Detection of hate-speech Text on Indonesian Twitter Social Media Using IndoBERTweet-BiLSTM-CNN

*No author information during review

Abstract—**Social media Twitter has become the second place in people's lives to express themselves. Social media users can comment on whatever they want, and it is not uncommon to find comments that contain hate-speech. If it is not stopped, hate-speech can spread quickly, therefore it is necessary to detect hate-speech. In this research, the detection of hate-speech was carried out using IndoBERTweet, which is a development of the BERT model that has been previously trained using data from Indonesian language Twitter, so it is suitable for classifying Indonesian language texts. BiLSTM and CNN are deep-learning methods that can be used for text classification. This research aims to detect hate-speech texts using these three methods and then combining them. To carry out optimization, experiments were carried out on batch size and learning rate values. With a batch size of 8 and a learning rate of 0.001, the best accuracy is 85.45%, and the F1-Score is 85.06%.**

Index Terms—**hate-speech, Text Classification, IndoBERTweet, BiLSTM, CNN.**

I. Introduction

Social media serves as a platform or online media that functions as a forum for users to participate, share, create, and exchange information through various forums and social networks [1]. One of the popular social media platforms in Indonesia is Twitter. By using Twitter, everyone is free to express and comment without clear restrictions. This lack of restrictions often leads to some individuals engaging in hate speech on Twitter, using the platform to demean or harm others. The issue of hate speech among users on the Twitter platform is a significant concern. Despite the freedom of expression it offers, Twitter also facilitates an environment where hate speech can thrive, impacting individuals and groups adversely. This unmoderated freedom often leads to hostile exchanges that can escalate into larger societal issues, necessitating a deeper understanding and management of such behaviors on social platforms.

Given the increasing and uncontrolled problem of hate speech, there is an urgent need for effective detection systems. These systems are essential not only for identifying and mitigating the spread of hate speech but also for maintaining the integrity and inclusiveness of online communication spaces. Developing robust detection mechanisms can help in preemptively identifying potential hate speech, thereby preventing it from causing harm or spreading further.

Detection of hate-speech on Indonesian Twitter has been done before with various method. Several studies use machine learning methods such as RFDT, SVM, Naive Bayes [2], [3], [4]. Then some use neural network methods such as BiLSTM, CNN, RNN, GRU [5], [6], [7] and produces quite good results compared to previous machine learning methods. Hate-speech detection is also carried out using BERT-based methods, specifically IndoBERT and IndoBERTweet for Indonesian language-based datasets and Indonesian Twitter [8] [9] [10] [11] . In this research, the performance of the IndoBERTweet method has been compared with methods based on neural networks and machine learning, and the results show that IndoBERTweet produces the best performance. Another research using the method of combining neural network layers was also carried out in the [12]. this research used the BiLSTM-CNN method to carry out text classification and produced an accuracy of 94% which is an outstanding result. However, this research was carried out using English-based dataset.

Previous research shows that there are many variations of methods for detecting hate speech using text classification. However, the most prominent one for classifying Indonesian-based hate-speech texts is IndoBERTweet, and combined methods such as BiLSTM-CNN are also very good at text classification. Therefore, this research proposed to combine the IndoBERTweet and BiLSTM-CNN methods with the hope of combining the advantages of IndoBERTweet in understanding the context in Indonesian very well and the ability of BiLSTM-CNN to carry out text classification.

II. Literatur Review

A. IndoBERT

In research [13] it is explained that IndoBERT is a transformer-based model in the BERT style, developed specifically for the Indonesian language. This model was trained as a masked language model using the Huggingface framework, with the default configuration for BERT-Base (uncased). IndoBERT was trained with a total of more than 220 million words from various sources, including Indonesian Wikipedia, news articles, and the Indonesian web corpus.

This model has 12 hidden layers, each with 768 dimensions, 12 attention heads, and a feed-forward with 3,072 dimensions. INDOBERT is trained using a batch size of 512 tokens per batch and utilizes the 31,923-token Indonesian WordPiece vocabulary [13]. The use of IndoBERT itself has been proven in research [5] that IndoBERT can perform Indonesian language sentiment analysis much better than the BERT model.

B. IndoBERTweet

IndoBERTweet is a model trained based on masked language model (MLM) tasks with the same configuration as the "indoor-base-uncased" model. IndoBERTweet is a transformer encoder with 12 hidden layers (dimensional=768), 12 attention heads, and 3 hidden feed-forward layers (dimensional=3,072), and uses 26 million Twitter data, with 409

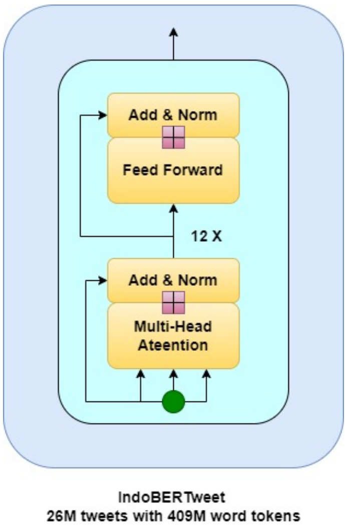

Fig. 1. IndoBERTweet Architecture

million tokens to pre-train the model [14]. Experiments [11] involved the use of the IndoBERTweet model for language tasks, especially in the Twitter environment, with variations in pretraining and vocabulary adaptation strategies in various domains, and the results showed better performance than the IndoBERT model. The architecture of IndoBerTweet can be seen in Figure 1.

