Journal of Applied Engineering and Technological Science
Vol 6(1) 2024: 225-237

SARA DETECTION ON SOCIAL MEDIA USING DEEP LEARNING ALGORITHM DEVELOPMENT

M. Khairul Anam[1*], Lucky Lhaura Van FC[2], Hamdani Hamdani[3], Rahmaddeni[4], Junadhi[5], Muhammad Bambang Firdaus[6], Irwanda Syahputra[7], Yuda Irawan[8]
Department of Informatics, Universitas Samudra, Langsa, Indonesia[1,7]
Department of Informatics Engineering, Universitas Lancang Kuning, Pekanbaru, Indonesia[2]
Department of Informatics, Universitas Mulawarman, Samarinda, Indonesia[3,6]
Department of Informatics Engineering, Universitas Sains dan Teknologi Indonesia, Pekanbaru, Indonesia[4,5]
Department of Informatics Engineering, Universitas Hang Tuah Pekanbaru, Indonesia[8]
khairulanam@unsam.ac.id[1]

Received: 13 June 2024, Revised: 04 October 2024, Accepted: 09 October 2024
**Corresponding Author*

ABSTRACT
Social media has become a key platform for disseminating information and opinions, particularly in Indonesia, where SARA (Ethnicity, Religion, Race, and Intergroup) issues can fuel social tensions. To address this, developing an automated system to detect and classify harmful content is essential. This study develops a deep learning model using Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) to detect SARA-related comments on Twitter. The method involves data collection through web scraping, followed by cleaning, manual labeling, and text preprocessing. To address data imbalance, SMOTE (Synthetic Minority Over-sampling Technique) is applied, while early stopping prevents overfitting. Model performance is evaluated using precision, recall, and F1-score. The results demonstrate that SMOTE significantly improves model performance, particularly in detecting minority-class SARA comments. CNN+SMOTE achieves a accuracy of 93%, and BiLSTM+SMOTE records a recall of 88%, effectively capturing patterns in SARA and non-SARA data. With SMOTE and early stopping, the model successfully manages class imbalance and reduces overfitting. This research supports efforts to curtail hate speech on social media, especially in the Indonesian context, where SARA-related issues often dominate public discourse.
Keywords: *Deep Learning, SARA Comments, SARA Detection, SMOTE, Social Media Classification.*

1. Introduction

Social media has become a primary space for users to express their opinions and views (Bailey et al., 2020; Konovalova et al., 2023). Among the various platforms, Twitter is one of the most frequently used to discuss current social and political issues (Casero-Ripollés, 2021). In Indonesia, especially in the context of the 2024 presidential election, Twitter has become the primary platform for netizens to voice support, criticism, or neutral opinions regarding presidential candidates. However, behind these expressions, comments often appear containing elements of SARA (Ethnicity, Religion, Race, and Intergroup), which have the potential to trigger social conflict and even violence (Prathama et al., 2022). Therefore, detecting and classifying comments related to SARA is important to prevent the escalation of social tensions during crucial periods such as the presidential election.

Although there have been several studies discussing the detection of hate speech and SARA on social media, the main challenge that still needs to be solved is the imbalance of data classes. For example, research conducted by (Malik et al., 2021) used Bidirectional Long Short-Term Memory (LSTM) to detect hate speech with an accuracy of 82%, while research by (Omran et al., 2023) used a Support Vector Machine (SVM) achieved an accuracy of 90%. Then, another study using a Convolutional Neural Network (CNN) obtained an accuracy of 95% (Fauzy & Setiawan, 2023). Furthermore, research conducted by (Rudiyanto & Setiawan, 2024) using CNN and Particle Swarm Optimization algorithms obtained an accuracy of 78%. Other researchers also used the Bidirectional Long Short–Term Memory Neural Network (BiLSTM) to detect hate speech and obtained an accuracy of 80% (Aurora et al., 2023). However, these studies have not completely resolved the problem of class imbalance, where the number of

classes in one is less than the other classes. This imbalance causes the classification model to tend to be more accurate in predicting the majority class but weak in detecting the minority class.

Previous studies have also yet to explore the use of deep learning algorithms such as CNN and BiLSTM fully to overcome this imbalance problem. While CNN and BiLSTM have been widely used in text classification, only a few have utilized oversampling techniques such as the Synthetic Minority Over-sampling Technique (SMOTE) to improve model performance on the minority class (Sharmin et al., 2020). In addition, few studies have integrated early stopping techniques to prevent overfitting, especially in the context of classifying comments containing SARA elements on Twitter.

In this study, we use deep learning algorithms CNN and BiLSTM to classify Twitter comments related to the 2024 presidential candidates, those containing SARA and those not. To overcome the problem of data imbalance, we apply the SMOTE method and early stopping techniques to prevent overfitting in the model (Anam et al., 2024). CNN and BiLSTM were chosen because of their proven ability to handle Twitter text, where the CNN model is very effective in identifying patterns in text (Aji et al., 2024), while BiLSTM can capture context more deeply by processing data in two directions (forward and backward) (Li et al., 2022).

This study aims to fill the gap in the existing literature by providing a more effective solution to detect racially and ethnically sensitive comments on Twitter during the 2024 presidential election. We also hope to develop a more balanced classification approach by applying SMOTE and early stopping, which are expected to improve the model's performance in detecting racially and ethnically sensitive comments.

