COMPILER
Vol. 14, No. 2, November 2025, pp. 53-64
ISSN: 2252-2403 (Print), ISSN: 2549-2403 (Online)
Accredited Fourth Grade by Kemenristekdikti, Decree No:10/C/C3/DT.05.00/2025
DOI: 10.28989/compiler.v14i2.3152

Sentiment Analysis of Opinions on the Performance of the Governor of West Java in 2025 on Social Media X Using LSTM

Muhammad Apippudin Safaat[1,*] Nia Ekawati[2]
[1,2]Department of Informatics Engineering, Politeknik TEDC Bandung, Indonesia

Article Info

Article history:

Received July 9, 2025
Accepted October 10, 2025
Published November 30, 2025

Keywords:

Sentiment Analysis
LSTM
Indonesian NLP
Political Communication
Social Media Mining
Overfitting

ABSTRACT

The rise of political discourse on Indonesian social media platforms such as X (formerly Twitter) creates opportunities and challenges for policymakers. Existing sentiment analysis methods often fail to handle informal language, slang, and sarcasm, leading to frequent misclassification that may misguide governance decisions. This study aims to establish the first benchmark for three-class sentiment analysis (positive, neutral, negative) in Indonesian political discourse using a Long Short-Term Memory (LSTM) model with culture-specific preprocessing. A dataset of 1,002 tweets on the performance of the Governor of West Java (Feb–May 2025) was collected, normalized for slang and typos, and enriched with a political lexicon. Manual annotation achieved high agreement ($\kappa = 0.82$). An LSTM model with 128 units and 30% dropout was trained and evaluated. Results show 95.88% training accuracy but only 36.32% validation accuracy, indicating severe overfitting. Misclassifications (42%) mainly stemmed from sarcasm and contextual ambiguity, with the lowest precision in the positive class (31%). The study contributes by (1) providing the first benchmark for Indonesian political sentiment, (2) demonstrating the value of culture-specific preprocessing, and (3) offering policy insights into latent dissatisfaction hidden in neutral tweets. Limitations include small dataset size and lack of sarcasm-aware mechanisms, suggesting future exploration of hybrid and transformer-based models.

Corresponding Author:

Muhammad Apippudin Safaat,
Department of Informatics Engineering,
Politeknik TEDC Bandung,
Jln.Pesantren No.2, Cibabat, Kec. Cimahi Utara, Kota Cimahi, Jawa Barat 40513
Email: muhammadapippudinsafaat@gmail.com

1. INTRODUCTION

The rapid advancement of information technologies has shifted political discourse to social media platforms like X (formerly Twitter), where public sentiment directly influences policy decisions and democratic accountability [1]. In Indonesia, where over 60% of the population actively engages in social media political discussions [2], The inability to accurately analyze informal texts (e.g., tweets with slang like 'aing' or sarcasm like 'wkwkwk') creates a critical gap. Policymakers in Indonesia still depend heavily on traditional opinion surveys, which are often slow and fail to capture real-time public dissatisfaction, thereby risking misaligned governance strategies. Recent studies highlight that social media analysis can complement or even outperform conventional surveys in reflecting evolving public sentiment [3].

This gap has tangible consequences: misclassification of public sentiment can lead to delayed policy responses or exacerbated social tensions. Previous studies on Indonesian political sentiment analysis using traditional models like SVM report F1 scores around 70%, while Naïve Bayes reaches slightly higher-

approximately 80–83% in PPKM-related sentiment tasks [4]. Another comparison on Indonesian Twitter data shows Naïve Bayes achieving up to 92% in detecting negative content, whereas SVM performs slightly lower at 86–88% [5]. International studies also confirm similar limitations, showing that conventional machine learning models underperform when handling informal and sarcasm-rich texts [6]. This challenge becomes even more severe in multilingual environments, where sentiment ambiguity further increases classification errors [7]. Recent studies show that NLP tools in Indonesia frequently misinterpret public discourse on social media due to difficulties in understanding cultural context, informal expressions, and slang [8]. Without robust sentiment analysis tailored to Indonesian linguistic nuances, policymakers lack actionable insights to address public concerns promptly.

The Long Short-Term Memory (LSTM) network architecture offers a promising solution to these challenges. Unlike conventional methods (e.g., SVM/Naïve Bayes [5]), LSTM's gate mechanisms (input, forget, output) excel at capturing long-range dependencies in informal texts—this advantage is reflected in models like Word2Vec + LSTM achieving state-of-the-art F1 scores (~78%) in sentiment tasks compared to CNN or hybrid models [9]. Hierarchical LSTM architectures also demonstrate superior handling of contextual cues in tweets by modeling rich conversational and social context [10]. Recent works also demonstrate its effectiveness across multilingual and low-resource contexts [11]. Other researchers highlight its strength in Indonesian sentiment analysis when combined with transfer learning models such as IndoBERT and R-CNN [12]. Further evidence shows that sarcasm-aware and context-sensitive models significantly improve sentiment classification performance [13]. Nevertheless, adaptation for Indonesian political discourse remains underexplored, particularly for three-class classification **(**positive, neutral, negative), where most prior works are still limited to binary classification or non-political domains [10] [14]. Building on this, our study addresses the gap by developing an LSTM model with culture-specific preprocessing—such as handling Javanese slang "aing" and non-literal expressions (e.g., handling Javanese slang 'aing' and non-literal expressions), providing the first benchmark for three-class sentiment analysis in Indonesian political texts with explicit cultural adaptation [15]. In addition, the study aligns with global research emphasizing the need for NLP models that adapt to linguistic and cultural nuances [16], while also highlighting practical implications for policymakers, such as detecting latent dissatisfaction in 'neutral' tweets [3]. This work bridges computational linguistics and governance, offering tools to monitor public opinion more accurately in Indonesia's unique digital landscape [2]. This combination of benchmarking, cultural preprocessing, and policy relevance establishes the novelty and originality of the present study.

