Improving Neutral Sentiment Classification in Indonesian E-Wallet Reviews Using Word2Vec and Easy Data Augmentation (EDA)

Muhammad Fattah Edric Camilo[1], Fatma Indriani[2*] , Mohammad Reza Faisal[3], Dwi Kartini[4], Dodon Turianto Nugrahadi[5]

[1,2,3,4,5]*Department of Computer Science, Universitas Lambung Mangkurat, Indonesia*

*corr-author: f.indriani@ulm.ac.id

Abstract - The rapid expansion of digital payments has produced massive volumes of user-generated reviews, making manual analysis impractical. This study focuses on the challenge of neutral sentiment classification in Indonesian e-wallet reviews, where neutral comments often contain ambiguous language and are underrepresented relative to positive and negative classes. A total of 26,537 preprocessed DANA application reviews were used to evaluate whether Word2Vec embeddings and Easy Data Augmentation (EDA) can improve neutral sentiment detection when combined with Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (BiLSTM) architectures. Experiments comparing eight model configurations showed that the combination of Word2Vec, EDA, and LSTM achieved the best performance, with 0.861 accuracy, 0.841 macro-F1, and 0.749 F1-score for the neutral class. These findings demonstrate that semantic representations and controlled lexical variation can jointly enhance minority-class recognition in short informal Indonesian text and highlight the importance of aligning embedding strategies with sequence architectures.

Keywords: Sentiment analysis, Word2Vec, easy data augmentation, LSTM, BiLSTM.

I. INTRODUCTION

The widespread adoption of electronic payment systems in Indonesia, especially QR-based payment services, has changed the way consumers perform daily transactions. This growth has increased the use of application-store review platforms, where users express satisfaction, complaints, and service expectations. These reviews provide valuable feedback on usability, service quality, and feature needs, but their volume makes manual analysis inefficient and inconsistent [1,2,3]. Sentiment analysis therefore offers an important automated approach for extracting actionable information from user opinions, and studies in related digital-service domains confirm that targeted model optimization can yield meaningful gains in classification performance [4].

Although sentiment analysis has progressed considerably, neutral sentiment classification remains particularly challenging in real-world review data. Neutral reviews often contain factual statements, clarifying questions, or mildly expressed concerns with limited emotional markers. As a result, they frequently overlap lexically with both positive and negative classes. The problem becomes more severe in imbalanced datasets, where neutral instances are substantially outnumbered and minority-class performance is often masked by high overall accuracy [2,3,5]. In e-wallet applications, however, neutral reviews remain important because they often indicate incomplete information, usability ambiguity, or unresolved service issues.

Previous studies on DANA app reviews have mainly used conventional machine-learning models such as Naïve Bayes, SVM, and Random Forest with bag-of-words or TF-IDF representations [6,7]. Other studies have explored convolutional neural networks for Indonesian sentiment analysis [8]. While these methods are useful, they are still limited in representing contextual similarity and subtle sentiment cues that frequently occur in short user-generated reviews.

Word2Vec has been widely used to reduce lexical sparsity and preserve semantic similarity among words [9], [10]. In parallel, recurrent architectures such as LSTM and BiLSTM have shown strong performance in sequence modeling because they can capture context across a review [11,12]. In addition, data augmentation has attracted growing attention as a practical way to improve robustness under limited and imbalanced training data [13]. Easy Data Augmentation (EDA) is especially attractive because it introduces simple lexical variations while retaining the original label when applied carefully.

Despite these developments, limited evidence is available on how Word2Vec and EDA interact for

neutral sentiment classification in Indonesian e-wallet reviews, especially when paired with different recurrent architectures. Therefore, this study aims to evaluate the individual and combined effects of Word2Vec and EDA on neutral-class detection and to compare their behavior in LSTM and BiLSTM models. The study contributes a systematic comparison across eight experimental settings and highlights an architecture-embedding interaction that is relevant for short, informal Indonesian text.

II. METHOD

This study applies multi-class sentiment classification to Indonesian reviews of the DANA e-wallet application. The overall workflow is presented in Fig. 1.

