Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871

Vol. 7, No. 3, June 2026, Page. 2746-2757
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5672

Improving RoBERTa Performance through Hyperparameter Optimization for Sentiment Analysis of Indonesian Tourism Reviews

Imamah[*1], Myo Thida[2], Fika Hastarita Rachman[3], Budi Dwi Satoto[4], Sri Herawati[5], Yeni Kustiyahningsih[6], Eka Mala Sari Rochman[7], Meita Lailatuz Zakiyah[8]

[1,4,5,6,8]Department of Information Systems, University of Trunodjoyo Madura, Bangkalan, Indonesia
[2]Department of Computer Science, University of Illinois, Chicago, USA
[3,7]Department of Informatics, University of Trunodjoyo Madura, Bangkalan, Indonesia

Email: [1]i2m@trunojoyo.ac.id

Received : Jan 29, 2026; Revised : Feb 1, 2026; Accepted : Feb 2, 2026; Published : Jun 15, 2026

Abstract

The performance of transformer models such as RoBERTa in sentiment classification is influenced by hyperparameter settings, especially the epoch and batch sizes. However, no previous study has examined the impact of changes in the number of epochs and batch sizes on the performance of each class in classification tasks, especially in Indonesian-language sentiment analysis of tourism reviews. Therefore, this study aims to fill this gap by analyzing the performance of RoBERTa and the impact of various hyperparameter settings on sentiment for each class. The dataset consists of 3,875 reviews from visitors to Lake Sarangan on Google Maps. The batch sizes used in this study are 8 and 16, and the epoch range is 2 to 4. There are three classes of sentiment: negative, neutral, and positive. The results demonstrate that increasing the batch size from 8 to 16 does not linearly improve model performance. The optimal combination of epoch=4 and batch size=8 achieved 91% accuracy, with significant improvements in recall and F1-score across all classes, especially in positive sentiment classification. This research offers valuable insights into fine-tuning RoBERTa for sentiment analysis in Indonesian contexts, providing recommendations for future sentiment analysis tasks in natural language processing.

Keywords : *Classification, Deep Learning, Sentiment Analysis, Text Mining, Touris Reviews.*

1. INTRODUCTION

Nowadays, sentiment analysis has become an important approach for understanding public opinion and user perceptions across a wide range of domains, including online marketplaces [1] and tourism attractions [2]. The tendency of consumers or users to express their opinions on online text-based platforms poses a challenge for obtaining suggestions for service improvement. However, manual sentiment classification is time-consuming and will delay service improvements. Hence, an automated approach using text mining within Natural Language Processing (NLP) is needed to address this issue. In addition, transformer-based models, such as BERT[3] and RoBERTa[4], are popular for improving the performance of automatic sentiment classification. Transformers are pre-trained models on large-scale data that have a better understanding of sentence context across many languages [5][6]. However, transformer models require fine-tuning to adapt to specific contexts, using relatively few epochs and small batch sizes [7].

In practice, the selection of hyperparameters, such as batch size and the number of epochs, significantly determines the model's performance. Any studies show that smaller batch sizes tend to yield better performance than larger ones on medium- or small-scale datasets [8]. The other study showed that using a batch size of 8 resulted in a higher F1-score than batch sizes of 16 or 32 when training BERT for software requirements classification [9]. In addition, reproducibility research on fine-

Vol. 7, No. 3, June 2026, Page. 2746-2757
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5672

tuning transformer models revealed that the stability of the fine-tuning process remains an issue and is influenced by hyperparameter choices, including epoch and batch size.

Although the literature has highlighted the importance of selecting hyperparameters such as batch size and epoch for fine-tuning transformer models, several gaps remain to be addressed. First, many studies have only explored hyperparameters in large models such as BERT or RoBERTa, and rarely examine the impact of epoch and batch size combinations on the performance of individual classes (negative, neutral, positive) in sentiment classification models. For example, existing studies demonstrate substantial variability in the performance of fine-tuned transformer models. Yet, they lack a detailed investigation of the effects of batch size and epoch variations on each-class performance. Second, in Indonesian domains or domains focused on local contexts, there is often a lack of cross-parameter exploration (epoch vs. batch size vs. subtask class). Hence, the adaptation of these hyperparameters still relies on general practice rather than specific empirical evidence. Third, although some studies have shown that small batch sizes can be more effective on small or medium-sized datasets, there are still few that compare the performance of small vs. large batch sizes with fixed epoch settings and attribute the differences to specific changes in individual classes, such as negative/positive recall rates.

