Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.5, October 2025, pp. 2562~2573
e-ISSN: 2548-6861

Integrating IndoBERTweet and GRU for Opinion Classification on X Towards Public Transportation in Jakarta

Fajria Ulumin Nafiah [1]*, Talitha Fujisai Panglima [2]*, Mohammad Idhom [3]*, Trimono [4]*
* Data Science, Universitas Pembangunan Nasional "Veteran" Jawa Timur
22083010081@student.upnjatim.ac.id [1], 22083010079@student.upnjatim.ac.id [2], idhom@upnjatim.ac.id [3], trimono.stat@upnjatim.ac.id [4]

Article Info

Article history:

Received 2025-08-09
Revised 2025-08-30
Accepted 2025-09-08

Keyword:

Gated Recurrent Unit (GRU),
IndoBERTweet,
Public Transportation,
Text Classification,
X

ABSTRACT

Jakarta, the capital of Indonesia, faces persistent challenges with its public transportation system due to rapid urbanization, increased use of private vehicles, and poor service quality. While social media platforms such as X (formerly Twitter) offer valuable insights into public opinion, their unstructured nature complicates analysis. This study uses deep learning models to categorize user sentiments into six labels that cover positive and negative aspects of comfort, safety, and punctuality. The results show that IndoBERTweet achieved the highest performance, with 95.43% accuracy and a macro F1-score of 0.9545. It also required the shortest training time, at six minutes and 30 seconds. IndoBERTweet+GRU followed closely behind with an accuracy of 94.62% and a macro F1-score of 0.9460 in six minutes and 50 seconds. This shows that adding a GRU layer provides competitive results, but does not surpass the baseline model. Error analysis revealed that, while the models performed well with explicit sentiments, the models struggled with implicit expressions, such as sarcasm and mixed opinions. These results demonstrate the potential of sentiment analysis in real-time monitoring systems, which could help policymakers identify urgent issues and support data-driven improvements in Jakarta's urban transportation services.

I. INTRODUCTION

Every tweet tells a story, and in Jakarta, those stories capture the pulse of Jakarta's people as they navigate the challenges of urban life, one post at a time. According to World Population Review [1], Jakarta's 2025 population is now estimated to be over 10 million residents. To address congestion and improve mobility, the government has introduced various public transit systems such as KRL, LRT, MRT, Jaklingko, and Transjakarta. However, despite these developments, the quality, accessibility, and integration of public transportation remain inconsistent, falling short of public expectations [2]. The discrepancy is frequently voiced on social media platforms like X, where users actively share their frustrations, experiences, and suggestions regarding daily commutes. This presents a critical gap, while people speak up, decision-makers lack structured insight to respond effectively [3]. If left unaddressed, it might result in ongoing public discontent, misinformed policy decisions, and eventually a transportation system that is unable to satisfy the demand of its users.

This research uses a Natural Language Processing (NLP) approach, which allows computers to process and interpret human language similarly to how people naturally communicate. Various NLP techniques have been widely applied in tasks such as sentiment analysis, and text classification [4], making it a suitable choice for analyzing public opinion regarding Jakarta's transportation system. To effectively implement this NLP approach, this study utilizes a combination of IndoBERTweet and Gated Recurrent Unit (GRU). IndoBERTweet is a language model pretrained for use on the Indonesian Twitter platform. The model is beneficial to address vocabulary mismatch issues, ensuring more effective tokenization and representation of domain-specific words [5]. The vanishing gradient problem is addressed by a type of recurrent neural network challenge Gated Recurrent Units (GRU), which is designed to capture long-range dependencies. GRUs incorporate two gating

mechanisms, the update gate, which determines which information to preserve, and the reset gate. These gates allow the model to selectively remember or discard information, making GRUs more computationally efficient than LSTMs [6].

In recent years, several researchers have employed deep learning approaches to analyze public opinion on transportation systems, highlighting both the effectiveness and limitations of current methods when applied to social media platforms. A study by Prawinata et al. [7] examined about 30,000 public opinions about Jakarta's transportation system employing the Long Short Term Memory (LSTM) classification technique from social media X. The research explored four scenarios combining CBOW and Skip-Gram feature extraction with 80:20 and 70:30 data splits, achieving high accuracy between 85.16% and 85.9%. The best result, 85.9%, was obtained using Skip-Gram with an 80:20 split. However, it also identifies several shortcomings, notably about the informal and colloquial vocabulary that characterises Indonesian Twitter. This limitation highlights the benefit of using IndoBERTweet-GRU. IndoBERTweet is pre-trained on Indonesian Twitter data, so it can capture colloquial and informal expressions more effectively. The GRU layer also improves the model's ability to learn sequential dependencies within contextual embeddings. Then, Anisa and Wiwik [8] conducted research on sentiment analysis to improve the quality of public transportation services, specifically the "Suroboyo Bus". In their study, they applied the Random Forest Classifier to perform the classification task. The dataset was split into 70% for training and 30% for testing. This approach resulted in an accuracy of 71.27%. Although this classical machine learning approach demonstrated reasonable performance, it struggles to capture deeper semantic and contextual nuances of social media text. IndoBERTweet-GRU, on the other hand, uses contextual embeddings from a Transformer model trained on Indonesian Twitter data and provides a more robust framework for handling informal, context-rich language in public opinion. Nadya and Hasanul [9] carried out a study aimed at capturing public sentiment toward Jakarta's public transportation system. To tackle the sentiment classification task, they implemented an LSTM model trained over 15 epochs. The model achieved a training accuracy of 98% and a test accuracy of 94%, with a loss value of 0.24. For comparison, they also tested a Support Vector Machine (SVM) on the same dataset, which only managed to reach a peak accuracy of 78%. The LSTM model demonstrated strong performance compared to the SVM baseline; however, its effectiveness may be limited by its reliance on word embeddings that are not well-suited to the informal and dynamic nature of social media language. IndoBERTweet-GRU addresses this issue by using contextual Indonesian Twitter embeddings and GRU to capture sequential patterns, resulting in more accurate sentiment classification in noisy text. Merdiansah et al. [10] analyzed public sentiment on electric vehicles in Indonesia using the IndoBERT model and found that training with IndoNLU data improved contextual understanding, achieving 98.54% accuracy. Their findings underscore the effectiveness of deep learning in sentiment analysis while highlighting challenges in processing informal language. Indriani [11] showed that IndoBERTweet outperformed IndoBERT and mBERT in a multilabel student feedback classification task. The best performance, using 64-token sequences with end truncation, achieved a macro F1-score of 0.8462 which is higher than IndoBERT (0.8432) and mBERT (0.8230). These findings further demonstrate the effectiveness of IndoBERTweet in processing Indonesian text. IndoBERTweet-GRU takes this a step further by adding a GRU layer to capture sequential dependencies beyond contextual embeddings. This offers the potential for even greater accuracy in complex sentiment classification tasks.

