Evaluating learning rate effects on long short-term memory for Indonesian sentiment classification

Serly Eldina, Tekad Matulatan, Novrizal Fattah Fahmitra
Department of Informatics Engineering, School of Electrical and Informatics, Universitas Maritim Raja Ali Haji, Tanjungpinang, Indonesia

Article Info

Article history:

Received Jul 14, 2025
Revised Jan 12, 2026
Accepted Jan 30, 2026

Keywords:

Adaptive moment estimation
Early stopping
Learning rate
Long short-term memory
One-way analysis of variance
Sentiment classification

ABSTRACT

Hyperparameter optimization is a crucial process for enhancing the performance of deep learning models, particularly in the context of Indonesian sentiment classification. This study examines the impact of varying learning rates on a long short-term memory (LSTM) architecture trained with the adaptive moment estimation (Adam) optimizer. The dataset comprises 9,295 Indonesian comments automatically labeled by the Indonesian Bidirectional Encoder Representations from Transformers (IndoBERT) model. Stratified k-fold cross-validation was employed to maintain class balance during training. Learning curves were analyzed to evaluate convergence and identify potential overfitting, while early stopping was applied when performance improvements became insignificant. The one-way analysis of variance (ANOVA) test (p-adj = 0.000575 < 0.05) revealed significant differences among the learning rate variations. Post-hoc analysis indicated the learning rates of 0.0001, 0.001, and 0.002 differ significantly from 0.02. Descriptive statistics showed that a learning rate of 0.001 was the most optimal, achieving the highest validation accuracy while maintaining a relatively low variance. Evaluation across two data categories demonstrated that lower learning rates (0.0001 and 0.002) achieved the best accuracy, 78.71% on in-domain data, whereas higher learning rates (0.01 and 0.02) performed better on cross-domain data with 36% accuracy. These findings highlight the crucial role of learning rate selection in determining model stability and generalization capability.

Corresponding Author:

Serly Eldina
Department of Informatics Engineering, School of Electrical and Informatics
Universitas Maritim Raja Ali Haji
Tanjungpinang, Indonesia
Email: serlyeldina03@gmail.com

1. INTRODUCTION

Sentiment analysis on social media has become a prominent research area that leverages natural language processing (NLP) and computational linguistics to extract subjective information from textual data, thereby improving the understanding of public opinion, brand perception, and social dynamics [1], [2]. It is widely used to classify Indonesian text into three sentiment categories: positive, negative, and neutral [3]. One of the most widely used models for this purpose is the long short-term memory (LSTM), a type of recurrent neural network (RNN) that is capable of capturing long-term dependencies within sequential data, allowing it to perform effectively across a wide range of text processing tasks [4]-[6].

Learning rate optimization plays a crucial role in deep learning, as it significantly affects model stability and convergence. Among various optimization algorithms, the adaptive moment estimation (Adam) optimizer is one of the most popular, as it adaptively adjusts the learning rate using both momentum and gradient mean estimation, enabling faster and more stable convergence [7]-[9]. The learning rate itself is a critical hyperparameter that influences the stability and speed of the training process. A high learning rate increases the risk of overshooting the optimal solution, whereas a low learning rate leads to gradual and slower convergence [10]. Proper tuning of this parameter is vital to achieving optimal model performance and to mitigating overfitting or underfitting. Therefore, additional strategies such as early stopping and dropout are often applied to prevent these issues [11].

Previous studies have explored the implementation of the LSTM model [12], and the use of the Adam optimizer for Indonesian sentiment analysis [13]. For example, a study analyzing various learning rate configurations reported that different learning rate values (0.1, 0.01, 0.001) and epoch settings (300, 500, 1000) influenced model stability and accuracy in time-series prediction of chlorophyll-a concentration, with suboptimal learning rates causing training instability [14]. Similarly, a study on the CIFAR-10 dataset showed that a learning rate of 0.001 combined with a dropout rate of 0.5 yielded the best performance in addressing overfitting and underfitting issues [10]. Another study on the Indonesian Bidirectional Encoder Representations from Transformers (IndoBERT) sentiment analysis model concluded that variations in the Adam learning rate significantly influenced stability and accuracy. The optimal learning rate of 2e-5 achieved an accuracy of 94.14%, while an excessively low value of 1e-7 caused instability and reduced accuracy to 69.76% [15].

Although previous studies have examined the impact of learning rate variations, there remains a research gap. Based on our current knowledge, no study has specifically investigated the effect of learning rate variations on the LSTM model for Indonesian text classification using automatically labeled data generated by a fine-tuned IndoBERT model. Most prior works focused on time-series, image, or sentiment datasets without performing statistical significance tests on the results. Therefore, this study aims to evaluate the effect of different Adam learning rate configurations on the performance of a text classification model.

The model was trained using 9,295 Indonesian comments, and Stratified k-fold cross-validation was applied to ensure result reliability and to address class imbalance by maintaining proportional class distributions in each fold [16]. Furthermore, a one-way analysis of variance (ANOVA) test with a significance level of (p-adj < 0.05) was conducted to determine statistical significance, followed by a post-hoc test to assess differences among learning rate variations [17], [18]. The evaluation was conducted using a confusion matrix under two data scenarios to assess the model's generalization ability: in-domain (comments from the Sirekap application on the Play Store) and cross-domain (news headlines related to the "Makanan Bergizi Gratis (MBG)" topic) [19].

