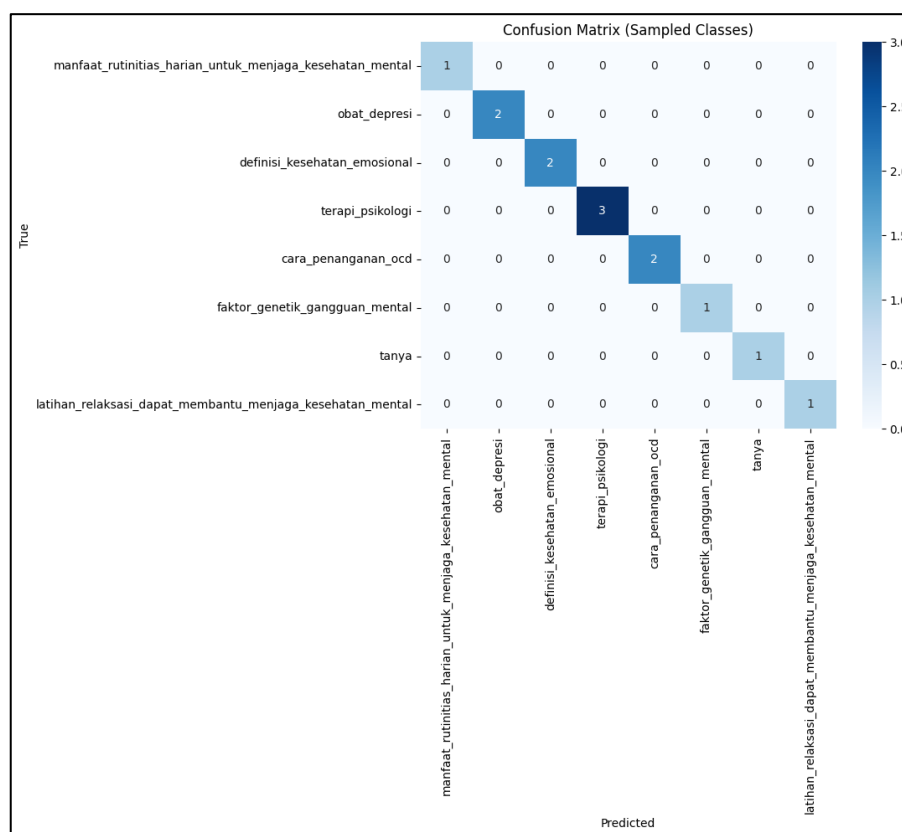


Fig. 11. Confusion Matrix of BiLSTM

Table 3. Classification Report (Sampled Classes)

Class	Precision	Recall	F1-Score	Support
<i>manfaat_rutinitias_harian_untuk_menjaga_kesehatan_mental</i>	1.000	1.000	1.000	1
<i>obat_depresi</i>	0.667	1.000	0.800	2
<i>definisi_kesehatan_emosional</i>	1.000	1.000	1.000	2
<i>terapi_psikologi</i>	1.000	1.000	1.000	3
....	....	....	....	....
<i>cara_penanganan_OCD</i>	1.000	1.000	1.000	2
<i>faktor_genetik_gangguan_mental</i>	1.000	1.000	1.000	1
<i>tanya</i>	1.000	1.000	1.000	1
<i>latihan_relaksasi_dapat_membantu_menjaga_kesehatan_mental</i>	0.500	0.500	0.500	2
accuracy			0.825	302
macro avg	0.837	0.848	0.820	302
weighted avg	0.858	0.825	0.823	203

Fig 11 and Table 3 display only a small subset of the 100 classes to simplify the visualization. However, the evaluation metrics calculations still include all 100 classes. The dataset already includes 100 diverse classes, providing a broad range of mental health topics for evaluation. While this ensures variability in the analysis, adding additional test classes that cover more specific or less common mental health conditions could further enhance the model's ability to generalize to a wider range of real-world scenarios. However, the current set of 100 classes is already sufficient for the purpose of this study. The evaluation results indicate that the BiLSTM algorithm achieved strong overall performance, with an

accuracy of 82.5%, an average macro precision of 83.7%, an average macro recall of 84.8%, and an average macro F1-score of 82.0%. The BiLSTM model demonstrated exceptional performance in classifying dominant classes such as 'emotional health definition', 'psychological therapy', and 'benefits of daily routines for mental health', achieving a perfect F1-score of 1.00. Additionally, BiLSTM performed well on the 'antidepressant medication' class, with an F1-score of 0.80. This suggests that the model's performance may be influenced by the amount of available data for each class, highlighting the importance of balancing class distribution to improve classification accuracy, especially for minority classes.