C. LSTM

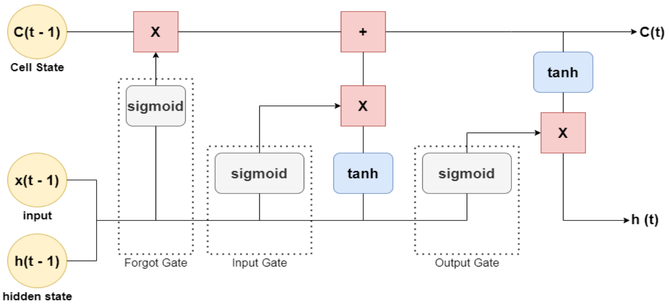

Fig. 2. LSTM Architecture

Long-short memory (LSTM) is a type of Recurrent Neural Network (RNN) with additional features (mathematical functions and special cell states) designed to be able to remember and store longer [15]. LSTM can ”remember” and ”use” information from previous input, even if the input consists of a long sequence or has a large distance between one element and the next. LSTM allows the model to understand patterns and relationships involving words or phrases that may be scattered throughout tweets, thereby improving the ability to detect hate-speech that may be contained in that content [16]. Mathematically, it can be seen in equations 1, 2, 3, 4, 5, and 6.

$$InputGate(i_t) = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) \tag{1}$$

$$ForgotGate(f_t) = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f) \tag{2}$$

$$OutputGate(O_t) = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o) \tag{3}$$

$$C't = \tanh(W_c \cdot [h_{t-1}, X_t] + b_c) \tag{4}$$

$$C_t = X_t * C_{t-1} + i_t * C' \tag{5}$$

$$h_t = Ot * \tanh(C_t) \tag{6}$$

σ is a sigmoid function that converts values to a range between 0 and 1. W_i, W_f, W_o and W_c are the weights applied to the combined vector of the output of the previous cell (hidden state) and the current input $([h_{t-1}, x_t])$. b_i, b_f, b_o, b_c are the biases added after multiplying the weights. The hyperbolic tangent function $(\tanh)$ is applied to the results of the combined weight and vector calculations to produce candidate values from the new memory. In the context of hate-speech on Twitter, the input gate will control how much information from the words or phrases in the tweet will be entered into the memory cells. This will help the model in selecting important and relevant words or content to understand the context of the tweet regarding hate-speech. The forget gate will help the model control how much information from previous tweets related to hate-speech needs to be forgotten. This could mean reducing the influence of words or content that are irrelevant or do not fit the characteristics of hate-speech. Meanwhile, the output gate will determine how much information from the memory cell will be used as output. This will allow the model to select relevant and important information from tweets related to hate-speech. $C't$ is Candidate Memory which will help the model in recognizing words or phrases in tweets that have certain characteristics or characteristics that indicate a tendency towards hate-speech. C_t is Cell State, namely the section that stores long-term information containing information related to words or phrases which cumulatively gives an idea of whether the tweet contains hate-speech or not. h_t is the output of the LSTM unit which is the result of a combination of information stored in the cell state that is relevant to the characteristics of hate-speech on Twitter. This could be the result of an analysis that shows the level of tendency for hate-speech in the tweet.

D. BiLSTM

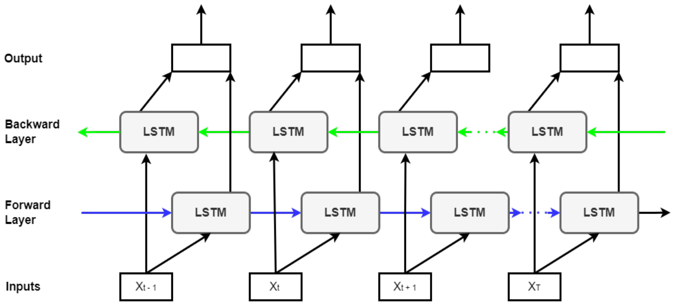

Fig. 3. BiLSTM Architecture

Bidirectional Long Short-Term Memory (BiLSTM) is an extension of LSTM which can process and model data sequences better because it can understand text context in both directions. In BiLSTM, two parts of LSTM work simultaneously, namely observing words from beginning to end to capture future context, while the other observes the sequence from end to beginning to understand the historical context.

$$h_i Forward = LSTMForward(X_{t-1}) \tag{7}$$

$$h_i Backward = LSTMBackward(X_{t+1}) \tag{8}$$

$$h_n = [h_i Backward, h_i Forward] \tag{9}$$

This process produces two hidden representations $h_i Forward$ and $h_i Backward$ as shown in equations (7) and (8) respectively. Next, BiLSTM combines the information from both LSTM networks to calculate the final representation as shown in equation (9). The data contained in BiLSTM includes context related to hate from both directions. The information generated by BiLSTM is then channeled to the attention layer to assign different weights to this hidden information. [17]