While existing studies have explored various machine learning and deep learning approaches, including LSTM, Random Forest, and IndoBERT, most have not yet adopted Large Language Models (LLMs) such as IndoBERTweet. These models are capable of capturing deeper semantic and contextual meaning, which is especially crucial in processing informal, user-generated content on social media. Furthermore, prior works rarely combine contextual embeddings with lightweight sequence modeling, limiting both depth of understanding and scalability. To address the identified gaps, this study proposes a hybrid framework that combines IndoBERTweet for context-rich representation with Gated Recurrent Units (GRU) to model the sequential nature of tweet data. This integration constitutes the main innovation of the research, aiming to improve the classification of informal, short-form, user-generated texts on social media, particularly in the context of public opinion regarding Jakarta's transportation system. The proposed method is designed to synergize the linguistic depth of IndoBERTweet with the computational efficiency and sequence modeling capabilities of GRU.

II. METHODS

The experimental setup, evaluation metrics, and hyperparameter configurations are detailed in the following subsections in Figure 1.

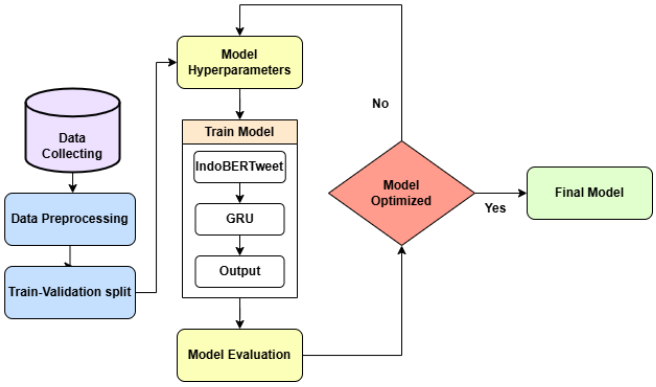

Figure 1. Research Flow

A. Data Collecting

This study collected data through web scraping from X (formerly Twitter) using a keyword-based filtering approach focused on Jakarta's public transportation modes: MRT, KRL, TransJakarta, Jaklingko, and LRT. The scraping was performed with a publicly available Selenium-based script from GitHub [12], which automated tweet loading and extraction through browser interaction. A total of 18,623 tweets from May 2020 to May 2025 were gathered and stored in CSV format for subsequent preprocessing and analysis.

B. Data Preparation

The dataset was labeled manually by two annotators, the first and second authors of this study. The annotation scheme consisted of six sentiment classes derived from three key dimensions: *Kenyamanan* (comfort), *Keamanan* (security), and *Ketepatan Waktu* (punctuality). Each dimension was categorized as either positive or negative. Although a formal inter-annotator agreement score was not computed, the annotators followed jointly defined guidelines and resolved disagreements through discussion to maintain consistency. The selection of three key dimensions is grounded in prior literature on public transportation service quality in Jakarta [13]. These dimensions are consistently highlighted as critical factors influencing user satisfaction and perception. These findings reinforce the relevance of adopting comfort, safety, and punctuality. Kusrowo et al. [14] also state in their research that public transportation must meet three basic criteria, comfort, safety, and punctuality. Additionally, class imbalance within the labeled dataset was addressed to improve the robustness and performance of the proposed model.

C. Data Preprocessing

This study starts by retaining only the content and label columns, followed by multi-stage text normalization. The preprocessing stage is carried out to prepare the dataset before model training. The main steps include case folding, where all text is converted to lowercase for consistency. The next step is normalization of informal language to achieve uniformity, such as spelling correction and abbreviation expansion using a publicly available Indonesian slang dictionary from GitHub [15] to deal with informal or unusual language that is commonly found on social media. After that, noise is removed, including punctuation, symbols, hashtags, and emojis, to simplify the vocabulary [16]. To minimize noise and ensure data consistency, irrelevant data such as spam advertisements, duplicate posts, and entries dominated by non-Indonesian languages were identified and excluded. The next step is stopword removal, to eliminate common words that carry minimal semantic weight and are often excluded to focus on more meaningful terms [17][18]. Once the text is cleaned, it undergoes tokenization. Tokenization is a process to split text into individual tokens that can be processed by the model [19], performed with the IndoBERTweet Tokenizer [20], which is specifically designed to handle the nuances of Indonesian informal and slang expressions commonly found on social media. These preprocessing steps ensure that the text is clean, standardized, and suitable for representation learning in subsequent modeling. The dataset is labeled into six sentiment classes representing three key dimensions, such as *Kenyamanan* (comfort), *Keamanan* (security), and *Ketepatan Waktu* (punctuality), each with positive and negative polarities, while class imbalance is addressed to enhance model performance.

D. Train-Validation Split

Ensuring proper dataset splitting is crucial to developing a good model to make accurate predictions on unseen data [21]. To find an optimal set of model parameters that could help prevent overfitting (the model does not generalize well), it's necessary to split the data into training and validation sets. The training set is the subset of data utilized by the model for learning. It gains knowledge by identifying patterns, connections, and links within this data portion. In contrast, validation is a subset of data used to assess the performance of different model configurations during model selection, which helps determine the optimal model parameters by evaluating prediction errors [22]. Usually, a bigger training set helps the model learn more effectively, enabling it to generalize better to unseen data.[23]. To sum up, having a bigger training set improves the model's learning ability, while a properly sized validation set ensures accurate monitoring of the model's performance during training.

E. Model Training

The model in this study combines IndoBERTweet and Gated Recurrent Units (GRU) to effectively handle informal Indonesian social media text. IndoBERTweet generates rich contextual embeddings tailored for tweets by leveraging a transformer architecture pre-trained on Indonesian-language data, making it well-suited for capturing nuances in informal expressions [24]. These embeddings are then fed into a GRU layer, which models the sequential dependencies of the text through update and reset gates, allowing the network to retain relevant context across tokens while mitigating vanishing gradient issues common in standard RNNs [25]. Finally, a dense layer processes the GRU output to classify sentiments. This hybrid approach leverages the strengths of both contextual embedding and sequence modeling, and prior studies have shown that combining transformer-based models with GRU layers can lead to better performance in sentiment analysis compared to using either model alone [26].