### 3.6 Deployment

The results from the chatbot system deployment phase, including its user interface and functionality, are presented in Fig 12. The chatbot is accessed via a website with a simple and user-friendly design, featuring a text input area for users and an output area displaying chatbot responses.



Fig. 12. Chatbot Interface

## 4. Conclusion

This study successfully implemented the Bidirectional Long Short-Term Memory (BiLSTM) algorithm in the development of a chatbot for mental health information services. The BiLSTM model demonstrated strong performance during training, achieving 98% accuracy. However, its accuracy on test data dropped to 82%, indicating that the model needs improvement in adapting to new data. Nevertheless, an 82% accuracy still highlights the model's potential in correctly classifying user inputs. To further enhance chatbot performance, future research is recommended to utilize a larger and more diverse dataset, as well as implement various optimization techniques to improve the model's capability. The development of this BiLSTM-based chatbot is expected to provide more accessible and responsive mental health information services for users.

## 5. References

- [1] F. Anwar and P. Julia, "Analisis Strategi Pembinaan Kesehatan Mental Oleh Guru Pengasuh Sekolah Berasrama Di Aceh Besar Pada Masa Pandemi," *J. Edukasi J. Bimbingan. Konseling*, vol. 7, no. 1, pp. 64–83, 2021.
- [2] I. A. Ridlo, "Pandemi COVID-19 dan Tantangan Kebijakan Kesehatan Mental di Indonesia,"

- 
- Insa. J. Psikol. dan Kesehat. Ment.*, vol. 5, no. 2, p. 162, 2020, doi: 10.20473/jpkm.v5i22020.162-171.
- [3] Alini and L. N. Meisyalla, "Gambaran Kesehatan Mental Remaja SMPN Bangkinang Kota Kabupaten Kampar," *J. Ners*, vol. 6, no. 23, pp. 80–85, 2022, [Online]. Available: <http://journal.universitaspahlawan.ac.id/index.php/ners>
- [4] C. T. Setiawan, S. G. Sijabat, Ervan, and Habibi, "Menjembatani Kesenjangan dalam Perawatan Kesehatan Mental: Pendekatan Baru untuk Diagnosis, Pengobatan, dan Pengurangan Stigma," *J. Multidisiplin West Sci.*, vol. 2, no. 08, pp. 660–667, 2023, doi: 10.58812/jmws.v2i08.579.
- [5] H. D. Kurniawan and A. L. Dwi, "Upaya Meningkatkan Kesehatan Mental Di Kalangan Remaja," *J. Abdi Masy.*, vol. 2, no. 2, pp. 564–568, 2024.
- [6] M. Fauzan, M. P. Wicaksana, and P. G. Rahardandi, "Literatur Review Penggunaan Chatbot Untuk Layanan Informasi," vol. 4, pp. 8316–8323, 2024.
- [7] M. E. Rianto and A. Furqon, "Impelementasi AI Chatbot Sebagai Support Assistant Website Universitas Nurul Jadid Menggunakan Algoritma BiLSTM".
- [8] Y. C. Lee, Y. Cui, J. Jamieson, W. Fu, and N. Yamashita, "Exploring Effects of Chatbot-based Social Contact on Reducing Mental Illness Stigma," *Conf. Hum. Factors Comput. Syst. - Proc.*, no. April, 2023, doi: 10.1145/3544548.3581384.
- [9] M. Casu, S. Triscari, S. Battiato, L. Guarnera, and P. Caponnetto, "AI Chatbots for Mental Health: A Scoping Review of Effectiveness, Feasibility, and Applications," *Appl. Sci.*, vol. 14, no. 13, 2024, doi: 10.3390/app14135889.
- [10] M. C. Klos, M. Escoredo, A. Joerin, V. N. Lemos, M. Rauws, and E. L. Bunge, "Artificial intelligence-based chatbot for anxiety and depression in university students: Pilot randomized controlled trial," *JMIR Form. Res.*, vol. 5, no. 8, 2021, doi: 10.2196/20678.
- [11] M. Ferian, R. Akbari, B. Rahayudi, and L. Muflikhah, "Implementasi Deep Learning menggunakan Algoritma EfficientDet untuk Sistem Deteksi Kelayakan Penerima Bantuan Langsung Tunai berdasarkan Citra Rumah di Wilayah Kabupaten Kediri," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 4, pp. 1817–1825, 2023.
- [12] M. H. Chatbot, S. P. Afrisia, F. M. Hana, and W. C. Wahyudin, "Implementasi Metode Long Short Term Memory ( LSTM ) pada Chatbot Kesehatan Mental Mahasiswa," vol. 21, no. 2, pp. 107–116, 2024, doi: 10.30595/sainteks.v21i2.23869.
- [13] D. R. Alghifari, M. Edi, and L. Firmansyah, "Implementasi Bidirectional LSTM untuk Analisis Sentimen Terhadap Layanan Grab Indonesia," *J. Manaj. Inform.*, vol. 12, no. 2, pp. 89–99, 2022, doi: 10.34010/jamika.v12i2.7764.
- [14] P. Anki and A. Bustamam, "Measuring the accuracy of LSTM and BiLSTM models in the application of artificial intelligence by applying chatbot programme," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, no. 1, pp. 197–205, 2021, doi: 10.11591/ijeecs.v23.i1.pp197-205.
- [15] F. Zakariya, J. Zeniarja, and S. Winarno, "Pengembangan Chatbot Kesehatan Mental Menggunakan Algoritma Long Short-Term Memory," *J. Media Inform. Budidarma*, vol. 8, no. 1, p. 251, 2024, doi: 10.30865/mib.v8i1.7177.
- [16] Y. Caesar, I. Sabastian, A. Kindarto, and A. Fathurrohman, "Analisis Sentiment Masyarakat Terhadap Clash of Champions Ruang Guru Menggunakan Metode Support Vector Machine ( SVM )," pp. 820–838, 2024.
- [17] M. A. C, "Analisis Sentimen pada Ulasan Penyedia Layanan Sentiment Analysis on Service Provider Reviews Using the C4 . 5 Algorithm," vol. 6, no. 2, pp. 1–5, 2023.
- [18] K. Setiadi, "Komputa : Jurnal Ilmiah Komputer dan Informatika Shopeefood Pada Media Sosial Twitter Menggunakan Algoritma Naive Bayes dan Support Vector Machine ( SVM ) Komputa : Jurnal Ilmiah Komputer dan Informatika," vol. 12, no. 1, pp. 29–38, 2023.
- [19] Aripin, Steven Adi Santoso, and Hanny Haryanto, "Mengoptimalkan Akurasi pada Klasifikasi Emosi Majemuk Berdasarkan Semantik Kalimat Menggunakan XLM-RoBERTa," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 12, no. 1, pp. 29–36, 2023, doi: 10.22146/jnteti.v12i1.6084.
- [20] F. A. Oktavirahani and R. Maharesi, "Implementasi Algoritma Decision Tree Cart Untuk