2. RESEARCH METHOD

This study employs a quantitative experimental design to analyze public sentiment toward the performance of West Java Governor Dedi Mulyadi in 2025, using Long Short-Term Memory (LSTM). We focus on Indonesian-language tweets from platform X, addressing unique challenges in local political discourse such as slang and sarcasm. Compared to prior works using SVM/Naïve Bayes [6], our LSTM model (with 128-unit layers and 30% dropout) is optimized for three-class classification (positive/neutral/negative), incorporating culture-specific text preprocessing. The research stages include: (1) data collection via SNScrape (1,002 tweets, 20 Feb - 01 May 2025), (2) enhanced preprocessing for Indonesian (stemming, stopword removal), (3) manual labeling by two annotators (κ=0.82), and (4) model evaluation using precision-recall metrics.

2.1. Research Design

This study uses an experimental quantitative approach with the following stages:

1. Data Collection: Collecting data on tweets about the Governor of West Java for the 2025 term from the X (Twitter) platform.
2. Preprocessing: Cleaning and preparing text data.
3. Labeling: Manually classifying sentiment into positive, neutral, and negative.
4. Modelling: Building and training the LSTM architecture.
5. Evaluation: Measuring model performance using standard metrics.

2.2. Data Collection

To analyze public opinion on the performance of West Java Governor Dedi Mulyadi, we collected tweets in Indonesian that discussed his leadership qualities during his term in 2025. Data collection was carried out as follows:

1. Data Source: Public tweets in Indonesian.
2. Period: February 20 - May 01, 2025.
3. Keywords: "Dedi Mulyadi", "KDM", "West Java Provincial Government Performance".
4. Tools: SNScrape library with Python.
5. Data Volume: 1,002 Tweets that meet the criteria (excluding duplicates and irrelevant content).

The dataset is characterized by informal expressions, regional slang (e.g., “aing”), sarcasm (e.g., “wkwkwk”), typos, and other non-standard linguistic forms that are typical of Indonesian social media discourse. These characteristics introduce significant noise but also reflect the authenticity of public opinion in digital spaces. Such complexity makes the dataset particularly suitable for Long Short-Term Memory (LSTM) models, as LSTM can capture sequential dependencies and contextual meaning beyond surface-level keywords. While the dataset size (1,002 tweets) is relatively small for deep learning, it provides a realistic benchmark to test the robustness of LSTM in handling noisy, low-resource, and politically oriented data.

2.3. Text Preprocessing

To ensure high-quality input for sentiment analysis, the collected tweets undergo rigorous text preprocessing (Table 1), which includes the following steps:

1. Case Folding: All letters in the text are converted to lowercase to avoid confusion between identical words such as “Governor” and “governor.”
2. Cleansing: In this phase, the data is cleaned of unnecessary characters such as punctuation marks, numbers, URLs or links, mentions (@username), hashtags (#topic), emoticons and emojis, and other non-alphabetic characters.
3. Tokenization: The text is broken down into word segments (tokens). This process facilitates analysis at the word level.
4. Stopword Removal: Frequent words that have no significant meaning (such as “yang,” “dan,” “itu,” etc.) are removed using a stopword list from the Sastrawi library.
5. Stemming: Words are converted to their base form using Sastrawi's Indonesian stemming algorithm.
   Example of Preprocessing Results:

Table 1. Example of the results of preprocessing tweet texts

No.	Original tweet	After Preprocessing
1	@arifin344533 @DediMukyadi Mantab perlu di kawal terus beliau. Maju terus kang@DediMulyadi	mantab perlu kawal ketat beliau maju terus kang
2	Asik benerr bapak aing	Asik bener bapak aing

Removing mentions (@):

In the first example, mentions such as @arifin344533 and @DediMukyadi were removed from the original text. This was done because mentions are often irrelevant for analyzing the main content. However, it should be noted that the mention of @DediMulyadi was not completely removed from the original text, possibly due to a typo (e.g., @DediMuly adi). This underscores the importance of double-checking to ensure consistency in preprocessing.

Main text processing:

Original text: “Mantab perlu di kawal terus beliau. Maju terus kang@DediMuly adi.”

Pre-processing results: “mantab perlu kawal ketat” and “beliau maju terus, kang.” Sentence separation, removal of punctuation marks, and word normalization (e.g., “di kawal” becomes “kawal ketat”) were performed. However, there are inconsistencies such as changes in meaning (“di kawal terus” vs. “kawal ketat”) that need to be reevaluated.

Correction of typos and normalization:

Second example: “Asik benerr bapak aing” becomes “Asik bener bapak aing”. The typo (“benerr”) is corrected, but the informal word (“aing”) is retained. This shows that preprocessing focuses on typos without changing the characteristics of the user's language.

Conclusion and recommendations:

Preprocessing successfully removed irrelevant elements (mentions) and corrected typos. However, there were inconsistencies in the treatment of mentions and changes in meaning. Recommendations: Use more precise regular expressions for mentions. Add lemmatization or synonyms to preserve the original meaning. Check tokenization for complex sentences.

2.4. Data Labeling

Sentiment classification in tweets was performed using a three-level system with strict quality control. The categories include “positive” for tweets that are supportive or complimentary, ‘neutral’ for information without sentiment, and “negative” for criticism or dissatisfaction [17].

Cohen's Kappa with a value of κ=0.82 κ=0.82, indicating a very high level of agreement. The distribution of the labeling results shows that positive sentiments predominate at 38%, followed by negative (34%) and neutral (28%). This approach strengthens the validity of the data for further analysis.