A. Data Collection

The dataset was obtained from Kaggle under the title "DANA App Sentiment Review on Playstore Indonesia" by Alex Mario Simanjuntak [14]. It contains approximately 48,000 labeled Google Play Store reviews discussing transaction processing, application functionality, service quality, and general user experience. The original class distribution is imbalanced: positive 53%, negative 34%, and neutral 13%. This imbalance makes neutral-class learning especially difficult because the class is both underrepresented and semantically ambiguous.

B. Data Preprocessing

1) Lowercasing: all review text was converted to lowercase to ensure consistent token representation.

2) Noise Removal: URLs, excessive punctuation, symbols, and irregular characters were removed because they did not contribute meaningful sentiment information.

3) Stopword Removal: common functional words were removed to emphasize content-bearing terms.

4) Stemming: Indonesian stemming was applied using the Sastrawi library to normalize inflected words to their base forms [15,16].

5) Single-Word Review Handling: single-word reviews were removed because they provide very limited context for multi-class classification.

6) Deduplication: duplicate reviews after preprocessing were removed to reduce redundancy and avoid training bias.

After preprocessing, the dataset decreased from 48,000 to 26,537 reviews. The reduction came from removing 130 empty entries, single-word reviews, and 2,129 duplicate instances. The final class distribution consisted of 13,544 negative, 8,453 positive, and 4,540 neutral reviews.

C. Dataset Splitting

The preprocessed dataset was divided into 80% training data and 20% testing data using stratified splitting so that the class distribution was preserved in both partitions. The test set was kept isolated throughout all experiments to ensure an unbiased evaluation.

D. Easy Data Augmentation on the Training Data

EDA was applied only to the training set to reduce the effect of limited minority-class data [17,18]. Three operations were used: synonym replacement, random swap, and random deletion. Random insertion was excluded because inserting arbitrary words may distort the sentiment of neutral reviews, which already contain subtle sentiment cues [19]. EDA was applied only to the neutral class, increasing the number of neutral training instances from 3,632 to 4,946 while leaving the negative and positive classes unchanged. Table I shows the class distribution before and after augmentation.

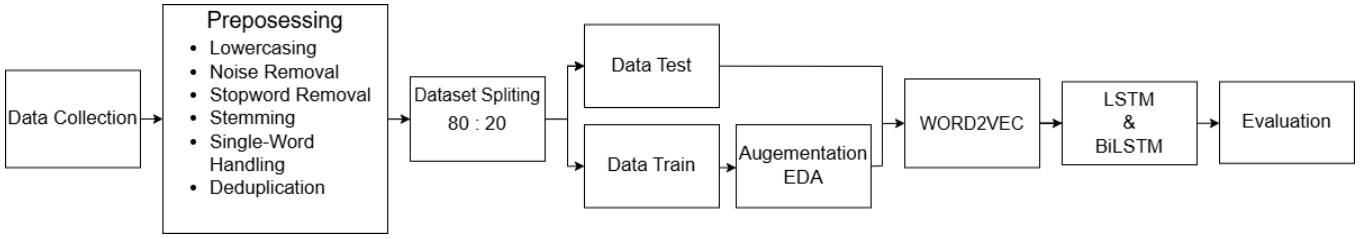

Fig. 1 Overall research methodology flowchart

TABLE I
CLASS DISTRIBUTION BEFORE AND AFTER EASY DATA AUGMENTATION (EDA)

Sentiment	Before	After	Change
Negative	10.835	10.835	0
Neutral	3.632	4.946	1.314
Positive	6.762	6.762	0
Total	21.229	22.543	1.314

E. Word Embedding Construction

Two embedding initialization strategies were compared. The baseline used random initialization, with embedding weights learned during model training. The alternative used Word2Vec embeddings trained on the preprocessed DANA review corpus with the Gensim library. Word2Vec was configured with 100-dimensional vectors, a context window of 10, a minimum word frequency of 2, and the skip-gram architecture [9]. In both settings, the embedding matrix was loaded into a trainable Keras Embedding layer so that the model could adapt the representation to the classification task.