The GRU has only two gates, the reset and update gates. Reset gates are represented by equation (1) below, update gates by equation (2), and hidden state candidates and the final hidden state are determined by equations (3) and (4) [27].

$$r_t = \sigma(x_t W_r + h_{t-1} W_r + b_r) \quad (1)$$

$$z_t = \sigma(x_t W_z + h_{t-1} W_z + b_z) \quad (2)$$

$$\underline{h}_t = tan\, h(x_t W_x + (r_t \odot h_{t-1})\, W_h + b) \quad (3)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \underline{h}_t \quad (4)$$

Then, in the model's output layer, a softmax activation function is used, a form of logistic regression that can be used in multi-class classification [28]. Softmax converts the network output into interpretable probability vectors, which allows mapping the network results into specific classes [29]. Equation (5) shows the activation function formula, softmax, which takes x_i as input the number of labels and j the class label [30].

$$S(x_i) = \frac{e^{x_i}}{\sum_j^n e^{x_j}} \quad (5)$$

The cross-entropy loss function is then employed as a metric to measure the dissimilarity between two probability distributions to maximize the model's performance in multi-class classification tasks [31]. The equation is shown in Equation (6) below, where p is the true distribution, q is the predicted distribution, and x ranges over all possible outcomes [32]. With that being said, according to PyTorch Contributors, the CrossEntropyLoss function integrates softmax activation function and negative log-likelihood loss in one class to effectively calculate classification loss for multi-class problems [33].

$$H(p,q) = -\sum_x p(x) \log_e q(x) \quad (6)$$

The step continues with optimizing the model's parameters using Adaptive Moment Estimation (Adam) for weight updates. The implementation of both adaptive momentum acceleration and learning rate by Adam involves the combination of the moving average exponentially on second momentum with the first momentum. In this research, the learning rate is set to 2e-5 or 0.00002. A small learning like this is commonly used when fine-tuning a large pre-trained model. It allows the model to adjust its weights gradually, preventing large updates that might destabilize training and helping preserve the pre-learned knowledge from the original IndoBERTweet model. Adam calculates adaptive learning rates for each weight parameter from the first moment and second moment estimates of the gradient [34]. Several hyperparameters must be determined, the learning rate (α), the exponential decay rate for the first and second momentum estimates (β), a very small number (close to 0) to prevent zero division (ε), and the assumption parameter to be optimized is θ_t, the gradient associated with the objective function parameter at timestep t will be fixed in Equation (7) [35].

$$g_t = \nabla_\theta f_t(\theta_{t-1}) \quad (7)$$

Then, the bias updates for the first and second momentum estimates are calculated as in Equations (8) and (9), respectively [36].

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (8)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g \odot g \quad (9)$$

The corrected biases in the first and second moments are calculated with Equations (10) and (11) [35].

$$\hat{m} = \frac{m_t}{1 - \beta_1^t} \quad (10)$$

$$\hat{v} = \frac{v_t}{1 - \beta_2^t} \quad (11)$$

Then, after getting the bias-corrected for both momentum estimates, the parameter improvement can be calculated as in Equation (12) [35].

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \varepsilon}} \quad (12)$$

F. Model Hyperparameters

Configuring hyperparameters plays a vital role in enhancing the effectiveness of machine learning and deep learning models [37]. Several hyperparameters, including learning rate, batch size, number of layers, epoch, and number of hidden units, are important in neural network architectures. When tuned properly, these hyperparameters can lead to faster convergence and better predictive performance [38]. An epoch represents one complete pass through the entire training dataset. During each epoch, the model processes every sample in the training set once, using forward and backward propagation to update its internal parameters based on the calculated errors [39]. The learning rate is particularly influential. It controls how much the model's weights are updated in response to the error during training. Therefore, selecting an appropriate learning rate is vital for achieving optimal training results [40].

In this study, a fixed set of hyperparameters was applied. These values were selected based on prior literature and empirical best practices for fine-tuning transformer-based models. Fine-tuning Transformer models typically uses learning rates around 2e-5, which provides a good trade-off between training speed and stability and moderate learning rates such as 2e-5 help to avoid divergence and ensure effective convergence [40]. Based on those statements, the learning rate was set to 2e-5. A batch size of 32 was chosen to ensure stable gradient updates while remaining within computational constraints. The IndoBERTweet max length of embedding is 64. The GRU hidden size was set to 128 units with a dropout rate of 0.1 to provide sufficient capacity to model temporal dependencies from contextual embeddings without overfitting. This fixed setup was chosen to prioritize computational efficiency and reproducibility, as the main goal of this study was to evaluate the performance of the IndoBERTweet-GRU architecture on the classification task rather than to explore exhaustive hyperparameter combinations.

G. Model Evaluation

Model evaluation is the process of ensuring the quality and reliability of the model and measuring the performance and ability of the model developed in making predictions or producing accurate outputs based on the data that has been studied [10]. This study's model evaluation framework is based on a confusion matrix. A confusion matrix is a tool used to understand the performance of a classification model, which stores four combinations of actual and predicted values

[41]. This matrix is used to derive performance metrics such as accuracy, precision, recall, and F1 score. These metrics provide a comprehensive view of classification performance, particularly with regard to the balance between correctly and incorrectly classified instances. Meanwhile, the research did not employ the area under the curve (AUC) metric because the evaluation focuses more on categorical prediction quality than on ranking probability outputs. Table I below presents an example of a confusion matrix.

TABLE I
CONFUSION MATRIX

		Predicted Class	
		Positive	**Negative**
Actual Class	Positive	TP	FP
	Negative	FN	TN

There are two types of predictions, correct and incorrect (errors). A True Positive (TP) occurs when the actual value and the prediction are positive. A False Positive (FP) occurs when the prediction is positive, but the actual value is negative. True Negative (TN) means both values are negative. False Negative (FN), although predicted to be negative, the sample is positive [41]. Evaluation is carried out using classification report metric, which includes accuracy, precision, recall, and F1-score [42]. The following formula is used to calculate accuracy, precision, recall, and F1-score:
Accuracy: measures the percentage of correct predictions relative to the total number of samples [43]. This can be expressed as Eq. (13):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FP} \quad (13)$$

Precision: measures the likelihood of correctly detected instances of actual events [43]. This can be expressed as Eq. (14):

$$\text{Precision} = \frac{TP}{TP + FP} \quad (14)$$

Recall: measures the proportion of predicted data in its class [43]. It can be expressed as Eq. (15):

$$\text{Recall} = \frac{TP}{TP + FN} \quad (15)$$

F1-Score: calculated the harmonic mean of precision and recall, providing insights into testing accuracy [43]. This can be expressed as Eq. (16):

$$\text{F1-Score} = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall} \quad (16)$$

III. RESULT AND DISCUSSION

A. Data Gathering

The data for this study is collected from X (formerly Twitter) via web scraping using a keyword-based filtering approach to ensure topic relevance. Specific keywords representing various modes of public transportation in Jakarta were selected, including MRT, KRL, TransJakarta, Jaklingko, and LRT. Scraping was conducted using a publicly available GitHub script that uses Selenium WebDriver to dynamically interact with the X interface. This enabled automated scrolling, loading, and extraction of relevant tweet content. A total of 18,623 tweets were collected and stored in CSV format. Each entry contains metadata such as the date, username, and tweet content for further preprocessing and analysis. The data is presented in Table II, which displays the top four rows of the data.