2. Literature Review

Research on the detection of hate speech and SARA has employed various machine learning and deep learning algorithms. Each algorithm in the table below has produced different levels of accuracy, depending on the context and methods applied. It is essential to understand why each algorithm performs at a certain level and how these approaches are relevant for detecting hate speech or SARA-related content. Table 1 presents previous research on the detection of hate speech.

Table 1 - Previous Research Related to Detection

Researcher	Algorithm	Object	Accuracy
(Malik et al., 2021)	LSTM	Hate Speech Detection	82%
(Lyrawati, 2022)	GRLVQ	Hate Speech Detection	70.74%
(Omran et al., 2023)	SVM	Hate Speech Detection	90%
(Siddiqui et al., 2021)	LSTM	Hate Speech Detection	72%
(Taradhita & Putra, 2021)	CNN	Hate Speech Classification	82.5%
(Aurora et al., 2023)	BiLSTM	Hate Speech Detection	80.25%
This Research	CNN + SMOTE + Early Stopping	SARA Detection	-
This Research	BiLSTM + SMOTE + Early Stopping	SARA Detection	-

Previous studies have used various algorithms to detect hate speech, each with its advantages and disadvantages. LSTM (Long Short-Term Memory), used by Malik et al. (2021) and Siddiqui et al. (2021), showed accuracy results of 82% and 72%. LSTM is a recurrent neural network designed to process sequential data, such as text while preserving temporal information from the sequence of words. The advantage of LSTM is its ability to handle long sequences and understand context, which is important for detecting hate speech. However, the disadvantage of LSTM lies in its slow training speed and sensitivity to imbalanced data, so its performance may decrease when faced with biased datasets.

GRLVQ (Generalized Relevance Learning Vector Quantization), used by Lyrawati (2022), recorded an accuracy of 70.74%. GRLVQ is a prototype-based algorithm that learns feature relevance to perform classification. While this algorithm is useful for smaller or structured datasets, it could be more effective for handling unstructured and complex text data, such as those found in hate speech on social media. This algorithm is less good at recognizing complex patterns than deep learning techniques such as CNN or LSTM.

Research by Omran et al. (2023) used SVM (Support Vector Machine) and showed the highest accuracy of 90%. SVM is effective in binary classification and can work well on small to medium datasets. SVM maximizes the margin between different classes, thus providing accurate classification results. However, SVM is only sometimes ideal for large or imbalanced datasets, as this algorithm tends to have difficulty handling data that is too much from one class compared to another, such as hate speech detection or SARA.

CNN (Convolutional Neural Network), used by Taradhita and Putra (2021), achieved an accuracy of 82.5% in hate speech classification. CNN is usually used for pattern recognition in visual data, but it also effectively captures local text features. CNN uses convolutional layers to extract features from text and recognize patterns that may be related to hate speech. However, CNN could be more optimal in understanding the context of word sequences, which can be a challenge in detecting contextual hate speech, such as SARA.

In a recent study, Aurora et al. (2023) used BiLSTM (Bidirectional Long Short-Term Memory) and achieved an accuracy of 80.25%. BiLSTM is a variant of LSTM that processes text in two directions, forward and backward, to capture the context of words from both sides. BiLSTM is very effective for processing text data with relationships between words in different orders. This algorithm is superior to standard LSTM in understanding the overall context of a sentence, especially in the case of hate speech, which requires a deeper understanding.

This study used CNN and BiLSTM to detect comments containing SARA elements on Twitter. These two algorithms are combined with the SMOTE (Synthetic Minority Over-sampling Technique) technique to overcome the problem of data imbalance, where comments containing SARA elements are fewer than non-SARA comments. SMOTE works by creating synthetic samples from the minority class so that the model has more data to learn patterns from racially motivated comments (Herianto et al., 2024). In addition, an early stopping technique is applied to prevent overfitting during model training. Early stopping stops training when the model does not improve the validation data so the model can still generalize well on new data.

This literature review shows that each algorithm has strengths and weaknesses in detecting hate speech or racially motivated speech. Algorithms such as SVM provide high accuracy for balanced datasets, while BiLSTM is superior in handling the context of word sequences. Using SMOTE and early stopping is expected to improve the algorithm's performance in this study, especially in handling imbalanced datasets and preventing overfitting.

3. Research Methods

Research methodology is the flow or stages that will be carried out in this research, which has been arranged according to the object to be studied. With the research methodology, it is hoped that the research will run as it should so that it can get quality results. The stages of research methodology are adjusted to the needs of the research. Figure 1 shows the flow of research that will be carried out in this study.

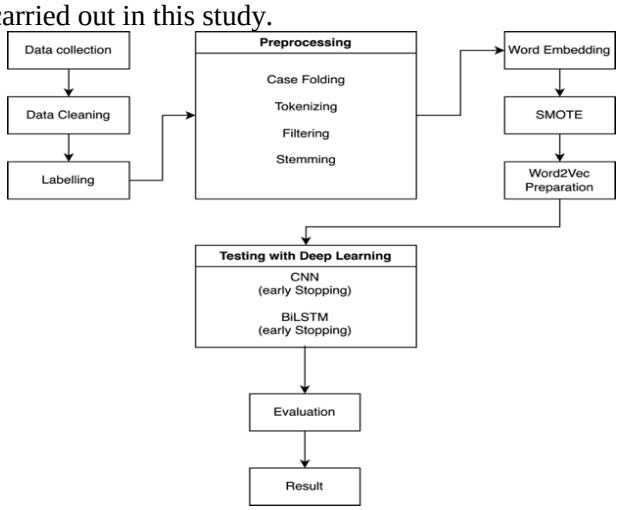

Fig. 1. Development of CNN and LSTM Models

A. Dataset

In this study, data collection was carried out using web scraping techniques to collect tweets related to the 2024 presidential election in Indonesia. The scraping process was carried out using Python's Scrapy and Tweepy libraries. Scrapy facilitates efficient data retrieval from web pages, while Tweepy utilizes the Twitter API to access tweets directly. We used keywords such as "2024 presidential election" and "2024 presidential candidates" to pull relevant tweets during a certain period, namely from November 2022 to March 2023.