F. Model Training with LSTM and BiLSTM

Fig. 2 illustrates the LSTM and BiLSTM architectures used in this study.

Both architectures start with a trainable embedding layer for padded sequences. This layer is followed by an LSTM or bidirectional LSTM layer with 128 units and return_sequences enabled, then a global max-pooling layer. The pooled representation passes through a dropout layer with a rate of 0.3, a dense layer with 64 units and ReLU activation, another dropout layer, and a final softmax output layer with three units. The models were trained using the Adam optimizer with a learning rate of 1×10^-4 and sparse categorical cross-entropy loss. All experiments used the same preprocessing pipeline and train-test split to ensure a controlled comparison.

G. Evaluation

Performance was evaluated on the held-out test set using accuracy, macro-F1, and class-specific F1-scores for the negative, neutral, and positive classes. Because the dataset is imbalanced, macro-F1 was used as the primary metric since it assigns equal weight to each class [5]. Confusion matrices were also examined to identify systematic errors in neutral-class prediction. In total, eight experimental configurations were compared by varying the embedding type, augmentation setting, and recurrent architecture.

III. RESULTS AND DISCUSSION

A. Experimental Results

Eight experimental configurations were evaluated to measure the effect of embedding initialization, data augmentation, and model architecture. Table II summarizes the results using accuracy, macro-F1, and class-specific F1-scores. The best configuration was LSTM with Word2Vec and EDA, which achieved 0.861 accuracy, 0.841 macro-F1, and 0.749 F1-score for the neutral class. Compared with the baseline configuration (LSTM with random embeddings and no augmentation), this result represents a 6.5% improvement in neutral-class F1-score.

```
model = Sequential([
    Embedding(input_dim=VOCAB_SIZE,
              output_dim=EMBEDDING_DIM,
              weights=[embedding_matrix],
              input_length=MAX_LEN,
              mask_zero=True,
              trainable=True),
    LSTM(128, return_sequences=True),
    GlobalMaxPooling1D(),
    Dropout(0.3),
    Dense(64, activation='relu'),
    Dropout(0.3),
    Dense(3, activation='softmax')
])

model.compile(
    loss='sparse_categorical_crossentropy',
    optimizer=Adam(learning_rate=0.0001),
    metrics=['accuracy']
)
```

```
model = Sequential([
    Embedding(input_dim=VOCAB_SIZE,
              output_dim=EMBEDDING_DIM,
              weights=[embedding_matrix],
              input_length=MAX_LEN,
              mask_zero=True,
              trainable=True),
    Bidirectional(LSTM(128, return_sequences=True)),
    GlobalMaxPooling1D(),
    Dropout(0.3),
    Dense(64, activation='relu'),
    Dropout(0.3),
    Dense(3, activation='softmax')
])

model.compile(
    loss='sparse_categorical_crossentropy',
    optimizer=Adam(learning_rate=0.0001),
    metrics=['accuracy']
)
```

Fig. 2 LSTM and BiLSTM model architecture

Three general patterns can be observed. First, configurations that combined Word2Vec and EDA consistently outperformed those using only one of the two techniques. Second, LSTM performed more consistently than BiLSTM when Word2Vec embeddings were used. Third, the neutral class remained the most difficult category, as indicated by the larger variation in F1-score across settings. Table II presents the complete comparison.

B. Impact of Word2Vec and Data Augmentation

Using LSTM with random embeddings and no augmentation as the baseline (F1-NEU = 0.704), Word2Vec alone increased neutral-class F1-score to 0.742, while EDA alone improved it to 0.734. These gains indicate that semantic representation and controlled lexical variation each contribute positively, but through different mechanisms.

Word2Vec reduces lexical sparsity by representing semantically related Indonesian terms in neighboring regions of the vector space. This helps the model capture subtle neutral cues that do not rely on explicit positive or negative words. EDA complements this behavior by exposing the model to varied surface forms of neutral expressions through synonym replacement, random swap, and random deletion without heavily distorting the original sentiment.

