Jurnal Teknik Informatika (JUTIF)
Vol. 5, No. 5, October 2024, pp. 1443-1450

DOI: https://doi.org/10.52436/1.jutif.2024.5.5.2396
p-ISSN: 2723-3863
e-ISSN: 2723-3871

LEARNING RATE AND EPOCH OPTIMIZATION IN THE FINE-TUNING PROCESS FOR INDOBERT'S PERFORMANCE ON SENTIMENT ANALYSIS OF MYTELKOMSEL APP REVIEWS

Muhammad Naufal Zaidan[*1], Yuliant Sibaroni[2], Sri Suryani Prasetiyowati[3]

[1,2,3]Informatics, School of Computing, Telkom University, Indonesia
Email: [1]zidanaufal@student.telkomuniversity,ac,id, [2]yuliant@telkomuniversity.ac.id,
[3]srisuryani@telkomuniversity.ac.id

(Article received: June 27, 2024; Revision: July 15, 2024; published: October 29, 2024)

Abstract

With the advancement of the digital era, the growth of mobile applications in Indonesia is rapidly increasing, particularly with the MyTelkomsel app, one of the leading applications with over 100 million downloads. Given the large number of downloads, user reviews become crucial for improving the quality of services and products. This study proposes a sentiment analysis approach utilizing the Indonesian language model, IndoBERT. The main focus is on optimizing the learning rate and epochs during the fine-tuning process to enhance the performance of sentiment analysis on MyTelkomsel app reviews. The IndoBERT model, trained with the Indo4B dataset, is the ideal choice due to its proven capabilities in Indonesian text classification tasks. The BERT architecture provides contextual and extensive word vector representations, opening opportunities for more accurate sentiment analysis. This study emphasizes the implementation of fine-tuning with the goal of improving the model's accuracy and efficiency. The test results show that the model achieves a high accuracy of 96% with hyperparameters of batch size 16, learning rate 1e-6, and 3 epochs. The optimization of the learning rate and epoch values is key to refining the model. These results provide in-depth insights into user sentiment towards the MyTelkomsel app and practical guidance on using the IndoBERT model for sentiment analysis on Indonesian language reviews.

Keywords: *Epoch, Fine-tuning, IndoBERT, Learning rate, MyTelkomsel, Sentiment Analysis*

1. INTRODUCTION

In this digital era, mobile applications have become one of the main tools used by many people, driven by the widespread use of internet facilities worldwide, particularly in Indonesia. In the past two years, 66.48 percent of the Indonesian population accessed the internet in 2022, and 62.10 percent in 2021 [1]. As internet usage increases, mobile applications are becoming increasingly popular among smartphone users, providing convenience in various aspects of life. Among the many applications that facilitate users, one is the MyTelkomsel application.

MyTelkomsel is a provider service application that facilitates users in paying mobile bills, providing access to data and credit information, and enhancing service usage visibility. By offering all these features in one application, MyTelkomsel creates better connectivity between users and service providers. According to research [2] The MyTelkomsel application has been downloaded by 100 million users and has a 4.5 star rating on Google Play Store. With so many users of this application, an accurate method is needed to improve the quality of MyTelkomsel's service. In addition, the frequent updates and new features added to the MyTelkomsel app indicate its growing importance and user engagement in the digital services sector [3].

To understand user sentiment towards the MyTelkomsel application, sentiment analysis methods can be used to analyze the feelings and opinions of the public towards an entity [4], In this context, the entity is the MyTelkomsel application. Research [5] reveals that sentiment analysis is a model that clearly identifies and distinguishes between positive and negative reviews. There are two main approaches often implemented in sentiment analysis: the machine learning approach and the lexicon-based approach [6]. The machine learning approach involves training algorithms on labeled data to predict sentiment, while the lexicon-based approach relies on predefined dictionaries of positive and negative words [7]. Several machine learning approaches are used in sentiment analysis. In previous research [8] one of the architectures from Transformer, IndoBERT, was used. The test results showed that the IndoBERT model is suitable for Indonesian text classification, with the highest F1 Score for class 1 being 0.9246 compared to other methods. This is achievable because IndoBERT is the first monolingual BERT model for the Indonesian

language that has proven to achieve the best performance on most Natural Language Processing (NLP) tasks [9]. Several aspects influence evaluation results, such as F1-score, precision, accuracy, and recall. One of them is the process of selecting hyperparameter values during the fine-tuning process. This argument is related to research [10] where the classification accuracy of each machine learning model can be further optimized through hyperparameter tuning to achieve better accuracy than that obtained using the model's default hyperparameter values. Moreover, hyperparameter tuning can significantly affect the performance of deep learning models, making it a crucial step in model development [11]. For instance, selecting appropriate learning rates and epochs can lead to improved model performance and faster convergence [12].