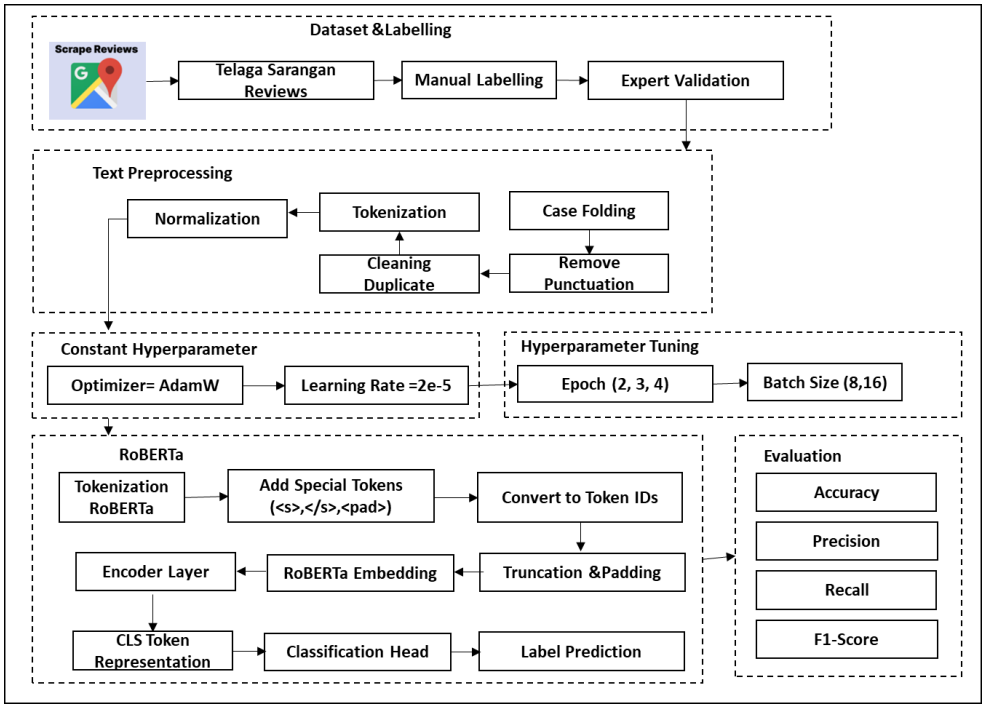

Figure 1. Architecture Systems.

For example, a study examining fine-tuning RoBERTa for mobile bug detection used batch sizes of up to 128 and epochs of up to 25. Still, it did not specifically attribute the influence of hyperparameters to each class in the classification task [10]. While previous research has explored hyperparameter optimization in large transformer models, few studies have systematically analyzed the combination of epoch and batch size in the context of Indonesian-language sentiment analysis for tourism reviews. This study aims to fill this gap by examining the performance of RoBERTa across different hyperparameter settings for each class sentiment (negative, neutral, and positive), and by explaining the performance differences between classes as a result of the combination of selected hyperparameters.

The format of this paper is as follows: Section 2 provides a detailed description of the materials and methods; Section 3 presents the results; Section 4 presents the discussions; and Section 4 presents the conclusions.

Vol. 7, No. 3, June 2026, Page. 2746-2757
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5672

2. METHOD

The architecture of this research is illustrated in Figure 1. The first step is collecting Sarangan Lake reviews; the next steps are manual labeling, followed by validation by an Indonesian expert, text preprocessing, and classification with RoBERTa. Hyperparameter tuning was used to assess the influence of each parameter on RoBERTa performance using accuracy, precision, recall, and F1-score evaluation metrics. The detailed stages of each process are described in the following subchapter.

2.1. Text Preprocessing

Text preprocessing is a process of converting an unstructured dataset into a structured dataset which is needed to prepare the text before proceeding with sentiment classification using machine learning. This process aims to clean and organize text, making it easier to convert into numerical data [11]. By preprocessing, important information can be separated from unnecessary terms, enabling a more effective and efficient understanding of the sentiments in the text [12]. The stages of preprocessing include labelling, cleaning, case folding, tokenization, normalization, stopword removal, and stemming. However, when using RoBERTa, it is not recommended to apply stopword removal or stemming, as these steps will remove the necessary features and affect the context understanding provided by tokenization and embeddings [13].

2.2. Robust Optimized BERT Pretraining Approach (RoBERTa)

Robustly Optimized BERT Pre-training Approach (RoBERTa) is the expansion of a modified BERT model through improvements in pre-training procedures [14]. RoBERTa was released by *Facebook AI Research* (FAIR) to improve the performance of the BERT model in NLP tasks. These modifications include increasing the batch size and the number of parameters, and applying *dynamic masking*, which allows the model to generalize better without predicting the next sentence.

RoBERTa has a 12-layer architecture with 768 *hidden states* per layer [15], aiming to overcome the limitations of transformer-based models through fine-tuning strategies [16]. RoBERTa is trained using *dynamic masking* of complete sentences without *Next Sentence Prediction* (NSP), *large* mini-batches, and *BPE byte-level* larger. As seen in Figure 2, RoBERTa works: the model receives a sentence as input, which is converted into tokens so it can serve as valid input to the model [17]. RoBERTa has three valid inputs: *input_ids*, attention_mask, and *token_type_ids* [18].

RoBERTa demonstrates strong generalization because it is trained on larger, more diverse data. Training on a broader, more varied dataset allows RoBERTa to understand a wide range of linguistic contexts in greater depth, thereby improving its ability to generalize to new data it has never seen before. Large amounts of data enable models to learn from more examples, ultimately reducing the likelihood of *overfitting* to patterns in the training data. A higher number of training iterations allows the model to capture the distribution of the underlying data better, improving its performance on various downstream tasks. The longer training time also allows RoBERTa to refine its internal representations, making it more accurate and effective at generalizing to new tasks [19].

The RoBERTa model employs dynamic masking, in contrast to the static masking. Before the data sequence is addressed, the data will be amplified by a factor of ten to increase the number of instances, with distinct words incorporated into each of the ten sequences. This aims to ensure that the model repeatedly processes the same sentence with different token masks. The model can encounter several masking patterns within the same sequence [20]. RoBERTa employs comprehensive sentence-level data as input to the model [21]. The data sample is continuous, and the chosen sample token length does not surpass 512 tokens per phrase. A specific document separator token is injected when the document boundary is traversed during sampling.

