Advance Sustainable Science, Engineering and Technology (ASSET)
Vol. 8, No.3, July 2026, pp. 0260308-01 ~ 0260308-016
ISSN: 2715-4211 DOI: https://doi.org/10.26877/asset.v8i3.2862

Comparative Evaluation of Automatic Labeling and Modeling Strategies for Indonesian Sentiment Analysis: Methodology and Performance Evaluation

Khoiriya Latifah[1,5*], Agung Handayanto[1], Nur Latifah Dwi M.S[1], Rahul Bhandari[2,] Ton Nguyen Trong Hien[3,5], Doston Pirnazarov[4,5]

[1]Faculty of Engineering and Informatics, Universitas PGRI Semarang, Jl. Sidodadi Timur No 24, Semarang, Central Java 50232, Indonesia

[2]Department School of Business and International Office, Jindal Global University, Sonipat Narela Road, Near Jagdishpur Village, Sonipat, Haryana 131001, India.

[3]Faculty of Business Administration, Van Lang University, 69/68 Dang Thuy Tram, Binh Loi Trung 72329, Ho Chi Minh City, Vie.

[4]Narpay Faculty of General Sciences, Samarkand State Foreign Languages Institute, Kamolot street, Narpay district, 141200, Uzbekistan

[5]Naveen Jindal Young Global Research Fellowship, O.P Jindal Global University, Haryana, India

*khoiriyalatifah@upgris.ac.id

Abstract. Sentiment analysis is vital for understanding consumer perception, yet Indonesian sentiment classification faces challenges due to labeled data scarcity and computational constraints. This study advances automatic labeling techniques and establishes performance benchmarks for Indonesian text. The research compares two labeling approaches InSet Lexicon and IndoBERT based Hugging Face pipeline on 8,447 Tapera-related opinions. Results show InSet Lexicon produced a highly skewed distribution (89.66% neutral), while the IndoBERT pipeline achieved a more balanced distribution (47.66% neutral, 38.43% positive, 13.91% negative).. Evaluation of various modeling strategies revealed that combining InSet Lexicon + TF-IDF with Naïve Bayes or Random Forest achieved scores above 85%. While RNN-LSTM reached >90% accuracy, it required significant resources. Notably, fine-tuning IndoBERT with optimal hyperparameters yielded the most robust performance, achieving 80–90% accuracy with a low validation loss of 0.1. The study concludes that for small datasets (<12,000 samples), the most effective strategies for Indonesian sentiment analysis are either the InSet Lexicon paired with traditional Machine Learning or automatic labeling using pre-trained models followed by rigorous fine-tuning.

Keywords: Low resources nlp, sentiment analysis, automatic labeling, vectorization, postagging

(Received 2025-10-11, Revised 2026-04-01, Accepted 2026-04-02, Available Online by 2026-05-24)

1. Introduction

Data labeling is a crucial yet highly tedious task, as it requires significant time and expertise in the domain related to the data's context. Manual labeling has limitations because humans are prone to interpretive errors, fatigue, and boredom, which can lead to mistakes in labeling. Additionally, the unstructured or ambiguous nature of data can make it challenging to label accurately, further increasing the likelihood of errors. To address the shortcomings of manual labeling, automated methods using algorithms and AI models can be employed. Automated labeling with AI assistants has been shown to improve labeling performance, demonstrating that labels predicted by models, even with minimal training, can significantly enhance both the accuracy and speed of labeling tasks [1]. AI-assisted labeling is effective because its results closely approximate those of human labeling [2,3], while also minimizing costs and time compared to human labeling [4–6]. Sentiment analysis, a crucial aspect of Natural Language Processing (NLP), detects emotions from text. In general, sentiment analysis utilizes two primary approaches: lexicon-based methods and machine learning-based methods. Lexicon-based techniques are further divided into Dictionary-based and Corpus-based approaches [7]. The machine learning method uses manually annotated datasets as training data to automatically classify text, while the lexicon-based method relies on a pre-established opinion dictionary (lexicon) for text categorization [8]. Research by Setiawan (2024) illustrates a practical consideration in lexicon-based sentiment analysis [9]. His study on public sentiment regarding a suicide bombing revealed that while the SVM algorithm with the VADER lexicon yielded a 94% accuracy marginally higher than the 93% achieved with the InSet lexicon the choice of tool depends on the language of the data. For Indonesian tweets, using InSet is more efficient as it avoids the ineffectiveness of translating data into English, a step necessary for VADER. This demonstrates the enduring utility of specialized lexicons in natural language processing, a tradition continued by researchers like Musfiroh et al. (2021) who also utilized the InSet lexicon on a Twitter dataset related to online lectures, achieving strong performance with an accuracy of 79.2%[10]. The Bidirectional Encoder Representations from Transformers (BERT) approach has gained widespread adoption in sentiment analysis to uncover human emotions from text. Sentiment analysis based on deep learning, particularly with BERT, holds immense potential for extracting valuable insights from textual data, benefiting a wide range of applications [11]. Compared to traditional models like SVM, Naive Bayes, and LSTM, BERT-based models demonstrate superior performance [12]. A study titled Sentiment Analysis of Customer Reviews on the Ruang Guru Application Using the BERT Method highlighted the high effectiveness of pre-trained BERT models for sentiment analysis [13]. Additionally, Ardiansyah et al. (2024) explored BERT's application in analyzing sentiments toward government policies in their study, Sentiment Analysis of Electronic System Operator (PSE) Policies Using the BERT Algorithm, achieving accuracy rates of 69%, 55%, and 55% [14]. Research involving BERT models has been advancing steadily in recent years. For example, Tabinda Kokab et al. (2022) performed a sentiment analysis on social media data, demonstrating that the CBRNN model, which is based on BERT, surpassed the performance of other models [15]. The next year, Chandradev et al. (2023) performed a similar study on review analysis using a simplified BERT model, achieving an impressive accuracy of 91.40% [16]. In other research, the IndoBERT model excelled in processing Indonesian text, with accuracies of 85% on validation training and 86% on testing training [17]. Developed as a deeply bidirectional pre-trained model, BERT utilizes unlabeled text data by integrating

both left and right contextual layers, allowing it to be fine-tuned for machine learning tasks with just one additional layer. Its swift advancement has enabled its application in sentiment analysis for the Indonesian language, yielding highly accurate results [18]. Fine tuning can reduce bias [19,20]. Numerous automated labeling methods can be utilized.