When the two techniques were combined, the neutral-class F1-score rose to 0.749, exceeding the improvement of either individual technique. This pattern suggests a complementary relationship: semantic embeddings help the model generalize across the lexical variations introduced by augmentation, while augmentation broadens the contexts in which semantic relationships are learned. Fig. 3 visualizes this effect.

C. Architecture Comparison: LSTM versus BiLSTM

With random embedding initialization and no augmentation, BiLSTM outperformed LSTM on the neutral class, which is consistent with the expectation that bidirectional context can help resolve ambiguity. However, the pattern changed when Word2Vec was introduced. Word2Vec improved LSTM performance from 0.704 to 0.742, but reduced BiLSTM performance from 0.740 to 0.707. This reversal indicates that pre-trained semantic embeddings were more compatible with LSTM than with BiLSTM in this dataset.

A plausible explanation is that Word2Vec was trained to capture forward semantic regularities through the skip-gram mechanism, whereas BiLSTM processes the sequence in both directions. In short and informal reviews, backward context may add limited new information while introducing representation mismatch. This explanation is also consistent with previous observations on Indonesian sentiment analysis using pre-trained embeddings [20].

The same tendency remained under augmentation. LSTM with Word2Vec and EDA still produced the best overall result, while BiLSTM with Word2Vec was among the weakest settings for neutral detection. Fig. 4 and Fig. 5 summarize these comparisons.

TABLE II
MODEL RESULTS COMPARISON

Embedding	Model	Augmentation	Macro F1	F1 NEG	F1 NEU	F1 POS	ACC
Word2Vec	LSTM	EDA	0.84053	0.89094	0.74938	0.88126	0.86096
Random	LSTM	EDA	0.82915	0.87810	0.73380	0.87557	0.84815
Word2Vec	LSTM	None	0.83851	0.88902	0.74230	0.88421	0.85678
Random	LSTM	None	0.82235	0.88421	0.70365	0.87918	0.84845
Word2Vec	BiLSTM	EDA	0.83469	0.88248	0.74127	0.88032	0.85399
Random	BiLSTM	EDA	0.83232	0.87879	0.73413	0.88403	0.85079
Random	BiLSTM	None	0.83578	0.88692	0.74031	0.88012	0.85692
Word2Vec	BiLSTM	None	0.82481	0.88573	0.70745	0.88124	0.84915