From an ethical perspective, all data collected is public and openly available on Twitter. However, researchers remain careful in handling sensitive data and avoid collecting personal information or user identities. In addition, this study follows data-driven research ethics guidelines and does not use information that could violate user privacy.

B. Labelling data and Preprocessing

Data collected through scraping requires further cleaning to remove irrelevant elements, such as mentions, hashtags, retweets, symbols, links, numbers, and emoticons. This cleaning process uses the re (regular expression) and pandas libraries to manipulate the data efficiently.

After the data is cleaned, the research team manually reviews each tweet. Tweets containing SARA elements are labelled as "SARA," while others are labeled "Non-SARA." The labeling criteria are based on local and international guidelines on hate speech and identity-based discrimination. We use previous hate speech cases as a guide in determining whether a tweet contains SARA elements or not.

The labeled data then goes through several stages of text preprocessing. The first step is case folding, where all text is converted to lowercase to maintain consistency (P. P. Putra et al., 2024). The next step is tokenization, where text is broken down into individual words using the NLTK library (Begum & Sree, 2023). After that, we filtered to remove unimportant words, such as conjunctions or articles, using a list of stop words from the same library (R. S. Putra et al., 2022).

Stemming was also used to reduce words to their basic form (Abidin et al., 2024). This stemming process uses Sastrawi, a stemming library specifically for Indonesians. This preprocessing technique was chosen to increase the model's efficiency in learning patterns from text and reduce the ambiguity that may arise due to variations in word usage or word forms (Rianto et al., 2021).

C. *Use of Word2vec*

We used Word2Vec with the Skip-gram approach for text representation, which can map each word into a vector. The Word2Vec training process was carried out with the Gensim library. The main parameters used in Word2Vec are as follows: the embedding size is 200, the window size is 5, the minimum word count is 2, and the skip-gram method is chosen to better capture the word's context. This configuration is validated using a test dataset to ensure the resulting vector representation is accurate enough to represent the SARA context.

D. SMOTE

To address the class imbalance between SARA and non-SARA data, we use the SMOTE. SMOTE generates synthetic samples from the minority class (SARA) by creating new data based on existing samples (Herianto et al., 2024). The parameters used in SMOTE include k-nearest neighbors (K), set at five so that each minority sample produces five nearest neighbors. This is expected to balance the data distribution between the SARA and non-SARA classes so that the model is not too biased towards the majority class.

E. Model Training

Model training uses two deep learning algorithms: CNN (Convolutional Neural Network) and BiLSTM (Bidirectional Long Short-Term Memory).

a. CNN

Convolutional Neural Network (CNN) is a deep learning algorithm widely used for image classification. However, this algorithm can also be applied to data in word form (Dharani et al., 2023; Muis et al., 2023). The CNN layers used are Conv1D, MaxPooling1D, and GlobalMaxPooling1D layers. The first thing to do is create instances of the Sequential class so that models can be built sequentially. The following CNN architecture used in this study is shown in Figure 2:

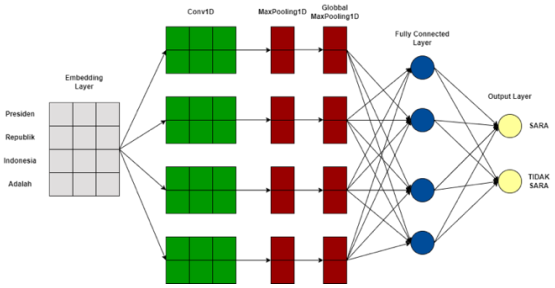

Fig. 2. Architecture CNN

Figure 2 is a CNN architecture focusing on text; the input layer in CNN for text consists of a vector representation of words in the text. This study uses Word2Vec, which converts words into numeric vectors. This layer receives text that has been processed and converted into a form that the CNN model can understand.

Next, the text converted into a vector will go through a convolution layer. This layer aims to capture important patterns or features of the text, such as phrases or words that often appear together. These features are important for detecting patterns related to hate speech or SARA. After the convolution layer, the data will be processed by the pooling layer, which reduces dimensions and filters out irrelevant information. Pooling helps the network more efficiently identify patterns without losing important information.

After the convolution and pooling processes, the data is forwarded to the fully connected layer, where all neurons are connected to the previous layer to compile the information taken from the text. This layer is tasked with making the final prediction, for example, whether the text is included in the SARA or non-SARA category. The model then produces this prediction in the output layer, providing the final classification decision based on previous training.

b. BiLSTM

Next is the BiLSTM algorithm used in this study.