Based on the aforementioned references, which methodology should be employed to achieve an optimal integration of models for sentiment analysis in Indonesian text? Furthermore which automatic labeling technique yields superior downstream classification performance? The objective of this research is to advance the state of the art in automatic labeling and establish effective model benchmarks. The resultant outputs form this sentiment analysis may be utilized by governmental entities, trough departments affiliated with Tapera to inform decision making process aligned with the dataset's contextual framework [21].

This study will use an Indonesian-language dataset consisting of text data related to Tapera. The research gap is when we used InSet Lexicon and pipeline automatic labeling and how to combine the appropriate automatic labeling methods for Indonesian text data with the right vectorization methods, as well as the proper selection of modeling techniques for sentiment analysis of Indonesian text data (case study using Tapera dataset). There are many challenges in generating text feature representations in the Indonesian language, such as ambiguous words, slang, and irrelevant words. Automatic labeling produces labeled datasets, which are then vectorized for training models like Naive Bayes, Random Forest, LSTM, or BERT (through transfer learning and fine-tuning). Therefore, the selection of effective text features is an important aspect to consider because the labeled text data needs to be converted into a numerical form to be input into machine learning (ML) or deep learning (DL) models. Automatic labeling accelerates data preparation, while vectorization ensures compatibility with modeling algorithms. Thus, the choice of appropriate automatic labeling and vectorization for machine learning or deep learning modeling will influence the model's performance results in classifying text or determining sentiment outcomes.

1.1. Related Research

Research on sentiment analysis of the government policy regarding TAPERA has been extensively conducted. Among them: Research conducted by Sihombing et al. (2024) used data from the social media platform Twitter, showing that the Naïve Bayes algorithm achieved the highest accuracy in sentiment analysis at 69.17%, followed by Support Vector Machine with an accuracy of 68.42%, and Random Forest with an accuracy of 66.17% [22]. Labeling was done using the VADER Lexicon, which has not been adapted to the Indonesian language, and text feature vectorization used TF-IDF. Positive sentiment accounted for 37.1%, negative for 35.4%, and neutral for 27.4%. Naïve Bayes is the most effective algorithm for sentiment analysis of the Tapera policy, especially in the context of complex text data from social media. Research conducted by Firdaus et al. (2025) resulted in a sentiment analysis of public opinion towards TAPERA using the IndoBERT Lite Large model [23]. From 14,618 YouTube comments collected, 13,766 comments were processed after the preprocessing stage. Labeling used IndoBERT Transfer Learning and text feature vectorization used TF-IDF. The sentiment labeling results showed a dominance of negative sentiment with 9,571 comments, reflecting public concern regarding the program's transparency, implementation, and communication. Positive sentiment reached 2,485 comments, indicating limited appreciation for the program, while 1,710 neutral comments suggested a need for clearer information. Based on evaluation using a confusion matrix, this model achieved an accuracy of 78%. Research conducted by Syahputra et al. (2024) used 16,641 data points from web scraping and the YouTube API [24]. Labeling used the polarity automatic labeling method, text vectorization used TF-IDF, and modeling used SVM. The model performance showed excellent accuracy of 99.81%, meaning the SVM model is effective for conducting sentiment analysis. The sentiment results showed that public opinion towards Tapera was 99.82% negative and only 2% had a positive opinion [25]. Research conducted by Muhammadi et al. (2022) used comment data from the YouTube API and TextBlob, with results showing that 76% of sentiments about Tapera were negative, 23% were positive, and 1% were neutral [26]. Research conducted by Muhandhis et al. (2025) used

Twitter data with a combination of automatic labeling using a Lexicon-based approach, SVM modeling, and TF-IDF text feature extraction. The model demonstrated good performance with an accuracy of 81.7% [27]. Research conducted by Isnaeni Muhandis used comments from TikTok videos. Labeling was done automatically using the pre-trained Hugging Face model 'ayameRushia/bert-base-indonesian-1.5G-sentiment-analysis-smsa' (IndoBERT), text data vectorization used TF-IDF, and modeling used Random Forest. The model performed well for sentiment analysis, yielding an accuracy of 89%, with sentiment analysis results of 82% negative, 10% positive, and 8% neutral.
This research will discuss the right combination when we use automatic labeling with the selection of appropriate text feature extraction and modeling that has good performance for conducting sentiment analysis on Indonesian text data. How to makes good combination of modeling and automatic labelling a sentiment analysis in Indonesian text.

2. Research Methods

Text classification is a fundamental Machine Learning problem with applications across various domains. Sentiment analysis is a common type of text classification, with the goal of identifying the polarity of a text or the type of opinion it expresses. Applications of sentiment analysis in this research include analyzing social media posts to determine public reception of the TAPERA policy or extrapolating general public opinion towards it. One established method for solving text classification problems is machine learning, which follows a high-level, end-to-end workflow. We will identify the optimal combination using automatic labeling with InSet Lexicon or Hugging Face pipeline, paired with the appropriate vectorization technique and modeling approach spanning traditional methods, deep learning, and fine tuning in this research.