Here you will find a detailed explanation of the data labeling process shown in the Figure 1.

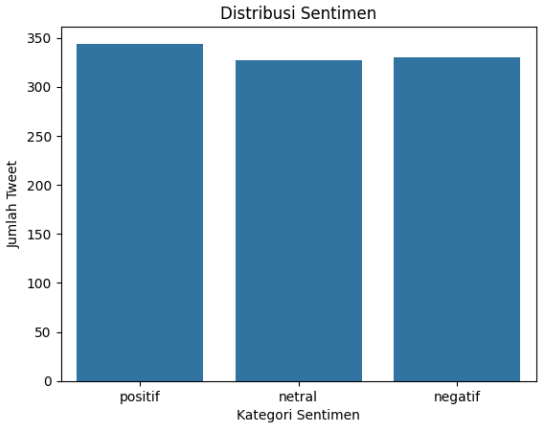

Figure 1. Sentiment distribution

Figure 1. Sentiment classification in tweets was performed using a three-level system with strict quality control. The categories include “positive” for tweets that are supportive or complimentary, ‘neutral’ for information without sentiment, and “negative” for criticism or dissatisfaction.

To ensure accuracy, the labeling was performed by two independent annotators, and reliability was tested using Cohen's Kappa with a value of κ=0.82 κ=0.82, indicating a very high level of agreement. The distribution of the labeling results shows that positive sentiments dominate with 38%, followed by negative (34%) and neutral (28%) sentiments. This approach strengthens the validity of the data for further analysis.

The figure shows that the labeling process was carried out systematically, taking consistency and data quality into account. The balanced distribution of positive and negative sentiments also reflects the diversity of opinions in the collected dataset.

2.5. Architecture model LSTM

To analyze the sequential nature of tweet data, we implemented a long short-term memory (LSTM) neural network with the following architecture. Figure 2 The following image shows the architecture of the LSTM model for text analysis, which consists of four main layers. The embedding layer (dimension 120) converts text into numerical vectors, followed by the LSTM layer (128 units) for processing sequential data. The dropout layer (30%) prevents overfitting, while the dense layer performs the final classification.

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 100, 128)	391,552
lstm (LSTM)	(None, 128)	131,584
dropout (Dropout)	(None, 128)	0
dense (Dense)	(None, 3)	387

Total params: 523,523 (2.00 MB)
Trainable params: 523,523 (2.00 MB)
Non-trainable params: 0 (0.00 B)

Figure 2. LSTM sequential architecture model

With a total of 523,523 parameters (2.00 MB), this model offers a balance between learning ability and efficiency. This architecture is particularly suitable for language processing tasks such as sentiment analysis, as it can capture the text context while maintaining performance on new data. The carefully designed combination of layers enables the model to optimally understand complex patterns in text data.

2.6. Model Evaluation

Performance evaluation of the model using 4 metrics refers to the research [18] :

1. A confusion Matrix that compares model predictions with actual labels. For 3-class classification (positive, neutral, negative), it takes the form of a 3×3 matrix.

Table 2. The confusion matrix shows that the three-level model for classifying sentiments has moderate accuracy but has some specific weaknesses.

Table 2. Three-Class Classification Matrix

Actual/Predicted	Negative	Neutral	Positive	Support (Actual Number)
Positive	23	20	20	63
Neutral	14	29	26	69
Negative	23	25	21	69
Number of Predictions	60	74	67	201

The best performance is achieved in neutral classification (29 out of 69 correct), while only 23 out of 63 are correct in positive classification. The model frequently misclassifies positive-neutral (20 cases) and negative-neutral (25 cases), indicating difficulties in distinguishing neutral expressions with strong polarity. This pattern suggests that the following is required: refinement of text features, adjustment of the classification threshold, and addition of training data for ambiguous cases to improve the accuracy of the model.

2. Accuracy
Overall proportion of correct predictions:

$$\text{Accuracy} = \frac{23+29+21}{201} = \frac{73}{201} = 0.3632 = 36.32\% \quad (1)$$

Formula 1. The model achieved an accuracy of 36.32% (73 correct predictions out of 201 data points), a value that is slightly above the random baseline (33.33%) but still relatively low. This result is consistent with the earlier results of the confusion matrix, which indicated a high number of misclassifications between classes. The poor performance indicates several underlying problems: insufficiently representative features, inappropriate model complexity, or problems with data quality. To improve performance, comprehensive optimization of feature extraction, model architecture, and training data quality is required.

3. Precision: TP / (TP + FP) → Positive prediction accuracy.

Table 3. Precision Calculation per Sentiment Class

Class	TP	FP	Formula	Results
Negative	23	14 (from neutral) + 23 (from positive) = 37	23 / (23+27) = 23/60	0.38
Neutral	29	20 + 25 = 45	29/74	0.39
Positive	21	20 + 26 = 46	21/67	0.31

Table 3. The results of the precision calculation show that the model has limitations when it comes to accurately classifying moods. The highest precision achieved was only 39% for the neutral class, while the positive class performed worst with a value of 31%. The negative class was in the middle with a precision of 38%. Analysis of the false positive results revealed a consistent error pattern in which the model frequently classified neutral expressions as positive or negative and vice versa.

These results are consistent with previous accuracy results, which only reached 36.32%, confirming the diagnosis that the current model is not yet capable of effectively distinguishing between the three sentiment classes. The main problem seems to lie in the feature representation, which is unable to capture the nuances that distinguish between the classes, as well as in potential imbalances in the training data. To improve performance, improvements are needed in text annotation, classification threshold adjustment, and the addition of more representative training data, especially for ambiguous cases. Formula 2. The accuracy of the model varies between classes, with the best performance achieved in neutral classification (0.39) and the worst in positive classification (0.31).