E. CNN

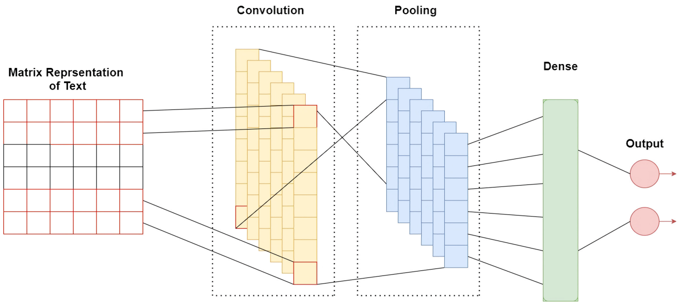

Fig. 4. CNN Architecture

Convolutional Neural Network (CNN) is a type of artificial neural network architecture that has become one of the most popular choices in the field of deep learning. CNNs were originally developed to handle computer vision tasks and have enabled achievements that were considered impossible in recent decades. Initially used for facial recognition, autonomous vehicles, and smart medical treatment, CNNs were later applied to various sequence modeling problems, such as text classification, sentiment analysis, and other prediction tasks [18]. The CNN architecture starts by representing the input sentences as a matrix in the “Sentence Representation” step. This matrix becomes the input for the convolution layer, where filters are used to generate a feature map taking into account important parameters such as the number and size of kernels. After convolution, a “Max Pooling” step is performed to reduce the data dimensionality and highlight significant features. The results are then connected to the fully connected layer to combine features. The dropout technique is used to overcome overfitting by randomly deactivating some neurons during training. The final process involves a softmax layer to produce output based on the highest probability of a particular class. Optimization is carried out to reduce losses during training, with optimization options such as Adam, Nadam, or SGD. Overall, this architecture forms a series of integral steps for feature extraction and classification of text data to produce optimal class predictions [19]. The architecture of CNN can be seen in Figure 4.

III. Research Methodology

A. Design system

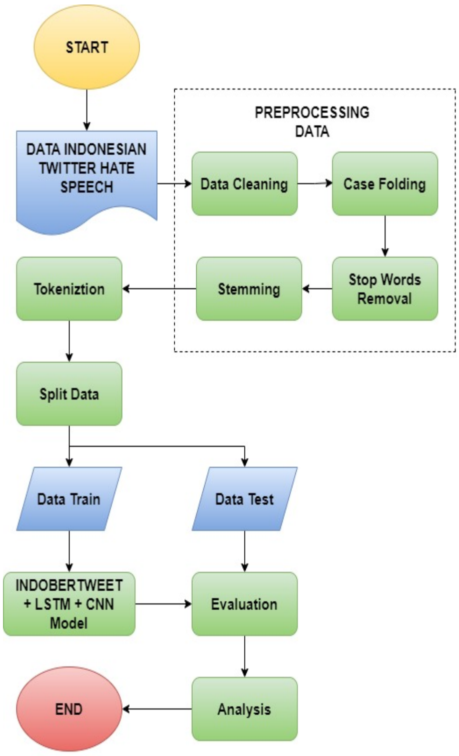

Fig. 5. System Flowchart

This research stage began by taking hate-speech data sets from Indonesian Twitter. Next, the preprocessing stage is carried out, namely, the stage to improve data quality. First, data cleaning is carried out by making text data easier to analyze by machines. After that, stemming and tokenization are carried out so that text analysis and NLP enable better processing and understanding of text data. Next, the data is split into train and test, where the train data will later be entered into modeling, while the test data will be used for evaluation. Then we enter the modeling stage where the IndoBERTweet, BiLSTM, and CNN models are combined. Finally, an evaluation is carried out to determine the performance of the model that has been created. The stages of more detailed research can be seen in Fig. 5.

B. Data set

These data come from previous research [20]. This includes Indonesian language Twitter comments labeled hate-speech and non hate-speech, consisting of 13169 tweets, 7608 labeled non hate-speech, and 5561 labeled hate-speech. The data set itself is not ready, so it cannot be used for modeling. For better data quality and ready to be included in the modeling, it is necessary to preprocess the data because there are still many words referring to USER and URL, as well as ambiguous words such as “/n6, 00/n, 44.”

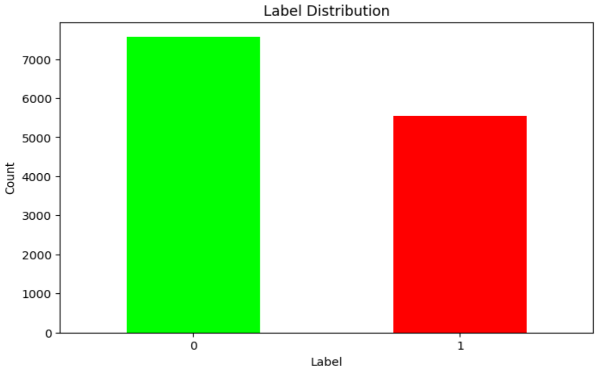

Fig. 6. Label Distribution

C. Preprocessing Data

Several preprocessing methods are used at this point to enhance the quality of the data. Tokenization, stopwords, data cleansing, and case folding are the methods that will be used [21].

a. Case folding : Case folding is a technique for doing lowercase, namely changing all uppercase letters to lowercase, this is done to make it easier for machines to analyze text because the difference in uppercase and lowercase letters can make machines read them with different meanings [22].

b. **Stopwords** : Next, stop words were eliminated since they frequently have little bearing on the evaluation of sentiment, particularly when combined with conjunctions or personal pronouns. Stopwords are removed so that search engines can concentrate more on sentimentally reflective terms.

c. Data Cleaning : There are many words that are not useful in sentiment analysis in this data set, such as "URL", "USER", `\xf0\x9f\x98\x8f`Therefore, these words will be removed to make it easier for machines to analyze sentiment, which can be seen in Table I.