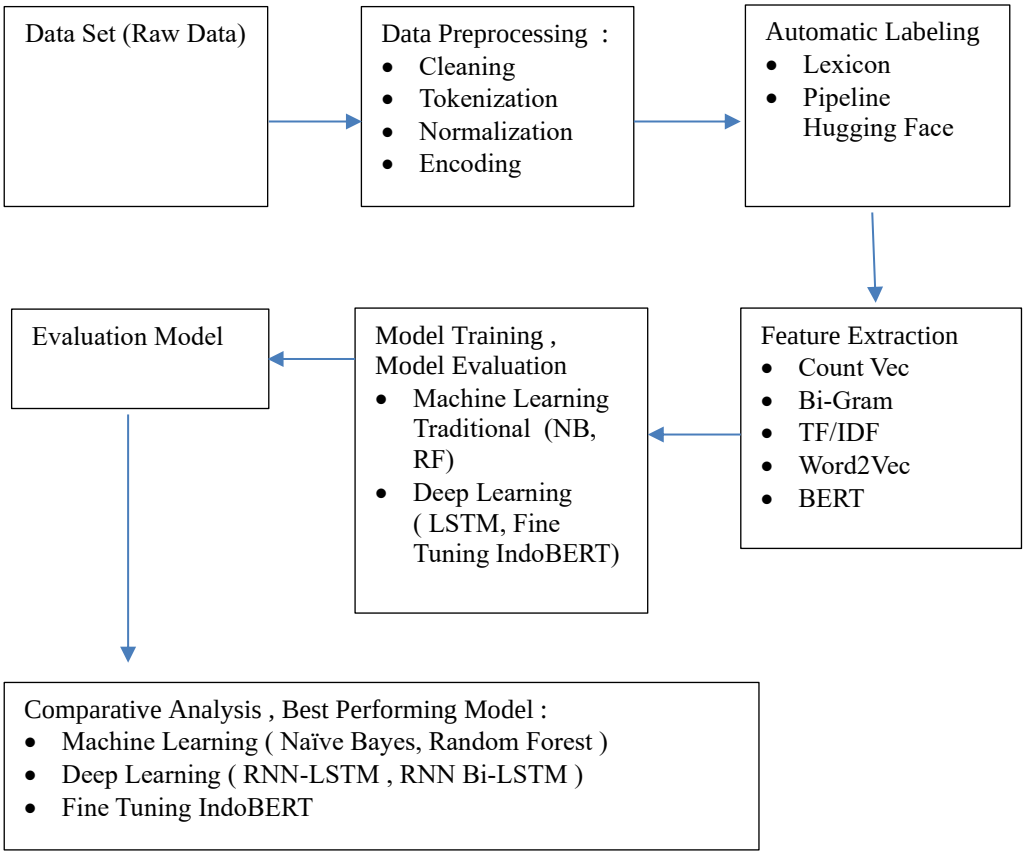

Figure 1. Research Metodology

2.1. Lyfe Cycle Natural Language Processing

The methodology employed in this research will utilize the NLP (Natural Language Processing) Life Cycle.

2.1.1. Data Collection and Description.

Data gathering is the most important step in solving any supervised machine learning problem, as a text classifier can only be as good as the dataset it is built from. Therefore, collecting the necessary data is crucial. In this study, the dataset consists of Indonesian public opinions regarding a government policy known as *Tabungan Perumahan Rakyat* (Tapera). The data comprises a total of 8,587 records, acquired through crawling the social media platform Twitter and supplemented with a dataset from Kaggle. (https://www.kaggle.com/datasets/unshoytable/twitter-tapera-dateset). This dataset consists of unstructured text data in the Indonesian language.

2.1.2. PreProcessing Data

Raw data crawled from Twitter (now X) is typically unstructured and noisy, containing elements such as user mentions, URLs, hashtags, emojis, slang, abbreviations, irregular capitalization, and duplicate entries. Text preprocessing is a critical step as it prepares the data for feature extraction. The results of the preprocessing will significantly impact the modeling outcome and its ability to achieve high performance. Text preprocessing is a series of steps performed on text data before analysis can be conducted. There are many challenges in processing unstructured text to produce clean text features, making it easier for algorithms to understand and perform tasks like text classification or sentiment analysis. Some challenges specific to preprocessing Indonesian text include the use of slang on social media, ambiguous words, numerous abbreviations, incorrect use of emoticons or symbols, and other irrelevant words. Proper text preprocessing significantly impacts the analysis results because human language contains context and meaning that can change even with slight variations in presentation. To prepare the data for subsequent analysis and modeling, a systematic preprocessing pipeline was applied. The pipeline consists of six sequential stages: case folding, data cleaning, normalization, tokenization, stopword removal, and stemming. This order ensures optimal effectiveness, as performing stemming or stopword removal before normalization often reduces accuracy in informal Indonesian text.
The preprocessing stages in this research are:

2.1.2.1. Remove Duplicate and Case Folding

The first preprocessing step is to remove duplicate data. And case folding The initial dataset of 8,587 records was reduced to 8,447 records after this step. Case folding converts all characters in the text to lowercase. This step eliminates redundancy arising from case variations (e.g., “Baik”, “BAIK”, and “baik” are treated as identical tokens). By standardizing capitalization, case folding reduces the overall vocabulary size, decreases computational complexity, and improves model efficiency. It serves as the foundational step in most text preprocessing pipelines to ensure consistency across the dataset

2.1.2.2. Data Cleaning

Data cleaning removes irrelevant elements and noise from the raw tweets using regular expressions. This process includes the removal of retweet indicators (RT patterns), user mentions (@username), URLs (http/https), emojis, emoticons, irrelevant special characters like currency symbols, hyphens, brackets, or other characters that can interfere with processing were removed., and numbers (unless contextually relevant)., hashtags (#hashtag) were optionally retained when they carried potential feature value. Extra whitespace, tabs, newlines, and duplicate tweets were also eliminated.
This stage eliminates non-semantic elements that do not contribute to the textual meaning, significantly reduces data dimensionality, prevents models from learning irrelevant patterns, and enhances the quality of subsequent preprocessing steps.

a. Removing Stopwords

Stopword removal eliminates common words that carry little semantic value, such as “yang”, “di”, “dan”, “itu”, and “dari”. Indonesian stopword lists from the Natural Language Toolkit (NLTK) or custom high-quality lists were utilized. This step dramatically reduces dataset dimensionality—often by 30–40%—and focuses the model on content-bearing words. Consequently, it enhances the performance of classification, clustering, and other downstream tasks.This involves eliminating common words that do not provide significant important information for text analysis or processing. These words frequently appear in text and tend to lack specific, relevant meaning in a given context. Removing stopwords is important as it can speed up text processing, reduce data dimensionality, and focus the analysis on more informative or significant words.

Implementation in this research **:**

Stopword removal was performed by creating custom stopwords using the postager from the nlp_id library to obtain highly meaningful stopwords, as some standard stopwords may be meaningful for our specific use case. Referring to nlp_id's documentation and considering our research context on the Tapera policy in Indonesia, the custom stopwords were defined as words with the following tags: NEG (negation), JJ (adjective), VB (verb), FW (foreign word), and NUM (numbers). Before implementing the custom stopwords with postager, the dataset was reviewed to ensure alignment with the research context.

b. *Cleaning Slang Words***:**

Slang words were cleaned using a dictionary, which can be found here: https://github.com/nasalsabila/kamus-alay/blob/master/colloquial-indonesian-lexicon.csv. replaces informal slang, abbreviations, typos, and colloquial Indonesian expressions with their formal or standard equivalents (e.g., “gw” → “saya”, “bgt” → “banget”, “lg” → “lagi”). A custom slang dictionary or publicly available Indonesian slang lexicon was employed for this purpose.Because Twitter data is highly informal and rich in slang, normalization is essential for effective stemming and stopword removal. This step bridges the gap between social media language and standard Indonesian vocabulary, thereby improving model accuracy by approximately 5–15% in many Indonesian NLP studies

c. *Removing Punctuation:*

Punctuation marks such as periods, commas, question marks, and exclamation points often do not contribute significantly to text understanding in many text analysis cases and were removed.