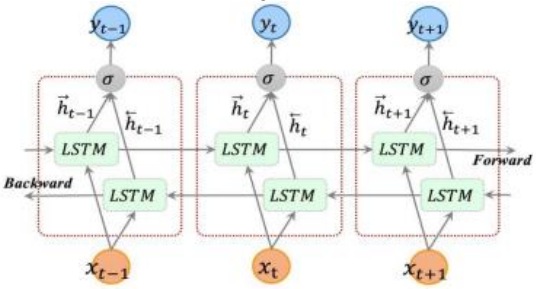

Fig. 3. Architecture BiLSTM

Figure 3 illustrates the architecture of Bidirectional Long Short-Term Memory (BiLSTM), which is a variant of LSTM that processes sequential data, such as text, in two directions: forward and backward. In this architecture, each time step t receives information from both the preceding sequence and the upcoming sequence.

In the forward flow, data is processed from the previous time x_{t-1} to the current time x_t to the future time x_{t+1}. This allows the model to capture context from the incoming data sequence progressively. On the other hand, the backward flow processes data from the future to the past, that is, from x_{t+1} to x_t to x_{t-1}. By processing data in two directions, BiLSTM can

consider information from both sides, thus providing a more complete understanding of the context in text or other sequential data.

Each LSTM block in the figure shows that at each time point *t*, the incoming information is processed through two LSTMs in parallel, one for the forward flow and one for the backward flow. The outputs of both LSTMs are then combined to produce a better final result, which is used to make predictions or classifications based on sequential data (Kowsher et al., 2021).

F. Model Evaluation

The model evaluation uses accuracy, precision, recall, and F1 score metrics to measure classification performance. To ensure that the evaluation results are significant, we performed a comparative analysis between the CNN and BiLSTM model results after applying SMOTE and early stopping. This was done to determine whether the performance improvements obtained from applying these techniques consistently showed better results

4. Results and Discussions

In this study, we collected 10,001 data points from Twitter using web scraping techniques with the keywords "2024 presidential election" and "2024 presidential candidates." From this data, we applied initial filtering to remove irrelevant elements such as retweets, mentions, links, and other symbols that have no direct relevance to the text content. After going through the filtering process, 4810 unique data were successfully obtained. This unique data was determined through duplication elimination, where each tweet was checked to ensure no repetition or excessive similarity between the data.

The criteria used in this filtering included excluding retweets and tweets that only contained links or symbols without meaningful text. This was done to ensure that only relevant text data was included in further analysis. This process helped clean the dataset from redundant and irrelevant data that could affect the quality of model training.

The research team manually labeled data using strict criteria to distinguish SARA content from non-SARA content. We used guidelines applied to local laws on hate speech and guidelines from the international community on identity-based discrimination. Tweets are categorized as "SARA" if they contain derogatory, insulting, or discriminatory comments against ethnicity, religion, race, and inter-group. Conversely, if a tweet does not contain these elements, it is categorized as "Non-SARA."

We implement a dual annotation process to improve labeling accuracy. Two independent annotators review and label the same tweet. If there is a difference in labeling between the two annotators, the final decision is made through joint discussion or involving a third annotator as a mediator. This process ensures that data labeling is carried out consistently and objectively. After the labeling process is carried out, the preprocessing process is carried out. The cleaned data is then processed further, namely the word weighting process using word2vec.

We use Word2Vec to convert text into a vector representation. The parameters used in Word2Vec training include EMBEDDING_SIZE of 200, WINDOW_SIZE of 5, MIN_WORD of 2, EPOCH of 30, SG (Skip-gram) set to 1, and NEGATIVE SAMPLING set to 10.

- EMBEDDING_SIZE: The vector dimensionality is set to 200 to capture a fairly rich semantic representation of the words in the text.
- WINDOW_SIZE: A window size of 5 is chosen to consider the nearest neighboring words around the target word. This helps in understanding the local context.
- MIN_WORD: Words that occur less than twice are ignored to reduce noise from infrequent words.
- SG: The Skip-gram model is chosen because it is better at predicting the context words of a target word, which is very important in Twitter texts that are often short.
- NEGATIVE SAMPLING: This parameter is set to 10 to speed up Word2Vec training and better handle negative samples.

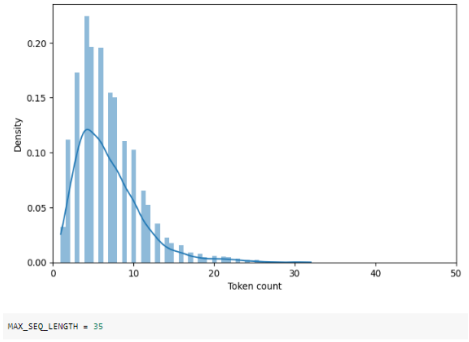

Fig. 4. Determine the maximum length of a sentence.

Figure 4 shows the maximum sentence length, defined as MAX_SEQ_LENGTH, at 35. Furthermore, data preparation was carried out for embedding layers and unique words or tokens were obtained in the Word2Vec dictionary that had been created as many as 4337. After that, the tokenized token is converted into a sequence of integer numbers based on the index dictionary. After being changed, padding and truncation are then carried out. In this process, padding is given to each sentence, and truncation removes tokens from sentences longer than max length.

The next process is to define X and y variables in the data. At this stage, the y variable is converted into a categorical with the help of a hardware library. The variable y, previously 1 dimension, is converted into 2 dimensions. Furthermore, because the data used is unbalanced, oversampling techniques are carried out using the SMOTE method to balance the labels on the data. Data that initially amounted to 4810 became as many as 9486 data. So, for the next process, the data used is as much as 9486.