Vol. 7, No. 3, June 2026, Page. 2746-2757
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5672

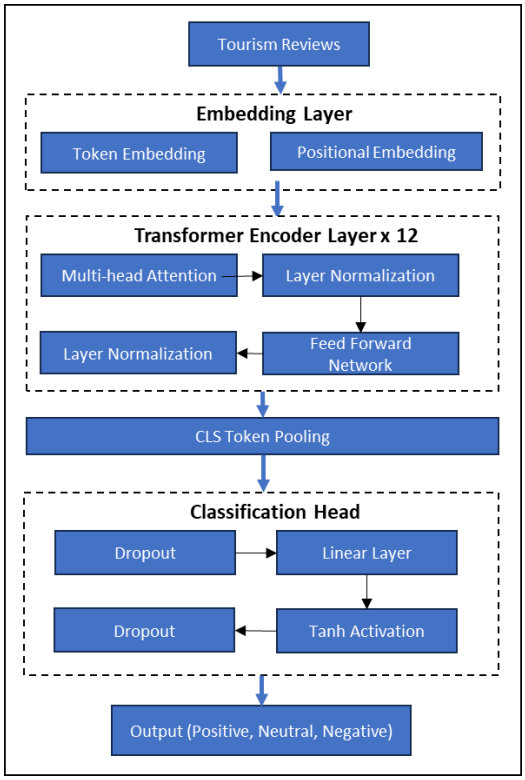

Figure 2. Architecture of RoBERTa

2.3. Hyperparameter RoBERTa

The RoBERTa model was trained for 125,000 steps with an effective batch size of approximately 8,000 sequences. It is doubtful that the batch size used in the RoBERTa model is far from ideal, yet it achieves better results than the base model. Hyperparameters are parameters that regulate how the model or learning algorithm operates during training, with the goal of finding the best combination of values that improve the model's performance in machine learning or deep learning [22]. Using appropriate hyperparameters can improve the effectiveness and accuracy of sentiment classification models. In this study, two parameters will be varied: Batch size, Epoch, and Learning Rate.

Batch size is a parameter that determines how much data is processed per training iteration in deep learning. If the number of batch sizes is too small, the model can become unstable, while a batch size that is too large risks memory overload [23].

Epochs are a parameter that controls how many times the model processes the entire dataset during training. The number of epochs also impacts the effectiveness of deep learning. If the number of epochs is too large, the model will overfit because each epoch becomes too focused on the training dataset [24]. If the number of epochs is too low, the model may not fit the data well enough, which is called underfitting.

In this study, the batch size and the number of epochs were systematically varied to examine their effects on model performance. The learning rate was kept constant at 2e-5 based on preliminary testing[25], and the model was fine-tuned for 2-4 epochs.

2.4. Evaluation

This study employed a confusion matrix to assess the performance of the Robustly Optimized BERT Pre-training Approach (RoBERTa). The accuracy metric assesses the model's ability to predict correct labels relative to expert annotations. Along with accuracy, the metric assessments used in this study include precision, recall, and F1 scores. The precision value denotes the ratio of correct predictions for a class to the total predictions made for that class. High precision denotes a model with a low

Vol. 7, No. 3, June 2026, Page. 2746-2757
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5672

prediction error rate for the specified class. The recall value denotes the proportion of accurate predictions for a class relative to the total actual instances of that class. High recall signifies that the model can identify the majority of pertinent data. The F1-score is a statistic that integrates precision and recall. This technique is advantageous for addressing class imbalance in a dataset. A high F1-score indicates that the model performs well in terms of predictive accuracy and comprehensiveness.

3. RESULT

This section describes the dataset used in the research, as well as the application of the method and its results**.**

3.1. Dataset

The dataset was collected by scraping visitor reviews of Lake Sarangan on Google Maps, yielding 3,875 reviews. The labeling process was conducted by an expert in Indonesian, who classified the reviews into three categories: positive, negative, and neutral. The label distribution consists of 2,617 positive, 717 negative, and 335 neutral samples.

Table 1. Sample review review of Telaga Sarangan

Ulasan	Process
Tempat tenang yang cocok untuk duduk dan menikmati pemandangan jln-jln mengelilingi telaga juga sangat disarankan sambil berfoto dengan background telaga. (a calm place that is suitable for sitting and enjoying the view walking around the lake is also highly recommended while taking photos with the lake background)	Labelling Positive
tempat tenang yang cocok untuk duduk dan menikmati pemandangan jalanjalan mengelilingi telaga juga sangat disarankan sambil berfoto dengan background telaga	Case Folding
['tempat', 'tenang', 'yang', 'cocok', 'untuk', 'duduk', 'dan', 'menikmati', 'pemandangan', 'jalanjalan', 'mengelilingi', 'telaga', 'juga', 'sangat', 'disarankan', 'sambil', 'berfoto', 'dengan', 'background', 'telaga']	Tokenization
['tempat', 'tenang', 'yang', 'cocok', 'untuk', 'duduk', 'dan', 'menikmati', 'pemandangan', 'jalan', 'jalan', 'mengelilingi', 'telaga', 'juga', 'sangat', 'disarankan', 'sambil', 'berfoto', 'dengan', 'background', 'telaga', 'buatkan', 'tokenisasi']	Normalization