The main contributions of this study are as follows: (i) providing an empirical analysis of the effect of learning rate variations on the accuracy, stability, and generalization ability of the LSTM model and (ii) presenting the first study that examines the impact of learning rate variations on an Indonesian LSTM model automatically labeled using the IndoBERT model, with validation through a one-way ANOVA statistical test and evaluation conducted on both in-domain and cross-domain datasets. The remainder of this paper is organized as: section 2 describes the research methodology, section 3 presents the experimental results and analysis, and section 4 concludes the study and provides directions for future research.

2. METHOD

This study employs a sentiment classification approach to analyze textual data systematically. The research methodology is shown in Figure 1, which presents the process from data collection and preprocessing to model training and evaluation. The Figure 1 provides a clear overview of the workflow used in this study.

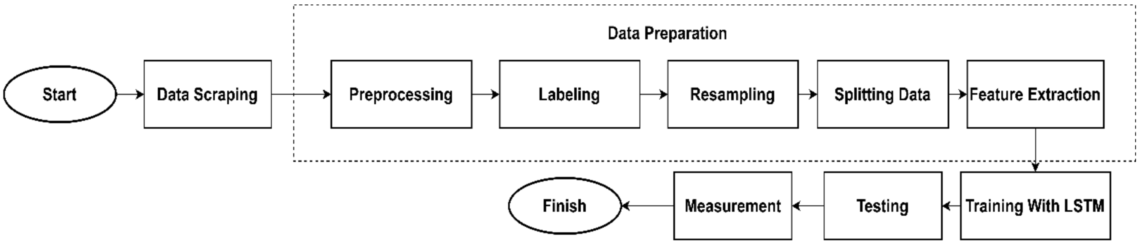


Figure 1. Research methodology

2.1. Data scraping

Data collection was performed through a web scraping technique using the Google Play Scraper [20], to systematically obtain user comments. In total, 19,936 entries were collected, focusing on “Most Relevant” reviews from the Sirekap application available on the Google Play Store.

2.2. Data preparation

The data preparation stage involves several essential processes to ensure that the dataset is suitable for analysis and modeling. This stage consists of preprocessing, labeling, feature extraction, and formatting, each contributing to improved data quality and model compatibility.

2.2.1. Preprocessing

Data preprocessing is a fundamental step in this study to ensure that the data used for analysis is clean, consistent, and properly prepared for further processing. This step is crucial in sentiment analysis, as it converts raw text into a structured format suitable for computational processing [21], [22]. The process generally involves several key operations, including redundancy removal [23], text cleaning, tokenization, stopword removal, case folding, and stemming [24], as well as handling missing values [23]. These preprocessing steps collectively improve the quality and reliability of the dataset for model training.

2.2.2. Labeling

The data labeling process was conducted using the IndoBERT model available on the Hugging Face platform, which facilitates community collaboration by providing open-source tools [25]. The pretrained model used for this task was mdhugol/indonesia-bert-sentiment-classification. Manual labeling was not performed due to limited linguistic expertise and the need to minimize annotator subjectivity. The IndoBERT model was chosen because it is a modern transformer-based model trained on a large-scale Indonesian corpus, enabling a deeper understanding of contextual meaning [26]. However, lexicon-based approaches, such as the Indonesian Sentiment Lexicon (InSet), have fundamental limitations because they rely on keyword matching within the lexicon and cannot capture semantic meaning patterns expressed in full sentences [27]. Valence aware dictionary and sentiment reasoner (VADER) and TextBlob require translation into English, which may result in the loss of the original meaning [28].

IndoBERT was used in this study to automatically label sentiments, creating three classes: negative, neutral, and positive. After the labeling process, a total of 17,152 text entries were obtained, consisting of 12,276 (71.6%) negative, 2,946 (17.2%) neutral, and 1,930 (11.2%) positive sentiments. The resulting labeled dataset was then used to train the LSTM model.

2.2.3. Resampling

The dataset generated from the IndoBERT-based labeling process was further refined through undersampling using the random under sampler technique to reduce class imbalance and minimize model bias toward the majority class [29]. After the undersampling process, the dataset consisted of 9,295 entries, with class distributions of 4,419 (47.5%) negative, 2,946 (31.7%) neutral, and 1,930 (20.8%) positive sentiments. The distribution results after resampling are shown in Figure 2.

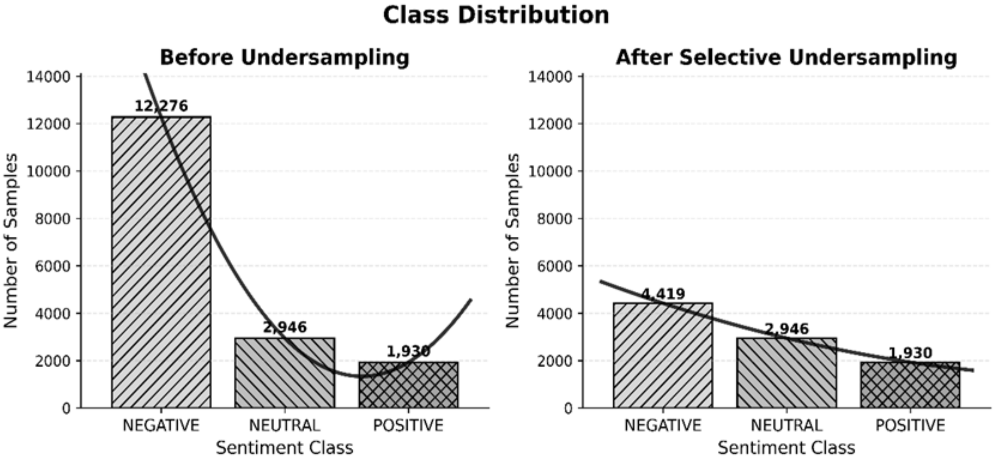


Figure 2. Class distribution comparison before and after undersampling

The dataset was stratified into 90% for training (8,365 entries) and 10% for testing (930 entries), ensuring that class proportions remained balanced across subsets. A stratified 10-fold cross-validation was then applied to the training set. Each fold produced training and validation subsets with balanced class distributions, enabling a comprehensive model evaluation and ensuring more reliable and representative results [16].