2.1.2.3. Normalization with Stemming and Lemmatization

Normalization is used to simplify the text, remove irrelevant variations, and ensure consistency in word representation.

a. Stemming:

Stemming reduces words to their root (stem) form by removing affixes (e.g., “bermain” → “main”, “memainkan” → “main”, “pemain” → “main”). For Indonesian text, the **Sastrawi (PySastrawi)** library was employed, as it is the most commonly used and effective stemmer for Bahasa Indonesia. Sastrawi applies a rule-based conflation approach based on the Nazief-Adriani algorithm and enhanced confix-stripping techniques, with repeated dictionary lookups to minimize over-stemming and under-stemming errors.This step reduces vocabulary size by mapping morphological variants to the same root word, lowers computational cost, and mitigates data sparsity issues. It is particularly important for Bahasa Indonesia, which features rich affixation patterns.

Stemming reduces words to their root or base form (stem) by removing affixes. It aims to treat different variations of similar words as the same entity in text analysis, which helps reduce dimensionality and can improve accuracy [28].

b. Lemmatization:

Lemmatization uses morphological information, dictionaries, or linguistic rules to reduce words to their canonical form or lemma (the dictionary form). It considers the context and part of speech to produce a valid base word, offering a more linguistically comprehensive approach than stemming.

2.1.2.4. Tokenization

Tokenization splits the cleaned text into individual word tokens. This process transforms complete sentences into discrete, processable units (tokens), enabling the precise application of subsequent techniques such as stopword removal, stemming, and feature extraction (e.g., TF-IDF or word embeddings).

Tokenization aims to break down the text into separate, individual units (tokens), such as words or terms, so it can be further processed for natural language processing and text analysis.

The initial tokenization is performed using word tokenization, which is the process of dividing text into smaller units based on words. This is done using the Tokenizer from the NLTK library in the Python programming language.

As we know, the process of text recognition in NLP involves converting a collection of text into a numerical representation so that it can be understood by a computer. Therefore, numerical representation is required to extract features, reduce dimensionality, and ensure model compatibility. Consequently, after text preprocessing, the next step is to analyze the tokenization results, which will then be used for text representation.

This research employs the Word Tokenization and Bi-Gram Tokenization methods, which will subsequently be analyzed for their preprocessing and tokenization outcomes. Since tokenization uses both Word Tokenization and Bi-Gram Tokenization, the next step is feature extraction into numerical vector form based on Count Vector (the frequency of word occurrences in the text) and Bi-Grams. The following is a visualization of the Word Tokenization and Count Vector vectorization: :

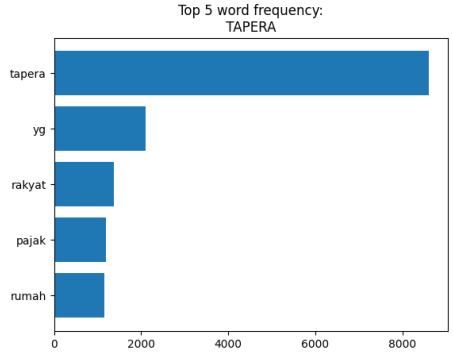

Figure 2. Top 5 Word Frequency

After preprocessing was completed, it can be observed that the most frequently appearing word is "Tapera" with 8,200 occurrences, followed by "yg" (an abbreviation for 'yang') with 2,200, "Rakyat" (people) with 1,300, and "Pajak" (tax) and "Rumah" (house) each with 1,200 occurrences. The conjunction "Yang" (Which/That) is still present.

The high frequency of the word "Tapera" (8,200 times in a dataset of 8,447 records) indicates that it is highly relevant to the data's topic. The subsequent words, "Rumah" (house), "Rakyat" (people), and "Pajak" (tax), suggest that Tapera is a government program intended for the people, related to housing and tax.

The presence of the word "Yang," which was not removed, remains useful as it can help detect and link words that may form positive or negative contextual phrases. For text feature extraction, Bi-Grams are used to detect phrases and negations, which enables the prediction of the next word within a sentence's context. The results can be observed in the graph below:

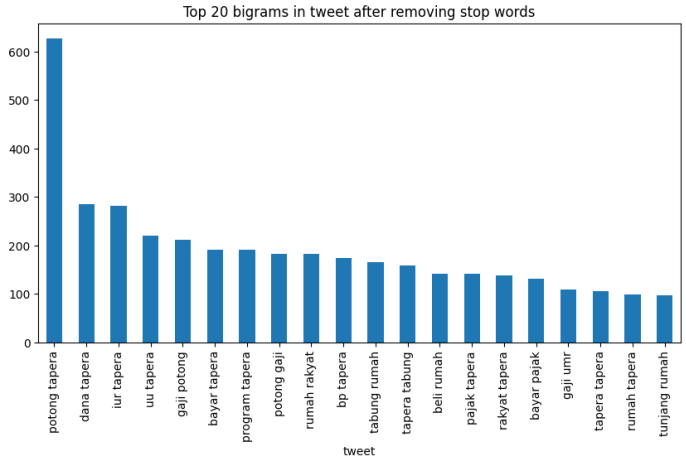

Figure 3. 20 Bi-Gram

From the text feature extraction using the Bi-Gram method, we can observe that all of the top 20 most frequent Bi-Grams contain the word "Tapera" and are associated with positive terms. This indicates that Tapera is strongly perceived in relation to how people can acquire housing through various methods, including salary deductions, fund deductions, contributions, and savings payments, all aimed at enabling the public to obtain decent housing.

This demonstrates that the preprocessing stage was successful and that the frequency-based text vectorization method effectively extracted the relevant features.

After the completion of the preprocessing stage and the analysis of its results using frequency-based text vectorization, the next step is to perform labeling. In this research, labeling will be conducted automatically, and the results will be compared to build a sentiment analysis model for Tapera.

2.1.3. Dataset Labeling

After obtaining high-quality data through several text data preprocessing stages, the next step in preprocessing involves labeling the data using automatic labeling, analyzing text data features, and selecting the appropriate modeling approach.After the data is cleaned, the next step is dataset labeling. Labeling the dataset is a critical step, as incorrect labels can directly impact the model's outcomes. There are two primary labeling methods: using manual annotators and automatic labeling.

