ADMI IJST International Journal Science and Technology

OPEN ACCESS

IJST Vol 4 No. 2 | July 2025 | ISSN: 2828-7223 (print), ISSN: 2828-7045 (online), Page 152-167

Comparison Machine Learning and Deep Learning in the Sentiment Analysis of Shopee Reviews

Kuwat Setiyanto[1]*, Azzahra Dania Indriyani[2]
Informatics, Faculty of Industrial Technology, Gunadarma University, Indonesia

Article History
Received : July 08th, 2025
Revised : July 19th, 2025
Accepted : July 21st, 2025
Published : July 31st, 2025

Corresponding author*:
kuwatsetiyanto@gmail.com

Cite This Article:
Kuwat Setiyanto, & Azzahra Dania Indriyani. (2025). Comparison of Machine Learning and Deep Learning in Shopee Review Sentiment Analysis. International Journal Science and Technology, 4(2), 152–167.

DOI:
https://doi.org/10.56127/ijst.v4i2.2225

Abstract: This study aims to compare the performance of Machine Learning (Random Forest and Support Vector Machine) and Deep Learning (Long Short-Term Memory) algorithms in analyzing the sentiment of Shopee app users. A total of 50,000 Indonesian language reviews were collected through web scraping from the Google Play Store. After going through the cleaning and feature extraction process, three models were developed and evaluated using accuracy, precision, recall, and F1-score metrics. The results showed that the LSTM model provided the best performance in classifying sentiment into three categories: positive, negative, and neutral. Furthermore, this model was implemented in a Streamlit-based sentiment analysis dashboard that allows interactive exploration and testing of sentiment. This study shows that the application of Machine Learning and Deep Learning technologies is effective in analyzing public opinion and can support strategic decision-making in the context of e-commerce.

Keywords: Deep Learning, Machine Learning, Model, Sentimen.

INTRODUCTION

The rapid development of information technology has driven a major transformation in the world of commerce, particularly through e-commerce platforms. In Indonesia, Shopee has become one of the most popular and widely used e-commerce applications. With millions of active users and increasing daily transactions, Shopee is not only a place to buy and sell, but also a platform for consumers to share their opinions, experiences, and reviews of the products or services they receive.

Customer reviews are a valuable source of data. Through analysis of these reviews, companies can understand customer perceptions and satisfaction and make strategic decisions to improve service quality. However, the large number of reviews received daily makes manual analysis inefficient and prone to bias. Therefore, an automated approach is needed that can accurately and efficiently categorize customer sentiment.

Sentiment analysis is a technique used to identify and classify a person's opinions or feelings toward an entity, such as a product or service. In this context, artificial intelligence-based approaches such as machine learning and deep learning are relevant solutions. Machine learning enables systems to learn from historical data and make predictions automatically, while deep learning provides deeper capabilities in understanding sentence context and complex meanings through artificial neural networks.

In this study, Machine Learning and Deep Learning approaches were used to evaluate and compare the performance of three sentiment classification methods: Random Forest, Support Vector Machine (SVM), and Long Short-Term Memory (LSTM). All three models were applied to classify Shopee customer review sentiment into three categories: positive, negative, and neutral. The results from the best-performing model were then implemented in an interactive sentiment analysis dashboard aimed at supporting data-driven decision-making on the e-commerce platform.

RESEARCH METHOD

The stages involved in this research are as follows:

1. Data Collection

   Shopee customer review data was collected using web scraping techniques, resulting in 50,000 rows of reviews in Indonesian. This data served as the basis for building a representative sentiment analysis model.

2. Exploratory Data Analysis (EDA)

   After data collection, exploratory analysis was performed to understand data characteristics, such as sentiment distribution and word frequency, and to identify missing data, duplication, or inconsistencies. Data cleaning involved removing irrelevant characters, tokenization, stopword removal, and stemming to prepare the data for further processing.

3. Feature Extraction

   Text features from customer reviews were extracted using two different methods. For the Machine Learning models (SVM and Random Forest), the Term Frequency-Inverse Document Frequency (TF-IDF) method was used. For the Deep Learning (LSTM) model, features were extracted using tokenization and embedding layer techniques, which allow the model to capture contextual relationships between words in a sentence.

4. Model Training

   At this stage, two main approaches are applied: Machine Learning algorithms (Random Forest and Support Vector Machine) and Deep Learning (Long Short-Term Memory). Machine Learning models are applied to predict sentiment based on text feature representations, while Deep Learning models are used to capture more complex word sequence patterns.

5. Model Evaluation

   After model training, evaluation was conducted using accuracy, precision, recall, and F1-score metrics. This evaluation aimed to measure each model's performance in classifying customer reviews into three sentiment categories: positive, negative, and neutral.