2.2.4. Splitting data

The dataset used in this study consisted of 9,295 entries, divided into a training set comprising 89.99% (8,365 entries) and a testing set comprising 10.01% (930 entries). The training data were subsequently partitioned using stratified 10-fold cross-validation, with 90% (7,528 entries) used for training and 10% (837 entries) for validation in each fold during model training [30].

2.2.5. Feature extraction

Feature extraction was conducted to convert textual data into numerical representations suitable for processing by the classification algorithm [31]. This process involved tokenization, in which the text was transformed into integer sequences using the Keras Tokenizer [32], and padding, which standardized the input length by truncating texts that exceeded the maximum length or appending zeros to shorter sequences [33]. The categorical sentiment labels were mapped into three classes and encoded using one-hot encoding. The feature extraction process is illustrated schematically, as shown in Figure 3.

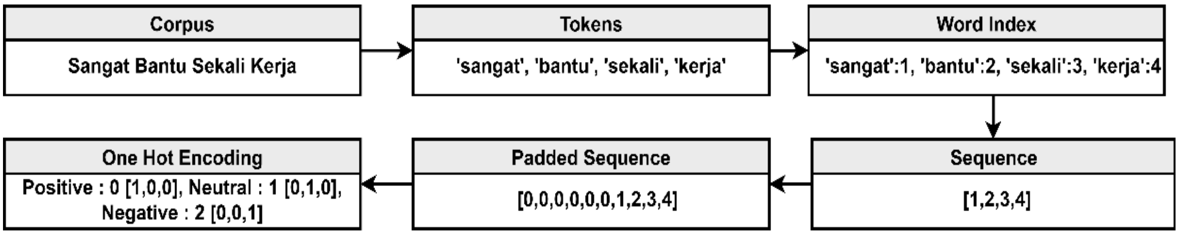

Corpus
Sangat Bantu Sekali Kerja
Tokens
'sangat', 'bantu', 'sekali', 'kerja'
Word Index
'sangat':1, 'bantu':2, 'sekali':3, 'kerja':4
Sequence
[1,2,3,4]
Padded Sequence
[0,0,0,0,0,0,1,2,3,4]
One Hot Encoding
Positive : 0 [1,0,0], Neutral : 1 [0,1,0],
Negative : 2 [0,0,1]

Figure 3. Illustration of the feature extraction process

2.3. Training with long short-term memory

The LSTM algorithm for sentiment analysis was trained using specific hyperparameter configurations, with a primary focus on optimizing the learning rate to ensure training stability. The evaluated learning rate values were 0.0001, 0.001, 0.002, 0.005, 0.01, and 0.02. During training, an early stopping mechanism was employed to terminate the process when improvements in validation performance became insignificant, thereby preventing overfitting and accelerating convergence [34].

The LSTM model was implemented using the Keras library in Python [35]. The model configuration in this study consisted of a 64-dimensional embedding, an LSTM layer with 32 units, and a dense layer with 3 output units, along with a dropout rate set to 0.2 and a batch size of 32. This configuration was adapted from previous studies, in which an embedding size of 64 and a dropout rate of 0.2 were found effective for sentiment analysis tasks [36], while the use of 32 LSTM units followed a hybrid architecture applied to hotel review sentiment analysis [37]. A batch size of 32 was consistently adopted in both prior studies.

2.4. Testing

The testing dataset was reserved exclusively for evaluating the final performance of the model and was not used during the training or validation phases to prevent bias [38]. The evaluation was conducted across two domains using a confusion matrix: in-domain, which includes 930 test samples from Sirekap application reviews on the Google Play Store (442 negative, 295 neutral, and 193 positive), and cross-domain, which includes 100 news headlines related to the MBG (30 negative, 40 neutral, and 30 positive).