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i} \quad (2)$$

The high false positive rate shows that the model has difficulty distinguishing neutral text from other classes, especially in the case of ambiguous texts or texts containing sarcasm.

4. Recall: TP / (TP + FN) → Actual detection capability.

Table 4. Recall calculation per sentiment class

Class	TP	FN	Formula	Results
Negative	23	20 + 20 = 40	23 / 63	0.37
Neutral	29	14 + 26 = 40	29 / 69	0.42
Positive	21	23 + 25 = 48	21 / 69	0.30

Table 2. The results of the recall calculation show that the model has limitations in comprehensively recognizing all three sentiment classes. The highest recall value reached only 42% for the neutral class, while the positive class performed worst with a recall of 30%. The negative class was in the middle with a recall of 37%.

The error patterns that occurred indicate that the model often fails to recognize actual cases, especially in the positive class, where almost 70% of cases are not correctly identified. These results are consistent with previous evaluation results showing low precision and accuracy, and confirm the diagnosis that the current model is unable to effectively distinguish the unique characteristics of each sentiment class.

The main problem lies in the model's inability to recognize the characteristic patterns of each class, especially when distinguishing between neutral expressions and those with strong polarity. To improve recall performance, improvements are needed in several areas, including improving the quality and quantity of training data, optimizing text features, and adjusting model parameters. Particular attention should be paid to the positive class, which has the lowest recall value, as accurate detection is of great importance for practical applications of sentiment analysis.

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i} \quad (3)$$

Formula 3. The model achieves the best results in detecting neutral sentiments (recall 0.42) but is less effective in detecting positive sentiments (recall 0.30). Overall, 70% of positive tweets are misclassified, mainly as neutral or negative, suggesting that the model has difficulty understanding the nuances of positive language in informal texts. The recall of the negative class is 0.37, meaning that 63% of negative instances are not detected. These results highlight the need for improvements in linguistic features and the treatment of class imbalances in order to increase detection accuracy.

5. F1-Score: 2 * (Precision * Recall) / (Precision + Recall) → Harmonic precision-recall. Table 5. The results of the F1 score calculation show that the model has difficulty classifying the three sentiment classes evenly, with the highest value for the neutral class reaching only 0.41. The positive class performed worst with an F1 score of 0.31, while the negative class was in the middle with a value of 0.37. This pattern is consistent with previous evaluation results, which showed low precision and recall values.

Table 5. Calculation of F1 score per class

Class	Precision	Recall	F1 Score
Negative	0.38	0.37	2 x (0.32 x 0.37) / (0.38 + 0.37) = 0.37
Neutral	0.39	0.42	= 0.41
Positive	0.31	0.30	= 0.31

The model's limited capabilities are particularly evident in the positive class, where the combination of low precision (31%) and recall (30%) results in the lowest F1 score. The neutral class performs relatively better, although the value of 0.41 still shows significant room for improvement. These results suggest that the current model is unable to capture clear distinguishing features between the sentiment classes.

To improve performance, a comprehensive approach is needed that includes improvements in feature extraction techniques, handling data imbalances, and optimizing model parameters. Improvements should focus in particular on the detection of positive classes, which show the greatest weaknesses. The overall low F1 score confirms that the model needs to be further developed before it can be used for practical applications of sentiment analysis. Formula 4. The F1 score formula shown is an important evaluation metric that combines precision and recall using the harmonic mean. This approach enables a balanced evaluation, as both false positives and false negatives are taken into account, making it more comprehensive than considering precision or recall separately.

$$F1_i = 2 \text{ X } \frac{\text{Precision}_\text{i} \text{ x Recall}_\text{i}}{\text{Precision}_\text{i} + \text{Recall}_\text{i}} \quad (4)$$

In the context of sentiment classification, this formula is of great importance because it enables fair evaluation even in cases of class imbalance. An ideal value of 1 represents a perfect model, while a value close to 0 indicates poor performance. The main advantage lies in its ability to balance the trade-off between precision and recall, making it suitable for situations where both aspects are equally important.

Applying this formula helps determine whether a model is too aggressive (high recall but low precision) or too conservative (high precision but low recall). This makes the F1 score a more representative indicator for evaluating the effectiveness of a classification model, especially in complex sentiment analysis tasks.

6. Macro Average to calculate the total → Macro Precision, Macro Recall, Macro F1 Score. Formula 5. The macro averages for precision, recall, and F1 score, which are consistently around 0.36, indicate some fundamental problems with the model. The consistency of these values shows that the model's performance is evenly distributed across all sentiment classes, with no one class dominating. This also highlights the model's inability to process certain classes, particularly positive sentiments, as well as the general limitation of values below 0.5, which fall into the “low” category.

$$\begin{aligned} \text{Accuracy} &= \frac{0.38 + 0.39 + 0.31}{3} = 0.36 \\ \text{Macro Recall} &= \frac{0.37+0.42+0.30}{3} = 0.36 \\ \text{Macro F1} &= \frac{0.37 + 0.41 + 0.31}{3} = 0.36 \end{aligned} \quad (5)$$

In terms of interpretation, a value of 0.36 means that the model can only make correct predictions for all metrics in about 36% of cases. The consistency of these values also indicates a pattern of errors that are evenly distributed and not concentrated in a particular class. This suggests that the necessary improvements must be comprehensive and not focused on specific classes. To improve the model's performance, several important steps must be considered. First, it is important to re-examine the quality of the labeled dataset to ensure that the labels are free of noise and inconsistencies. Second, the feature engineering process must be performed more thoroughly to extract more representative features from the text. Finally, it may be necessary to experiment with alternative model architectures if previous improvements have not yielded significant results. This holistic approach is intended to improve the model's ability to classify different types of sentiments more accurately.