TABLE II
SAMPLE DATA

Name	**Handle**	**Timestamp**	**Content**	**Label**
DHIZUN	@Adhiz Buana	2025-04-22T07:39:28.000Z	*Serius nanya, ini fungsinya buat …*	*Keamanan Negatif*
Luki Fachrizal	@_lukifachrizal	2025-02-01T09:12:09.000Z	*JakLingko kaya angkot lama …*	*Keamanan Negatif*
Gats	@Gatsme	2025-01-23T08:03:42.000Z	*Kota tuh terlalu sempit jalannya …*	*Keamanan Negatif*
´-	@everyellouz	2024-12-10T09:51:55.000Z	*antrian jaklingko ga tertib. nyebelin …*	*Keamanan Negatif*

B. Data Pre-processing

The data obtained from Twitter scraping totaled 18,623 rows. Each entry included metadata such as date, username, and tweet content. Preprocessing steps were then applied to remove noise from the data, including choosing the usable variables (text), stopword removal, punctuation, emoji cleaning, hashtag removal, and spelling correction. After that, the data was labeled manually, and the text was tokenized using the IndoBERTweet pre-trained model. The labels were

also encoded. Table III below shows examples of tweets before and after the cleaning process.

TABLE III
CLEANED DATA RESULT

Content	Label	cleaned_ content	tokens	encoded_ ids
mending naik tj soalnya pas gw ama najla	*Ketepatan Waktu Negatif*	*mending naik tj soalnya pas sama najla naik ja...*	[[CLS], *mending, naik, tj, soalnya, pas, sama,...*	[3, 7160, 3493, 9665, 9951, 1746, 1959, 14904,..
ngakak bgtt woii. labubu kejepit#la bubu #krl #kai	*Keamanan Negatif*	*mengakak banget woii labubu kejepit*	[[CLS], *mengak, ##ak, banget, woii, lab, ##ubu...*	[3, 3874, 1484, 10218, 20806, 2665, 25493, 262...
sepenuh-penuhnya krl di hari kerja tetep lebih...	*Keamanan Positif*	*sepenuh penuhnya krl hari kerja tetap lebih ny..*	[[CLS], *sepenuh, penuh, ##nya, krl, hari, kerj...*	[3, 20191, 3301, 1519, 15922, 1843, 2533, 2261...
sepi wae tuh.. ga ngerti jg sholat dimana dia....´-	*Kenyamanan Positif*	*sepi wae enggak mengerti sholat dimana biasany...*	[[CLS], *sepi, wae, enggak, mengerti, sholat, d...*	[3, 10369, 11100, 14863, 6572, 11607, 3357, 22...

The original label distribution in the dataset was imbalanced, with some classes having significantly more samples than others. This can cause the model to be biased toward the majority classes, resulting in poor generalization and reduced performance when predicting the minority classes. To address this issue, a random undersampling technique was applied. This method reduces the number of samples in the majority classes, balancing them with the minority class. With undersampling, the model should learn equally from all classes, improving the fairness and reliability of the classification results across the six labels. Table IV below illustrates the difference in data counts for each label before and after random undersampling is applied.

TABLE IV
LABEL DISTRIBUTION BEFORE AND AFTER UNDERSAMPLING

Label	Before Undersampling	After Undersampling
Ketepatan Waktu Negatif	2732	2732
Ketepatan Waktu Positif	2580	2580
Keamanan Negatif	2524	2524
Kenyamanan Positif	3993	2500
Kenyamanan Negatif	4069	2500
Keamanan Positif	2365	2365

Figure 2 shows the visualization of label distribution before and after undersampling. The resulting balanced dataset was then used in the model training process.

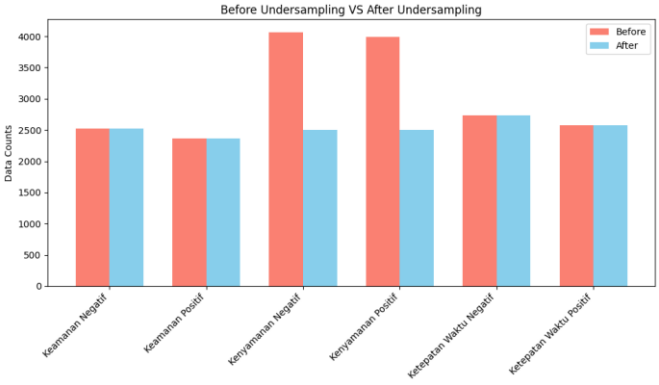

Figure 2. Label Distribution Before and After Random Undersampling

C. Train-Validation Split

After applying random undersampling, the dataset became balanced across all label classes, resulting in a total of 15201 data points (around 2500 samples per class). This balanced set of data was then split into training and testing sets using a method that split the data evenly to make sure the class distribution stayed the same in both sets. As a result, 80% of the data (12160 samples) is used to train, and the remaining 20% (3041 samples) is used to check the results. During training, a portion of the training data was split into a validation set (10% of the training set) to adjust hyperparameters and prevent overfitting. This approach ensures that the model is evaluated on both the test set and unseen data during training. Initially, we identified potential data leakage, as 461 duplicate tweets were present across the training and test sets. This issue was resolved by performing a global duplicate removal, ensuring that no overlap remained between the training, validation, and test sets. Additionally, k-fold cross-validation was not employed in this study due to the computational cost of training deep learning models.

Instead, a single stratified train, validation, and test split was used.

D. Model Training

The training was carried out using the IndoBERTweet-GRU model. The model had predefined hyperparameters. These included a learning rate of 2e-5, a batch size of 32, and a maximum sequence length for IndoBERTweet embeddings was 64. GRU hidden size of 128, and 3 training epochs. To illustrate the data flow during training, Figure 4 shows an example of how a sample tweet is processed by the model. The input text is tokenized using the IndoBERTweet tokenizer, transformed into embeddings, passed through the GRU layer, and finally classified into one of the labels using a dense layer. The model outputs raw logits, which are passed to the CrossEntropyLoss function during training. In the inference phase, the logits are converted into probabilities using a softmax function to determine the final predicted sentiment.