TABLE I
DATA CLEANING

Before Cleaned	**After Cleaned**
Setidaknya gw punya jari tengah buat lu, sebelum gw ukur nyali sama bacot lu `\xf0\x9f\x98\x8f`	Setidaknya gw punya jari tengah buat lu, sebelum gw ukur nyali sama bacot lu
`\xe2\x80\x9c` Tidak!`\xe2\x80\x9d` kata salah seorang Khalifah Rasyidin itu `\xe2\x80\x9c`kami dahulu memerangi orang kafir bersama mereka `\xe2\x80\x9d. \xe2\x80\x9c`Kalau begitu, mereka orang-orang munafik `\xe2\x80\x9d. \xe2\x80\x9c` Tidak juga. Sebab, orang munafik	kata salah seorang Khalifah Rasyidin itu, dahulu memerangi orang kafir bersama mereka. begitu, mereka orang-orang munafik. juga. sebab, orang munafik
USER USER USER USER Gue saranin gk perlu bnyk bacot maling ayat, langsung to the point aja lu jadi orang... Yg tdk sepaham lu teriakin aja kafir sesat, liberal, pemuja	Gue saranin gk perlu bnyk bacot maling ayat, langsung to the point aja lu jadi orang... Yg tdk sepaham lu teriakin aja kafir sesat, liberal, pemuja
USER USER USER `.\nOnta` mati klo kebanyakan Minum	Onta mati klo kebanyakan Minum

d. Tokenization : tokenization is dividing text or documents into smaller units called "tokens." [23]

e. Stemming : In text processing, stemming is the process of turning a written word into a base word. This procedure is very important for reducing data dimensionality and increasing throughput in various language learning applications. Stemming facilitates the analysis and indexing of word variants in Indonesian, which have a rough morphology. Algorithms like Sastrawi. This algorithm also ensures that the text preprocessing procedure is more accurate and efficient. [24]

D. Indobertweet-BiLSTM-CNN

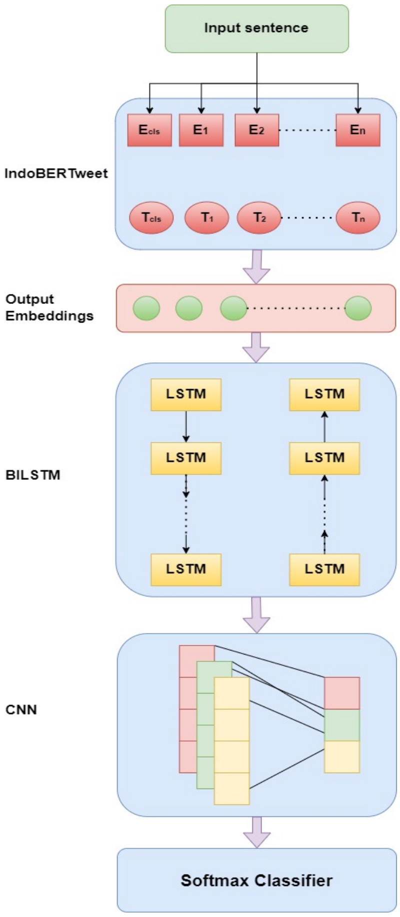

Fig. 7. IndoBERTweet-BiLSTM-CNN Architecture

Utilizing the model of Indobertweet-BiLSTM-CNN for the Indonesian Twitter dataset is a sensible approach since Indobertweet has been trained using Indonesian language data, enabling more accurate understanding of the language and the structure of Indonesian tweets. Combining CNN and BiLSTM allows the model to effectively extract contextual and spatial information from the text, increasing the capacity for sentiment analysis of informal, pendek, and khas text from Indonesian Twitter users as well as the ability for sound generalization of all text on Twitter.

IndoBERTweet is used by the model to improve its ability to learn Indonesian text features from Twitter. It offers vector representations that improve the model's comprehension of numerical connections in text. Using the [CLS] token outputs as inputs into the BiLSTM, which uses a sigmoid activation function for efficient feature extraction from both forward and backward text contexts, the integration of BERT with

BiLSTM begins at the output of the last BERT layer [25]. Additionally, the model combines CNN and BiLSTM to leverage temporal dynamics and local feature dependencies. It starts with a word embedding and uses BiLSTM processing to generate a rich vector representation. A CNN layer using 1-max pooling then refines the representation to extract the most important features for classification [12].

The model will be created by combining ideas from the research of [25] and [12]. The pre-trained transformer model ”indolem/indobertweet-base-uncased” is integrated into the model architecture along with a hybrid deep learning framework for text categorization, which is especially good at handling tweet text in Indonesian. Tokenization and uniform tweet padding are the first steps. Next, an embedding layer employing the transformer's pre-trained weights—which are not trainable—is used to keep the contextual embeddings that have been learned. A set of convolutional and max-pooling layers that improve and compress the feature set are added to a bidirectional 128-unit LSTM layer to capture both forward and backward contextual dependencies. To improve feature extraction, convolutional neural network (CNN) components are added after the bidirectional long short-term memory (LSTM) layer. To extract spatial hierarchies from the sequential data, a Conv1D layer with 128 filters and a kernel size of 5 is specifically used. A MaxPooling1D layer is then added to lower the spatial dimensions, effectively compressing the data and emphasizing the most important aspects. This process is furthered by additional convolutional and pooling layers, which improve the feature set and ready to add a sigmoid output layer for binary classification.