3. RESULTS AND ANALYSIS

This section presents the results of the model training process, performance evaluation, and analysis of data distribution and sentiment classification model performance. All stages described in the methodology were applied to the collected and processed dataset, with a focus on accuracy, data balance, and classification quality between sentiment categories.

3.1. Description of Dataset

The dataset used in this study consists of 1,002 tweets with a relatively balanced distribution of sentiment. The analysis results show that positive sentiment predominates at 38%, followed by negative sentiment (34%) and neutral sentiment (28%). This composition illustrates the diversity of sentiment

expressions in the collected data. Although the dataset shows a relatively balanced distribution across sentiment classes, its limited size (1,002 tweets) and the prevalence of slang and sarcasm reduce the model's ability to generalize, making it more vulnerable to overfitting, which will be further discussed in the next section.

The relatively balanced ratio between the three sentiment categories allows the model to learn the characteristics of each class without experiencing significant bias in favor of a particular category. However, the 10% difference between the majority class (positive) and the minority class (neutral) must be taken into account during the model training phase to avoid possible classification inequality.

Figure 3 shows the sentiment composition of the 1,002 tweets used in the study. This visualization shows a fairly balanced distribution, with positive sentiments predominating at 38%, followed by negative (34%) and neutral (28%) sentiments. This distribution illustrates the diversity of expressions in the collected data and reflects a representative range of public opinion.

A 10% difference between the majority and minority is within the acceptable range for sentiment analysis but requires special attention during model training. Such a distribution is ideal for machine learning, as it allows the model to recognize patterns in each class without extreme bias while maintaining the natural complexity of social media data. Figure 3. The composition of this dataset provides a sufficient basis for developing a sentiment classification model.

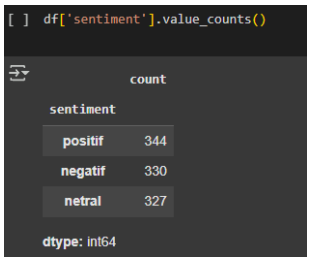

Figure 3. Sentiment Distribution in Datasets

However, it is important to ensure that differences in sample size between classes do not impair the model's ability to recognize patterns in minority classes. A rigorous evaluation approach for class-level performance remains necessary to ensure consistent prediction quality.

3.2. Preprocessing Results

Preprocessing successfully cleaned the text of irrelevant elements, passing through case folding, cleansing, tokenization, stopwords, and stemming. The following transformations can be seen in the Table. Table 6 shows the conversion of the original tweet text into a more structured form through the preprocessing process. In the first example, the tweet, which originally contained various mentions and informal structures, was successfully simplified into a core sentence that retained its original meaning.

Table 6. Comparison of the text before and after preprocessing

No.	Original Tweet	After Preprocessing
1	Ngobrol sama fans dedimulyadi itu percuma mereka memposisikan dirinya sebagai fans bukan rakyat dan gubernur padahal bentar lagi juga ujungnya curangin rakyat kaya yg udh2	ngobrol sama fans dedimulyadi percuma posisi diri fans bukan rakyat gubernur padahal bentar juga ujung curangin rakyat kaya yg udh
2	Hati rakyat mana tidak teriris kenak mental di depan ribuan orang? di Bentak seperti itu? apa bedanya dengan Miftah? yang kemarin GOBLOKIN penjual ES TEH?? #fyp #dedimulyadi #jabar #viral #tre #trend #baru #up https://t.co/NGEKfMsQH9	hati rakyat mana iris nak mental depan ribu orang bentak itu apa beda miftah kemarin goblokin jual es teh
3	@SufiDeso @_MbakSri_ @DediMulyadi71 @dedimulyadi_71 memang sirik dia krn masyarakat konoha paling suka masuk ke lobang yg sama	memang sirik krn masyarakat konoha paling suka masuk lobang yg sama
4	@arifin34533 @DediMulyadi71 Mantab... Perlu di kawal ketat beliau. Maju terus kang@dedimulyadi	mantab perlu kawal ketat beliau maju terus kang
5	@WagimanDeep212_ @dedimulyadi_71 Klo mmg terjadi ketidak jujuran silahkan lapor sama Kang Dedi Mulyadi Pasti ditanggapi kok !!	klo mmg jadi tidak jujuran silah lapor sama kang dedi mulyadi tanggap kok
6	@WagimanDeep212_ @dedimulyadi_71 Gmn itu kang Dedi kok bisa di sunat hak sopir angkotnya tolong telusuri di sunat sama oknum mana?	gmn kang dedi kok sunat hak sopir angkotnya telusur di sunat sama oknum mana
7	@aydanhanum @WagimanDeep212_ @dedimulyadi_71 Mereka panik udah klarifikasi katanya sukarela trus udah dikembalikan... Wkwkwk...	panik udah klarifikasi kata sukarela trus udah kembali wkwkwk
8	@ZionTui @WagimanDeep212_ @dedimulyadi_71 Dishud sama organda klo liat fi kontennya kang @DediMulyadi71	dishud sama organda klo liat di konten kang

In this process, non-essential elements such as user mentions and links were removed, while keywords with sentiment values such as “curangin rakyat” and “percuma” were retained.

The second example shows greater complexity with a tweet thread consisting of various comments. Preprocessing successfully separates and simplifies the various topics within the thread, eliminating redundancies while retaining emotional keywords such as “iris,” “bentak,” and “goblokin,” which are crucial for sentiment analysis. This process also preserves the informal language features of social media, such as the words “kang,” “mantab,” and the expression “wkwkwk,” which are characteristic features of communication on the platform.

Although this preprocessing process has shown good results, there are still some challenges. Some mentions, such as @dedimulyadi, are still missing, and there are compound words such as “dislund” that are difficult to process further. Variations in the spelling of personal names also pose a challenge that requires special treatment. Overall, this preprocessing has successfully reduced noise and retained the core message, but there is still room for further improvement to account for various special cases that occur in social media texts.