6. Presentation of Results

   The analysis results are displayed through data visualization using Matplotlib and Seaborn, as well as through an interactive Streamlit-based dashboard that allows users to directly explore the sentiment classification results.

RESULTS AND DISCUSSION

Data Preparation

Data Preparation is the initial stage of preparing data for use in model training. This process includes data cleaning, transforming text into numerical representations, and determining appropriate labels for sentiment classification.

a. Feature Extraction and Labeling

The initial stage of model training involves preparing features and labels. Features are obtained from the clean_content column, which has undergone text preprocessing. The feature representation uses the TF-IDF technique due to its ability to weight important words and reduce the influence of common words, so the data is ready to be used as model input.

Table 1. Separation of Features and Labels

Feature	Label
Clean Content (Clean Content)	• 2 (Positive) • 1 (Neutral) • 0 (Negative)

b. Feature Engineering

In the Feature Engineering stage, the cleaned text data is transformed and prepared to meet the requirements of the algorithm to be used. This process involves several text representation methods, including vectorization using TF-IDF, as well as tokenization and padding, specifically for LSTM models.

c. TF-IDF Vectorization

Text features are then represented using the TF-IDF Vectorizer with the parameter max_features=10000, which limits the number of features to the 10,000 most significant words. Using TF-IDF allows for emphasis on common words and higher weighting of less frequent but important words in the document.

Table 2. Summary of TF-IDF Feature Dimensions

Number of Samples (Rows)	Number of Features (Columns)
50.000	10.000

d. Tokenization and Padding for LSTM

In the LSTM model, text data is converted into a sequence of numeric tokens through tokenization, where each word is mapped to a unique index in the vocabulary. The tokenizer is used with a limit of 10,000 most frequently occurring words. The text is then converted into a sequence of numbers using texts_to_sequences.

Since LSTM requires inputs of equal length, padding is performed to equalize the length of each sequence to 100 tokens, by adding leading zeros if necessary.

Table 3. Summary of Tokenization and Padding

Description	Mark
Number of Samples	50.000
Maximum Sequence Length	100 Token
Vocabulary Size	10,000 Unique Words

e. Displaying Vocabulary and Class Info

At this stage, the number of sentiment classes, the feature data dimension after padding, and the vocabulary size from tokenization were checked. The number of unique classes found was 3 (positive, neutral, negative). The padded feature data consisted of 50,000 samples with a length of 100 tokens per sample, indicating the data was ready for model training.

Table 4. Summary of Vocabulary and Class

Information	Mark
Number of Classes	3
Feature Dimension (Sample * Token)	50.000 * 100
Vocabulary Size	18,788 words

Model Training

The model training phase involves feeding feature and label data to the algorithm to build a classification model. Several models are tested to achieve the best results based on accuracy and performance.

a. Model Deep Learning (LSTM)

At this stage, the preprocessed feature and label data is separated into three subsets: 70% training data, 20% testing data, and 10% validation data. This division is intended to train the model, test its performance, and avoid overfitting with validation during training.

Table 5. Distribution of LTSM Training Sample Data

Dataset	Number of Samples
Training Set	35.000
Validation Set	5.000
Test Set	10.000

Training Process Model LSTM

The LSTM model is trained using the backpropagation through time (BPTT) algorithm, which optimizes network weights by minimizing a loss function. A frequently used loss function for multi-class classification is categorical cross-entropy.

```
# Inisialisasi Model LSTM dengan Embedding dan Tuning Hyperparameter
model_lstm = Sequential([
    Embedding(input_dim=10000, output_dim=128, input_length=100),
    SpatialDropout1D(0.4),
    LSTM(128, dropout=0.3, recurrent_dropout=0.3),
    Dense(3, activation='softmax')
])

# Kompilasi Model
model_lstm.compile(loss='sparse_categorical_crossentropy', optimizer=Adam(), metrics=['accuracy'])

# Callback (EarlyStopping & ReduceLROnPlateau)
callback_early_stopping = EarlyStopping(monitor='val_loss', patience=2, restore_best_weights=True)
callback_lr_reduction = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=2)

callbacks = [callback_early_stopping, callback_lr_reduction]

# Melatih Model
riwayat_latihan = model_lstm.fit(fitur_latih_lstm, label_latih_lstm,
                                 validation_data=(fitur_validasi_lstm, label_validasi_lstm),
                                 epochs=15,
                                 batch_size=64,
                                 verbose=1,
                                 callbacks=callbacks)
```

```
Epoch 1/15
547/547 ━━━━━━━━━━━━━━━━━━━━ 204s 363ms/step - accuracy: 0.7529 - loss: 0.5486 - val_accuracy: 0.9474 - val_loss: 0.1513 - learning_rate: 0.0010
Epoch 2/15
547/547 ━━━━━━━━━━━━━━━━━━━━ 198s 357ms/step - accuracy: 0.9627 - loss: 0.1091 - val_accuracy: 0.9680 - val_loss: 0.1020 - learning_rate: 0.0010
Epoch 3/15
547/547 ━━━━━━━━━━━━━━━━━━━━ 208s 369ms/step - accuracy: 0.9846 - loss: 0.0495 - val_accuracy: 0.9764 - val_loss: 0.0852 - learning_rate: 0.0010
Epoch 4/15
547/547 ━━━━━━━━━━━━━━━━━━━━ 186s 339ms/step - accuracy: 0.9917 - loss: 0.0282 - val_accuracy: 0.9792 - val_loss: 0.0805 - learning_rate: 0.0010
Epoch 5/15
547/547 ━━━━━━━━━━━━━━━━━━━━ 218s 368ms/step - accuracy: 0.9942 - loss: 0.0210 - val_accuracy: 0.9762 - val_loss: 0.0837 - learning_rate: 0.0010
Epoch 6/15
547/547 ━━━━━━━━━━━━━━━━━━━━ 195s 355ms/step - accuracy: 0.9956 - loss: 0.0142 - val_accuracy: 0.9782 - val_loss: 0.0815 - learning_rate: 0.0010
```