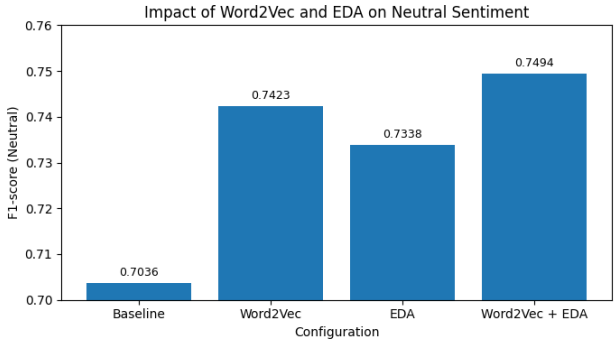

Fig. 3 Impact of Word2Vec and EDA on neutral-class performance

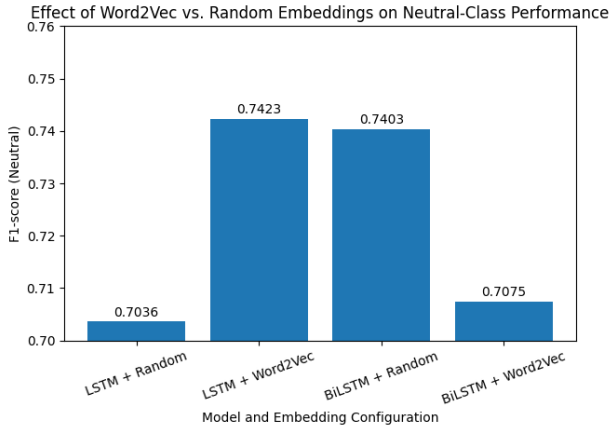

Fig. 4 Effect of Word2Vec versus random embeddings on neutral-class performance

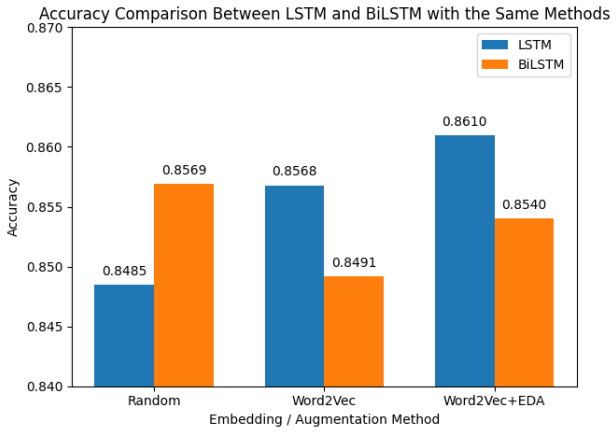

Fig. 5 Comparison of LSTM and BiLSTM under different settings

D. Error Analysis and Model Interpretation

The confusion matrix of the best model, LSTM with Word2Vec and EDA, is shown in Fig. 6. Negative and positive reviews are relatively well separated, while the neutral class still produces the largest number of errors. Of 908 actual neutral instances, 712 were classified correctly, while 119 were predicted as negative and 77 as positive. This confirms that neutral sentiment remains the hardest class despite the improvements achieved by Word2Vec and EDA.

Table III lists representative misclassifications from the Indonesian review data. The original Indonesian review texts are presented in italics to distinguish non-English research data from the English explanation. The errors mainly fall into three patterns. First, factual reports of system problems are often misclassified as neutral because they lack strong negative affect markers. Second, very short neutral phrases are easily mapped to positive or negative classes because the available context is extremely limited. Third, negation, sarcasm, and mixed expressions remain difficult for the model because they combine conflicting sentiment cues in a short text [21].

Table III illustrates representative misclassification cases, revealing three primary error patterns:

1) Negative misclassified as Neutral: The most frequent error occurs when reviews describe system problems using factual, non-emotional language. Examples include *tidak aplikasi* (no application) and *aplikasi dana ok tidak buka iya* (DANA app is okay but cannot be opened), which express frustration through brief statements lacking explicit negative affect terms. These reviews resemble neutral factual reports linguistically, making them difficult to distinguish from genuine neutral expressions.

2) Neutral misclassified as Positive or Negative: Short phrases containing minimal context force the model to infer sentiment from sparse cues. Examples such as *cicil bantu* (installment help) and *jaga aman* (keep safe) trigger incorrect associations with polarized categories based on word-level sentiment patterns rather than true contextual meaning.

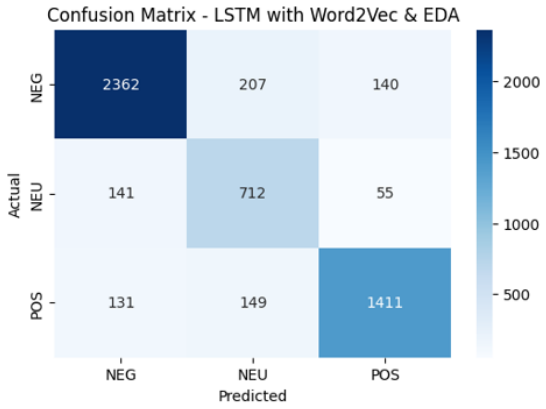

Fig. 6 Confusion matrix of LSTM with Word2Vec and EDA

TABLE III
REPRESENTATIVE MISCLASSIFICATIONS OF INDONESIAN REVIEW DATA

Review Text	Sentiment	Predicted
gmn ni udah update x dana msh ga d buka suruh update lg gmn ni dana	NEGATIVE	NEUTRAL
tidak aplikasi	NEGATIVE	NEUTRAL
susah login bikin pusing	NEUTRAL	NEGATIVE
aplikasi dana ok tidak buka iya	NEGATIVE	NEUTRAL
hallo admin aplikasi tidak tolong bantu bbrp	NEGATIVE	NEUTRAL
x top up saldo dana eh saldo nya ga masuk nya	NEGATIVE	POSITIVE
cicil bantu	NEUTRAL	POSITIVE
jaga aman	NEUTRAL	POSITIVE
mantap gak hilang iklan	POSITIVE	NEGATIVE
aplikasi busukkkkkdi pakai berat biaya kirim mahalketerlaluan kayak rampok	NEGATIVE	POSITIVE