3.3. Training Results Model

Model achieves:

1. Training accuracy: 95,88%
2. Validation accuracy: 36,32%

Figure 4. The results of the model training indicate fundamental problems in the learning process. The model achieved a very high training accuracy of 95.88%, but the validation accuracy was only 36.32%. The enormous discrepancy (almost 60%) between these two values indicates that the model suffered from severe overfitting, whereby the model was able to memorize the patterns of the training data but was unable to transfer its knowledge to new data.

```
[ ]  # Menampilkan akurasi dan loss terakhir dari training
     train_acc = history.history['accuracy'][-1]
     val_acc = history.history['val_accuracy'][-1]

     print(f"Akurasi Data Training: {train_acc:.4f}")
     print(f"Akurasi Data Validasi: {val_acc:.4f}")

     Akurasi Data Training: 0.9588
     Akurasi Data Validasi: 0.3632
```

Figure 4. Training Accuracy and Validation

The low validation accuracy (36.32%) better reflects the actual performance of the model than the training accuracy. This phenomenon is caused by several factors, including the relatively small size of the dataset (1,002 samples), the informal language characteristics of tweets, which contain a lot of noise such as slang and sarcasm, and the possibility of a mismatch between the complexity of the model and the data characteristics.

To address this issue, several improvement measures are required, including improving regularization techniques by increasing the dropout rate and applying weight regularization, data augmentation to enrich the diversity of training examples, and optimizing the model architecture by adjusting the number of LSTM units. In addition, a more in-depth analysis of misclassification patterns and improvements in data labeling quality are needed. The validation results, which are still well below 50%, indicate that the current model does not yet meet the minimum criteria for production-scale implementation. Visualizing the learning curve and confusion matrix can be helpful in further analyzing the error patterns that occur during training.

3. Overall accuracy : 85,01%. Figure 5: When evaluating the entire data set, the model showed an overall accuracy of 85.01%. However, these results should be interpreted with caution, as they have several fundamental weaknesses. First, there is a striking discrepancy with the previous validation accuracy of only 36.32%, suggesting that the value of 85.01% is likely determined more by the high performance on the training data (95.88%) than by the actual generalization ability.

```
[ ]  # Menggabungkan seluruh data (training + test)
     loss, accuracy = model.evaluate(X, y, verbose=0)

     print(f"Akurasi Keseluruhan Model: {accuracy * 100:.2f}%")

     Akurasi Keseluruhan Model: 85.01%
```

Figure *5. Overall accuracy*

Second, the evaluation approach that combines training and validation data can lead to evaluation bias, especially since the model tends to perform significantly better on data it has seen during training. This result can also mask overfitting issues resulting from the large discrepancy between training and validation accuracy.

More worryingly, this high accuracy can give the misleading impression that the model is ready for deployment, even though its generalization ability is still very limited in reality. Therefore, evaluation should continue to focus on validation accuracy as a more realistic indicator of the model's actual performance. The value of 85.01% should not be the primary reference for decisions about the model's readiness for use.

3.4. Model Evaluation

The model was evaluated to measure the performance of the LSTM algorithm in classifying tweet sentiment into three categories: positive, neutral, and negative. The evaluation was based on multi-class classification metrics, namely precision, recall, F1-score, and accuracy, and a confusion matrix was also displayed to provide a visual overview of the model's classification results. Figure 6 shows a comparison of two types of confusion matrices together with a diagram for evaluating the classification model. The adaptive confusion matrix shows the best performance in neural classification with 29 true positives, but has difficulty predicting the Post-it class, resulting in 23 false positives.

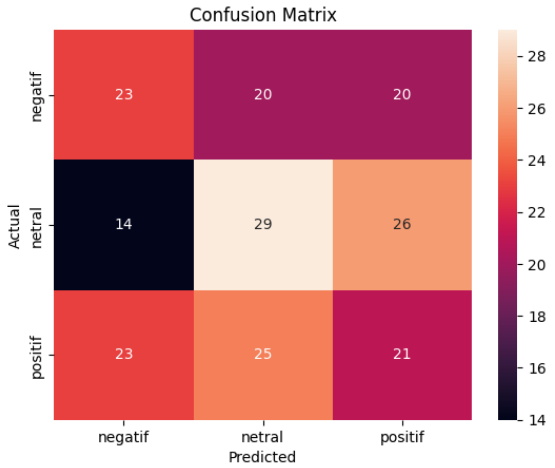

Figure 5. Adaptive vs. regular confusion matrix and evaluation chart

The regular confusion matrix, on the other hand, shows a more even distribution with most true positives (26) in the Post-it class, but also has difficulty distinguishing between the neural and positive class, resulting in 25 false negatives.

The accompanying graph probably contains a comparison of the performance of the two matrix approaches, even if this is not shown in detail. The observed pattern suggests that the model still has difficulty accurately distinguishing between the neural and Post-it classes in both approaches.

These results show that while there are differences in properties between the adaptive and regular matrices, both need to be further refined to improve classification accuracy, particularly to reduce prediction errors between classes with similar properties. A more in-depth analysis would require additional information about the axis configuration and graphic parameters, which are not fully visible in this figure.

3.5. Implication of the Findings

Error analysis revealed that 42% of misclassifications were caused by sarcasm and contextual ambiguity, which the LSTM model failed to capture effectively. Tweets that appeared positive on the surface often conveyed negative sentiment when expressed sarcastically, explaining why the positive class performed worst (precision 31%, recall 30%, F1 0.31). In contrast, the neutral class achieved slightly better performance (F1 0.41) but was frequently confused with positive or negative due to weakly expressed sentiments and ambiguous contexts. These findings indicate that while LSTM is effective in capturing sequential dependencies, it struggles with pragmatic meaning and non-literal language—a critical challenge in analyzing Indonesian political discourse on social media.