Before the dataset is used as input to RoBERTa, it is necessary to preprocess the review data to produce structured, noise-free data that improves data quality and reduces complexity. The results of the text preprocessing are shown in Table 1. Each review has a different word length as seen in Figure 3. The minimum review length is 1 word, while the maximum review length is 58 words. An example of a review containing only one word is 'recommended', whereas the review with the maximum length consists of 58 words is: "*Turing di mari enak gas lah ada perahu cepat muter danau 15rb 2x muter danau 200rb 3x muter pt pt aja biar jadi murah naek minimal 3 max 5 dah bagi rata jd terjangkau ada kuda muter 80rb bebek motor ada jg tp santuy bet jalan nya muterin air danau gas ajah perjalanan pegunungan cek kondisi kendaraan anda*"

(*Touring here is enjoyable. You can take a fast boat to go around the lake—15,000 IDR for two rounds and 200,000 IDR for three rounds. Just negotiate to get a cheaper price; the minimum is three people and a maximum of five, so the cost can be shared and become more affordable. There are also horses for riding around the area for 80,000 IDR. Pedal boats and motorboats are also available, but the atmosphere is relaxed. The road goes around the lake, and the journey passes through mountainous areas, so make sure to check your vehicle's condition*).

Vol. 7, No. 3, June 2026, Page. 2746-2757
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5672

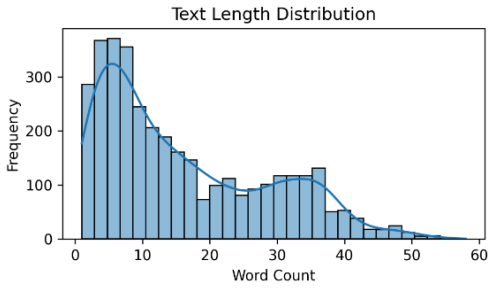

Figure 3. Text Length Distribution

After preprocessing, the next step is to split the data into training and testing sets. The training data is used to train the model, enabling it to learn patterns, while the testing data is used to evaluate the model's performance on previously unseen data. In this research, the data is split into 80% for training and 20% for testing.

3.2. RoBERTa Architexture

This study employed the RoBERTa architecture, comprising an embedding layer, 12 transformer encoder layers featuring multi-head self-attention, and a classification head consisting of two fully connected layers utilizing dropout and tanh activation. This framework enables the model to grasp long-range dependencies in text and efficiently translate contextual information into class predictions.

Table 4 presents the parameters of the RoBERTa architecture used for the text classification task in this study. Each layer has a specific function and a specific number of parameters that contribute to the model's ability to understand language context in depth. The first is the input layer, which serves only as a receiver of raw data and consists of ID tokens with a sequence length of 128. Because it only holds data without a learning process, this layer has no trainable parameters. The next layer is the embedding layer, which converts each token into a 768-dimensional vector representation. The number of parameters in this layer is V×768, where V denotes the vocabulary size, which is around 50,000, including both token and positional embeddings. This layer provides a basis for capturing the semantic representations of words.

After the embedding process, the input is passed through 12 Transformer encoder layers, which is the core of the RoBERTa architecture. Each encoder layer consists of a multi-head self-attention mechanism with 12 attention heads and a feed-forward network with an intermediate dimension of 3,072. This encoder block contains approximately 86–90 million parameters, making this layer the largest component of the model.

Table 2. Model Summary RoBERTa

Layer (Type)	Shape Output (with seq_len=128)	Param (≈)
Input	(B, 128)	0
Embeddings (token + position)	(B, 128, **768**)	~V×768
Encoder x12 (Transformer)	(B, 128, 768)	~86–90M
Pooling (CLS)	(B, 768)	0
Dropout	(B, 768)	0
Dense 768→768	(B, 768)	**590,592**
Dropout	(B, 768)	0
Output Dense 768→3	(B, 3)	**2,307**

The self-attention mechanism enables the model to capture both short- and long-range contextual dependencies in parallel, allowing each token representation to incorporate rich contextual information.

Vol. 7, No. 3, June 2026, Page. 2746-2757
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5672

Finally, the encoded representations are summarized using the [CLS] pooling strategy, where the representation of the first token is used as a global summary of the input sentence. A dropout layer with no parameters is then applied to prevent overfitting. After that, the Dense layer 768→768 (590,592 parameters) transforms the representation to be more discriminative. A tanh activation is added to introduce non-linearity, followed by a second dropout for additional regularization. The final stage is the Dense Output 768→3 (2,307 parameters), which converts the representation vector into three-class logits for sentiment classification. Although the classification head has relatively few parameters, it strongly determines the final prediction outcome. In total, this architecture has about 125 million parameters, with the majority in the embedding and encoder. The combination of powerful contextual representations and efficient classification heads makes RoBERTa particularly effective for natural language understanding and advanced text classification.

3.3. Result Analysis

After applying the RoBERTa model for sentiment analysis of Sarangan Lake, a comparative evaluation of model performance will be conducted to assess the differences between the results before and after hyperparameter tuning.