3) Complex sentiment expressions: Reviews containing sarcasm, negation, or mixed sentiment signals prove challenging. The phrase *mantap gak hilang iklan* (great, the ads did not disappear) uses a positive word sarcastically to express negative sentiment, while *x top up saldo dana eh saldo nya ga masuk nya* (top up was attempted, but the balance did not enter the account) combines a transaction action with a negative outcome, confusing the classifier.

IV. CONCLUSION

This study demonstrates that combining Word2Vec embeddings with Easy Data Augmentation improves neutral sentiment classification in an imbalanced Indonesian e-wallet review dataset. The best configuration, LSTM with Word2Vec and EDA, achieved 0.861 accuracy, 0.841 macro-F1, and 0.749 F1-score for the neutral class, representing a 6.5% improvement over the baseline. The results also show that the benefits of Word2Vec depend on the model architecture: LSTM consistently benefited from semantic initialization, whereas BiLSTM did not. These findings suggest that alignment between embedding strategy and sequence architecture is important for minority-class performance in short, informal Indonesian text. This study is limited to one dataset and one augmentation setting, and future work should evaluate other embedding models, augmentation strategies, and transformer-based classifiers.

ACKNOWLEDGMENT

The authors would like to thank the Data Science Lab, Department of Computer Science, Universitas Lambung Mangkurat, for computing support and research resources.

REFERENCES

[1] Kementerian Komunikasi dan Digital (Komdigi), "Transaksi QRIS Melonjak 226,54%, Revolusi Pembayaran Digital di Indonesia," 2024. [Online]. Available: https://www.komdigi.go.id/berita/ekonomi-digital/detail/transaksi-qris-melonjak-22654-revolusi-pembayaran-digital-di-indonesia

[2] S. Henning, W. Beluch, A. Fraser, and A. Friedrich, "A Survey of Methods for Addressing Class Imbalance in Deep-Learning-Based Natural Language Processing," in Proc. 17th Conf. Eur. Chapter Assoc. Comput. Linguistics, 2023, pp. 523-540, doi: 10.18653/v1/2023.eacl-main.38.

[3] K. L. Tan, C. P. Lee, and K. M. Lim, "A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research," Applied Sciences, vol. 13, no. 7, p. 4550, 2023, doi: 10.3390/app13074550.

[4] A. Fadlil, I. Riadi, and F. Andrianto, "Non-linear Kernel Optimisation of Support Vector Machine Algorithm for Online Marketplace Sentiment Analysis," JUITA: Jurnal Informatika, vol. 12, no. 1, pp. 29-38, 2024, doi: 10.30595/juita.v12i1.19798.

[5] J. Opitz, "A Closer Look at Classification Evaluation Metrics and a Critical Reflection of Common Evaluation Practice," Trans. Assoc. Comput. Linguist., vol. 12, pp. 820–836, 2024, doi: 10.1162/tacl_a_00675.

[6] G. G. Warow and H. Pandia, "Analisis Sentimen Aplikasi Dana Menggunakan Naïve Bayes Classifier dan Support Vector Machine," Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi, vol. 13, no. 1, 2024, doi: 10.35889/jutisi.v13i1.1893.

[7] F. A. Larasati, D. E. Ratnawati, and B. T. Hanggara, "Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 6, no. 9, pp. 4305–4313, 2022. [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/11562

[8] A. A. Achmad, I. Kurniasari, and I. Yanuartanti, "Analisis Klasifikasi Sentimen Berbasis Topik pada Ulasan Layanan Dana dan Sakuku dengan Convolutional Neural Network," INFORMASI (Jurnal Informatika dan Sistem Informasi), vol. 15, no. 2, 2023, doi: 10.37424/informasi.v15i2.267.