In this study, the author will implement the IndoBERT model by optimizing the selection of hyperparameter values for learning rate and epoch in the fine-tuning process to present better evaluation results. Previous studies have shown that optimizing these parameters can lead to substantial improvements in model accuracy and efficiency [13]. The IndoBERT method is chosen because it is effective in handling Indonesian text classification, making it suitable for implementation on the MyTelkomsel customer review dataset taken from the Play Store, where most of the data is in Indonesian. The availability of extensive user review data in Indonesian provides a rich source for training and validating the model, enhancing its robustness and applicability [14]. Furthermore, using a model specifically designed for the Indonesian language ensures better understanding and processing of linguistic nuances and context [15].

2. RESEARCH METHOD

To achieve the objectives of this research, the author designed a general framework for sentiment analysis of MyTelkomsel application reviews on the Google Play Store using the IndoBERT model. Figure 1 illustrates the general design flow used in this study.

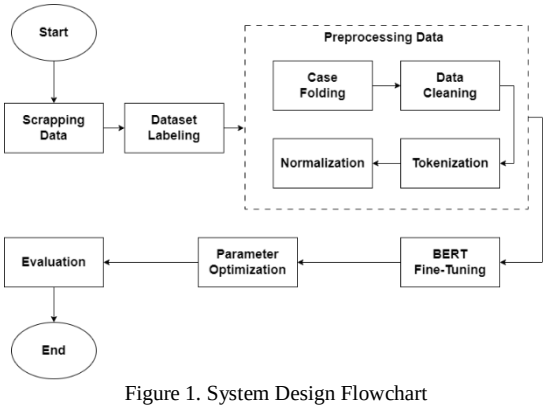

Figure 1. System Design Flowchart

2.1. Scrapping Data

Data scraping is the stage of data collection obtained from the Google Play Store in the form of user reviews of the MyTelkomsel application developed by Telkomsel. The scraping process works with Python using the Google Play Scraper library. The resulting dataset contains 9,000 review data. All collected data is then combined and stored in comma-separated values (.csv) format.

2.2. Dataset Labeling

Labeling is crucial because the IndoBERT model requires labeled data with known classifications. Annotation is done by assigning a negative label to reviews with sentences indicating dissatisfaction with the quality or service of the application, and a positive label to reviews that show user satisfaction. The labeling process is done manually to ensure an accurate match between the label and the review text, reducing errors and bias in the labeling process. Of the total 9,000 reviews, 36% were labeled positive and 64% were labeled negative, providing high-quality training data for the model.

2.3. Preprocessing Data

Data preprocessing is a stage carried out to tidy up unstructured datasets, making it easier for the data to be processed by the model. The preprocessing process is done using text processing techniques, which involve several stages: case folding, data cleaning, tokenization, and word normalization.

2.3.1. Case Folding

The case folding stage involves a series of steps to convert all characters to lowercase. This process is necessary because the data obtained does not always have a regular and consistent structure in the use of capital letters. Therefore, case folding is applied to standardize the use of capital letters by using the lower() function.

2.3.2. Data Cleaning

Data cleaning is used to remove attributes that affect the quality of data analysis by eliminating certain attributes such as numbers, symbols, URLs, usernames (@username), hashtags (#), excessive spaces, punctuation, emojis, and repeated characters in sentences [16], This process uses the Python RegEx library to identify the characters that need to be removed. Table 1 represents the characters that are cleaned.

Table 1 Proses Data Cleaning

Sentences	Deleted Character
Aplikasi MyTelkomsel sangat keren!!! 👍👍👍	!!! 👍👍👍
...	...
@user2711 Tolong bantu saya tidak bisa login 😣	@user2711 😣

2.3.3. Tokenization

This process aims to separate sentences into smaller units called "tokens." Tokens can be words, phrases, or characters depending on the type of tokenization used. The implementation of tokenization is done using the word_tokenize function provided by the Natural Language Toolkit (NLTK) library. The importance of tokenization in natural language processing and text analysis lies in its ability to break down text into tokens. This way, models or algorithms can more effectively understand and process the information contained in the text [17] [18].