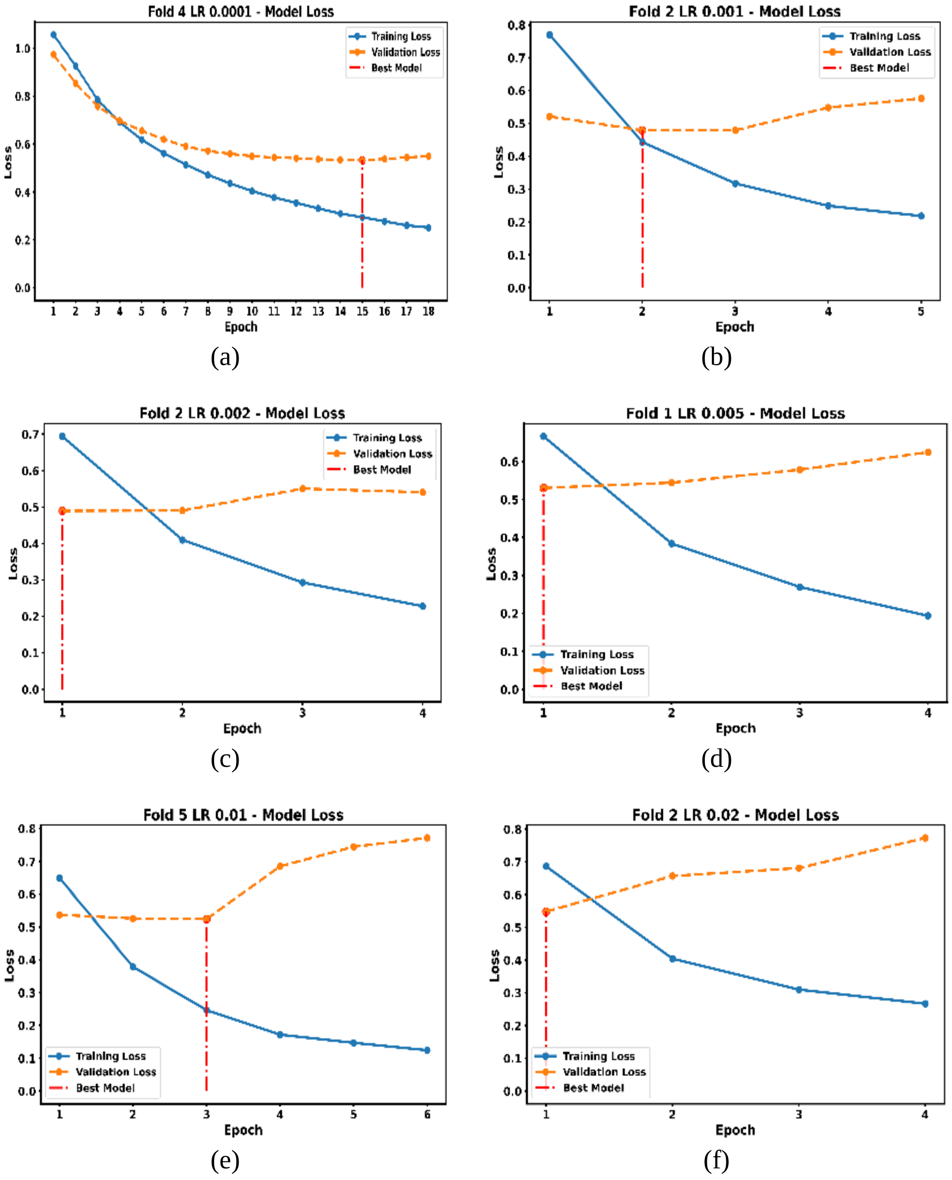
2.5. Measurement

Model evaluation was conducted through cross-validation on the training data, followed by a one-way ANOVA test to determine the significance of performance differences among various learning rate configurations. A post-hoc test was subsequently applied to further examine pairwise differences [39]. To validate the statistical results, a final evaluation was conducted on the testing dataset under two scenarios: comments from the Sirekap application on the Google Play Store and news headlines related to the MBG program. The representative model with the best performance from each learning rate configuration was selected based on the optimal fold. Model performance was measured using standard classification metrics derived from the confusion matrix, including accuracy, precision, recall, and F1-score.

3. RESULTS AND DISCUSSION

3.1. Experimental results and analysis

The experiments were conducted on a text classification task using a LSTM architecture optimized with the Adam optimizer. An early stopping mechanism with a patience of three epochs was applied to prevent overfitting [34]. Training was halted when no improvement in validation performance was observed for three consecutive epochs, and the best-performing model was selected based on the lowest validation loss before reaching the patience threshold. Figure 4 shows the validation loss curve of the model achieving the highest validation accuracy, presented as a representative example since all folds exhibit similar patterns. The red dash-dot line indicates the selected model prior to the onset of increasing loss, illustrating the loss behavior before potential overfitting.



(a) (b)

(c) (d)

(e) (f)

Figure 4. Representative loss history for selected learning rates: (a) 0.0001, (b) 0.001, (c) 0.002, (d) 0.005, (e) 0.01, and (f) 0.02

Based on the loss curves shown in Figure 4, the model with the smallest learning rate (0.0001) required more epochs to reach convergence due to the limited number of weight updates, resulting in a slower, more stable optimization process. In contrast, higher learning rates (0.001–0.02) accelerated the reduction of training loss and caused an earlier increase in validation loss, indicating the onset of overfitting. These configurations achieved their best performance in fewer epochs, exhibiting faster, less stable convergence patterns. In Figure 4(a) corresponds to 0.0001, while Figures 4(b) to (f) represent learning rates of 0.001, 0.002, 0.005, 0.01, and 0.02, respectively.

The results shown in Figure 5 indicate that a learning rate of 0.001 provides the best overall performance, with high accuracy, low variance, and consistent results across folds. This learning rate also converges faster than 0.0001 while maintaining similar accuracy levels, demonstrating an optimal balance between training stability and convergence speed. Although the dataset and model architecture differ [10], the findings consistently indicate that a moderate learning rate improves convergence and enhances the model's learning capability. Based on the validation training results, to assess the statistical significance of performance differences among learning rates, a one-way ANOVA was conducted using the 10-fold cross-validation results, with accuracy values averaged across folds. Assumption tests confirmed that the data were normally distributed (Shapiro-Wilk W = 0.9614, p = 0.0553) and had homogeneous variances ($Levene\ F = 1.3639$, p = 0.2524).