3.3.1. RoBERTa Without Hyperparameter Tuning

The first test was carried out with epoch 1 and a batch size of 8. This means the model performs a single iteration over the entire dataset. If batch_size = 8, the model processes 8 data samples simultaneously in a single forward and backward pass. One forward and backward pass means the model processes one batch of data (e.g., 8 samples at once), generates predictions, calculates the error (loss), and then updates its weights based on the error. So if there are 800 data samples and batch size = 8, then in one epoch the model will process 100 batches, since each batch processes 8 data samples. The model also updates its weight 100 times per epoch. The performance of RoBERTa without hyperparameter tuning is shown in Table 3. The recall is higher than the accuracy, precision, and F1 score on negative data, indicating that the model is good at predicting negative labels. Lower precision means that all data is predicted as negative labels, even though not all of them are completely negative. A lower F1-score suggests the model struggles to maintain a balance between precision and recall for negative labels. Although this model has a recall of 87% for negative labels, it only achieves a total recall of 82%. The accuracy, precision, and F1 score also reach 82%. It indicates that the model performs well for one class but not for another.

Table 3. RoBERTa without hyperparameter tuning

Label	Precision	Recall	F1-score
Negative	0.83	0.87	0.85
Neutral	0.85	0.79	0.82
Positive	0.79	0.81	0.80
Accuracy			**0.82**
Macro Avg	0.82	0.82	0.82
Weighted Avg	0.82	0.82	0.82

3.3.2. RoBERTa with Hyperparameter Tuning

In the previous evaluation, we found that RoBERTa's accuracy without hyperparameter tuning was 82%. The next stage is to evaluate the impact of hyperparameter tuning on model performance. The parameters tuned in this study were only the epoch and batch size, with values [2, 3, 4] and [8, 16], respectively. RoBERTa is a transformer trained on a very large amount of data, so it already has an understanding of the language from the beginning. Therefore, RoBERTa requires only simple fine-tuning and does not train the model from scratch. So the number of epochs used is usually small because

Vol. 7, No. 3, June 2026, Page. 2746-2757
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5672

the model adapts quickly and performance stabilizes quickly. If the epoch is too large, there is a risk of overfitting and excessive computational resources. Therefore, RoBERTa is generally trained for only 2–5 epochs to achieve optimal results.

Table 4. Hyperparameter epoch= 2 and batch size 8 and 16

Label	**Precision**		**Recall**		**F1-score**	
Batch	**8**	**16**	**8**	**16**	**8**	**16**
Negative	0.91	0.88	0.94	0.94	0.92	0.91
Neutral	0.84	0.90	0.96	0.93	0.90	0.91
Positive	0.92	0.89	0.77	0.80	0.84	0.84
Accuracy					**0.89**	**0.89**
Macro Avg	0.89	0.89	0.89	0.89	0.88	0.89
Weighted Avg	0.89	0.89	0.89	0.89	0.88	0.89

In Table 4. Tests were carried out with epoch=2 and batch size=8. The accuracy value was 0.7 times higher than with a model using epoch=1. However, the lowest recall value was obtained for the positive data, which reached only 0.77. This suggests that there are many prediction errors in the positive class. Even though the dataset has the most positive classes, it does not require balancing the class distribution. Based on Sanjaya et al, this condition is caused by a shift in the decision boundaries towards the minority class in SMOTE [26]. SMOTE generates synthetic samples by linearly interpolating between the minority class's nearest neighbors, without considering their proximity to the majority class. In datasets with high feature overlap, this approach can generate synthetic samples in the inter-class boundary region. As a result, the feature space previously dominated by the Positive class is contaminated by synthetic data from the Negative and Neutral classes, making the model more sensitive to minority classes. The direct impact of this phenomenon is an increase in the number of False Negatives in the Positive class, thereby lowering the Recall value of the Positive class. These findings confirm that although SMOTE balances class distributions quantitatively, the quality of data representation declines, and the model tends to overgeneralize to minority classes. The recall value on positive data rises to 0.80 when the batch size is increased to 16. Overall, precision, recall, and F1 Score showed better values at batch size=16.

Table 5. Hyperparameters epoch=3 and batch sizes 8 and 16

Label	**Precision**		**Recall**		**F1-score**	
Batch	**8**	**16**	**8**	**16**	**8**	**16**
Negative	0.94	0.84	0.94	0.95	0.94	0.89
Neutral	0.82	0.82	0.98	0.97	0.89	0.89
Positive	0.96	0.97	0.77	0.66	0.86	0.79
Accuracy					**0.90**	**0.86**
Macro Avg	0.91	0.88	0.90	0.86	0.90	0.86
Weighted Avg	0.91	0.88	0.90	0.86	0.90	0.86

Table 6. Hyperparameter epoch=4 and batch size 8 and 16

Label	**Precision**		**Recall**		**F1-score**	
Batch	**8**	**16**	**8**	**16**	**8**	**16**
Negative	**0.93**	0.88	**0.95**	0.97	**0.94**	0.92
Neutral	**0.86**	0.91	**0.97**	0.97	**0.91**	0.94
Positive	**0.94**	0.95	**0.80**	0.78	**0.86**	0.85
Accuracy					**0.91**	**0.91**
Macro Avg	0.91	0.91	0.90	0.91	0.90	0.91
Weighted Avg	0.91	0.91	0.91	0.91	0.90	0.91

Vol. 7, No. 3, June 2026, Page. 2746-2757
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5672

Furthermore, the dataset was tested at epoch=3 and batch sizes 8 and 16. The model's best performance is achieved at epoch=3 and batch size=8, as shown in Table 5. Although the accuracy value increased to 0.90, the recall value on positive data did not change (0.77). The accuracy value dropped by 4% to 0.86 when the batch size was 16. The recall for the negative class increased by 1%, whereas the recall for the neutral class decreased by 1%, and the recall for the positive class decreased by 11%.