2.3.4. Normalization

The normalization stage in text processing involves a series of steps to standardize or transform text into a more consistent form that adheres to established rules, referring to the use of NLTK with the Colloquial Indonesian Lexicon [19]. This involves transforming the dataset, which initially contains non-standard words such as "tdk," "bgs," "cpt," "ga," "enggak," "ngga," "gak," into their standard forms or correct spellings. Normalization is necessary to address variations in text representation, making it easier for further analysis or processing. Without normalization, the system might treat these words as having different meanings, even though they actually have the same meaning.

2.4. indoBERT Fine Tuning

IndoBERT is an Indonesian language model pre-trained based on the BERT architecture. This model is trained using a large-scale Indonesian dataset called Indo4B. The IndoBERT model is available in both base and large sizes and was trained using SentencePiece with byte pair encoding tokenizer as the vocabulary formation method [20].

Before implementing the IndoBERT model, the initial stage is dataset splitting. This process aims to evaluate the model's performance by dividing the data to be processed into three parts: training, testing, and validation. Among these divisions, the training data is the primary focus used to train the model.

Furthermore, before training the indoBERT model, the dataset needs to be adjusted to match the input format that BERT can process. To do this, BertTokenizer is used, a tokenization tool specifically designed to split sentences into tokens and generate inputs that match the required format [21]. This process is crucial because BERT uses a specific vocabulary depending on the model used. Thus, the tokenizer is responsible for preparing sentences so they can be processed as input representations by BERT, Figure 2 represents the BERT input.

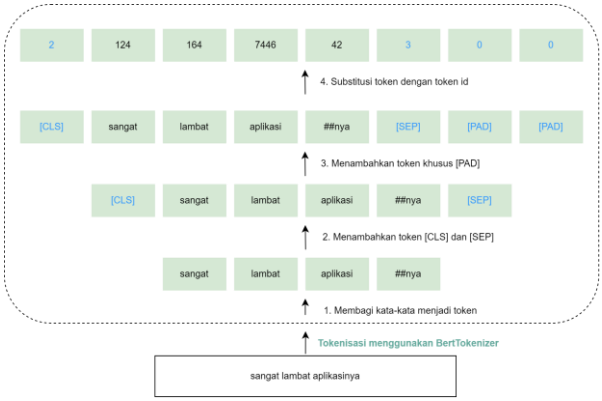

Figure 2. Ilustration of BERT Input

In the next stage, the Document Sentiment Dataset class is applied to load the data, where the subwords from the dataset can have different lengths. To enable parallel processing, standardizing the length of subwords through truncation or adding padding tokens is necessary. The implementation of the Document Sentiment Data Loader processes the list of subwords and sentiments, producing padded subwords, masks, and sentiments for the fine-tuning process as a training technique. Here, the model is not built from scratch but uses a pre-trained model, such as IndoBERT-base, and is partially updated for a new task. The BERT fine-tuning process, as illustrated in Figure 3, involves 12 encoders within the model, with primary attention on the output of the first token ([CLS]). This token is considered the average representation of the word tokens to obtain a sentence representation vector, which is used as input for the classifier. The classifier generates logits as rough probability predictions for sentence classification, and the softmax process converts the logits into predicted probabilities for a specific category in the context of sentiment analysis.

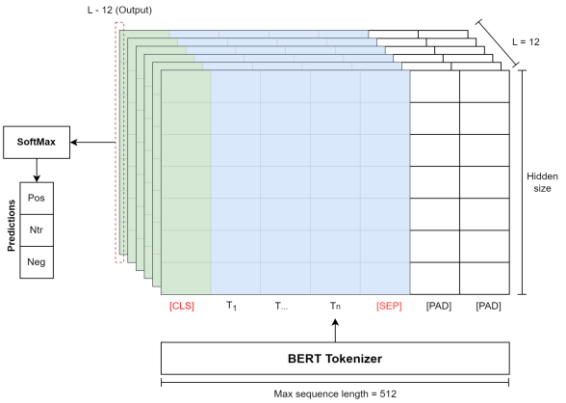

Figure 3. Sentiment Classification using the BERT Fine-Tuning Process

2.5. Parameter Optimization

Based on the explanation in the research [22], the BERT fine-tuning process involves adjusting hyperparameters. The hyperparameters that need attention in BERT training are batch size, epoch, and