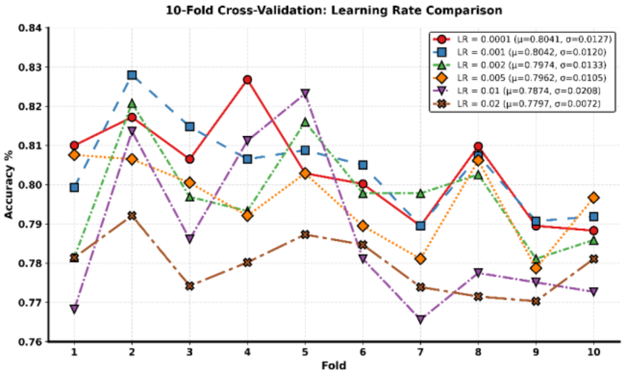


Figure 5. Comparison of learning rates using stratified 10-fold cross-validation

The ANOVA results showed a significant effect of learning rate on model accuracy ($F(5, 54) = 5.1989$, $p = 0.000575$, $\eta^2 = 0.3250$), with eta squared indicating a large effect size, suggesting statistically significant differences among learning rate configurations. The use of one-way ANOVA is consistent with previous studies which also assessed significance ($p < 0.05$) [17], [39]. Furthermore, the descriptive statistics of model performance across the tested learning rates are summarized in Table 1. As shown, the lower learning rates (0.0001 and 0.001) yielded the highest mean validation accuracy with relatively low variance, indicating stable model behavior. In contrast, higher learning rates (≥ 0.01) resulted in decreased accuracy.

Table 1. Descriptive statistics of validation results across various learning rates

Learning rate	Mean	Std dev	Min	Max
0.0001	0.8041	0.0127	0.7883	0.8268
0.001	0.8042	0.0120	0.7895	0.8280
0.002	0.7974	0.0133	0.7811	0.8208
0.005	0.7962	0.0105	0.7787	0.8076
0.01	0.7874	0.0208	0.7656	0.8232
0.02	0.7797	0.0072	0.7703	0.7921

The post-hoc analysis [18] revealed that significant differences in model accuracy were found only for the highest learning rate (0.02) compared with 0.0001, 0.001, and 0.002. Table 2 summarizes these comparisons, while the remaining pairwise comparisons did not show significant differences at a significance level of α = 0.05.

Table 2. Post-hoc analysis of differences between learning rates

Group 1	Group 2	Mean difference	P-adj	Lower	Upper	Significant
0.0001	0.02	-0.0244	0.0020	-0.0421	-0.0067	Yes
0.001	0.02	-0.0245	0.0019	-0.0422	-0.0068	Yes
0.002	0.02	-0.0177	0.0499	-0.0354	0	Yes

3.2. Evaluation

Evaluation was conducted to validate the training results using the test dataset [38]. Lower learning rates (0.0001 and 0.001) exhibited more stable convergence, with a one-way ANOVA indicating a significant effect on accuracy. The model's generalization capability was further assessed through confusion matrix analysis under two scenarios: in-domain and cross-domain [19]. The detailed evaluation results are shown in Table 3.

Table 3. Classification report: in-domain and cross-domain

Domain	Learning rate	Accuracy	Macro average			Weight average		
			Precision	Recall	F1-score	Precision	Recall	F1-score
In-domain	**0.0001**	**0.7871**	0.7735	0.7809	**0.7768**	0.7896	0.7871	**0.7879**
In-domain	**0.001**	**0.7849**	0.7701	0.7827	**0.7753**	0.7878	0.7849	**0.7856**
In-domain	**0.002**	**0.7871**	0.7820	0.7829	**0.7810**	0.7921	0.7871	**0.7881**
In-domain	0.005	0.7817	0.7720	0.7743	0.7730	0.7814	0.7817	0.7814
In-domain	0.01	0.7710	0.7563	0.7628	0.7593	0.7729	0.7710	0.7716
In-domain	0.02	0.7731	0.7693	0.7603	0.7619	0.7789	0.7731	0.7735
Cross-domain	0.0001	0.3300	0.6597	0.3611	0.2112	0.6604	0.3300	0.1994
Cross-domain	0.001	0.3400	0.7708	0.3722	0.2321	0.7938	0.3400	0.2185
Cross-domain	0.002	0.3300	0.3015	0.3528	0.2581	0.3026	0.3300	0.2502
Cross-domain	0.005	0.3300	0.6053	0.3611	0.2118	0.5947	0.3300	0.1997
Cross-domain	**0.01**	**0.3600**	0.6073	0.3861	**0.2868**	0.5966	0.3600	**0.2781**
Cross-domain	**0.02**	**0.3600**	0.5012	0.3722	**0.3253**	0.4895	0.3600	**0.3231**

The results show that, for in-domain data, learning rates of 0.0001, 0.001, and 0.002 achieved the best performance, with accuracies between 78.49% and 78.71% and F1-scores around 0.78. These results indicate a good balance between precision and recall under the imbalanced data condition. In contrast, for cross-domain data, the overall accuracy decreased. Lower learning rates (0.0001–0.002) reached only 33%–34%, while higher learning rates (0.01 and 0.02) slightly improved accuracy to about 36%, with F1-scores ranging from 0.27 to 0.32. Although the differences were not substantial, the results suggest that higher learning rates provide slightly better generalization to other domains, while lower learning rates perform better for in-domain data. This difference indicates the model's limited generalization ability, which may be affected by the distinct characteristics of the datasets: user comments are typically informal and context-dependent, whereas news titles tend to be more formal and topic-focused.