A. Model Evaluation

Model training was performed using CNN and BiLSTM. We used Adam optimization for model training with a learning rate of 0.001 and a batch size of 32. The model was trained for a maximum of 20 epochs with early stopping applied. Early stopping stops training if there is no increase in accuracy on the validation data for five consecutive epochs. Early stopping helps prevent overfitting, which will be discussed further in the next section, namely, model evaluation.

Model evaluation is the final stage that aims to measure the performance of the model that has been created and determine whether the model has met the needs and objectives of this study. This study used SMOTE to overcome unbalanced data. The training was conducted for Bidirectional Long Short Term Memory (BiLSTM) and CNN algorithms, with the maximum epoch limit used being 20 epochs with a batch size of 35. A comparison of the results of accuracy, precision, recall and f1-score data that have applied SMOTE and have not applied SMOTE to each algorithm is presented in Table 2.

Table 2 – Comparison of Results Using SMOTE

	Without SMOTE		With SMOTE	
	Bi-LSTM	CNN	Bi-LSTM	CNN
Accuracy	99%	99%	91%	92%
Precision	49%	49%	91%	92%
Recall	50%	50%	91%	92%
F1-score	50%	50%	91%	92%

Based on Table 2, the accuracy of the algorithm without SMOTE is better compared to after applying SMOTE. However, this accuracy cannot be used as a reference to measure the model's performance due to data imbalance. Model bias occurs because the model tends to predict the majority label, which is NO SARA, with high accuracy, while ignoring the minority data labeled SARA. This is evident from the imbalance between accuracy and the resulting precision, recall, and F1-score.

The testing results with the confusion matrix for the BiLSTM algorithm show that no test data labeled SARA were predicted as SARA. All test data labeled SARA were predicted as NO SARA due to insufficient training of the model on SARA-labeled data. For the CNN algorithm,

it was somewhat effective in predicting SARA-labeled data as SARA. However, due to extensive training on NO SARA-labeled data, most of the SARA-labeled data were predicted as NO SARA. This is caused by the imbalance between SARA and NO SARA-labeled data, making it necessary to balance the test data using SMOTE. Therefore, it is proven that data balancing with SMOTE can affect the algorithm's performance.

After data balancing, training on the built model applies early stopping to stop running training to prevent the model from overfitting. The maximum epoch limit used is 20, with a batch size of 35. Using Early Stopping, BiLSTM model training stopped at the 4th epoch with a loss of 0.2773, val_loss of 0.2584, accuracy of 88.87%, and val_accuracy of 87.35%. The time used for training the BiLSTM model is 4 minutes and 9 seconds. Next, plotting on loss and accuracy is done to see the training results.

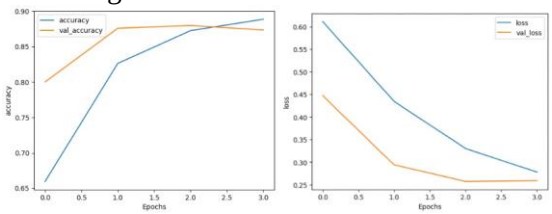

Fig. 5. Plotting Result Model BiLSTM

Figure 5 shows that the val_loss value is not higher than the loss value, so it can be concluded that Early Stopping can prevent overfitting. Furthermore, model performance is measured using a confusion matrix. The confusion matrix calculates the value of accuracy, precision, recall and f1-score on the model created. In addition to Bidirectional Long Short-Term Memory (BiLSTM), training was conducted using a Convolutional Neural Network (CNN). Convolutional Neural Network (CNN) model training stopped at the 11th epoch with a loss of 0.2008, val_loss of 0.2100, accuracy of 91.34%, and val_accuracy of 91.74%. The time used for training the Convolutional Neural Network (CNN) model was 2 minutes 34 seconds. Next, plotting is carried out on loss and accuracy to see the training results.

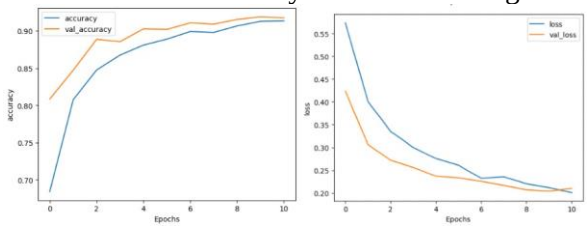

Fig. 6. Plotting Result Model CNN

Figure 6 shows that the val_loss value is not much greater than the loss value, so it can be concluded that early stopping can also prevent overfitting for the CNN algorithm. Furthermore, model performance is measured using a confusion matrix. Testing using balanced data and adding Early Stopping to the algorithm training showed that the CNN algorithm produced higher accuracy than the BiLSTM model. These results were obtained with different epochs. Then, the performance will be seen from the many epochs used. In this case, 3 epoch experiments were carried out, namely with 4 epochs, 11 epochs and 20 epochs on each algorithm. The determination of many epochs was based on the training results with the addition of Early Stopping to both the algorithm and the epoch limits specified in the training model in this study.

Table 3 – Comparison of Accuracy Results Based on Epoch

Algorithm	Accuracy		
	Epoch = 4	Epoch = 11	Epoch = 20
BiLSTM	89%	92%	91%
CNN	86%	93%	92%

Based on Table 3, it can be concluded that the epoch affects the accuracy produced. The algorithm that produces the highest. Accuracy is the CNN algorithm, which has an epoch of 20 and achieves an accuracy of 92%. As more epochs are used, the model will continue to practice, increasing accuracy. However, overfitting of the built model can occur in the absence of early stopping.