learning rate. Batch size refers to the number of samples fed into the network before weight adjustments are made. The larger the batch size, the longer it takes to complete one batch. An epoch is the number of times the network sees the entire dataset, and one epoch occurs when all examples have passed through the network during both the forward pass and backward pass [23]. Learning rate plays a role in determining how much the weights in the neural network will change. The higher the learning rate, the faster the gradient moves towards the landscape [24]. In this study, the thing that is very concerned is the adjustment of hyperparameter weights in the learning rate and Epoch where the hyperparameter optimization process is carried out using the Adam Optimizer method.

2.6. Evaluasi Metrik

Classifier performance evaluation involves various metrics that help measure how well the model can predict the target class on data test, here are some common performance evaluation metrics for classifiers:

Accuracy, measures the overall percentage of correct predictions using the following formula:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} + \frac{n(n-1)x^2}{2!} \quad (1)$$

Precission, measures how much of the model's positive predictions are actually correct using the following formula:

$$Precission = \frac{TP}{TP+FP} \quad (2)$$

Recall, measures the extent to which the model is able to find positive class of the entire supposedly positive instance using the following formula:

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

F-1 Score, is a useful combination measure to balance both aspects between precision and recall using the following formula:

$$F1 = 2\,x\,\frac{Precissionn\ x\ Recall}{Precission+Recall} \quad (4)$$

In the evaluation stage, the model is tested using dataset testing. The results of model testing with test data will be shown with a confusion matrix to provide a detailed description as in Table 2 regarding the results of model predictions on test datasets by dividing them into four main categories, namely, True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

Table 2. Confusion Matrix

		Predicted class	
		Positive	**Negative**
True Class	Positive	TP	FN
	Negative	FP	TN

3. RESULT AND DISCUSSION

3.1. Preparation Data

This research began with a data crawling process that successfully collected 9000 MyTelkomsel App review data. the dataset contains a collection of user reviews. Table 3 shows the results of manual dataset labeling where negative reviews are labeled with a "0" and positive labels with a "1".

Table 3. Labeled Text Review

Text	**Label**
Kualitas tidak baik, sering error	0.0
...	...
Telkomsel semakin ok	1.0

Then the results of the labeling that have been carried out are 5794 data labeled negatively and 3206 data labeled positive, the results of which can be seen in Figure 4. And the results of the WordCloud visualization show the words that often appear in the review, as listed in Figure 5.

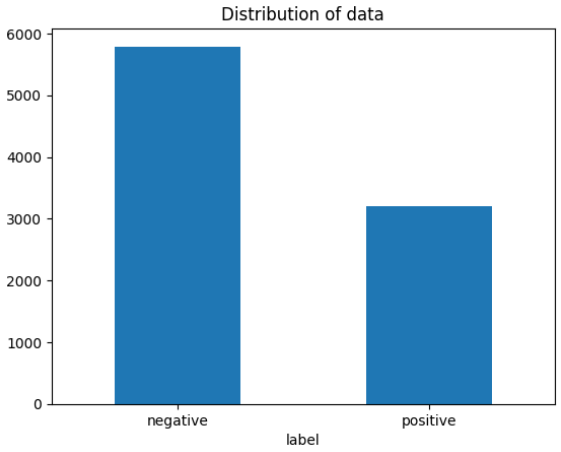

Figure 4. Results of Positive and Negative Review Data Distributions

Gambar 5. Word Cloud of MyTelkomsel App Reviews

The data preprocessing stage includes several important steps such as case folding, data cleaning, tokenization, and word normalization, to ensure the data is ready for further analysis.

Due to the imbalance between positive and negative labels, the author handles the data imbalance with an oversampling technique using RandomOverSampler. Figure 6 shows that after the resampling process, the positive and negative labeled review data has been balanced as opposed to the initial condition.