3.3. Performance

The model performance for different learning rates is shown in Tables 4 and 5. During the experiments, the training time was recorded using the Python time module, and memory usage was monitored through the psutil library in Google Colaboratory. The findings reveal that a learning rate of 0.002 provided the most efficient training time, while a learning rate of 0.001 achieved the lowest memory consumption without significantly increasing the training duration. Overall, the performance across in-domain and cross-domain datasets remained relatively consistent, with only minor variations observed despite differences in the number of test samples and evaluation periods.

Table 4. Average training time and memory usage for 10-fold cross-validation

Learning rate	Average time (s)	Average memory (MB)
0.0001	259.5	13.3
0.001	82.1	5.3
0.002	69.9	15.4
0.005	78.1	9.2
0.01	74.1	5.23
0.02	66.3	14.8

Table 5. Model performance comparison on in-domain and cross-domain testing

Dataset	Best fold	Learning rate	Time (s)	Memory (MB)
In-domain	4	0.0001	0.79	19.13
In-domain	2	0.001	0.79	0.38
In-domain	2	0.002	0.81	29.00
In-domain	1	0.005	0.78	6.25
In-domain	5	0.01	0.80	1.75
In-domain	2	0.02	0.78	18.63
Cross-domain	4	0.0001	2.96	0.75
Cross-domain	2	0.001	0.56	16.32
Cross-domain	2	0.002	0.54	9.88
Cross-domain	1	0.005	0.54	0.25
Cross-domain	5	0.01	0.52	12.01
Cross-domain	2	0.02	0.52	6.38

4. CONCLUSION

This study analyzed the effect of learning rate variations on the stability and accuracy of LSTM-based text classification models using the Adam optimizer. The results demonstrate that learning rate significantly influences model performance. Based on 10-fold cross-validation and a one-way ANOVA test, significant differences were observed among learning rates, with post-hoc analysis indicating that a learning rate of 0.02 differed significantly from 0.0001, 0.001, and 0.002. Analysis of the training and validation loss curves showed that a learning rate of 0.0001 produced the most stable convergence, albeit requiring more epochs. Learning rates of 0.0001 and 0.002 achieved the highest in-domain accuracy of 78.71%, while a learning rate of 0.001 provided an optimal balance between stability and training efficiency, reaching 78.49% in-domain accuracy. Higher learning rates (0.01–0.02) demonstrated better adaptability to cross-domain data, achieving 36% accuracy, indicating potential for improving generalization despite lower absolute accuracy. Therefore, a learning rate of 0.001 is recommended for achieving both high accuracy and efficient training, whereas 0.0001 is preferable when training stability is prioritized. Higher learning rates (0.01–0.02) can be leveraged to enhance adaptability on cross-domain data. These findings provide practical guidance for NLP practitioners in Indonesia on effectively tuning learning rates for Indonesian sentiment analysis tasks. Future research may explore adaptive learning rate mechanisms or scheduler-based optimization and extend experiments to other deep learning architectures, such as bidirectional or attention-based LSTM models, to further improve cross-domain generalization.

ACKNOWLEDGMENTS

We acknowledge the Department of Informatics Engineering, Faculty of Engineering and Maritime Technology, Universitas Maritim Raja Ali Haji, for providing the facilities and academic environment that enabled the completion of this research.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Serly Eldina	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Tekad Matulatan	✓	✓		✓	✓	✓		✓		✓		✓		✓
Novrizal Fattah Fahmitra	✓			✓	✓	✓		✓		✓		✓		

C :	**C**onceptualization	I :	**I**nvestigation	Vi :	**Vi**sualization
M :	**M**ethodology	R :	**R**esources	Su :	**Su**pervision
So :	**So**ftware	D :	**D**ata Curation	P :	**P**roject administration
Va :	**Va**lidation	O :	Writing - **O**riginal Draft	Fu :	**Fu**nding acquisition
Fo :	**Fo**rmal analysis	E :	Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data supporting the experimental findings of this study, including the model results on training and test data as well as all related experiment graphs, are publicly available in the project repository: [https://github.com/Serly-Eldina/LSTM-LearningRate-Indonesia-Sentiment].

REFERENCES

[1] T. Joseph, "Natural language processing (NLP) for sentiment analysis in social media," *International Journal of Computing and Engineering*, vol. 6, no. 2, pp. 35–48, 2024, doi: 10.47941/ijce.2135.

[2] J. Jose and R. Simritha, "Sentiment analysis and topic classification with LSTM networks and TextRazor," *International Journal of Data Informatics and Intelligent Computing*, vol. 3, no. 2, pp. 42–51, 2024, doi: 10.59461/ijdiic.v3i2.115.

[3] A. H. Nasution, A. Onan, Y. Murakami, W. Monika, and A. Hanafiah, "Benchmarking open-source large language models for sentiment and emotion classification in Indonesian tweets," *IEEE Access*, vol. 13, pp. 94009–94025, 2025, doi: 10.1109/ACCESS.2025.3574629.

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[5] S. M. Al-Selwi *et al.*, "RNN-LSTM: from applications to modeling techniques and beyond—systematic review," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 5, 2024, doi: 10.1016/j.jksuci.2024.102068.

[6] I. S. B. Azhar, W. K. Sari, and N. K. P. Gumay, "A comparative study of deep learning's performance methods for news article using word representations," *Sistemasi*, vol. 14, no. 2, p. 932, 2025, doi: 10.32520/stmsi.v14i2.5090.