[9] C. A. N. Agustina, R. Novita, Mustakim, and N. E. Rozanda, "The Implementation of TF-IDF and Word2Vec on Booster Vaccine Sentiment Analysis Using Support Vector Machine Algorithm," Procedia Computer Science, vol. 234, pp. 156-163, 2024, doi: 10.1016/j.procs.2024.02.162.

[10] N. P. Doholio, A. Samratulangi, and A. Setiawan, "Comparison of Word2Vec and CountVectorizer with Mutual Information in Support Vector Machine for Public Sentiment Analysis," J. Math. Comput. Stat., vol. 8, no. 1, pp. 12-23, 2025, doi: 10.35580/jmathcos.v8i1.6640.

[11] U. B. Mahadevaswamy and P. Swathi, "Sentiment Analysis using Bidirectional LSTM Network," Procedia Computer Science, vol. 218, pp. 45-56, 2023, doi: 10.1016/j.procs.2022.12.400.

[12] M. R. R. Rana, M. A. Al-Sabaawi, and A. M. Al-Hindawi, "A BiLSTM-CF and BiGRU-Based Deep Sentiment Analysis Model to Explore Customer Reviews for Effective Recommendations," Eng. Technol. Appl. Sci. Res., vol. 13, no. 5, pp. 11739-11746, 2023, doi: 10.48084/etasr.6278.

[13] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy, "A Survey of Data Augmentation Approaches for NLP," in Findings of ACL-IJCNLP, 2021, doi: 10.18653/v1/2021.findings-acl.84.

[14] A. M. Simanjuntak, "DANA App Sentiment Review on Playstore Indonesia," Kaggle Dataset, 2024. [Online]. Available: https://www.kaggle.com/datasets/alexmariosimanjuntak/dana-app-sentiment-review-on-playstore-indonesia

[15] A. S. Rizki, N. M. Aristi, M. N. Ridha, A. F. Zulfahri, and D. A. Wibowo, "Implementation of the Indonesian Language Stemming Algorithm in Twitter Data Preprocessing: Case Study of Twitter Wargabanua and Instakalsel," Fidelity, vol. 5, no. 3, pp. 175–183, Sep. 2023, doi: 10.52005/fidelity.v5i3.170.

[16] J. Pardede and D. Darmawan, "Perbandingan Algoritma Stemming Porter, Sastrawi, Idris, dan Arifin & Setiono pada Dokumen Teks Bahasa Indonesia," Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 12, no. 1, pp. 69-76, 2025, doi: 10.25126/jtiik.2025128860.

[17] A. N. Azizah, M. F. Asy'ari, I. W. D. Prastya, and D. Purwitasari, "Easy Data Augmentation untuk Data yang Imbalance pada Konsultasi Kesehatan Daring," Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 10, no. 5, pp. 1095-1104, 2023, doi: 10.25126/jtiik.2023107082.

[18] B. Li, Y. Hou, and W. Che, "Data Augmentation Approaches in Natural Language Processing: A Survey," AI Open, vol. 3, pp. 71-90, 2022, doi: 10.1016/j.aiopen.2022.03.001.

[19] A. Karimi, L. Rossi, and A. Prati, "AEDA: An Easier Data Augmentation Technique for Text Classification," in Findings of EMNLP, 2021, doi: 10.18653/v1/2021.findings-emnlp.234.

[20] I. Yanti and E. Utami, "Sentiment Analysis of Indonesia's Capital Relocation Using Word2Vec and Long Short-Term Memory Method," Jurnal Teknik Informatika (JUTIF), vol. 6, no. 1, pp. 149-158, 2025, doi: 10.52436/1.jutif.2025.6.1.2712.

[21] S. Mohammad, "Sentiment Analysis: Detecting Valence, Emotions, and Other Affectual States from Text," in Handbook of Natural Language Processing, 3rd ed. Elsevier, 2022, pp. 301-325, doi: 10.1016/B978-0-12-821124-3.00011-9.