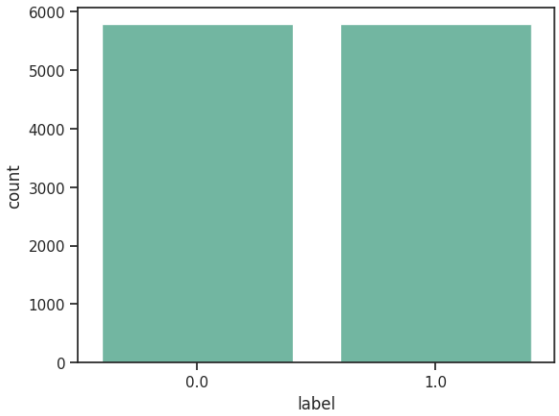

Figure 6. Review Data after Imbalance Handling Process

Data that has passed the preprocessing stage is divided into three parts: data train, data validation, and data test. In the data splitting process, the author uses a 70:20:10 comparison ratio which is used for training and validating the model.

3.2. Model Optimization

The model was then trained using fine-tuning techniques on the BERT (Bidirectional Encoder Representations from Transformers) architecture. The BERT model used was initialized using indobert-base-p2.

The model optimization process is performed using Adam's algorithm which is known for its convergence speed in training neuron networks. In addition, during training, the model performance is evaluated on the validation set to monitor and ensure that the model does not overfit the training data.

This study emphasizes the focus of parameter optimization on the fine tuning process, namely the Learning Rate and Epoch hyperparameters. The author searches for the best parameters using several trial combinations of each parameter value, and the results serve as comparisons to determine the combination of hyperparameters that yield the best accuracy.

a) Learning Rate Configuration

Configuration is done to understand the crucial role of the learning rate in the model training process. The learning rate determines how large a step the model takes in updating its weights at each iteration. The correct choice of learning rate significantly affects the convergence and final performance of the model. In this test, the author configured five ranges of learning rates with the epoch value set to 1.

Table 4. Learning Rate Parameter Configuration Results

Learning Rate	Accuracy
1e-5	0.9413
1e-6	0.8883
1e-7	0.6976
2e-5	**0.9414**
5e-5	0.9009

Based on Table 4, five configuration experiments have been conducted to determine the optimal learning rate parameter. The learning rate values tested were 1e-5, 1e-6, 1e-7, 2e-5, and 5e-5. Each learning rate provided a different accuracy level for the tested model. The learning rate of 2e-5 gave the highest accuracy of 0.9414, while the value of 1e-7 showed the lowest accuracy of 0.6976, indicating that the optimization step was too small for the model to achieve convergence.

From the results of the tests that have been carried out, the best learning rate configuration for this model is 2e-5 because it provides the highest accuracy compared to other learning rate values. Using the optimal learning rate is essential to ensure the model learns quickly and effectively.

b) Epoch Value Configuration

After obtaining the test results for the learning rate values, the next step is to configure the epoch values to achieve the highest accuracy. This configuration aims to find the perfect combination with the previously tested learning rate. The author used three ranges of epoch values and combined them with the five previously tested learning rate values.

Table 5 Average Train Accuracy Results

Learning Rate	1e-5	1e-6	1e-7	2e-5	5e-5
Epoch					
1	0.93	0.93	0.81	0.86	0.52
2	0.95	0.95	0.84	0.88	0.58
3	0.94	**0.96**	0.90	0.87	0.52

The test results are shown in Table 5. For a learning rate value of 1e-5, the highest accuracy of 0.95 was achieved at the 2nd epoch but slightly decreased to 0.94 at the 3rd epoch. This indicates that after reaching its peak at the 2nd epoch, the model started to overfit at the 3rd epoch. For a learning rate value of 1e-6, the highest accuracy of 0.96 was achieved at the 3rd epoch, with a consistent increase in accuracy from the 1st to the 3rd epoch. This shows that the model continued to learn effectively up to the 3rd epoch.

At a learning rate value of 1e-7, the highest accuracy of 0.90 was also achieved at the 3rd epoch, showing gradual and consistent improvement during the training process. For a learning rate value of 2e-5, the highest accuracy of 0.88 was reached at the 2nd epoch but slightly decreased to 0.87 at the 3rd epoch, also indicating possible overfitting after the 2nd epoch. Meanwhile, for a learning rate value of 5e-5, the highest accuracy of 0.58 was achieved at the 2nd epoch, with a significant drop at the 3rd epoch to 0.52, indicating that this learning rate value might be too high, causing the model to learn too quickly and then overfit.

From these results, it can be concluded that the combination of learning rate 1e−6 and the 3 epoch provides the highest accuracy of 0.96. This indicates that this combination is the best configuration for the model tested, resulting in the most optimal accuracy compared to other combinations tested. This combination ensures that the model is able to learn

effectively without overfitting too quickly, thus providing the best performance.