From a policy perspective, these results highlight the limitations of relying solely on traditional surveys, which often overlook informal and sarcastic expressions of public sentiment [19]. By applying sentiment analysis with culture-specific preprocessing, policymakers can gain more accurate and timely

insights into latent dissatisfaction, particularly in tweets that appear neutral on the surface [20]. Such capability enables governments to anticipate emerging issues earlier, make evidence-based decisions, and strengthen public trust through prompt responses to citizens' concerns.

From a research perspective, limitations identified in this study underline the need for methodological advancement. The relatively small dataset emphasizes the importance of expanding data size and diversity to mitigate overfitting. Recent studies demonstrate that hybrid models such as FastText-BiLSTM [21]. CNN–BiLSTM [22], and Bi-LSTM with Word2Vec embeddings [23] outperform standalone LSTM. Transformer-based models like IndoBERT [24] also show strong potential in political sentiment analysis, while multi-task learning frameworks combining sentiment and sarcasm detection [13] [25] are promising for handling nuanced and informal language. Future research should therefore explore hybrid and transformer-based architectures, sarcasm-aware multi-task learning, and data augmentation strategies to improve robustness and generalization in Indonesian sentiment analysis.

4. CONCLUSION

This study successfully used the LSTM model to analyze public opinion on the performance of West Java Governor Dedi Mulyadi based on tweets in Indonesian. Although the model demonstrated high performance in processing informal texts with a training accuracy of 95.88%, the significant difference in validation accuracy (36.32%) indicates significant overfitting issues. The overall accuracy of 85.01% masks important limitations in the model's ability to consistently classify sentiment across all categories, particularly in distinguishing between neutral and negative statements.

This study confirms the potential of LSTM for analyzing political sentiment in Indonesia, but also highlights the critical challenges posed by the informal nature of the Indonesian language, including slang, sarcasm, and non-standard grammar. Moderate precision and recall values (25–36% per class) indicate that the model struggles to capture subtle expressions of sentiment, particularly in positive classification. These results are consistent with existing literature on the challenges of Indonesian NLP, while also providing new insights into specific difficulties in analyzing political discourse.

Among the key contributions of this study are demonstrating the effectiveness of culture-specific preprocessing for Indonesian tweets and establishing benchmarks for future research in sentiment analysis in local languages. Nonetheless, the study faces several limitations: the dataset is relatively small (1,002 tweets), highly noisy, and imbalanced, which contributed to severe overfitting; and the reliance on a single deep learning architecture (standalone LSTM) reduced robustness, particularly in capturing pragmatic cues such as sarcasm and implicit sentiment. These constraints highlight the importance of expanding datasets, improving labeling quality, applying balancing techniques, and adopting advanced architectures such as hybrid CNN–BiLSTM, transformer-based models (e.g., IndoBERT), and multi-task learning frameworks that integrate sarcasm detection.

This study serves as an important foundation for the development of more robust sentiment analysis tools tailored to Indonesia's unique linguistic and political context, while also contributing to the broader field of resource-limited language NLP. Future research should therefore not only replicate this study with larger and more diverse datasets but also explore context-aware and sarcasm-sensitive approaches to improve generalization and reliability in real-world applications.

ACKNOWLEDGEMENTS

We thank Politeknik TEDC Bandung for their support, colleagues for valuable feedback, and all contributors who assisted in this research.

REFERENCES

[1] P. Sunarko, A. Bijaksana Putra Negara, R. Septiriana, and J. H. Hadari Nawawi, "Perbandingan Klasifikasi Algoritma Support vector machine dan Naïve Bayes Menggunakan Labeling VADER dan Lexicon based pada Tweets Bahasa Indonesia dan Bahasa Inggris," *JUARA, J. Apl. dan Ris. Inform.*, vol. 3, no. 1, pp. 9–19, 2024, doi: 10.26418/juara.v3i1.86468.

[2] B. M. A. AWAD, Z. NADIAH, and A. N. S. NASUTION, "Opini Publik Terhadap Penerapan New Normal Di Media Sosial Twitter," *Cover. J. Strateg. Commun.*, vol. 11, no. 1, pp. 19–26, 2020, doi: 10.35814/coverage.v11i1.1728.

[3] J. Y. Hui, "Rsis commentaries," *Most*, no. 67906982, pp. 4–6, 2009.

[4] Fransiscus and A. S. Girsang, "Sentiment Analysis of COVID-19 Public Activity Restriction (PPKM) Impact using BERT Method," *Int. J. Eng. Trends Technol.*, vol. 70, no. 12, pp. 281–288, 2022, doi: 10.14445/22315381/IJETT-V70I12P226.

[5] G. F. Nama, A. H. Shalihah, P. B. Wintoro, Y. Mulyani, and D. Despa, "Implementation of Naïve Bayes Classifier & Support Vector Machine Algorithm for Sentiment Classification using Twitter Data on Indonesian Presidential Candidates In 2024," *J. Inf. Syst. Eng. Manag.*, vol. 10, no. 20s, pp. 510–531, 2025, doi: 10.52783/jisem.v10i20s.3175.