The model highlights the architecture's ability to reliably handle and analyze social media text through sophisticated feature extraction and sequence modeling techniques. It is compiled with the Adam optimizer and binary cross-entropy loss, and it includes an early stopping mechanism to prevent overfitting during training.

E. Evaluation System

Assessments This is done by calculating the F1-Score and Accuracy of a certain model. After the classification procedure, the classification model assigns a label (positive or negative) to each sample, based on the analysis of each individual data point [26]. Finally, each participant might go to one of the following three cases:

True Positives (TP): Actual positives that were correctly predicted.
False Negatives (FN): Actual positives that were incorrectly predicted.
True Negatives (TN): Actual negatives that were correctly predicted.
False Positives (FP): Actual negatives that were predicted incorrectly.

This information can be represented in a Confusion Matrix (M),

$$\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix}$$

Accuracy is the percentage of network activities that are correctly classified. It is measured using the ratio of the correct predictions (both true positives and true negatives) to the total number of predictions made F1 Score is described as a measure that combines precision and recall (sensitivity) into one metric using the harmonic mean of both [27]. Mathematically, accuracy and F1-score can be written in 10 and 11 respectively.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad (10)$$

$$\text{F1-score} = \frac{2 * TP}{2 * TP + FP + FN} \quad (11)$$

IV. Experiments and Results

A. Model Performance

All Models have been trained using the same parameters to ensure fair comparison between models. These parameters include a batch size of 16 and Adam algorithm as an optimization method. Adam is a popular optimizer used in training neural network models because of its efficiency in handling sparse gradients and its good adaptation to various types of data. This uniform use of parameters and optimizers allows an objective evaluation of the performance of each model architecture in processing and predicting data.

TABLE II
Comparison of Model Performance

Model	F1-Score	Accuracy
BiLSTM	0.8157	0.8202
CNN	0.7882	0.7939
BiLSTM-CNN	0.8070	0.8069
IndoBERTweet	0.8080	0.8080
IndoBERTweet-BiLSTM	0.8440	0.8476
IndoBERTweet-CNN	0.8378	0.8385
IndoBERTweet-BiLSTM-CNN	**0.8463**	**0.8495**

From the table given, we can see that the performance of the IndoBERTweet-BiLSTM-CNN model stands out as the best among other models. However, it is important to also compare and understand the performance of other models presented in the table.

The BiLSTM model independently has quite high accuracy, namely 82.02%. This model is effective because of LSTM's ability to remember long-term information, especially useful in sentiment analysis or understanding the context of text data sequences. The CNN model, which has an accuracy of 79.39%, shows superiority in feature extraction through convolutional processing which can identify local patterns in text data, but appears to be less effective compared to models that use sequential or contextual components.

The BiLSTM-CNN combination achieved an accuracy of 80.69%, indicating that the combination of these two methods provides an improvement compared to using CNN alone, but it is not very significant when compared to using BiLSTM alone. This may indicate that while CNN provides good feature extraction, the sequentiality of data managed by BiLSTM makes a greater contribution to the understanding of context in text.

The IndoBERTweet model itself provides better accuracy than CNN but is lower than BiLSTM, with a value of 80.80%. This could be because transformer models like IndoBERTweet rely heavily on the volume and variety of training data to optimize deeper context understanding. The combination of IndoBERTweet with BiLSTM increases accuracy to 84.76%, and IndoBERTweet with CNN reaches 83.85%, both of which provide significant improvements compared to using IndoBERTweet alone. This indicates that the addition of sequential or convolutional structures can help in improving the model's understanding of complex data structures, such as text.

As a comparison, research [20] wichh is the source of the dataset , used Random Forest Decision Tree (RFDT) classifier and achieved the best accuracy of 77.36%. The significant differences in the accuracy of the best models from this study demonstrate advances in natural language processing techniques, particularly through the use of more complex and sophisticated deep learning architectures. This improvement may be due to the ability of deep learning-based models such as IndoBERTweet to understand the context in text more deeply, which cannot be fully achieved by tree-based methods such as RFDT. Additionally, the adaptability of deep learning models in dealing with variability in text data may also contribute to this performance improvement. Another research with same dataset [28] used GloVe-BiLSTM classifier and achieved the best accuracy of 82.31%, and research [29] used IndoBERTweet-BiGRU and achieved the best accuracy of 84.77%. The previous research shows that neural network models can produce better performance than machine learning models for classifying Indonesian-based texts, and also IndoBERTweet can improve the performance of a model, especially for Indonesian text classifiers.

The IndoBERTweet-BiLSTM-CNN achieved a good performance in text classification, particularly demonstrating its efficacy in hate speech detection in Indonesian text. By achieving an accuracy of 84.95% , this model illustrates the benefits of combining diverse neural network architectures to capitalize on their strengths in language understanding and processing. The integration of the IndoBERTweet model with transformer technology allows for an enhanced understanding of contextual nuances critical in text classification. Additionally, the BiLSTM component of the model helps in processing text data sequentially, and The CNN layers further contribute by extracting key features and patterns essential for identifying specific text classes effectively. This model not only surpasses traditional machine learning models such as the RFDT, which showed a maximum accuracy of 77.36%, but also demonstrates marked improvements over more sophisticated neural network setups like the GloVe-BiLSTM and IndoBERTweet-BiGRU. The robustness of the IndoBERTweet-BiLSTM-CNN model in handling the variability inherent in natural language makes it particularly adaptable to diverse linguistic expressions found in extensive and complex datasets.