Meanwhile, increasing the batch size does not yield a linear improvement in model performance, as it directly affects the model's learning process. The analysis of why it could happen is that when batch size = 8, weight updates are performed more frequently, allowing the model to adapt to data variations and, as a result, increasing accuracy. But when the batch size is increased to 16, the weight updates become less frequent, making the model less responsive. As a result, performance decreased, with accuracy dropping to 4% and recall in positive classes also decreasing significantly. This occurred because large batch sizes made the learning process less adaptive to data variation and distribution.

In subsequent trials, the epoch value was increased to 4, with batch size values of 8 and 16. The results of the study are shown in Table 6. The recall value for positive data, which reached 0.66, has increased to 0.80 with a batch size of 8. The accuracy value also increased by 1% to 0.91. This value represents the best accuracy across all hyperparameters. Although in batch 16 the accuracy reached 0.91, the recall only reached 0.78. As previously explained, a declining recall indicates that the model is not able to predict well on positive data.

Based on the research conducted, it can be observed that changes in the epoch and batch sizes have a significant effect on RoBERTa's performance. Increasing the batch size from 8 to 16 does not necessarily lead to an improvement in performance. In some scenarios, the F1-score accuracy increases, but in others, certain metrics decrease, especially in the tense class, such as the positive class. One of the main causes is the difference in learning nature between small and large batches. When using small batches such as 8, the learning process involves *a higher noise gradient*. This noise naturally acts as a form of regularization, helping the model be more stable and generalizable to new data. Small batch sizes also tend to produce a better precision and recall across classes. This is due to the more varied gradient characteristics during training, which prevent the model from converging too quickly to a local solution that benefits only the majority class. This gradient variation encourages the model to learn more diverse patterns across each class, including the minority class, thus improving overall classification performance.

Although using a small batch size results in slower per-epoch training times than a large batch size, the final results show better precision and recall. Large batch sizes produce more stable gradients, but they force the model to focus on dominant patterns in the data, thereby ignoring the characteristics of certain classes and leading to performance imbalances between them. Thus, using a small batch size increases the model's sensitivity to data variation and class distribution, albeit at the cost of slower training time.

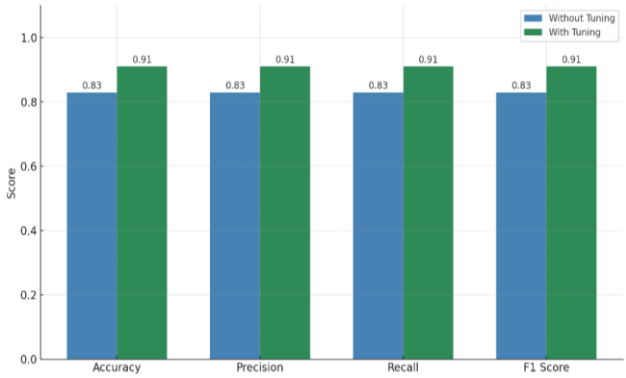

Figure 3. Model Performance Comparison: Without vs With Hyperparameter Tuning

Vol. 7, No. 3, June 2026, Page. 2746-2757
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5672

Conversely, when the batch size is increased to 16, the gradient becomes more stable and "clean". In some cases, this is advantageous, as the model can achieve more consistent, faster parameter updates towards convergence. If the dataset is relatively balanced and the patterns between classes are clear, this condition actually helps improve performance. However, in datasets with more diverse or unbalanced distributions, increasing the batch size can cause the model to lose sensitivity to minor variations, especially for more difficult-to-recognize classes. As a result, while accuracy can remain high, even the recall on positive classes tends to decline. The results of the comparison of RoBERTa performance with and without hyperparameter tuning are shown in Figure 2.

Based on Figure 3, the accuracy of the RoBERTa model increased by 8% after hyperparameter tuning. Furthermore, the RoBERTa model with the highest accuracy has been deployed using Streamlit, as shown in Figure 4. The application was tested to detect new data not contained in the training or testing data, and it succeeded in correctly predicting with a confidence level of 90.98%. This application can be developed and used to help tourist attraction owners see the polarity of sentiment for the tourist attractions they manage, thereby improving their service and management.