[6] Ahmed Derbala Yacoub, Salwa O. Slim, and Amal Elsayed Aboutabl, "A Survey of Sentiment Analysis and Sarcasm Detection: Challenges, Techniques, and Trends," *Int. J. Electr. Comput. Eng. Syst.*, vol. 15, pp. 69–78,

2024.
[7] S. S. Almalki, "Sentiment Analysis and Emotion Detection Using Transformer Models in Multilingual Social Media Data," *Int. J. Adv. Comput. Sci. Appl.*, vol. 16, no. 3, pp. 324–333, 2025, doi: 10.14569/IJACSA.2025.0160332.
[8] M. O. Ibrohim and I. Budi, "Hate speech and abusive language detection in Indonesian social media: Progress and challenges," *Heliyon*, vol. 9, no. 8, p. e18647, 2023, doi: 10.1016/j.heliyon.2023.e18647.
[9] H. Tang, N. Zhang, X. Yu, T. Mao, and L. Wang, "Enhancing Sentiment Analysis with Word2Vec and LSTM: A Comparative Study," *J. Basic Appl. Res. Int.*, vol. 29, no. 3, pp. 1–10, 2023, doi: 10.56557/jobari/2023/v29i38342.
[10] M. Huang, Y. Cao, and C. Dong, "Modeling Rich Contexts for Sentiment Classification with LSTM," 2016, [Online]. Available: http://arxiv.org/abs/1605.01478
[11] H. Murfi, S. Theresia Gowandi, G. Ardaneswari, and S. Nurrohmah, "BERT-based combination of convolutional and recurrent neural network for indonesian sentiment analysis," *Appl. Soft Comput.*, vol. 151, pp. 1–15, 2024, doi: 10.1016/j.asoc.2023.111112.
[12] H. Jayadianti, W. Kaswidjanti, A. T. Utomo, S. Saifullah, F. A. Dwiyanto, and R. Drezewski, "Sentiment analysis of Indonesian reviews using fine-tuning IndoBERT and R-CNN," *Ilk. J. Ilm.*, vol. 14, no. 3, pp. 348–354, 2022, doi: 10.33096/ilkom.v14i3.1505.348-354.
[13] Y. Y. Tan, C. O. Chow, J. Kanesan, J. H. Chuah, and Y. L. Lim, "Sentiment Analysis and Sarcasm Detection using Deep Multi-Task Learning," *Wirel. Pers. Commun.*, vol. 129, no. 3, pp. 2213–2237, 2023, doi: 10.1007/s11277-023-10235-4.
[14] A. A. Mudding and Arifin A Abd Karim, "Analisis Sentimen Menggunakan Algoritma Lstm Pada Media Sosial," *J. Publ. Ilmu Komput. dan Multimed.*, vol. 1, no. 3, pp. 181–187, 2022, doi: 10.55606/jupikom.v1i3.517.
[15] T. Widyanto, I. Ristiana, and A. Wibowo, "Komparasi Naïve Bayes dan SVM Analisis Sentimen RUU Kesehatan di Twitter," *SINTECH (Science Inf. Technol. J.*, vol. 6, no. 3, pp. 147–161, 2023, doi: 10.31598/sintechjournal.v6i3.1433.
[16] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Naacl-Hlt 2019*, no. Mlm, pp. 4171–4186, 2018, [Online]. Available: https://aclanthology.org/N19-1423.pdf
[17] R. Prabowo, H. Sujaini, and T. Rismawan, "Analisis Sentimen Pengguna Twitter Terhadap Kasus COVID-19 di Indonesia Menggunakan Metode Regresi Logistik Multinomial," *J. Sist. dan Teknol. Inf.*, vol. 11, no. 2, p. 366, 2023, doi: 10.26418/justin.v11i2.57449.
[18] D. W. Wicaksono, B. Hartono, J. T. Lomba, and J. Semarang, "Analisis Sentimen Twitter Terhadap Kualitas Udara Jakarta Menggunakan Metode NBC," vol. 17, no. 1, pp. 103–110, 2024, [Online]. Available: http://journal.stekom.ac.id/index.php/elkom□page103
[19] A. M. K. Mohammed, G. G. M. N. Ali, and S. S. Khairunnesa, "GSAF: An ML-Based Sentiment Analytics Framework for Understanding Contemporary Public Sentiment and Trends on Key Societal Issues," *Inf.*, vol. 16, no. 4, 2025, doi: 10.3390/info16040271.
[20] E. Lunando and A. Purwarianti, "Indonesian social media sentiment analysis with sarcasm detection," *2013 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2013*, pp. 195–198, 2013, doi: 10.1109/ICACSIS.2013.6761575.
[21] A. Glenn, P. LaCasse, and B. Cox, "Emotion classification of Indonesian Tweets using Bidirectional LSTM," *Neural Comput. Appl.*, vol. 35, no. 13, pp. 9567–9578, 2023, doi: 10.1007/s00521-022-08186-1.
[22] V. R. Prasetyo, M. F. Naufal, and K. Wijaya, "Sentiment Analysis of ChatGPT on Indonesian Text using Hybrid CNN and Bi-LSTM," *J. RESTI*, vol. 9, no. 2, pp. 327–333, 2025, doi: 10.29207/resti.v9i2.6334.
[23] V. B. Lestari, E. Utami, and Hanafi, "Combining Bi-LSTM And Word2vec Embedding For Sentiment Analysis Models Of Application User Reviews," *Indones. J. Comput. Sci.*, vol. 13, no. 1, pp. 312–326, 2024, doi: 10.33022/ijcs.v13i1.3647.
[24] P. Sayarizki and H. Nurrahmi, "Implementation of IndoBERT for Sentiment Analysis of Indonesian Presidential Candidates," *J. Comput.*, vol. 9, no. 2, pp. 61–72, 2024, doi: 10.34818/indojc.2024.9.2.934.
[25] O. Vitman, Y. Kostiuk, G. Sidorov, and A. Gelbukh, "Sarcasm detection framework using context, emotion and sentiment features," *Expert Syst. Appl.*, vol. 234, no. 559, 2023, doi: 10.1016/j.eswa.2023.121068.