The CNN algorithm produces good performance and is balanced with 11 epochs where the val_loss value is not much greater than the loss value. This indicates that there is no overfitting of the model training. In addition, the BiLSTM algorithm produces good performance and is balanced with 4 epochs. As for training with 20 epochs, the val_loss value is much higher than the loss value, which means that the model training for both algorithms is overfitted. In addition to affecting accuracy, epochs also affect model training time.

Table 4 – Model Training Time Comparison Results

Algorithm	Training Time		
	Epoch = 4	Epoch = 11	Epoch = 20
BiLSTM	3 Minute 53 Second	11 Minute 45 Second	17 Minute 18 Second
CNN	56 Second	2 Minute 55 Second	3 Minute 57 Second

Table 4 shows that the longer the model's training time, the more epochs are used. The CNN algorithm is much faster at model training than the BiLSTM algorithm. After several experiments, 2 factors affect the model's performance.

1. Data balance greatly affects model performance. Models can get high accuracy with unbalanced data, but this is because algorithms focus more on majority data and ignore minority data. As a result, the precision, *recall* and *f1-score* values in model evaluation are less satisfactory.
2. The number of *epochs* also affects the model's performance. More epochs do result in higher accuracy. However, when viewed from model training plotting, high accuracy tends to cause *overfitting*.

B. Discussion

The results of this study show that *BiLSTM* and *CNN* architectures can be used to classify comments containing SARA and not SARA on the topic of 2024 presidential candidates on Twitter social media. The model built can classify SARA comments with fairly good accuracy. There were 5 training experiments with the algorithm used. The first experiment is to use data that has not been modified with SMOTE; the second experiment is to use data that has been modified with SMOTE; the third experiment is to use data that has been modified with SMOTE and also with the addition of *Early Stoping*, the fourth experiment is to use data that has been modified with SMOTE and epoch as many as 11. The fifth experiment uses data modified with SMOTE and epoch as many as 4 on model training. A summary of the model evaluation results is presented in Table 5 to facilitate the evaluation of the results obtained.

Table 5 – Summary of Model Evaluation Results

Performance	Without SMOTE Epoch = 20		With SMOTE Epoch = 20		With SMOTE and Early Stopping		With SMOTE Epoch = 4		With SMOTE Epoch = 11	
	CNN	BL	CNN	BL	CNN	BL	CNN	BL	CNN	BL
Accuracy	99%	99%	92%	91%	92%	87%	86%	89%	93%	92%
Precision	73%	49%	92%	91%	92%	88%	87%	89%	93%	92%
Recall	66%	50%	92%	91%	92%	87%	89%	89%	92%	92%
F1-Score	69%	50%	92%	91%	92%	87%	86%	89%	92%	92%
Time Training	2 M 20 S	9 M 18 S	3 M 57 S	17 M 18 S	1 M 2 S	1 M 2 S	1 M	4 M 20 S	2 M 34 S	11 M
Loss	0.0291	0.0387	0.1572	0.2824	0.2008	0.2773	0.3083	0.2548	0.1895	0.1307
Val_loss	0.0663	0.0491	0.2058	0.2408	0.2100	0.2589	0.2839	0.2481	0.2067	0.2263

Note: BL=BiLSTM

From Table 5, it is evident that there are differences in the performance of CNN and BiLSTM models under various conditions for detecting offensive content. These conditions include training with and without SMOTE, the use of early stopping, and different epochs, specifically 4, 11, and 20 epochs. The impact of these conditions is evaluated through metrics such as accuracy, precision, recall, F1-score, training time, loss, and validation loss (Val_loss).

Overall, Table 5 highlights that CNN models generally maintain high accuracy and precision across various conditions, whereas BiLSTM models require more training time and

exhibit greater variability in precision and recall depending on the training conditions. The use of SMOTE and early stopping proves beneficial as they help improve model performance by addressing class imbalance and preventing overfitting. The comprehensive performance metrics provide clear indications of how different training strategies affect the effectiveness of CNN and BiLSTM models in detecting offensive content.

The highest accuracy achieved by both algorithms is 99% with 20 epochs. However, when looking at the loss and Val_loss values, it is evident that the Val_loss is higher than the training loss, indicating that overfitting is occurring under these conditions.

The best CNN model is identified as the one trained with SMOTE and 11 epochs. This model has slightly lower accuracy compared to the one trained with 20 epochs but exhibits a smaller difference between loss and Val_loss, indicating it is not overfitting. Additionally, it has a faster training time of 2 minutes and 34 seconds, making it more efficient and practical compared to other models.

For BiLSTM, the best model is obtained with SMOTE and 11 epochs. This model achieves an accuracy of 92% with a small difference between loss and Val_loss, indicating it does not overfit. Despite having a longer training time of 11 minutes, this model provides the best balance between performance and stability.

The classification results with *the CNN* algorithm show that SARA data is classified as many as 1353 and NO SARA as many as 1259. Meanwhile, with the BiLSTM algorithm, data labeled SARA is classified as much as 1313 and NO SARA as much as 1173. This study aimed to improve model performance by adding several techniques. The comparison results with previous studies are shown in Table 5.