B. Hyperparameter Test

The hyperparameter itself has a significant impact on determining a neural network model's performance [30]. The hyperparameters that will be examined are learning rate and batch size. Batch size refers to the number of sample data points that are used in a single iteration to update the batch size of the model, while learning rate is a parameter that indicates the number of significant batch sizes in the neural network that are adjusted during the backpropagation process using the loss function's gradient [31].

The batch sizes that were used for this experiment were 10, 16, 32, and 64. The learning rates were 0.1, 0.01, 0.001, and 0.0001. To learn a deep learning model, a combination of various learning rates and batch sizes is applied. The first loop begins with the list of learning rate, and the second loop, which is nested in the first loop, begins with the list of batch size. Models are created, combined, and then trained using data for every combination of final learning objectives and batch size. For a more thorough analysis, the validation accuracy of each combination is displayed in a DataFrame table. This makes it possible for researchers to evaluate parameter combinations that provide the best possible work quality on scientific data. When training a neural network model, early stopping is used to prevent overfitting for a given number of epochs, especially on relatively small or large datasets. This also helps to anticipate scenarios when learning models need a lot of information or noise from training data in order to properly generalize to new data.

TABLE III
PARAMETER EXPERIMENT ACCURACY

Batch Size	Learning Rate			
	0.01	0.001	0.0001	0.00001
8	0.564787	0.8545	0.815927	0.595771
16	0.564787	0.8495	0.810594	0.592988
32	0.564787	0.8430	0.794588	0.564787
64	0.564787	0.8364	0.785823	0.564787

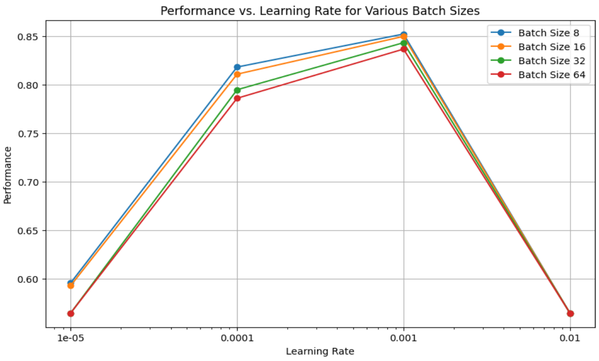

Fig. 8. Model Architecture

It can be seen that the model performance increases significantly when the learning rate fills from 0.00001 to 0.001, showing consistent performance improvements across all tested batch sizes. This improvement peaks at a learning rate of 0.001, where all models, regardless of batch size, achieve the highest performance. However, when the learning rate was further increased to 0.01, there was a drastic drop in performance, indicating that too high a learning rate may cause instability in the training process, perhaps due to too large steps in weight updates causing the model to exceed the global minimum or optimal.

This behavior is consistent with the common understanding in machine learning that appropriate learning rates are crucial in training neural models. A smaller learning rate may be too slow in convergence, requiring more epochs to reach maximum performance, while a learning rate that is too large may risk divergence in training due to too aggressive weight changes. Optimal performance around a learning rate of 0.001 for all batch sizes indicates that this is a balance point between efficient convergence and training stability.

The best results obtained were a combination of batch size 8 and learning rate of 0.001, with accuracy of 85.45%. This shows that the model that has been created will obtain optimal accuracy when the batch size value is reduced.

C. Discussion

After getting the most optimal results, a comparison was made between the performance of the model that had been built and the performance of previous research studies. As a comparison, research [20] wichh is the source of the dataset , used Random Forest Decision Tree (RFDT) classifier and achieved the best accuracy of 77.36%. Another research with same dataset [28] used GloVe-BiLSTM classifier and achieved the best accuracy of 82.31%, and research [29] used IndoBERTweet-BiGRU and achieved the best accuracy of 84.77%. The previous research shows that neural network models can produce better performance than machine learning models for classifying Indonesian-based texts, and also IndoBERTweet can improve the performance of a model, especially for Indonesian text classifiers.

TABLE IV
COMPARISON OF MODEL PERFORMANCE

Model	Accuracy
RFDT [20]	0.7736
GloVe-BiLSTM [28]	0.8231
IndoBERTweet [29]	0.8477
IndoBERTweet-BiLSTM-CNN	**0.8545**

The IndoBERTweet-BiLSTM-CNN achieved a good performance in text classification, particularly demonstrating its efficacy in hate speech detection in Indonesian text. By achieving an accuracy of 85.45% , this model illustrates the benefits of combining diverse neural network architectures to capitalize on their strengths in language understanding and processing. The integration of the IndoBERTweet model with transformer technology allows for an enhanced understanding of contextual nuances critical in text classification. Additionally, the BiLSTM component of the model helps in processing text data sequentially, and The CNN layers further contribute by extracting key features and patterns essential for identifying specific text classes effectively. This model not only surpasses traditional machine learning models such as the RFDT, which showed a maximum accuracy of 77.36%, but also demonstrates marked improvements over more sophisticated neural network setups like the GloVe-BiLSTM and IndoBERTweet-BiGRU. The robustness of the IndoBERTweet-BiLSTM-CNN model in handling the variability inherent in natural language makes it particularly adaptable to diverse linguistic expressions found in extensive and complex datasets.