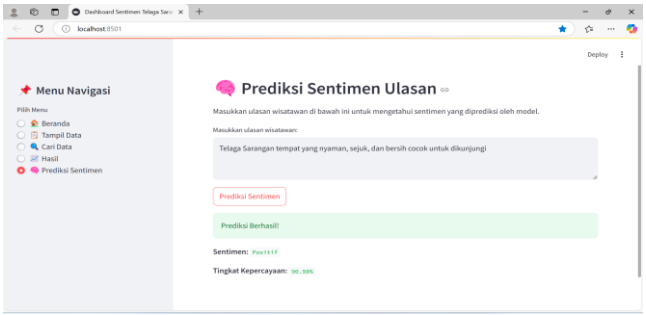

Figure 4. Tourism Sentiment APP with RoBERTa

4. DISCUSION

The results of the experiment show that hyperparameter optimization can improve the performance of the RoBERTa model for sentiment analysis in Indonesian tourism reviews. Using the evaluation metrics of accuracy, precision, recall, and F1 Score, it was found that RoBERTa Performance increased from 83% to 91% for all metrics. These findings are in line with previous studies that stated that the number of epochs and batch sizes influences transformer-based models [10]. Usman et al. used batch sizes (8, 32, 64, 128) and epochs (3, 9, 20, 25), and another study conducted by Wang et al. used a combination of epochs (3, 4, 5) and batch sizes (16, 32) [27]. The results of our study are in line with those of Usman and Wang, showing that smaller batch sizes (batch size = 8) with a moderate number of epochs (≈2–5) on average produce better results for the Roberta model. Likewise, combining these two values also improves Roberta's performance on the Indonesian-tourism reviews dataset. This is because smaller batches introduce stochasticity that benefits optimization and allow more parameter updates per epoch, which helps generalization on a medium-sized corpus [10].

Another finding in our study is that model performance is influenced by how imbalanced classes are handled, especially by the selection of an appropriate balancing method. In this study, the SMOTE balancing method was applied, resulting in decision rounds in which the recall of the positive class (the original majority class) was lower than that of the class whose sample size was increased through SMOTE over-sampling. Based on research by Sanjaya et al. [26], this phenomenon is called decision boundaries, and they state that SMOTE-TOMEK is better than SMOTE for handling imbalances without causing decision boundaries. In addition, class weighting, targeted data augmentation for underperforming classes, calibration of decision thresholds, or the use of task-specific adapters are recommended to further improve minority-class recall without sacrificing precision [28]. Overall, our

Vol. 7, No. 3, June 2026, Page. 2746-2757
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5672

results confirm that hyperparameter optimization, particularly epoch tuning and batch size, provides replicable benefits to improve RoBERTa performance on Indonesian domain data.

5. CONCLUSION

Based on the research results, it can be concluded that RoBERTa's performance is highly dependent on the balance of hyperparameters. If the number of epochs is too large, the model will accelerate overfitting. A large batch size can reduce the model's sensitivity to minority classes and data variations. Therefore, the combination of low epochs (2–4) and batch sizes of 8 and 16, according to this research, is the optimal fine-tuning configuration for RoBERTa. This combination provides a balance between learning speed, model stability, and generalizability to new data. The best RoBERTa model was obtained with epoch=4 and batch size=8, yielding an 8% increase in accuracy compared to the model's performance before hyperparameter tuning.

This study's findings can be applied to real-time sentiment analysis on tourism platforms, enabling businesses to gather instant customer feedback. Future research may explore applying similar hyperparameter tuning strategies to multi-lingual sentiment analysis models, such as XLM-RoBERTa, and implement a balancing model, except SMOTE, to avoid decision boundaries in the majority class.

ACKNOWLEDGEMENT

The authors would like to thank Universitas Trunodjoyo Madura (UTM) for supporting this research through the Research Group Grant 2022, under Contract Numbers SP DIPA-023.17.2.677535/2022.

REFERENCES

[1] P. Sharma, P. Tomar, and D. Mukherjee, "Sentiment Analysis on Amazon Dataset using Transfer Learning," in *2022 International Conference on Fourth Industrial Revolution Based Technology and Practices (ICFIRTP)*, 2022, pp. 160–165, doi: 10.1109/ICFIRTP56122.2022.10059413.

[2] I. Imamah, H. Husni, E. M. Rohman, I. O. Suzanti, and F. A. Mufarroha, "Text mining and Support Vector Machine for Sentiment Analysis of tourist Reviews in Bangkalan Regency," *J. Phys. Conf. Ser.*, vol. 1477, no. 2, pp. 0–6, 2020, doi: 10.1088/1742-6596/1477/2/022023.

[3] R. Qasim, W. H. Bangyal, M. A. Alqarni, and A. Ali Almazroi, "A Fine-Tuned BERT-Based Transfer Learning Approach for Text Classification," *J. Healthc. Eng.*, vol. 2022, 2022, doi: 10.1155/2022/3498123.

[4] D. T. Putra and E. B. Setiawan, "Sentiment Analysis on Social Media with Glove Using Combination CNN and RoBERTa," *J. RESTI*, vol. 7, no. 3, pp. 457–563, 2023, doi: 10.29207/resti.v7i3.4892.

[5] W. Suwarningsih, R. A. Pratama, F. Y. Rahadika, and M. H. A. Purnomo, "RoBERTa: language modelling in building Indonesian question-answering systems," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 20, no. 6, pp. 1248–1255, 2022, doi: 10.12928/TELKOMNIKA.v20i6.24248.

[6] Y. Liu *et al.*, "Roberta: A robustly optimized bert pretraining approach," *arXiv Prepr. arXiv1907.11692*, 2019.

[7] U. K. Immanuel, "Sentiment Analysis of Public Opinions Regarding Ideas of Presidential Candidates in YouTube Video Comments with Robustly Optimized BERT Pretraining Approach," 2024.

[8] N. A. Semary, W. Ahmed, K. Amin, P. Pławiak, and M. Hammad, "Improving sentiment classification using a RoBERTa-based hybrid model," *Front. Hum. Neurosci.*, vol. 17, pp. 1–10, 2023, doi: 10.3389/fnhum.2023.1292010.