Table 6 – Comparison with Previous Research Results

Model	Accuracy
CNN (Adam & Setiawan, 2023)	91.93%
BiLSTM (Lestari et al., 2024)	87%
BiLSTM (Chihab et al., 2022)	75%
LSTM (Yang, 2023)	89%
ERNIE + BiLSTM (Hsieh & Zeng, 2022)	89%
CNN + SMOTE	93%
BiLSTM + SMOTE	92%

Based on Table 6, the algorithm that produces the highest accuracy is the Convolutional Neural Network (CNN) algorithm with a combination of SMOTE and *Early Stopping.*

C. Research Limitations

The main limitation of this study is that it is limited to data taken from Twitter alone, so it may only represent some forms of hate speech on other platforms. In addition, manual labeling may cause annotator bias, although mitigation efforts were made through the double annotation process and discussion. Another challenge is that while SMOTE helps balance the data, the synthetic results generated may only sometimes represent the natural variation of SARA content, affecting the model's ability to detect more ambiguous comments.

5. Conclusion

This study aims to develop a classification model that can detect SARA-related comments on Twitter using the Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) approaches, as well as to overcome challenges to data measurement by implementing SMOTE (Synthetic Minority Over-sampling Technique) and early stop to prevent overfitting. The study results show that applying SMOTE successfully improves the model's ability to handle imbalanced data, especially in detecting SARA comments, which are often in the minority. Using CNN+SMOTE, the model achieved a precision of 93%, while BiLSTM+SMOTE provided an accuracy result of 92%, indicating that both models effectively capture hate-related text patterns.

From the research data obtained, using SMOTE and early stop is important in improving model performance, especially in the context of significant class relatedness between SARA and non-SARA data. CNN excels in terms of precision, which means this model is better at reducing false positives, while BiLSTM provides better recall results, meaning less SARA data is missed.

This study implies that the combined approach of deep learning models and data balancing techniques can be applied to detect hate speech more accurately on social media platforms. In Indonesia, where SARA issues often trigger conflicts, this model can be the basis for developing a better automated monitoring system. Recommendations for further research are to apply this model to other social media platforms and use a more diverse dataset so that the generalizability of the results can be improved and the model can handle language variations or other contexts that may arise outside of Twitter.

References

Abidin, Z., Junaidi, A. & Wamiliana. (2024). Text Stemming and Lemmatization of Regional Languages in Indonesia: A Systematic Literature Review. *Journal of Information Systems Engineering and Business Intelligence*, *10*(2), 217–231. https://doi.org/10.20473/jisebi.10.2.217-231

Adam, A. Z. R. & Setiawan, E. B. (2023). Social Media Sentiment Analysis using Convolutional Neural Network (CNN) dan Gated Recurrent Unit (GRU). *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika (JITEKI)*, *9*(1), 119–131. https://doi.org/10.26555/jiteki.v9i1.25813

Aji, N. B., Kurnianingsih, Masuyama, N. & Nojima, Y. (2024). CNN-LSTM for Heartbeat Sound Classification. *International Journal on Informatics Visualization*, *8*(2), 735–741. https://doi.org/10.62527/joiv.8.2.2115

Anam, M. K., Defit, S., Haviluddin, Efrizoni, L. & Firdaus, M. B. (2024). Early Stopping on CNN-LSTM Development to Improve Classification Performance. *Journal of Applied Data Sciences*, *5*(3), 1175–1188. https://doi.org/10.47738/jads.v5i3.312

Aurora, E., Zahra, A., Sibaroni, Y., Sri, & & Prasetyowati, S. (2023). Classification of Multi-Label of Hate Speech on Twitter Indonesia using LSTM and BiLSTM Method. *JINAV: Journal of Information and Visualization*, *4*(2), 2746–1440. https://doi.org/10.35877/454RI.jinav1864

Bailey, E. R., Matz, S. C., Youyou, W. & Iyengar, S. S. (2020). Authentic self-expression on social media is associated with greater subjective well-being. *Nature Communications*, *11*(1). https://doi.org/10.1038/s41467-020-18539-w

Begum, S. G. & Sree, P. K. (2023). Drug Recommendation Using a "Reviews and Sentiment Analysis" By a Recurrent Neural Network. *Indonesian Journal of Multidisciplinary Science*, *2*(9), 3085–3094. https://doi.org/10.55324/ijoms.v2i9.530

Casero-Ripollés, A. (2021). Influencers in the political conversation on twitter: Identifying digital authority with big data. *Sustainability (Switzerland)*, *13*(5), 1–14. https://doi.org/10.3390/su13052851

Chihab, M., Chiny, M., Boussatta, N. M. H., Chihab, Y. & Youssef Hadi, M. (2022). BiLSTM and Multiple Linear Regression based Sentiment Analysis Model using Polarity and Subjectivity of a Text. *IJACSA) International Journal of Advanced Computer Science and Applications*, *13*(10), 436–442. https://doi.org/10.14569/IJACSA.2022.0131052

Dharani, R., Revathy, S. & Danesh, K. (2023). Fuzzy Genetic Particle Swarm Optimization Convolution Neural Network Based on Oral Cancer Identification System. *Journal of Applied Engineering and Technological Science*, *5*(1), 150–169. https://doi.org/10.37385/jaets.v5i1.2874

Fauzy, A. R. I. & Setiawan, E. B. (2023). Detecting Fake News on Social Media Combined with the CNN Methods. *JURNAL RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *7*(2), 271–277. https://doi.org/10.29207/resti.v7i1.4889