D. Confusion Matrix

The classification model has been evaluated to compare the text ”Hate” with ”Non-Hate.” The results show that the model has significant thresholds for pressure, recall, and f1-score for each of the two categories, indicating its effectiveness in identifying and comparing the two types of texts.

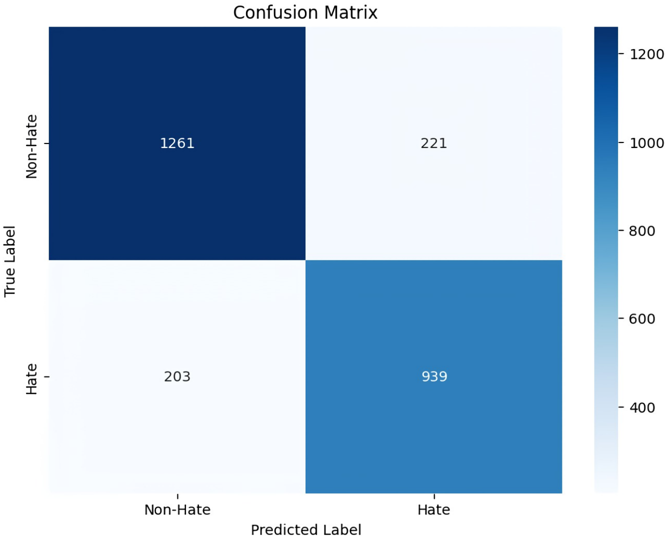

Fig. 9. Result of Confusion Matrix

Regarding the ”Non-Hate” category, the model achieves a precision of 0.8683 and a recall of 0.8583, resulting in a f1 score of 0.8633. This indicates that the model is quite effective in classifying the text ”Non-Hate,” with nearly equal ability to determine a reliable label and minimal reluctance to classify the text ”Hate” as ”Non-Hate.”

On the other hand, for the ”Hate” category, the model yields a f1 score of 0.8249 by indicating a precision of 0.8188 and a recall of 0.8310. Even while this score is lower than that of the ”Non-Hate” category, it nevertheless indicates that the model is rather effective in identifying the ”Hate” text. However, a somewhat more conservative hypothesis indicates that there are a few ’Non-Hate’ texts that are very certainly classified as ’Hate.’ This is supported by the recall test’s results, which show that the model successfully identified most of the major ’Hate’ texts.

In general, the model’s accuracy of 0.8545 indicates a very good workflow for classifying texts into two categories. The macro avg for precision, recall, and f1-score are, respectively, 0.8528, 0.8489, and 0.8506, indicating good balance between the two classes. The slightly higher weighted avg value (0.8542 for precision and 0.8541 for f1-score) indicates that the weight or proportion of samples in each class has been well taken into account in model evaluation.

V. CONCLUSIONS

The combined IndoBERTweet-BiLSTM-CNN model can detect Indonesian hate speech text really well. The teks processing carried out was data cleaning, lowercase, stopwords removal, and stemming for Indonesian text. After comparing the combined model with other models like Bilstm, CNN, and BiLSTM-CNN, it can be concluded that the created model outperforms the others. Comparisons have also been made with previous research that used method like RFDT, GloVe-BiLSTM, IndoBERTweet-BiGRU methods, and the combined model has also exceeded the performance of previous research. Parameter testing was also conducted to determine the best combination of batch size and learning rate, and the optimal values were found to be 8 and 0.001, respectively.The best accuracy and F1-Score produced were 85.45% and 85.06% . For further research, it is recommended to use a longer data set, because the model created is quite complex so it is usually used for datasets that contain longer and more complex text, such as product reviews or film reviews, and also another algorithm like IndoBERTweet-CNN-BiLSTM or improve this method by make a different architecture and layer units. Data balancing method such as SMOTE and under sampling can also be an option to improve this research in the future.

References

[1] G. Z. Nabiilah, S. Y. Prasetyo, Z. N. Izdihar, and A. S. Girsang, "Bert base model for toxic comment analysis on indonesian social media," *Procedia Computer Science*, vol. 216, pp. 714–721, 2023.

[2] K. M. Hana, S. Al Faraby, A. Bramantoro, *et al.*, "Multi-label classification of indonesian hate speech on twitter using support vector machines," in *2020 International Conference on Data Science and Its Applications (ICoDSA)*, pp. 1–7, IEEE, 2020.

[3] S. Kurniawan and I. Budi, "Indonesian tweets hate speech target classification using machine learning," in *2020 Fifth International Conference on Informatics and Computing (ICIC)*, pp. 1–5, IEEE, 2020.

[4] D. C. Asogwa, C. I. Chukwuneke, C. Ngene, and G. Anigbogu, "Hate speech classification using svm and naive bayes," *arXiv preprint arXiv:2204.07057*, 2022.

[5] A. P. J. Dwitama, D. H. Fudholi, S. Hidayat, *et al.*, "Indonesian hate speech detection using bidirectional long short-term memory (bi-lstm)," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 2, pp. 302–309, 2023.

[6] D. A. N. Taradhita and I. Darma Putra, "Hate speech classification in indonesian language tweets by using convolutional neural network.," *Journal of ICT Research & Applications*, vol. 14, no. 3, 2021.