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

di mana:

- C adalah jumlah kelas (3 kelas: positif, netral, negatif)
- y_i adalah label sebenarnya dalam format one-hot
- $\hat{y}_i$ adalah probabilitas prediksi model untuk kelas ke-i

Figure 1. Code and Formula for Calculating LSTM Model Training

b. Model Machine Learning (SVM)

The Support Vector Machine (SVM) model is used as a machine learning approach for sentiment classification. Before training, the data, transformed into feature representations using TF-IDF, is divided into two parts: 80% for training and 20% for testing.

Table 6. Number of SVM Data Samples

Type	Number of Samples
Training Set	40.000
Test Set	10.000
Total	50.000

This division aims to ensure that the model can be trained with enough data, and then tested for performance on data it has never seen before.

SVM Model Training Process

The SVM model was trained using the One-vs-Rest (OvR) approach for three-class classification (negative, neutral, positive), with a linear kernel. The training objective was to find the optimal hyperplane that separates the classes in the TF-IDF feature space.

```python
# Inisialisasi Model SVM dengan Kernel Linear
model_svm = SVC(kernel='linear', probability=True)

# Latih Model
model_svm.fit(X_latih_svm, y_latih_svm)

# Lakukan Prediksi
y_pred_svm = model_svm.predict(X_uji_svm)
```

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i$$

dengan syarat:

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

di mana:

- $\mathbf{w}$ adalah vektor bobot model
- b adalah bias
- C adalah parameter regulasi (penyeimbang antara margin maksimum dan penalti kesalahan)
- ξ_i adalah slack variable untuk data yang tidak terpisah secara sempurna
- y_i adalah label kelas sebenarnya (biasanya ± 1)
- $\mathbf{x}_i$ adalah fitur input untuk sampel ke-i
- n adalah jumlah total sampel pelatihan

Figure 2. SVM Model Training Code and Formula

c. Model Machine Learning (RF)

The Random Forest (RF) model was chosen as one of the machine learning methods for sentiment classification in this study. The data, represented in the form of TF-IDF features, was then divided into two subsets: 70% for training and 30% for testing. This data division allows the model to learn from a large amount of data while simultaneously being tested on previously unseen data, allowing for objective measurement of model performance.

Table 7. Number of Random Forest Data Samples

Type	Number of Samples
Training Set	35.000
Test Set	15.000
Total	50.000

RF Model Training Process

The Random Forest model is an ensemble learning method that combines prediction results from several independent decision trees to improve classification accuracy and stability. Mathematically, Random Forest builds T decision trees, where each t-th tree produces a prediction ht(x) for an input feature x. The final model prediction y^ is obtained using a majority voting mechanism, which selects the class most frequently chosen by all trees in the forest.

```
# Inisialisasi Model Random Forest dengan Penyesuaian Hyperparameter
rf_model = RandomForestClassifier
 (n_estimators=300, max_depth=None, min_samples_split=5, min_samples_leaf=2, random_state=42)

# Latih Model
rf_model.fit(X_latih_rf, y_latih_rf)

# Lakukan Prediksi
y_pred_rf = rf_model.predict(X_uji_rf)
```

$$\hat{y} = \arg\max_{c \in C} \sum_{t=1}^{T} \mathbf{1}(h_t(x) = c)$$

di mana:

- C adalah himpunan kelas target (misalnya kelas sentimen negatif, netral, positif),
- $\mathbf{1}(\cdot)$ adalah fungsi indikator yang bernilai 1 jika prediksi pohon $h_t(x)$ sama dengan kelas c, dan 0 jika tidak,
- T adalah jumlah pohon keputusan dalam ensemble.