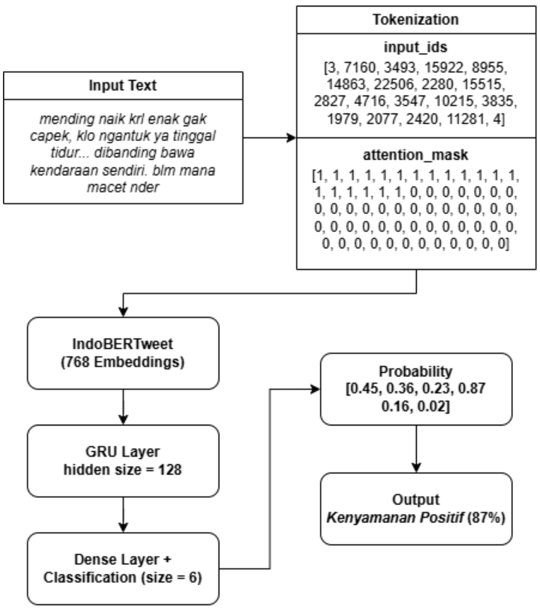

Figure 3. Training Workflow

Figure 3 above shows the workflow of the text classification process using the architecture of the IndoBERTweet hybrid model and combined with the Gated Recurrent Unit. The process involves several steps, as follows:

1) *Text Input*: The first step was to write with sentences of opinion about public transport in Jakarta efficiency, written in Indonesian language, such as *"mending naik krl enak gak capek, klo ngantuk ya tinggal tidur… dibanding bawa kendaraan sendiri. blm mana macet nder"*.
2) *Tokenization*: The text then goes through a tokenization process using the IndoBERTweet tokenizer, which converts words into numerical representations (input_ids) and generates attention_mask to mark important tokens and padding.
3) *IndoBERTweet Embedding*: The processed tokens are put into the IndoBERTweet model, which is a transformer model that's been specially trained on Indonesian Twitter data. The output from this stage is a 768-dimensional embedding vector for each token.
4) *GRU Layer*: The embeddings are then processed by a Gated Recurrent Unit (GRU) layer with a hidden size of 128. This layer is responsible for capturing sequential and contextual relationships between tokens within the text.
5) *Dense Layer and Classification*: The GRU output is passed to a fully connected (dense) layer that performs classification into one of six predefined categories. The output is a probability distribution over the six classes.
6) *Output Prediction*: The model outputs probability scores for each class. The class with the highest probability is selected as the final prediction. In the provided example, the highest score (0.87) corresponds to the category *Positive Comfort*, which is then returned as the classification result.

The model's learning progress was tracked through training and validation loss values recorded after each epoch.

TABLE V
EPOCH RECORD RESULT

Epoch	Training Loss	Training Accuracy	Validatio n Loss	Validation Accuracy
1	0.5668	0.8138	0.2386	0.9260
2	0.2087	0.9337	0.1499	0.9525
3	0.1499	0.9525	0.1667	0.9517

Based on Table V above, at the first epoch, the training loss was relatively high at 0.5668 with an accuracy of 81.38%. In comparison, the validation loss was much lower at 0.2386 with an accuracy of 92.60%, indicating the model started to learn well on unseen data. By the second epoch, the training loss decreased significantly to 0.2087, and the accuracy improved to 93.37%. Validation loss also decreased to 0.1499, with validation accuracy increasing to 95.25%, showing notable performance improvement on both datasets. In the last epoch, the training loss further dropped to 0.1499 with a training accuracy of 95.25%. Validation loss slightly increased to 0.1667, and validation accuracy slightly decreased to 95.17%. Despite this small fluctuation, the

model's performance remained stable, and the high validation accuracy indicates no significant overfitting occurred during training.

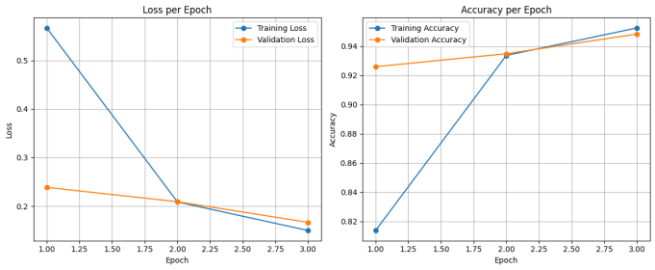

Figure 4. Training and Validation Performance per Epoch

A comprehensive evaluation of the model's training progress, as illustrated by the loss and accuracy in Figure 4, indicates the development of strong generalization capabilities. This is shown by a slow decrease in both training and validation loss, along with a steady increase in accuracy over time. The similarity between the training and validation accuracy and loss in the final results suggests that the model did not overfit.

E. Model Evaluation

Critical insight into the model's classification capabilities and overall robustness is provided by evaluating its performance across multiple sentiment classes. The stability is further confirmed by the classification report, which evaluates the model's ability to accurately predict the six distinct sentiment classes, associated with high values in precision, recall, and F1-score. An overall accuracy of 94.84% was demonstrated by the model, indicating that approximately 95% of the 3041 test samples were correctly classified by it. However, the average and weighted average for precision, recall, and F1-score are all about 0.95, which shows that the model always performs well across all types of sentiments, even when some classes have more data than others. These findings indicate the model's resilience and accuracy in addressing complex multi-class sentiment classification tasks. A detailed classification of the sentiment classes is presented in Table VI below.

TABLE VI
CLASSIFICATION REPORT

	Precision	**Recall**	**F1-Score**	**Support**
0	0.9826	0.9565	0.9694	414
1	0.9555	0.9839	0.9695	436
2	0.9464	0.9523	0.9493	482
3	0.9299	0.9667	0.9479	480
4	0.9763	0.8988	0.9360	504
5	0.9240	0.9593	0.9413	418
Accuracy			0.9517	2734
Macro Avg	0.9524	0.9529	0.9522	2734
Weighted Avg	0.9525	0.9517	0.9516	2734

Based on Table VI above, the performance of each label demonstrates strong results. All precision, recall, and F1-score values are above 0.9, with support values of around 400-500. These outcomes indicate that the model achieves balanced performance across labels. This balanced performance may be related to the use of random undersampling before training, which resulted in a more proportionate distribution across labels. The comparison of label distributions before and after undersampling is provided in Table IV.

Regarding classification problems, determining how well a model performs involves more than just calculating accuracy. One of the most effective tools for performance evaluation in such cases is the confusion matrix. A confusion matrix presents a detailed breakdown of the model's predictions compared to the actual ground truth labels, organized in a tabular format. Each row of the matrix represents the actual class, while each column represents the predicted class, allowing for a comprehensive view of where the model performs well and where it struggles. It shows how the model's predictions compare to the actual results. This matrix is a powerful tool for visualizing and interpreting the outcomes of a classifier. It helps to uncover patterns of misclassification or error and evaluate metrics. The confusion matrix below in Figure 5 shows the results of a multiclass classification model across six classes, labeled from 0 to 5.