[7] E. Sazany and I. Budi, "Hate speech identification in text written in indonesian with recurrent neural network," in *2019 International Conference on Advanced Computer Science and information Systems (ICACSIS)*, pp. 211–216, IEEE, 2019.

[8] P. H. Zakaria, D. Nurjannah, and H. Nurrahmi, "Misogyny text detection on tiktok social media in indonesian using the pre-trained language model indobertweet," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 7, no. 3, pp. 1297–1305, 2023.

[9] S. Saadah, K. M. Auditama, A. A. Fattahila, F. I. Amorokhman, A. Aditsania, A. A. Rohmawati, *et al.*, "Implementation of bert, indobert, and cnn-lstm in classifying public opinion about covid-19 vaccine in indonesia," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 4, pp. 648–655, 2022.

[10] H. M. Lee and Y. Sibaroni, "Comparison of indobertweet and support vector machine on sentiment analysis of racing circuit construction in indonesia," *Jurnal Media Informatika Budidarma*, vol. 7, no. 1, pp. 99–106, 2023.

[11] A. D. Maulana and K. M. Lhaksmana, "Sentiment analysis on tweets of kanjuruhan tragedy using deep learning indobertweet," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 7, no. 3, pp. 948–955, 2023.

[12] R. Ma, S. Teragawa, and Z. Fu, "Text sentiment classification based on improved bilstm-cnn," in *2020 Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, pp. 1–4, IEEE, 2020.

[13] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "Indolem and indobert: A benchmark dataset and pre-trained language model for indonesian nlp," *arXiv preprint arXiv:2011.00677*, 2020.

[14] F. Koto, J. H. Lau, and T. Baldwin, "Indobertweet: A pretrained language model for indonesian twitter with effective domain-specific vocabulary initialization," *arXiv preprint arXiv:2109.04607*, 2021.

[15] K. K. Dubey, R. Nair, M. U. Khan, and P. S. Shaikh, "Toxic comment detection using lstm," *2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, pp. 1–8, 2020.

[16] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *Proceedings of the 26th international conference on World Wide Web companion*, pp. 759–760, 2017.

[17] S. Khan, M. Fazil, V. K. Sejwal, M. A. Alshara, R. M. Alotaibi, A. Kamal, and A. R. Baig, "Bichat: Bilstm with deep cnn and hierarchical attention for hate speech detection," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 7, pp. 4335–4344, 2022.

[18] E. F. Ayetiran, "Attention-based aspect sentiment classification using enhanced learning through cnn-bilstm networks," *Knowledge-Based Systems*, vol. 252, p. 109409, 2022.

[19] S. Imron, E. I. Setiawan, J. Santoso, M. H. Purnomo, *et al.*, "Aspect based sentiment analysis marketplace product reviews using bert, lstm, and cnn," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 3, pp. 586–591, 2023.

[20] M. O. Ibrohim and I. Budi, "Multi-label hate speech and abusive language detection in indonesian twitter," in *Proceedings of the third workshop on abusive language online*, pp. 46–57, 2019.

[21] D. Fimoza, A. Amalia, and T. H. F. Harumy, "Sentiment analysis for movie review in bahasa indonesia using bert," in *2021 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, pp. 27–34, IEEE, 2021.

[22] D. Jurafsky and J. H. Martin, "Speech and language processing (3rd (draft) ed.)," 2019.

[23] M. Mashuri *et al.*, "Sentiment analysis in twitter using lexicon based and polarity multiplication," in *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT)*, pp. 365–368, IEEE, 2019.

[24] B. Siswanto and Y. Dani, "Sentiment analysis about oximeter as covid-19 detection tools on twitter using sastrawi library," in *2021 8th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE)*, pp. 161–164, IEEE, 2021.

[25] R. Cai, B. Qin, Y. Chen, L. Zhang, R. Yang, S. Chen, and W. Wang, "Sentiment analysis about investors and consumers in energy market based on bert-bilstm," *IEEE access*, vol. 8, pp. 171408–171415, 2020.

[26] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC genomics*, vol. 21, pp. 1–13, 2020.

[27] A. O. Alzahrani and M. J. Alenazi, "Designing a network intrusion detection system based on machine learning for software defined networks," *Future Internet*, vol. 13, no. 5, p. 111, 2021.

[28] R. A. Ilma, S. Hadi, and A. Helen, "Twitter's hate speech multi-label classification using bidirectional long short-term memory (bilstm) method," in *2021 International Conference on Artificial Intelligence and Big Data Analytics*, pp. 93–99, IEEE, 2021.

[29] A. Marpaung, R. Rismala, and H. Nurrahmi, "Hate speech detection in indonesian twitter texts using bidirectional gated recurrent unit," in *2021 13th International Conference on Knowledge and Smart Technology (KST)*, pp. 186–190, IEEE, 2021.

[30] L. Geni, E. Yulianti, and D. I. Sensuse, "Sentiment analysis of tweets before the 2024 elections in indonesia using indobert language models," *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika (JITEKI)*, vol. 9, no. 3, pp. 746–757, 2023.

[31] D. Granziol, S. Zohren, and S. Roberts, "Learning rates as a function of batch size: A random matrix theory approach to neural network training," *Journal of Machine Learning Research*, vol. 23, no. 173, pp. 1–65, 2022.