Using human annotators can introduce subjectivity into the labeling process. Furthermore, the results can be influenced by human factors such as fatigue or poor psychological condition, potentially affecting label consistency and accuracy.

This research will employ automatic labeling methods using:

- A lexicon for the Indonesian language, specifically the InSet Lexicon.
- A Hugging Face pipeline using the IndoBERT model.

Previous studies provide numerous references on automatic labeling. However, none have explicitly discussed the criteria for choosing between a Lexicon-based automatic labeling approach and a Hugging Face pipeline method.

As we know, processing unstructured text data is highly dependent on its vectorization how the numerical representation preserves semantic information. Therefore, this research will investigate whether a relationship exists between the choice of vectorization method (frequency-based vs. prediction-based) when automatic labeling is performed using a Lexicon versus a Hugging Face pipeline. This analysis will be followed by modeling with both Machine Learning and Deep Learning approaches, specifically using a Fine-Tuned IndoBERT model.

2.1.3.1. Lexicon Labeling

In this research, the InSet Lexicon is used for labeling. This is a lexicon specifically adapted for the Indonesian language and is available as a Python library. It is a sentiment analysis tool that produces four main metrics: positive (pos) = the proportion of positive words, negative (neg):= the proportion of negative words, neutral (neu) = the proportion of neutral words and compound = a normalized composite score ranging from -1 to +1. This is the result of automatic labwling using InSet Lexicon

```
neutral     7574 (89,66%)
positive     600 ( 7,10%)
negative     273 ( 3,23%)
```

The compound score is the most important metric for determining the overall sentiment, categorized as follows: ≥ 0.05: POSITIVE Sentiment (Good), > -0.05 and < 0.05**:** NEUTRAL Sentiment, ≤ -0.05: NEGATIVE Sentiment (Bad). The reliability of this compound score measurement was assessed using a Kappa Score, which yielded a value of 0.7. This indicates that the compound score is reasonably reliable for performing sentiment analysis on the dataset used in this study.

2.1.3.2. Hugging Face Pipeline Labeling

The second labeling method is automatic labeling using a Hugging Face pipeline with the pre-trained BERT model indolem/indobertweet-base-uncased. The result of automatic labeling as following below :

```
neutral     4026 (47,66%)
positive    3246 (38,43%)
negative    1175 (13,91%)
```

The evaluation results using the Kappa score matrix yielded a value of 0.76, indicating that the labeling is of high quality.

2.1.4. Data Modeling

Data modeling is the stage of selecting the Machine Learning and Deep Learning models to be used in this research. The purpose of modeling is to analyze the text data using the appropriate algorithms to accurately determine sentiment

2.1.5. Model Evaluation

Evaluation is the stage for assessing the model's performance. It determines whether the model correctly classifies the text data into the appropriate sentiment categories. Therefore, the model must be measured using performance metrics. In this research, the F1-Score is used as the primary measurement.

3. Result and Discussion

In this research the approach demonstrated the importance of sentiment analysis or text classification in detecting opinion, as understanding the context of text enables more effective intervention strategies to analyse of the sentiment or opinon positive, negative or neutral. We comparison of three combination methods.

The first method utilizes InSet Lexicon for labeling and employs Machine Learning for modeling, with text features based on Frequency-Based Text Vectorization.

The modeling results obtained the values shown in the following table 1.:

Table 1. Scenario 1 Combination of Lexicon, TF/IDF Vectorization, Naïve Bayes, Random Forest Modeling

Labeling	Vectorization	Method	Tested Parameter	Tested Value	Evaluation Metrics
InSet Lexicon	TF/IDF	Naïve Bayes	-	-	F-1 Score :0,89; Precsion : 0,89 Recall : 0,90;

				Accuracy L 0,89
	Random Forest	-	-	F-1 Score:0,95; Precision : 0,94; Recall: 0,92; Accuracy : 0,95
	RNN LSTM	Epoch	6	Val Acc 0,95; Val Loss: 0,1; Test Acc :0,89; Test Loss: 0,30
		Batch Size	32	
		Embedding	300	
		LSTM	64	
		Dense	256	
		Dropout	0,3	
		Activation	Relu	
		Optimazer	Adam	
		LR	0,001	

Following tokenization and the analysis of text features based on Frequency-Based Text Vectorization results using Count Vectorization and Bi-Gram Vectorization, the next step is to perform modeling with the dataset that has been automatically labeled using the InSet Lexicon. This is combined with TF-IDF-based vectorization and traditional Machine Learning modeling using Naïve Bayes and Random Forest.

All combinations of InSet Lexicon labeling, TF-IDF, and Machine Learning models yielded excellent evaluation results, with F1-Score, Precision, Recall, and Accuracy > 0.8. This demonstrates that this specific combination can effectively enhance the accuracy of Indonesian text sentiment analysis. The combination successfully meets the text recognition requirements of the algorithms, which is supported by the dataset of 8,447 records and a token count featuring the word "Tapera" 8,200 times.

Applying TF-IDF after CountVectorizer weights the counts by reducing the influence of frequent but less informative words across the corpus. It emphasizes rare, discriminative terms, making the representation more meaningful. In practice, this is equivalent to using TF IDF Vectorizer, which combines both steps. This pipeline helps by downweighting common noise and highlighting sentiment-bearing terms. However, it produces fixed-length vectors per document, losing sequential information, which is a limitation when feeding into sequence models like LSTM

When modeling the results from InSet Lexicon labeling using RNN-LSTM model that utilizes an embedding layer to understand word context, the model can effectively discover pattern recognition. This allows it to capture complex patterns from lexicon features and non-linear relationships to understand interactions between words, as well as improve generalization, meaning it performs better at applying learned knowledge to new, unseen data. The LSTM architecture is specifically designed to address the limitations of a standard RNN, which struggles to predict words based on information from the distant past. LSTM networks are capable of remembering information stored over long sequences while also discarding information that is no longer relevant. This makes LSTMs more efficient at processing, predicting, and classifying data based on specific time sequences.