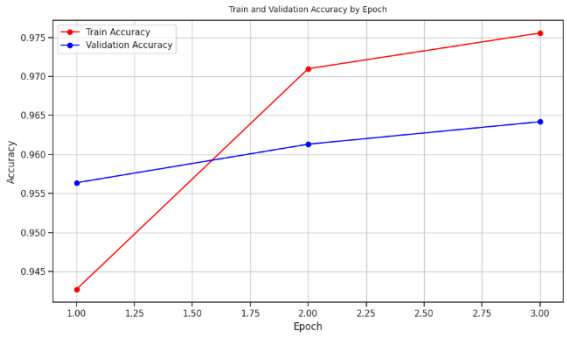

Figure 7. Accuracy Results Graph Based on Train and Validation Data

Figure 7 illustrates the progression of model accuracy during the training and validation phases over three epochs and a learning rate of 1e-6. The red line represents training accuracy, while the blue line represents validation accuracy. In the first epoch, the training accuracy was recorded at 94.3%, then increased significantly to 97.1% in the second epoch, and reached 97.6% in the third epoch. Meanwhile, the validation accuracy started at 95.6% in the first epoch, then gradually increased to 96.1% in the second epoch, and slightly increased to 96.4% in the third epoch.

The increase in accuracy indicates that the model is undergoing an effective learning process. The model demonstrates good generalization with consistently high and stable accuracy. These findings suggest that the model has adequate and reliable performance..

3.3. Metric Evaluation Result

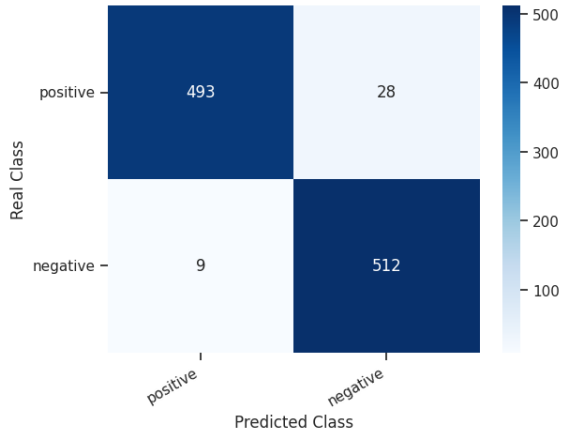

Figure 8. Evaluation Results of the Confusion Matrix

The evaluation metrics results in Figure 8 show that there are 493 positive class samples correctly predicted by the model as positive (TP). On the other hand, there are 28 samples that are actually negative but incorrectly predicted by the model as positive (FP).

Furthermore, there were 9 samples of positive classes that were incorrectly predicted by the model as negative (FN), indicating that there were some cases where the model failed to correctly identify positive samples, and there were 512 samples of negative classes that were correctly predicted by the model as negative (TN), indicating that the model was also quite reliable in recognizing negative samples.

Overall, the performance of this model can be considered good because the high TP and TN values indicate that the model can accurately distinguish between positive and negative classes. The low FP and FN values also indicate that the model makes few errors in its predictions. In other words, this model has good precision and sensitivity, making it reliable for the given classification task. These results provide confidence that the model is quite effective and efficient in handling complex classification tasks.

Table 6. Classification Report

	Precision	**Recall**	**F-1 Score**	**support**
Negative	0.95	0.98	0.97	521
Positive	0.98	0.95	0.96	521
Accuracy			0.96	1042
Macro Avg	0.97	0.96	0.96	1042
Weighted Avg	0.97	0.96	0.96	1042

From the classification report results, the model achieved high scores on both negative and positive sentiments. Table 6 shows overall that the classification model has excellent performance, with high metric values across all categories and classes, as well as a good balance between Precision and Recall, resulting in an optimal F1-Score. Its strong performance in both classes confirms its reliability in the given classification task.