[7] J. L. Ba and D. P. Kingma, "Adam: a method for stochastic optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15, 2015.

[8] W. Cheng, R. Pu, and B. Wang, "AMC: adaptive learning rate adjustment based on model complexity," *Mathematics*, vol. 13, no. 4, 2025, doi: 10.3390/math13040650.

[9] P.-J. Chiang, "Adaptive penalty method with an Adam optimizer for enhanced convergence in optical waveguide mode solvers," *Optics Express*, vol. 31, no. 17, p. 28065, 2023, doi: 10.1364/oe.495855.

[10] C. Peng, "Comprehensive analysis of the impact of learning rate and dropout rate on the performance of convolutional neural networks on the CIFAR-10 dataset," in *Applied and Computational Engineering*, 2024, pp. 183–192, doi: 10.54254/2755-2721/102/20241161.

[11] A. K. P. Anil and U. K. Singh, "An optimal solution to the overfitting and underfitting problem of healthcare machine learning models," *Journal of Systems Engineering and Information Technology (JOSEIT)*, vol. 2, no. 2, pp. 77–84, 2023, doi: 10.29207/joseit.v2i2.5460.

[12] N. O. Aljehane, "A new approach to sentiment analysis on twitter data with LSTM," *2023 3rd International Conference on Computing and Information Technology (ICCIT)*. IEEE, pp. 657–663, Sep. 13, 2023, doi: 10.1109/iccit58132.2023.10273876.

[13] S. I. Wiguna, A. Erfina, and C. Warman, "Sentiment analysis of public opinion on Pi network on Reddit using FinBERT," *Indonesian Journal of Data and Science*, vol. 6, no. 3, pp. 426–432, 2025, doi: 10.56705/ijodas.v6i3.342.

[14] K. Shirini, M. B. Kordan, and S. S. Gharehveran, "Impact of learning rate and epochs on LSTM model performance: a study of chlorophyll-a concentrations in the Marmara Sea," *Journal of Supercomputing*, vol. 81, no. 1, p. 265, 2025, doi: 10.1007/s11227-024-06806-2.

[15] M. N. Zaidan, Y. Sibaroni, and S. S. Prasetyowati, "Learning rate and epoch optimization in the fine-tuning process for Indobert's performance on sentiment analysis of Mytelkomsel app reviews," *Jurnal Teknik Informatika (Jutif)*, vol. 5, no. 5, pp. 1443–1450, 2024, doi: 10.52436/1.jutif.2024.5.5.2396.

[16] S. Szeghalmy and A. Fazekas, "A comparative study of the use of stratified cross-validation and distribution-balanced stratified cross-validation in imbalanced learning," *Sensors*, vol. 23, no. 4, 2023, doi: 10.3390/s23042333.

[17] K. Anam and A. Al-Jumaily, "Improved myoelectric pattern recognition of finger movement using rejection-based extreme learning machine," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 19, no. 1, pp. 134–145, 2021, doi: 10.12928/TELKOMNIKA.V19I1.16566.

[18] R. F. Rojas, C. Joseph, G. Bargshady, and K. L. Ou, "Empirical comparison of deep learning models for fNIRS pain decoding," *Frontiers in Neuroinformatics*, vol. 18, 2024, doi: 10.3389/fninf.2024.1320189.

[19] P. Joshi, S. Mundra, and A. Mundra, "Proto-Att-FSL: enhanced prototypical network for cross-domain few-shot airline sentiment classification," *Social Network Analysis and Mining*, vol. 15, no. 1, 2025, doi: 10.1007/s13278-025-01436-9.

[20] H. Aldabbas, A. Bajahzar, M. Alruily, A. A. Qureshi, R. M. A. Latif, and M. Farhan, "Google Play content scraping and knowledge engineering using natural language processing techniques with the analysis of user reviews," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 192–208, 2020, doi: 10.1515/jisys-2019-0197.

[21] M. F. Abdillah and K. Kusnawi, "Comparative analysis of long short-term memory architecture for text classification," *ILKOM Jurnal Ilmiah*, vol. 15, no. 3, pp. 455–464, 2023, doi: 10.33096/ilkom.v15i3.1906.455-464.

[22] C. Salazar, E. Montoya-Múnera, and J. Aguilar, "Sentiment analysis in learning resources," *Journal of Computers in Education*, vol. 10, no. 4, pp. 637–662, 2023, doi: 10.1007/s40692-022-00237-9.

[23] S. Garg, D. Chaturvedi, T. Jain, A. Mishra, and A. Kapoor, "Sentiment analysis: techniques, limitations, and case studies in data extraction and classification," *International Research Journal of Engineering Science*, vol. 9, no. 3, pp. 1–6, 2023.

[24] M. Afrad, D. C. Febrianto, S. Wijayanto, and M. Y. Fathoni, "Sentiment analysis of visitor reviews on Baturaden tourist attraction using machine learning methods," *Edu Komputika Journal*, vol. 11, no. 1, pp. 57–64, 2024, doi:10.15294/edukom.v11i1.10561.

[25] U. R. Pol, "Hugging face: revolutionizing AI and NLP," *International Journal for Research in Applied Science and Engineering Technology*, vol. 12, no. 8, pp. 1121–1124, 2024, doi: 10.22214/ijraset.2024.64023.

[26] H. Ahmadian, T. F. Abidin, H. Riza, and K. Muchtar, "Hybrid Models for emotion classification and sentiment analysis in indonesian language," in *Applied Computational Intelligence and Soft Computing*, Yogyakarta, Indonesia, 2024, pp. 209–214, doi: 10.1155/2024/2826773.