Herianto, Kurniawan, B., Hartomi, Z. H., Irawan, Y. & Anam, M. K. (2024). Machine Learning Algorithm Optimization using Stacking Technique for Graduation Prediction. *Journal of Applied Data Sciences*, *5*(3), 1272–1285. https://doi.org/10.47738/jads.v5i3.316

Hsieh, Y. H. & Zeng, X. P. (2022). Sentiment Analysis: An ERNIE-BiLSTM Approach to Bullet Screen Comments. *Sensors*, *22*(14), 1–15. https://doi.org/10.3390/s22145223

Konovalova, E., Le Mens, G. & Schöll, N. (2023). Social media feedback and extreme opinion expression. *PLoS ONE*, *18*(11), 1–20. https://doi.org/10.1371/journal.pone.0293805

Kowsher, M., Tahabilder, A., Sanjid, M. Z. I., Prottasha, N. J., Uddin, M. S., Hossain, M. A. & Jilani, M. A. K. (2021). LSTM-ANN & BiLSTM-ANN: Hybrid deep learning models for enhanced classification accuracy. *Procedia Computer Science*, *193*, 131–140. https://doi.org/10.1016/j.procs.2021.10.013

Lestari, V. B., Utami, E. & Hanafi. (2024). Combining Bi-LSTM And Word2vec Embedding For Sentiment Analysis Models of Application User Reviews. *Indonesian Journal of Computer Science*, *13*(1), 312–326. https://doi.org/10.33022/ijcs.v13i1.3647

Li, X., Lei, Y. & Ji, S. (2022). BERT- and BiLSTM-Based Sentiment Analysis of Online Chinese Buzzwords. *Future Internet*, *14*(11), 1–15. https://doi.org/10.3390/fi14110332

Lyrawati, D. P. N. (2022). Hate Speech Detection on Twitter Approaching The Indonesian Election Using Machine Learning. *The Journal on Machine Learning and Computational Intelligence*, *2*(1), 26–31. https://doi.org/10.26740/vol2iss1y2022id20

Malik, P., Aggrawal, A. & Vishwakarma, D. K. (2021). Toxic Speech Detection using Traditional Machine Learning Models and BERT and fastText Embedding with Deep Neural Networks. *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, 1254–1259. https://doi.org/10.1109/ICCMC51019.2021.9418395

Muis, A., Yudhana, A. & Dahlan, A. (2023). Comparison Analysis of Brain Image Classification Based on Thresholding Segmentation With Convolutional Neural Network. *Journal of Applied Engineering and Technological Science*, *4*(2), 664–673. https://doi.org/10.37385/jaets.v4i2.1583

Omran, E., Al Tararwah, E. & Al Qundus, J. (2023). A comparative analysis of machine learning algorithms for hate speech detection in social media. *Online Journal of Communication and Media Technologies*, *13*(4), 1–11. https://doi.org/10.30935/ojcmt/13603

Prathama, N. A., Hasani, M. R. & Akbar, M. I. (2022). SARA Hoax: Phenomena, Meaning, and Conflict Management. *Jurnal ASPIKOM*, *7*(2), 129. https://doi.org/10.24329/aspikom.v7i2.1117

Putra, P. P., Anam, M. K., Defit, S. & Yunianta, A. (2024). Enhancing the Decision Tree Algorithm to Improve Performance Across Various Datasets. *INTENSIF: Jurnal Ilmiah Penelitian Dan Penerapan Teknologi Sistem Informasi*, *8*(2), 200–212. https://doi.org/10.29407/intensif.v8i2.22280

Putra, R. S., Agustin, W., Anam, M. K., Lusiana, L. & Yaakub, S. (2022). The Application of Naïve Bayes Classifier Based Feature Selection on Analysis of Online Learning Sentiment in Online Media. *Jurnal Transformatika*, *20*(1), 44–56. https://doi.org/10.26623/transformatika.v20i1.5144

Rianto, Mutiara, A. B., Wibowo, E. P. & Santosa, P. I. (2021). Improving the accuracy of text classification using stemming method, a case of non-formal Indonesian conversation. *Journal of Big Data*, *8*(1). https://doi.org/10.1186/s40537-021-00413-1

Rudiyanto, R. A. & Setiawan, E. B. (2024). Sentiment Analysis Using Convolutional Neural Network (CNN) and Particle Swarm Optimization on Twitter. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, *9*(2), 188–195. https://doi.org/10.33480/jitk.v9i2.5201

Sharmin, T., Di Troia, F., Potika, K. & Stamp, M. (2020). Convolutional neural networks for image spam detection. *Information Security Journal*, *29*(3), 103–117. https://doi.org/10.1080/19393555.2020.1722867

Siddiqui, J. A., Yuhaniz, S. S., Memon, Z. A. & Amin, Y. (2021). Improving Hate Speech Detection Using Machine and Deep Learning Techniques: A Preliminary Study. *Open International Journal of Informatics (OIJI)*, *9*(2), 21–34. https://doi.org/10.11113/oiji2021.9nSpecial Issue 2.143

Taradhita, D. A. N. & Putra, I. K. G. D. (2021). Hate speech classification in Indonesian language tweets by using convolutional neural network. *Journal of ICT Research and Applications*, *14*(3), 225–239. https://doi.org/10.5614/itbj.ict.res.appl.2021.14.3.2

Yang, Y. (2023). Application of LSTM Neural Network Technology Embedded in English Intelligent Translation. *Computational Intelligence and Neuroscience*, *2023*, 1–1. https://doi.org/10.1155/2023/9764613