4. DISCUSSION

The authors obtained sentiment analysis results for the MyTelkomsel application reviews using IndoBERT, with accuracy, precision, recall, and F1-score each at 96.3%. The parameters used include a learning rate of 1e-6, 3 epochs, a batch size of 16, Adam optimization, and Random Over Sampling to handle data imbalance. Compared to previous research, this study shows better results. Study [25]achieved an accuracy of 83.% and an F1-score of 91% with CNN algortihm based on IndoBERT model. Research [26] reached an accuracy of 84% with fine-tuning BERT on Google Play reviews. And the study [27] achieved an accuracy of 92% in multi-aspect sentiment analysis of the Bibit application using the IndoBERT model. Overall, IndoBERT with the right parameters and data balancing techniques is effective for this study, demonstrating improved performance compared to previous research. This study confirms the importance of selecting the appropriate parameters and handling imbalanced data to maximize model performance in natural language processing.

5. CONCLUSION

This study successfully analyzed the sentiment of MyTelkomsel application reviews using the IndoBERT model with very satisfactory results. The model achieved an accuracy, precision, recall, and F1-score of 96.3% each. By crawling data, 9000 review data were collected, consisting of 5794 negative reviews and 3206 positive reviews. After data preprocessing and balancing with RandomOverSampler, the data was divided into training, validation, and testing sets with a ratio of 70:20:10.

The model was trained using fine-tuning techniques on the IndoBERT-base-p2 architecture with optimization using the Adam algorithm. Various combinations of hyperparameter learning rate and epochs were tested to find the best configuration. Test results showed that a learning rate of 1e-6 and 3 epochs provided the highest accuracy of 96%.

Metric evaluation showed that the model correctly predicted 493 positive samples (TP) and 512 negative samples (TN), with only 28 negative samples incorrectly predicted as positive (FP) and 9 positive samples incorrectly predicted as negative (FN). These results indicate that the model has high precision and sensitivity, and is able to distinguish well between positive and negative classes..

Overall, this study confirms that IndoBERT, with proper parameter adjustments and effective data balancing techniques, can achieve excellent performance in sentiment analysis. This highlights the importance of selecting optimal parameters and handling imbalanced data to maximize model performance in natural language processing. These findings provide a strong foundation for further development in the sentiment analysis of application reviews in Indonesia.

REFERENCES

[1] T. Sutarsih and K. Maharani, *STATISTIK TELEKOMUNIKASI INDONESIA 2022*. Indonesia: Badan Pusat Statistik, 2023. Accessed: Nov. 15, 2023. [Online]. Available: https://www.bps.go.id/publication/2023/08/31/131385d0253c6aae7c7a59fa/statistik-telekomunikasi-indonesia-2022.html

[2] O. : Asep and I. Nugraha, "FAKTOR-FAKTOR YANG MEMPENGARUHI PENGGUNAAN SMARTPHONE DALAM AKTIVITAS BELAJAR MAHASISWA TEKNOLOGI PENDIDIKAN UNIVERSITAS NEGERI YOGYAKARTA FACTORS AFFECTING USE OF SMARTPHONE IN STUDENTS LEARNING ACTIVITIES."

[3] A. Morgan-Thomas, L. Dessart, and C. Veloutsou, "DIGITAL ECOSYSTEM AND CONSUMER ENGAGEMENT: A SOCIO-TECHNICAL PERSPECTIVE Author Details DIGITAL ECOSYSTEM AND CONSUMER ENGAGEMENT: A SOCIO-TECHNICAL PERSPECTIVE DIGITAL ECOSYSTEM AND CONSUMER ENGAGEMENT: A SOCIO-TECHNICAL PERSPECTIVE."

[4] A. Ligthart, C. Catal, and B. Tekinerdogan, "Systematic reviews in sentiment analysis: a tertiary study," *Artif Intell Rev*, vol. 54, no. 7, pp. 4997–5053, Oct. 2021, doi: 10.1007/s10462-021-09973-3.

[5] H. M. Keerthi Kumar, B. S. Harish, and H. K. Darshan, "Sentiment Analysis on IMDb Movie Reviews Using Hybrid Feature Extraction Method," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 5, pp. 109–114, 2019, doi: 10.9781/ijimai.2018.12.005.

[6] S. Pandya and P. Mehta, "A Review On Sentiment Analysis Methodologies, Practices And Applications," *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH*, vol. 9, p. 2, 2020, [Online]. Available: www.ijstr.org

[7] M. Darwich, S. A. Mohd Noah, N. Omar, and N. A. Osman, "Corpus-Based Techniques for Sentiment Lexicon Generation: A Review," *Journal of Digital Information Management*, vol. 17, no. 5, p. 296, Oct. 2019, doi: 10.6025/jdim/2019/17/5/296-305.