[27] Y. Fauziah, B. Yuwono, and A. S. Aribowo, "Lexicon based sentiment analysis in indonesia languages : a systematic literature review," in *RSF Conference Series: Engineering and Technology*, 2021, pp. 363–367, doi: 10.31098/cset.v1i1.397.
[28] K. Barik and S. Misra, "Analysis of customer reviews with an improved VADER lexicon classifier," *Journal of Big Data*, vol. 11, no. 1, 2024, doi: 10.1186/s40537-023-00861-x.
[29] S. N. Almuayqil, M. Humayun, N. Z. Jhanjhi, M. F. Almufareh, and N. A. Khan, "Enhancing sentiment analysis via random majority under-sampling with reduced time complexity for classifying tweet reviews," *Electronics (Switzerland)*, vol. 11, no. 21, 2022, doi: 10.3390/electronics11213624.
[30] I. O. Muraina, "Ideal dataset splitting ratios in machine learning algorithms: general concerns for data scientists and data analysts," *7th International Mardin Artuklu Scientific Researches Conference*, pp. 496–504, 2022.
[31] A. Maiti, A. Abarda, and M. Hanini, "The impact of feature extraction techniques on the performance of text data classification models," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 35, no. 2, pp. 1041–1052, 2024, doi: 10.11591/ijeecs.v35.i2.pp1041-1052.
[32] L. Davoodi, J. Mezei, and M. Heikkilä, "Aspect-based sentiment classification of user reviews to understand customer satisfaction of e-commerce platforms," *Electronic Commerce Research*, 2025, doi: 10.1007/s10660-025-09948-4.
[33] J. Shobana and M. Murali, "An efficient sentiment analysis methodology based on long short-term memory networks," *Complex and Intelligent Systems*, vol. 7, no. 5, pp. 2485–2501, 2021, doi: 10.1007/s40747-021-00436-4.
[34] M. K. Anam, S. Defit, Haviluddin, L. Efrizoni, and M. B. Firdaus, "Early stopping on CNN-LSTM development to improve classification performance," *Journal of Applied Data Sciences*, vol. 5, no. 3, pp. 1175–1188, 2024, doi: 10.47738/jads.v5i3.312.
[35] A. C. M. V. Srinivas, C. Satyanarayana, C. Divakar, and K. P. Sirisha, "Sentiment analysis using neural network and LSTM," in *IOP Conference Series: Materials Science and Engineering*, 2021, p. 012007, doi: 10.1088/1757-899x/1074/1/012007.
[36] R. K. Das, M. Islam, M. M. Hasan, S. Razia, M. Hassan, and S. A. Khushbu, "Sentiment analysis in multilingual context: Comparative analysis of machine learning and hybrid deep learning models," *Heliyon*, vol. 9, no. 9, p. e20281, 2023, doi: 10.1016/j.heliyon.2023.e20281.
[37] Y. A. Singgalen, "BiLSTM-LSTM Hybrid model with glove embeddings for hotel review sentiment analysis," *Journal of Information System Research (JOSH)*, vol. 6, no. 2, pp. 857–868, 2024, doi: 10.47065/josh.v6i2.6420.
[38] Z. B. Kizilkan, M. S. Sivri, I. Yazici, and O. F. Beyca, "Neural networks and deep learning," *Springer Series in Advanced Manufacturing. Springer International Publishing*, pp. 127–151, 2022, doi: 10.1007/978-3-030-93823-9_5.
[39] Z. Zmirli, S. El Harfaoui, A. Mohssine, A. Driouich, H. Chaair, and B. Sallek, "Less energy and aluminum electrode consumption for fluoride removal using electrocoagulation process: optimization by one-way ANOVA analysis and experimental design," *Desalination and Water Treatment*, vol. 292, pp. 185–205, 2023, doi: 10.5004/dwt.2023.29504.

BIOGRAPHIES OF AUTHORS

Serly Eldina she is currently pursuing a Bachelor of Engineering (S.T.) degree in Informatics Engineering at the School of Electrical and Informatics, Universitas Maritim Raja Ali Haji (UMRAH). She has an interest in artificial intelligence (AI), particularly in natural language processing (NLP) and its applications in text analysis. Her research areas include machine learning and deep learning. She can be contacted at email: serlyeldina03@gmail.com.

Tekad Matulatan a tenure lecturer in the Informatics Department, School of Electrical and Informatics, Universitas Maritim Raja Ali Haji (UMRAH), Province of Riau Islands. Graduated from Flinders University of South Australia (2005) and had undergone 1 year of pre-doctoral education at Arizona State University (2015). As PI in the field of educational data mining (EDM) 2014-2017, and research assistant in the fields of multimodal NLM, generative AI, and visual computational. He can be contacted at email: tekad.matulatan@umrah.ac.id.

Novrizal Fattah Fahmitra received his Bachelor's Degree in Informatics Engineering from STMIK AMIKOM Yogyakarta. After that, he took the Informatics Study Program - Master's Program for Medical Informatics Concentration at the Islamic University of Indonesia. In 2024, he joined Universitas Maritim Raja Ali Haji (UMRAH) as a lecturer for the Bachelor's Degree of Informatics Engineering Study Program. Now his field of science is system information technology administration, his research concludes system analyst, usability analysis, IT audit, and IT administration. He can be contacted at email: nffahmitra@umrah.ac.id.