This combination can enhance model performance by providing better feature representations. TF-IDF reduces dimensionality and noise, leading to faster training and higher accuracy in sentiment prediction. For lexicon-labeled data where labels are derived from dictionaries like InSet, this helps mitigate label noise, as lexicons may mislabel ambiguous contexts. CountVectorizer (a precursor to TF-IDF) with ensemble models handled imbalanced lexicon-like labels effectively via oversampling, achieving high precision. Combination of TF-IDF + LSTM can learn to refine these through sequence modelling, LSTM adds robustness to sequence-dependent sentiments missed by lexicons. This approach positively influences sentiment analysis by improving feature quality and model robustness, especially for lexicon-labeled Indonesian data, leading to higher accuracy and better handling of linguistic nuances.

Lexicon-based labeling combined with traditional Machine Learning provides solid results (>85%), while with Deep Learning (RNN LSTM) it can achieve >90%, albeit with a trade-off of higher complexity and resource requirements. Accuracy 5-10% improvement over traditional Machine Learning, better handling for complex or ambiguous cases, **s**peed **s**lower in training, moderate in inference.

The second method involves labeling using a Hugging Face pipeline combined with Machine Learning modeling and text feature extraction. IndoBERT is a transformer model that has been pre-trained specifically for the Indonesian language. BERT possesses the capability for Contextual Understanding: IndoBERT comprehends sentence context deeply using an attention mechanism. It benefits from Pre-trained Knowledge: The model has already acquired linguistic understanding of Indonesian from a large training dataset. It produces a sentiment label (positive/negative/neutral) along with a confidence score. This labeling method was measured using a Kappa Score, yielding a value of 0.79, which indicates that it produces high-quality sentiment analysis results. The modeling results using TF/IDF vectorization, Word2Vec, and embedding layers are as follows:

Table 2. Combination of Pipeline Hugging Face IndoBert Labeling, TF IDF, Word2Vect Vectorization , Random Forest, LSTM

Labeling	Vect	Method	Tested Parameter	Tested Value	Result
Pipeline IndoBERT Hugging Face indolem/indobertweet-base-uncased	TF/IDF	Random Forest	Optimazer N estimator = 100		Acc: 0,68 (the model misclassifies positive class the most)
			n-iteration Cross vold Random state Split,leaf, max_dept	300 5 28 10,5,20	
	TF/IDF	LSTM	Optimazer Random Seach Max words Biderectional LSTM Dense, activation Optimazer epoch	 5000 32, 64, step=16 3, Softmax Adam 10	Val Acc: 0.60
	Word2vec	Random Forest	Min_count window Vector_size method Hyperparameter n-estimator, cv Spilt, leaf	4 7 100 Skip gram 100, 5 3 , 4	Acc: 0,65,

	LSTM	Vector-size	100, 1	Val acc: 06, test acc: 0,6 ; val loss: 0,5; test loss: 0,45
		Vocab-size	10000	
		Embedding dim	32	
		Max len	100	
		Lstm	16, 8	
		Learning rate	1e-3	
		Epoch	3	
		Optimazer	AdamW	
		Learning Rate	2e-5	

We can observe that all model evaluation metrics only range from 0.6 to 0.7. This indicates that modeling using either Machine Learning or Deep Learning is less suitable for automatic labeling with the Hugging Face pipeline. The modeling utilized a TF-IDF Random Forest model with a Random Search optimization method. This involved a very thorough hyperparameter optimization configuration, testing 300 random parameter combinations, each evaluated with 5-fold cross-validation and using all available CPU cores for maximum efficiency. The configuration used random feature selection with a subset of 10 features per split and a minimum of 5 samples per leaf. When using Word2Vec with the aforementioned configuration, a vector size of 200 strikes a balance between expressiveness and efficiency, while a window size of 7 captures sufficient context. Additionally, a minimum count of 4 effectively filters out rare words. For the Random Forest model, n_estimators = 100 represents an optimal estimate given the approximately 8,000 data points, and max_depth = 15 helps prevent overfitting, as the Random Forest algorithm is inherently robust against this issue.

Meanwhile, a BiLSTM (Bidirectional Long Short-Term Memory) network was used to obtain better features for understanding context, as its layers process sequences in both forward and backward directions. We can see that all evaluations of the model measurement matrix only range from 0.6 to 0.7.

This indicates that modeling using either Machine Learning or Deep Learning is less suitable for automatic labeling using the Hugging Face pipeline when combined with frequency-based text vectorization or Sentence Prediction-based methods for conducting sentiment analysis. Pre-trained on general Indonesian corpora but may struggle with domain-specific data like Tapera policy tweets if not fine-tuned. Policy-related texts often include technical terms (e.g., "tabungan perumahan") or nuanced sentiments (e.g., sarcasm), leading to noisy pseudo-labels. But if the Hugging Face pipeline uses a default sentiment model without fine-tuning (e.g., on a Tapera-specific subset), it may misclassify ambiguous phrases (e.g., "Tapera membantu tapi ribet"), reducing label quality. Loss of Sequential Information: TF-IDF creates sparse, high-dimensional vectors (e.g., 500-5000 features) that ignore word order, which is critical for Indonesian sentiment (e.g., "tidak bagus" vs. "bagus"). This can limit LSTM's effectiveness if TF-IDF vectors are fed directly without sequence-aware preprocessing, leading to poor feature quality and accuracy Random Forest with TF-IDF's sparse features but may not leverage IndoBERT's contextual labels fully without additional embeddings, capping performance. LSTM Configuration: LSTM (or Bi-LSTM) requires sequence inputs (e.g., word embeddings like IndoBERT or Word2Vec) rather than TF-IDF vectors alone. If TF-IDF is fed directly, LSTM loses sequential context, reducing accuracy to 50-65%. Unoptimized LSTM (e.g., no dropout, small hidden layers) further lowers performance. Tapera policy texts may include technical jargon or mixed sentiments, confusing IndoBERT's labels and TF-IDF's features.. Feature Dimensionality: An oversized TF-IDF vocabulary (>10,000 features) causes RF overfitting and LSTM input issues, reducing accuracy to 60%. Conservative settings (max_depth=10, high split/leaf) may underfit complex datasets, capping accuracy at 60%. Random Forest handles TF-IDF well but may not fully leverage IndoBERT's contextual labels without embeddings, limiting performance. LSTM Requires sequence inputs (e.g.,

IndoBERT/Word2Vec embeddings). Feeding TF-IDF directly loses context, dropping accuracy to ~50-65%. Unoptimized LSTM (e.g., no dropout) exacerbates this.

The third method is to use the Hugging Face pipeline for labeling and fine-tuning to perform sentiment analysis. The training data results can be seen in the following table.