The results demonstrated a very good performance of the model. Class 0 is correctly classified 483 times, with a few misclassifications as Class 1 and 5. Class 1 exhibits a similar pattern, with 459 correct predictions and a low number of misclassifications as Class 2. In Class 3, the model was able to make 459 correct predictions. Although Class 4 achieved 495 correct predictions, it exhibits slightly more confusion than the other classes, particularly with classes 2 and 5. Then, with over 506 accurate predictions, Class 5 has the highest number of correct predictions. Overall, the confusion matrix results suggest that the model achieves balanced performance, with each class showing a high number of correct predictions (ranging approximately from 450 to 500).

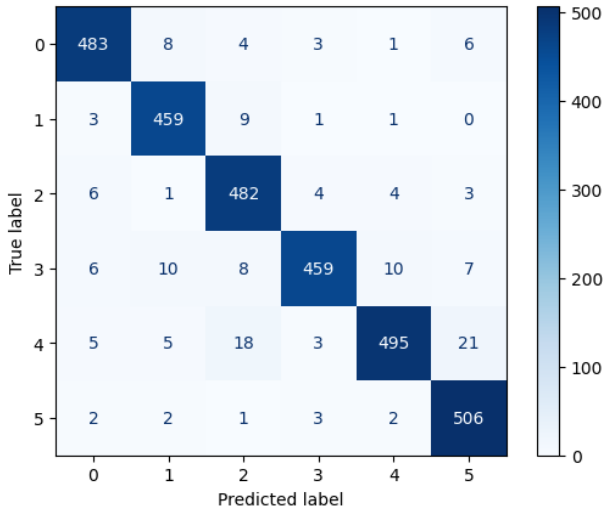

Figure 5. Confusion Matrix Result

To further examine the results presented in the confusion matrix for actual Twitter data, Table VI provides a detailed overview of the model's predictions. It includes a selection of input texts, their true labels, and the labels predicted by the model. This comparison highlights both correct and incorrect classifications and helps identify cases where subtle or misleading text features influenced the model's predictions. Analyzing these examples offers deeper insight into the model's decision-making process and points to areas where its accuracy may be improved

TABLE VII
PREDICTION RESULTS TABLE

Text	True Label	Prediction
maaf tp kdg sebel sm org yg "knp sih org2 ga pake transum pdhl nyaman" karena ORG2 EMG PAKE TRANSUM EMG PAKE TJ KRL LRT MRT dan itu jg SUMPEK alias penuh terus bos berdiri rapet2 ga duduk. / Sorry, sometimes I get annoyed with people saying "why don't people use public transport even though it's comfortable" because PEOPLE ARE ACTUALLY USING TRANSJAKARTA, TJ, KRL, LRT, MRT and it's also CROWDED, so everyone's standing really close, no seats.	*Kenyamanan Negatif* / Comfort Negative	*Kenyamanan Negatif* / Comfort Negative
krl rangkaian baru enak jg yah 🥰✋tapi gabisa nyenderrin kepala jauh bgt ga enak 🥵🥵🥵 / The new KRL series is nice and comfortable 🥰✋ but you can't rest your head, it's too far, really uncomfortable 😭😭😭	*Kenyamanan Negatif dan Positif* / Comfort Negative and Positive	*Kenyamanan Negatif dan Positif* / Comfort Negative and Positive
Asli jaklingko hari ini susah banget.. Karna udah 7.57 akhirnya dari halte ke kantor LARIII.. Trus absen jam 7.59 Ngos2an asli.. / Honestly, JakLingko was really difficult today… Since it was already 7:57, I finally ran from the bus stop to the office… Checked in at 7:59, totally out of breath…	*Ketepatan Waktu Negatif* / Timeliness Negative	*Keamanan Negatif* / Security Negative

Based on Table VII above, some errors and observations in the model can be attributed to the inherent challenges in the dataset. In certain cases, tweets contain mixed sentiments, such as the second example where user expresses both positive and negative aspects of comfort. Since the model is constrained to assign a single label per tweet, it primarily captures the dominant sentiment, potentially overlooking secondary aspects. Interestingly, in the first example, the tweet is about negative comfort, but contains phrases like "why don't people use public transport even though it's comfortable." Despite this conflicting wording, the model correctly classified the tweet as negative, demonstrating its ability to capture contextual cues. On the other hand, some tweets are less explicit, such as the third example describing a delayed commute with JakLingko. The lack of clear mentions of lateness or system performance makes it difficult for the model to classify it under timeliness negative, leading to misclassification into a different category. These cases highlight the limitations of the model's ability to infer implicit information from the text.

F. Model Comparison and Analysis

Besides highlighting the proposed model, we also conducted experiments with other hybrid architectures such as IndoBERTweet-GRU and IndoBERTweet-LSTM, as well as the base IndoBERTweet model. All models were trained using the same parameters as the proposed model to ensure a fair comparison. This section presents the comparison and analysis of these models. Table VIII summarizes their performance based on several indicators (e.g., Accuracy, Macro Average, and Execution Time).

TABLE VIII
MODEL COMPARISON TABLE

Model	Accuracy (Macro)	F1-Score (Macro)	Precision (Macro)	Execution Time
IndoBERTweet	0.9543	0.9545	0.9545	6m 30s

IndoBERTweet+LSTM	0.9473	0.9475	0.9470	7m 10s
IndoBERTweet+GRU	0.9517	0.9524	0.9622	6m 50s

Based on Table VII, IndoBERTweet achieved the highest performance among the three models with an accuracy of 0.9543 and a macro F1-Score of 0.9545, despite having the shortest execution time of 6 minutes 30 seconds. This indicates that the base IndoBERTweet model is already highly effective without additional recurrent layers. IndoBERTweet+GRU also showed competitive results with an accuracy of 0.9517 and macro F1-Score of 0.9524, slightly lower than the base model. However, its execution time (6 minutes 50 seconds) was relatively close, which suggests that the addition of GRU did not provide a significant performance improvement while adding extra complexity. On the other hand, IndoBERTweet+LSTM obtained the lowest scores, with an accuracy of 0.9473 and macro F1-Score of 0.9475, while also requiring the longest execution time (7 minutes 10 seconds). This implies that the integration of LSTM did not contribute positively to the model's performance and instead increased computational cost. In summary, the results suggest that the plain IndoBERTweet model performs best for this task. The addition of GRU and LSTM does not substantially improve performance. In this case, LSTM even reduces efficiency.

G. Application and Policy Insights

Analysis of Twitter data collected between 2020 and 2025 indicates that the category with the highest frequency is negative comfort, comprising approximately 4,000 tweets. This finding suggests that comfort is the primary concern expressed by the public regarding Jakarta's public transportation system. Consequently, policymakers should prioritize improvements in comfort-related aspects, including cleanliness, seating conditions, air conditioning systems, and the availability of vehicles during peak hours. These insights provide an evidence-based foundation for developing policies aimed at enhancing the quality of public transportation services in Jakarta.