- Merekomendasikan Ukuran Baju," *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 1, p. 138, 2022, doi: 10.30865/jurikom.v9i1.3838.
- [21] L. Augustyniak, T. Kajdanowicz, and P. Kazienko, "Aspect detection using word and char embeddings with (Bi) LSTM and CRF," *Proc. - IEEE 2nd Int. Conf. Artif. Intell. Knowl. Eng. AIKE 2019*, no. August, pp. 43–50, 2019, doi: 10.1109/AIKE.2019.00016.
- [22] E. Subowo, F. Adi Artanto, I. Putri, and W. Umaedi, "Algoritma Bidirectional Long Short Term Memory untuk Analisis Sentimen Berbasis Aspek pada Aplikasi Belanja Online dengan Cicilan," *J. Fasilkom*, vol. 12, no. 2, pp. 132–140, 2022, doi: 10.37859/jf.v12i2.3759.
- [23] D. I. Puteri, "Implementasi Long Short Term Memory (LSTM) dan Bidirectional Long Short Term Memory (BiLSTM) Dalam Prediksi Harga Saham Syariah," *Euler J. Ilm. Mat. Sains dan Teknol.*, vol. 11, no. 1, pp. 35–43, 2023, doi: 10.34312/euler.v11i1.19791.
- [24] A. Nurdin, B. Anggo Seno Aji, A. Bustamin, and Z. Abidin, "Perbandingan Kinerja Word Embedding Word2Vec, Glove, Dan Fasttext Pada Klasifikasi Teks," *J. Tekno Kompak*, vol. 14, no. 2, p. 74, 2020, doi: 10.33365/jtk.v14i2.732.
- [25] R. Faurina, M. J. Gazali, and I. D. A. Herani, "Implementasi Deep Feed-Forward Neural Network pada Perancangan Chatbot Berbasis Web di UPPIK RSUD M. YUNUS," *Komputika J. Sist. Komput.*, vol. 12, no. 2, pp. 11–20, 2023, doi: 10.34010/komputika.v12i2.8914.
- [26] D. P. Sidik, F. Utamingrum, and L. Muflikhah, "Penggunaan Variasi Model pada Arsitektur EfficientNetV2 untuk Prediksi Sel Kanker Serviks," ... *Teknol. Inf. dan Ilmu ...*, vol. 7, no. 5, pp. 2116–2121, 2023, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/12656>