Table 3. Comparation of Pipeline Hugging Face Labeling, BERT Vectorization, Fine Tuning IndoBERT

Labeling	Vect	Method	Tested Parameter	Tested Value	Result
Fine Tuning IndoBert	BertForSequenceClassification	indolem/indobertweet-base-uncased	Hyperparameter tuning		Acc: 0.70; Precison=0,7; Recall=0,7; eval_loss': 0.4, 'eval_steps_per_second': 0.141,Training loss 0,5
			Padding sequence	128	
			learning_rate	1e-5	
			train_batch_size	16	
			eval_batch_size	16	
			train_epochs	4	
			weight_decay	0,01	
			logging_steps	10	
	BERT Text Classifier	cahya/bert-base-indonesian-522M	Hyperparameter Tuning		Val Acc : 0,80; Train Acc : 0,89; Test Acc : 93; Train Loss : 01; Val Loss : 0,3
			Encoding Tokenizer	32, SEP, CLS, Softmax	
			Sequence Length	512	
			Max Sequence Length	150	
			Batch Size	16	
			Epoch	3	
			Optimazer	AdamW	
			Learning Rate	2e-5	

Deep Learning methods are data hungry , need more than 50.000 data items for training.. The distributions of the source and target data must be the same. Labeled data in the target domain may be limited. This problem is typically addressed with transfer learning. Reuse and adapt already learning model. This research using fine tuning parameter on the IndoBERT model. BERT is based on the Seq2Seq & Transformer model. When an input is processed by the Encoder, it converts an arbitrary-length input into a fixed-length hidden representation. The output is generated by the Decoder, which produces an arbitrary-length output. The purpose is to understand the context of words based on their position in a sentence, find the focus of the sentence, and determine the relationship between words in

a sentence. This process is applied between every word in the sentence, which is why it is also called multi-headed self-attention.

As can be seen in the table, labeling using the Hugging Face pipeline and modeling with BERT, along with fine-tuning hyperparameters, will produce a high-quality model for sentiment analysis [29]. This is evidenced by its high accuracy and low $_{validation}$ loss and test loss. Padding sequence standardizes the length of all text inputs to the fixed length required by IndoBERT by adding a special [PAD] token in empty positions. Learning rate is a hyperparameter that controls the size of the steps the model takes when updating parameters (weights) during training. It determines the model's learning speed. Weight decay is a regularization technique that adds a penalty based on the magnitude of the model parameters to prevent overfitting. Batch size refers to the number of input data samples processed in one training iteration, while an epoch refers to the number of complete passes through the entire training dataset. Logging steps is a parameter that determines how frequently the model records (logs) training information, such as loss, learning rate, and other metrics, during the IndoBERT training process.

4. Conclusion

This study demonstrates that for sentiment analysis on small to medium Indonesian datasets (< 12.000 samples), the choice of automatic labeling technique, vectorization method, and modeling approach significantly influences overall performance. The most effective combinations are: (1) InSet Lexicon combined with TF-IDF vectorization and traditional machine learning models (Naive Bayes or Random Forest), which consistently deliver robust results exceeding 85% across key evaluation metrics while remaining computationally efficient; and (2) automatic labeling via the Hugging Face IndoBERT pipeline followed by fine-tuning the IndoBERT model, which achieves strong performance (80–90% Accuracy, Precision, Recall, and F1-Score) with notably low validation loss. In contrast, combining Hugging Face pipeline labeling with traditional vectorization techniques (TF-IDF or Word2Vec) and conventional ML/DL models yields suboptimal results. For researchers working with limited Indonesian text data, we recommend adopting InSet Lexicon with TF-IDF and classical machine learning models when computational resources and time are constrained. When higher semantic understanding is required and GPU resources are available, fine-tuning a pre-trained IndoBERT model after initial labeling with the Hugging Face pipeline offers superior performance. A hybrid approach—using lexicon-based labeling and traditional models as a baseline before advancing to transformer-based fine-tuning provides a comprehensive and methodologically sound framework for Indonesian sentiment analysis.

Declaration of AI and AI assisted technologies in the writing process

During the preparation of this work the author(s) used Google Collab and Jupyter Notebook in order to make codes and train models. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We express our sincere gratitude to the LPPM Universitas PGRI Semarang, Central Java, Indonesia, and the Naveen Jindal Young Global Research Fellowship, O.P. Jindal Global University, Haryana, India, for their generous support in making this research possible.

References

[1] Desmond M, Muller M, Ashktorab Z, Dugan C, Duesterwald E, Brimijoin K, et al. Increasing the speed and accuracy of data labeling through an ai assisted interface. Proceedings of the 26th International Conference on Intelligent User Interfaces, 2021, p. 392–401. https://doi.org/10.1145/3397481.3450698.

[2] Desmond M, Brachman M, Duesterwald E, Dugan C, Joshi NN, Pan Q, et al. AI assisted data labeling with interactive auto label. Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, 2022, p. 13161–3. https://doi.org/10.1609/aaai.v36i11.21714.

[3] Julianto IT, Kurniadi D, Balilo Jr BB, Rohman F. A Comparative Study of Alternative Automatic Labeling Using AI Assistant. Sinkron: Jurnal Dan Penelitian Teknik Informatika 2024;8:2125–33. https://doi.org/10.33395/sinkron.v8i4.13950.

[4] Biswas S, Young K, Griffith J. A comparison of automatic labelling approaches for sentiment analysis. ArXiv Preprint ArXiv:221102976 2022.

[5] Nema S, Vachhani L. Surgical instrument detection and tracking technologies: Automating dataset labeling for surgical skill assessment. Front Robot AI 2022;9. https://doi.org/10.3389/frobt.2022.1030846.

[6] Imtihan K, Mutawali L, Bagye W, Tantoni A. Automated Label Extraction for Sentiment Analysis in Indonesian Text. Int J Adv Sci Eng Inf Technol 2025;15. https://doi.org/10.18517/ijaseit.15.3.20602.

[7] Thomas S, Yuliana, Noviyanti. P. Study Analisis Metode Analisis Sentimen pada YouTube. Journal of Information Technology 2021;1:1–7. https://doi.org/10.46229/jifotech.v1i1.201.

[8] Faizal A, Irawan ASY, Juardi D. Perbandingan Lexicon Based Dan Naïve Bayes Classifier Pada Analisis Sentimen Pengguna Twitter Terhadap Gempa Turki. INTECOMS: Journal of Information Technology and Computer Science 2023;6:1037–48. https://doi.org/10.31539/intecoms.v6i2.7360.