Beyond model performance, the proposed approach demonstrates significant potential for real-time monitoring of public transportation services. By integrating this model with social media data streams, such as Twitter, user opinions can be automatically classified into specific categories (e.g., comfort, punctuality, and security). The results could then be visualized on an interactive dashboard, offering comprehensive and timely insights into public sentiment. Such integration would support policymakers and transportation authorities in identifying recurring issues and implementing data-driven interventions while simultaneously empowering commuters with real-time information about service conditions.

Overall, the proposed approach not only achieves robust model performance but also provides actionable insights for both decision-makers and end-users, highlighting its practical relevance beyond academic evaluation.

IV. CONCLUSION

This study investigated text classification of Twitter data related to services of public transportation in Jakarta, focusing on comfort, safety, and punctuality, each with positive and negative labels. Among the models evaluated, IndoBERTweet achieved the highest performance with an accuracy of 0.9543 and a macro F1-Score of 0.9545, while also having the shortest execution time of 6 minutes 30 seconds. This demonstrates that the base IndoBERTweet model is highly effective without additional recurrent layers. IndoBERTweet+GRU showed competitive results (accuracy 0.9517, macro F1-Score 0.9524) but did not significantly outperform the base model, despite slightly increased computational cost. IndoBERTweet+LSTM obtained the lowest scores (accuracy 0.9473, macro F1-Score 0.9475) and required the longest execution time, suggesting that adding LSTM did not contribute positively and even reduced efficiency.

The proposed model effectively captured explicit sentiment cues, even when tweets contained conflicting or additional information. Nevertheless, challenges remained in cases of mixed sentiments or implicit expressions, such as sarcasm or subtle complaints, leading to occasional misclassifications. The error analysis highlighted these limitations and emphasized that while IndoBERTweet+GRU can handle direct sentiment well, it struggles with nuanced language, reflecting both dataset and model constraints.

Overall, the findings confirm that leveraging hybrid language models like IndoBERTweet-GRU provides strong performance for Indonesian sentiment classification. For future work, multi-label classification could address overlapping sentiments, advanced attention mechanisms could better capture context, and expanding the dataset with implicit sentiment examples may further improve robustness. These steps would enhance model applicability in real-world monitoring of public transportation services, supporting both policymakers and commuters with actionable insights.

REFERENCES

[1] World Population Review, "Jakarta Population 2025." [Online]. Available: https://worldpopulationreview.com/world-cities/jakarta-population.

[2] M.R. Rahmatullah, M. Alimuddin, and P. Lestari, "Evaluating Jakarta's Public Transportation Services Using Passenger Feedback on Twitter," Journal of Urban Mobility and Smart Cities, vol.3, no.2, pp. 45-54, 2021.

[3] I.M. Putri, P. Wulandari, and E.Suryani, "Sentiment Analysis on Twitter using LSTM and Word2Vec for Public Opinion Monitoring," in Proc. 4th Int. Conf. Data Science and Information Technology (DSIT 2022), 2022.

[4] H. K. Dixit, "Natural Language Processing (NLP) and Understanding," International Journal of Advanced Research in