[9] A. A. Azhari, Y. Sibaroni, and S. S. Prasetiyowati, "Detection of Indonesian Hate Speech in the Comments Column of Indonesian Artists' Instagram Using the RoBERTa Method," *JIPI*, vol. 8, no. 3, pp. 764–773, 2023, doi: 10.29100/jipi.v8i3.3898.

[10] M. Usman *et al.*, "Fine-Tuned RoBERTa Model for Bug Detection in Mobile Games: A

Vol. 7, No. 3, June 2026, Page. 2746-2757
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5672

Comprehensive Approach," *Computers*, vol. 14, no. 4, p. 113, 2025.
[11] D. M. Rathod, K. Patel, A. J. Goswami, S. Degadwala, and D. Vyas, "Exploring Drug Sentiment Analysis with Machine Learning Techniques," in *2023 International Conference on Inventive Computation Technologies (ICICT)*, 2023, pp. 9–12, doi: 10.1109/ICICT57646.2023.10134055.
[12] C. P. Chai, "Comparison of text preprocessing methods," *Nat. Lang. Eng.*, vol. 29, no. 3, pp. 509–553, 2023.
[13] M. Pfeifer and V. P. Marohl, "CentralBankRoBERTa: A fine-tuned large language model for central bank communications," *J. Financ. Data Sci.*, vol. 9, p. 100114, 2023.
[14] F. I. Kurniadi, N. L. P. S. P. Paramita, E. F. A. Sihotang, M. S. Anggreainy, and R. Zhang, "BERT and RoBERTa Models for Enhanced Detection of Depression in Social Media Text," *Procedia Comput. Sci.*, vol. 245, pp. 202–209, 2024, doi: https://doi.org/10.1016/j.procs.2024.10.244.
[15] Z. Huang, T. Ban, and Y. Zhang, "A novel approach for malicious URL detection using RoBERTa and sparse autoencoder," *J. Inf. Secur. Appl.*, vol. 94, no. September, p. 104214, 2025, doi: 10.1016/j.jisa.2025.104214.
[16] A. Jabbary, R. Boostani, F. A. Alenizi, and A. Salih, "RoBERTa , ResNeXt and BiLSTM with self-attention : The ultimate trio for customer sentiment analysis," *Appl. Soft Comput.*, vol. 164, no. November 2023, p. 112018, 2024, doi: 10.1016/j.asoc.2024.112018.
[17] R. Mohawesh, H. Bany, Y. Jararweh, and M. Alkhalaileh, "Fake review detection using transformer-based enhanced LSTM and RoBERTa," *Int. J. Cogn. Comput. Eng.*, vol. 5, no. June, pp. 250–258, 2024, doi: 10.1016/j.ijcce.2024.06.001.
[18] M. A. A. Yani and W. Maharani, "Analyzing cyberbullying negative content on twitter social media with the RoBERTa method," *JINAV J. Inf. Vis.*, vol. 4, no. 1, pp. 61–69, 2023.
[19] L. Yang, J. Wang, and W. Qiu, "RoBERTa-based Multi-Feature Integrated BiLSTM and CNN Model for Ceramic Review Analysis," *IEEE Access*, 2025.
[20] O. Ozyegen *et al.*, "Classifying multi-level product categories using dynamic masking and transformer models," *J. Data, Inf. Manag.*, vol. 4, no. 1, pp. 71–85, 2022.
[21] M. Straka, J. Náplava, J. Straková, and D. Samuel, "RobeCzech: Czech RoBERTa, a monolingual contextualized language representation model," in *International conference on text, speech, and dialogue*, 2021, pp. 197–209.
[22] X. D. J. Nguyen and Y. A. Liu, "Methodology for hyperparameter tuning of deep neural networks for efficient and accurate molecular property prediction," *Comput. \& Chem. Eng.*, vol. 193, p. 108928, 2025.
[23] X. Piao, D. Synn, J. Park, and J. K. Kim, "Enabling Large Batch Size Training for DNN Models Beyond the Memory Limit While Maintaining Performance," *IEEE Access*, vol. 11, no. September, pp. 102981–102990, 2023, doi: 10.1109/ACCESS.2023.3312572.
[24] J.-Y. Ong, L.-Y. Ong, and M.-C. Leow, "Addressing overfitting in comparative study for deep learning-based classification," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 23, no. 3, pp. 673–681, 2025.
[25] I. Gambo, R. Massenon, R. Oluwaseun, S. Agarwal, and W. Pak, "Identifying and resolving conflict in mobile application features through contradictory feedback analysis," *Heliyon*, vol. 10, no. 17, p. e36729, 2024, doi: 10.1016/j.heliyon.2024.e36729.
[26] U. P. Sanjaya *et al.*, "XGBoost for Educational Performance: Comparing SMOTE and SMOTE-TOMEK on Imbalanced Data," *ICCMS (Proceeding Int. Collab. Conf. Multidiscip. Sci.*, vol. 2, no. 2, pp. 271–281, 2024.
[27] S. Wang, "Development of an automated transformer-based text analysis framework for monitoring fire door defects in buildings," *Sci. Rep.*, vol. 15, no. 1, pp. 1–22, 2025, doi: 10.1038/s41598-025-27648-9.
[28] N. Bölücü, M. Rybinski, X. Dai, and S. Wan, "An adaptive approach to noisy annotations in scientific information extraction," *Inf. Process. Manag.*, vol. 61, no. 6, p. 103857, 2024, doi: 10.1016/j.ipm.2024.103857.