[9] Setiawan A. Analisis Sentimen Masyarakat Di Twitter Terhadap Kejadian Bom Bunuh Diri Polsek Astana Anyar Menggunakan Algorithma SVM Dengan Leksikon Vader Dan Inset. Skripsi. UIN Syarif Hidayatullah Jakarta, 2024

[10] Musfiroh D, Khaira U, Utomo PEP, Suratno T. Analisis Sentimen terhadap Perkuliahan Daring di Indonesia dari Twitter Dataset Menggunakan InSet Lexicon: Sentiment Analysis of Online Lectures in Indonesia from Twitter Dataset Using InSet Lexicon. MALCOM: Indonesian Journal of Machine Learning and Computer Science 2021;1:24–33. https://doi.org/10.57152/malcom.v1i1.20.

[11] Sriyanti ZA, Kartika DSY, Najaf A. Implementasi Model BERT Pada Analisis Sentimen Pengguna Twitter Terhadap Aksi Boikot Produk Israel. Jurnal Informatika Dan Teknik Elektro Terapan 2024;12. https://doi.org/10.23960/jitet.v12i3.4743.

[12] Sahoo C, Wankhade M, Singh BK. Sentiment analysis using deep learning techniques: a comprehensive review. Int J Multimed Inf Retr 2023;12:41. https://doi.org/10.1007/s13735-023-00308-2.

[13] Rachmawati F, Azmi U, Azwarini R. Comparison of Lexicon-Based Methods and Bidirectional Encoder Representations for Transformers Models in Sentiment Analysis of Government Debt Market Movements. International Journal of Engineering and Computer Science Applications (IJECSA) 2025;4:13–28. https://doi.org/10.30812/ijecsa.v4i1.4832.

[14] Ardiansyah, Adika Sri Widagdo, Krisna Nuresa Qodri, Saputro FEN, Nisrina Akbar Rizky P. Analisis sentimen terhadap pelayanan Kesehatan berdasarkan ulasan Google Maps menggunakan BERT. Jurnal Fssilkom 2023;13:326–33. https://doi.org/10.37859/jf.v13i02.5170.

[15] Tabinda Kokab S, Asghar S, Naz S. Transformer-based deep learning models for the sentiment analysis of social media data. Array 2022;14:100157. https://doi.org/10.1016/j.array.2022.100157.

[16] Vidya Chandradev, I Made Agus Dwi Suarjaya, I Putu Agung Bayupati. Analisis Sentimen Review Hotel Menggunakan Metode Deep Learning BERT. Jurnal Buana Informatika 2023;14:107–16. https://doi.org/10.24002/jbi.v14i02.7244.
[17] Susanto J. Analisis Sentimen Pengguna Aplikasi Udemy Dengan Menggunakan Metode Naïve Bayes. Doctoral dissertation. Universitas Duta Bangsa Surakarta, 2024
[18] Malasari N, Ramli M. Analisis Sentimen Media Sosial Menggunakan Algoritma BERT dan LSTM. Journal of Computer Science and Information Technology 2025;1:85–92. DOI: https://doi.org/10.70716/jocsit.v1i3.318.
[19] Amien M, Gunawan GF. BERT dan Bahasa Indonesia: Studi tentang Efektivitas Model NLP Berbasis Transformer. ELANG: Journal of Interdisciplinary Research 2024;1:132–40. https://doi.org/10.32664/elang.v1i02.
[20] Ogbuokiri B, Obaido G, Kamalu C, Aruleba K, Achilonu O, Mienye ID, et al. Cross-domain fairness audit of sentiment label bias in foundation models: Comparing human and machine annotations on tweets and reviews. Machine Learning with Applications 2025;21:100717. https://doi.org/10.1016/j.mlwa.2025.100717.
[21] Afifah SN, Prabowo MA, Agustina AY, Razik MA. The Effectiveness of the Tapera Program in Improving the Welfare of Government Employees: Media Ethnography Analysis. Innovation Business Management and Accounting Journal 2024;3:533–43. https://doi.org/10.56070/ibmaj.2024.057.
[22] Sihombing E, Halmi Dar M, Aini Nasution F. Comparison of Machine Learning Algorithms in Public Sentiment Analysis of TAPERA Policy. International Journal of Science, Technology & Management 2024;5:1089–98. https://doi.org/10.46729/ijstm.v5i5.1164.
[23] Firdaus MP, Trisnawarman D. Analisis Sentimen Publik terhadap Program Tabungan Perumahan Rakyat Menggunakan Model IndoBERT Lite pada Komentar YouTube. MALCOM: Indonesian Journal of Machine Learning and Computer Science 2025;5:359–68. https://doi.org/10.57152/malcom.v5i1.1744.
[24] Syahputra RA, Arifin R, . S, Iqbal M. Sentiment Analysis on Tabungan Perumahan Rakyat (TAPERA) Program by using Support Vector Machine (SVM). Journal of Applied Informatics and Computing 2024;8:531–41. https://doi.org/10.30871/jaic.v8i2.8694.
[25] Faturohman MI, Arifin M. Text Blob-Based Sentiment Analysis Of Tabungan Perumahan Rakyat (TAPERA) Policy: A Public Perceptron Study. Jti Undip: Jurnal Teknik Industri 2025;20:11–20. https://doi.org/10.14710/jati.20.1.11-20.
[26] Muhammadi RH, Laksana TG, Arifa AB. Combination of Support Vector Machine and Lexicon-Based Algorithm in Twitter Sentiment Analysis. Khazanah Informatika : Jurnal Ilmu Komputer Dan Informatika 2022;8:59–71. https://doi.org/10.23917/khif.v8i1.15213.
[27] Muhandhis I, Ritonga AS. Public sentiment analysis on TikTok about Tapera policy using Random Forest classifier. Sistemasi: Jurnal Sistem Informasi 2025;14:354–65. https://doi.org/10.32520/stmsi.v14i1.4878.
[28] Rianto, Mutiara AB, Wibowo EP, Santosa PI. Improving the accuracy of text classification using stemming method, a case of non-formal Indonesian conversation. J Big Data 2021;8:26. https://doi.org/10.1186/s40537-021-00413-1.
[29] Ridwan Petervan Siburian F, Suharjito. Boosting-Based Machine Learning Models and Hyperparameter Tuning for Predicting Vehicle Carbon Dioxide Emission. Advance Sustainable Science Engineering and Technology 2025;7:02504019. https://doi.org/10.26877/asset.v7i4.2097.