Electrical, Electronics and Instrumentation Engineering, 2025, doi: 10.15662/IJAREEIE.2025.1402025.
[5] F. Koto, J. H. Lau, and T. Baldwin, “IndoBERTweet: A Pretrained Language Model for Indonesian Twitter with Effective Domain-Specific Vocabulary Initialization,” in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 10660–10668. doi: 10.18653/v1/2021.emnlp-main.833.
[6] K. L. Tan, C. P. Lee, and K. M. Lim, “RoBERTa-GRU: A Hybrid Deep Learning Model for Enhanced Sentiment Analysis,” Applied Sciences, vol. 13, no. 6, Mar. 2023, doi: 10.3390/app13063915.
[7] D.A. Prawinata, A.D. Rahajoe, and I. G. S. M. Diyasa, ‘Analisis Sentimen Kendaraan Listrik Pada Twitter Menggunakan Metode Long Short Term Memory,” SABER: Jurnal Teknik Informatika, Sains dan Ilmu Komunikasi, vol. 2, no. 1, pp. 300-313, Jan 2024.
[8] A. Kumalasari and W. Handayani, “Sentiment Analysis to Improve the Quality of Public Transportation Services "Suroboyo Bus”, Indonesian Interdisciplinary Journal of Sharia Economics (IIJSE) , vol. 7, no. 3, pp. 6407-6426, Aug. 2024.
[9] N. R. Djodjobo and H. Fahmi, “Understanding Public Sentiment on Jakarta Public Transportation Using LSTM”, SINTECH Journal, vol. 8, no. 1, pp. 38–51, Apr. 2025.
[10] R. Merdiansah, S. Siska, and A. Ali Ridha, “Analisis Sentimen Pengguna X Indonesia Terkait Kendaraan Listrik Menggunakan IndoBERT,” Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI), vol. 7, no. 1, pp. 221–228, Mar. 2024. [In Indonesian]
[11] F. Indriani, R. A. Nugroho, M. R. Faisal, and D. Kartini, “Comparative Evaluation of IndoBERT, IndoBERTweet, and mBERT for Multilabel Student Feedback Classification,” Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 8, no. 6, pp. 748–757, Dec. 2024.
[12] Godkingjay, "selenium-twitter-scraper," GitHub, [Online]. Available: https://github.com/godkingjay/selenium-twitter-scraper.
[13] Rachman, F. F., Nooraeni, R., & Yuliana, L. (2021). Public Opinion of Transportation Integrated (Jak Lingko), in DKI Jakarta, Indonesia. Procedia Computer Science, 179, 696–703. https://doi.org/10.1016/j.procs.2021.01.057
[14] Pavliuk, Baibuz, and Honcharova, “TEXT PREPARATION FOR NATURAL LANGUAGE PROCESSING,” in Proceedings of the XIX International Scientific and Practical Conference, Dnipro, Ukraine: International Science Group, May 2024, pp. 223–225.
[15] Fendiirfan, “Kamus-Alay”, GitHub, [Online]. Available: https://github.com/fendiirfan/Kamus-Alay.
[16] Y. HaCohen-Kerner, D. Miller, and Y. Yigal, “The influence of preprocessing on text classification using a bag-of-words representation,” PLOS ONE, vol. 15, no. 5, p. e0232525, May 2020, doi: 10.1371/journal.pone.0232525.
[17] M. Gerlach, H. Shi, and L. A. N. Amaral, “A universal information theoretic approach to the identification of stopwords,” Nature Machine Intelligence, vol. 1, no. 12, pp. 606–612, Dec. 2019, doi: 10.1038/s42256-019-0112-6.
[18] K. Juluru, H.-H. Shih, K. N. Keshava Murthy, and P. Elnajjar, “Bag-of-Words Technique in Natural Language Processing: A Primer for Radiologists,” RadioGraphics, vol. 41, no. 5, pp. 1420–1426, Sep. 2021, doi: 10.1148/rg.2021210025.
[19] G. Grefenstette, "Tokenization," in Syntactic Wordclass Tagging, H. van Halteren, Ed. Dordrecht: Springer, 1999, pp. 117–133. doi: 10.1007/978-94-015-9273-4_9.
[20] Indolem, "IndoBERTweet," GitHub, [Online]. Available: https://github.com/indolem/IndoBERTweet.
[21] P. Sayarizki, Hasmawati, H. Nurrahmi, “Implementation of IndoBERT for Sentiment Analysis of Indonesia Presidential Candidates”, Indonesia Journal of Computing, vol. 9, no. 2, pp. 61-72, August. 2024
[22] Y. Xu and R. Goodacre, “On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning,” Journal of Analysis and Testing, vol. 2, no. 3, pp. 249–262, Jul. 2018, doi: 10.1007/s41664-018-0068-2.
[23] M. Sivakumar, S. Parthasarathy, and T. Padmapriya, “Trade-off between training and testing ratio in machine learning for medical image processing,” PeerJ Computer Science, vol. 10, Sep. 2024, doi: 10.7717/peerj-cs.2245.
[24] A. Wibowo, A. S. Utomo, and A. Purwarianti, "IndoBERTweet: A Pretrained Language Model for Indonesian Social Media Texts," in Proceedings of the 2021 International Conference on Asian Language Processing (IALP), 2021, pp. 123–128.
[25] K. Cho et al., “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation,” in EMNLP, 2024, pp. 1724–1734.
[26] S. Liu, Y. Chen, and X. Zhang, “Combining BERT and GRU for Sentiment Analysis on Social Media,” in Proc. 2021 Int. Conf. on Computational Linguistics and Intelligent Systems, 2021, pp. 250–258.
[27] M. Cho, C. Kim, K. Jung, and H. Jung, “Water Level Prediction Model Applying a Long Short-Term Memory (LSTM)–Gated Recurrent Unit (GRU) Method for Flood Prediction,” Water, vol. 14, no. 14, p. 2221, Jul. 2022, doi: 10.3390/w14142221.
[28] M. R. Dwimahendra et al., “Klasifikasi Jenis Kayu Berdasarkan Citra Serat Kayu Menggunakan Convolutional Neural Network,” JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika), vol. 10, no. 1, pp. 72-80, March. 2024. [In Indonesian] doi: 10.29100/jipi.v10il.5726
[29] T. Garai, S. Dalapati and F. Smarandache., “Softmax Function Based Neutrosophic Aggregation Operators and Application in Multi-Attribute Decision Making Problem,” Neutrosophic Sets and Systems, vol. 56, no. 1, 2023.
[30] M. F. Naufal and S. F. Kusuma, “Analisis Perbandingan Algoritma Machine Learning dan Deep Learning untuk Klasifikasi Citra Sitem Isyarat Bahasa Indonesia (SIBI),” Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), vol. 10, no. 4, pp. 873-882, August. 2023. doi: 10.25126/jtiik.2023106828
[31] E. Matsuyama, M. Nishiki, N. Takahashi, and H. Watanabe, “Using Cross Entropy as a Performance Metric for Quantifying Uncertainty in DNN Image Classifiers: An Application to Classification of Lung Cancer on CT Images,” Journal of Biomedical Science and Engineering, vol. 17, no. 01, pp. 1–12, 2024, doi: 10.4236/jbise.2024.171001.
[32] E. Matsuyama, H. Watanabe, and N. Takahashi, “Performance Comparison of Vision Transformer- and CNN-Based Image Classification Using Cross Entropy: A Preliminary Application to Lung Cancer Discrimination from CT Images,” Journal of Biomedical Science and Engineering, vol. 17, no. 09, pp. 157–170, 2024, doi: 10.4236/jbise.2024.179012.
[33] PyTorch Contributors, PyTorch Documentation: torch.nn. CrossEntropyLoss, PyTorch, [Online]. Available:https://docs.pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html.
[34] N. Andriani, B. Warsito and R. Santoso, “Analisis Sentimen Aplikasi Microsoft Teams Berdasarkan Ulasan Google Play Store Menggunakan Model Neural Network Dengan Optimasi Adaptive Moment Estimation (ADAM),” Jurnal Gaussian, vol. 13, no. 1, 2024. [In Indonesian]
[35] M. S. Haqqi and B. Kusumoputro, “Komparasi Metode Optimasi Adam dan SGD dalam Skema Direct Inverse Control untuk Sistem Kendali Data Sikap dan Ketinggian Quadcopter,” ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika, vol. 10, no. 2, April. 2022. DOI: http://dx.doi.org/10.26760/elkomika.v10i2.458 [In Indonesian]
[36] E. Bartz, T. Bartz-Beielstein, M. Zaefferer, and O. Mersmann, Hyperparameter Tuning for Machine and Deep Learning with R. Singapore: Springer Nature Singapore, 2023. doi: 10.1007/978-981-19-5170-1.
[37] J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian Optimization of Machine Learning Algorithms.”
[38] Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016. Accessed: May 22, 2025. [Online]. Available: http://www.deeplearningbook.org
[39] H. Zhang, G. Li, J. Li, Z. Zhang, Y. Zhu, and Z. Jin, “Fine-Tuning Pre-Trained Language Models Effectively by Optimizing Subnetworks

Adaptively," Nov. 2022, [Online]. Available: http://arxiv.org/abs/2211.01642

[40] F. Nurqoulby, A. A. Arifiyanti and D. S. Y. Kartika, "Analysis Sentiment Of Users Internet Service Providers In Indonesia On Social Media X Using Support Vector Machine," Data Science: Journal of Computing and Applied Informatics, vol. 8, no. 2, pp. 88-95, Jul. 2024.

[41] S. Sathyanarayanan and B.R Tantri, "Confusion Matrix-Based Performance Evaluation Metrics," African Journal of Biomedical Research, vol. 27 no. 4s, pp 4023-4031, Nov. 2024.

[42] F. S. Mulyo, "Building a Sentiment Classification Model Using IndoBERT", Medium, Dec 26, 2024. [Online]. Available: https://medium.com/%40fadilsatriomulyo/building-a-sentiment-classification-model-using-indobert-22ba010a1257

[43] A. Muzakir, K. Adi, and R. Kusumaningrum, "Short Text Classification Based on Hybrid Semantic Expansion and Bidirectional GRU (BiGRU) Based Method to Improve Hate Speech Detection," International Information *and Engineering Technology Association*, vol. 37, no. 6, pp. 1471-1481, Dec 2023. https://doi.org/10.18280/ria.370611