[8] J. Tugas, A. Fakultas, H. K. Putra, M. Arif Bijaksana, and A. Romadhony, "Deteksi Penggunaan Kalimat Abusive Pada Teks Bahasa Indonesia Menggunakan Metode IndoBERT."

[9] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," Nov. 2020, [Online]. Available: http://arxiv.org/abs/2011.00677

[10] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, "Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis," *Informatics*, vol. 8, no. 4, Dec. 2021, doi: 10.3390/informatics8040079.

[11] T. Yu and H. Zhu, "Hyper-Parameter Optimization: A Review of Algorithms and Applications," Mar. 2020, [Online]. Available: http://arxiv.org/abs/2003.05689

[12] S. Mohammadi and M. Chapon, "Investigating the performance of fine-tuned text classification models based-on BERT," in *2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science*

and Systems (HPCC/SmartCity/DSS), 2020, pp. 1252–1257.

[13] M. Wortsman *et al.*, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time." [Online]. Available: https://github.

[14] B. Ferdinandy *et al.*, "Challenges of machine learning model validation using correlated behaviour data: Evaluation of cross-validation strategies and accuracy measures," *PLoS One*, vol. 15, no. 7, Jul. 2020, doi: 10.1371/journal.pone.0236092.

[15] D. Sebastian, H. D. Purnomo, and I. Sembiring, "Bert for natural language processing in bahasa Indonesia," in *2022 2nd International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)*, 2022, pp. 204–209.

[16] M. Hosseinzadeh *et al.*, "Data cleansing mechanisms and approaches for big data analytics: a systematic study," *J Ambient Intell Humaniz Comput*, vol. 14, no. 1, pp. 99–111, Jan. 2023, doi: 10.1007/s12652-021-03590-2.

[17] D. Miller, "Leveraging BERT for Extractive Text Summarization on Lectures." [Online]. Available: https://github.com/dmmiller612/lecture-summarizer.

[18] C. Chantrapornchai and A. Tunsakul, "Information Extraction based on Named Entity for Tourism Corpus," Jan. 2020, doi: 10.1109/JCSSE.2019.8864166.

[19] F. Millstein, *Natural language processing with python: natural language processing using NLTK*. 2020.

[20] B. Wilie *et al.*, "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding," Sep. 2020, [Online]. Available: http://arxiv.org/abs/2009.05387

[21] A. Nayak, H. P. Timmapathini, K. Ponnalagu, and V. Venkoparao, *Domain adaptation challenges of BERT in tokenization and sub-word representations of Out-of-Vocabulary words*. Association for Computational Linguistics, 2020. [Online]. Available: https://github.com/

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Oct. 2018, [Online]. Available: http://arxiv.org/abs/1810.04805

[23] I. G. T. Isa and B. Junedi, "Hyperparameter Tuning Epoch dalam Meningkatkan Akurasi Data Latih dan Data Validasi pada Citra Pengendara," *Prosiding Sains Nasional dan Teknologi*, vol. 12, no. 1, p. 231, Nov. 2022, doi: 10.36499/psnst.v12i1.6697.

[24] H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and T. Zhao, "SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization," Nov. 2019, doi: 10.18653/v1/2020.acl-main.197.

[25] A. Zevana and D. Riana, "TEXT CLASSIFICATION USING INDOBERT FINE-TUNING MODELING WITH CONVOLUTIONAL NEURAL NETWORK AND BI-LSTM," *Jurnal Teknik Informatika (Jutif)*, vol. 4, no. 6, pp. 1605–1610, Jan. 2024, doi: 10.52436/1.jutif.2023.4.6.1650.

[26] K. S. Nugroho, A. Y. Sukmadewa, H. Wuswilahaken Dw, F. A. Bachtiar, and N. Yudistira, "BERT Fine-Tuning for Sentiment Analysis on Indonesian Mobile Apps Reviews," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Sep. 2021, pp. 258–264. doi: 10.1145/3479645.3479679.

[27] Serly Setyani, "Multi Aspect Sentiment Analysis of Mutual Funds Investment App Bibit Using BERT Method," *International Journal on Information and Communication Technology (IJoICT)*, vol. 9, no. 1, pp. 44–56, Jul. 2023, doi: 10.21108/ijoict.v9i1.718.