Figure 3. Random Forest Model Training Code and Formula

Model Evaluation

Model evaluation is performed to assess the performance of the trained machine learning algorithm, using metrics such as accuracy, precision, recall, and F1-score. The evaluation process aims to ensure the model's ability to effectively classify previously unseen data. Evaluation results on test data provide an overview of the model's generalization ability in accurately and consistently classifying negative, neutral, and positive sentiments.

a. LSTM Model Evaluation

After training, the LSTM model was evaluated on the test data using accuracy metrics and classification reports (precision, recall, f1-score) for three sentiment classes: negative, neutral, and positive. The model achieved a training accuracy of 99.46% and a test accuracy of 97.29%, demonstrating good generalization ability.

```
Evaluasi Model LSTM
Akurasi pelatihan per epoch: 0.9956
Akurasi uji: 0.9774
313/313 ━━━━━━━━━━━━━━━━━━━━ 15s 45ms/step

Laporan Klasifikasi:
              precision    recall  f1-score   support

     negatif       0.95      0.93      0.94       999
      netral       0.97      0.98      0.98      4619
     positif       0.99      0.99      0.99      4382

    accuracy                           0.98     10000
   macro avg       0.97      0.97      0.97     10000
weighted avg       0.98      0.98      0.98     10000
```

Figure 4. Results of LSTM Model Evaluation on Test Data

The classification report shows high performance across all classes, with balanced precision and recall, indicating the model is capable of accurately and completely classifying data. The loss and accuracy graphs during training indicate a stable process without overfitting.

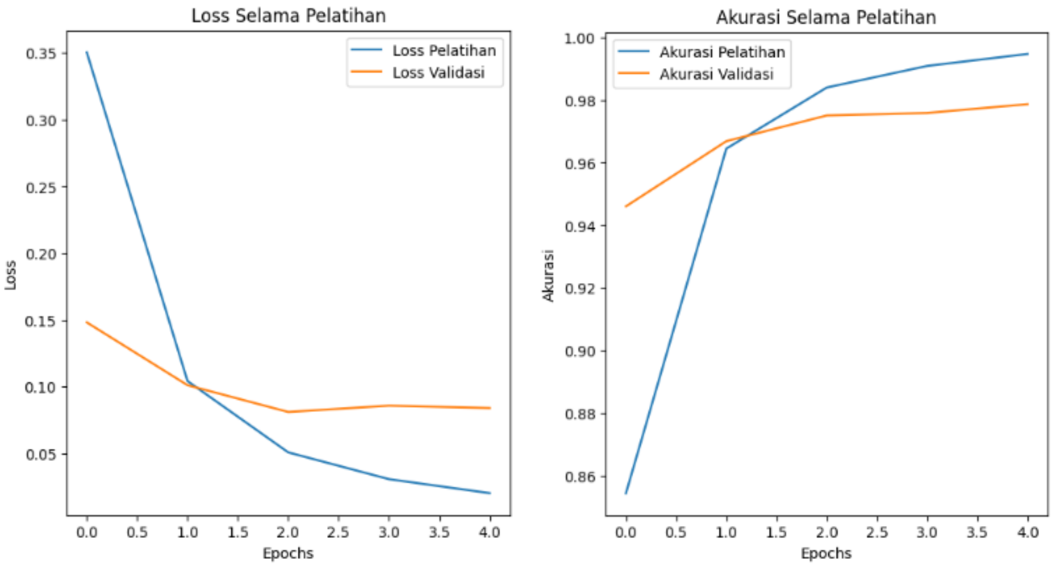

Figure 5. Loss and Accuracy Graph During Training

b. SVM Model Evaluation

After training, the SVM model was evaluated on test data using accuracy metrics and classification reports (precision, recall, f1-score) for three sentiment classes:

negative, neutral, and positive. The model achieved a training accuracy of 98.33% and a testing accuracy of 96.05%, demonstrating excellent and stable generalization performance on previously unseen data.

```
Evaluasi Model SVM + TF-IDF

Akurasi Latih: 0.9833
Akurasi Uji: 0.9605

Laporan Klasifikasi:
              precision    recall  f1-score   support

     negatif       0.94      0.83      0.88       999
      netral       0.94      0.98      0.96      4619
     positif       0.99      0.97      0.98      4382

    accuracy                           0.96     10000
   macro avg       0.95      0.93      0.94     10000
weighted avg       0.96      0.96      0.96     10000
```

Figure 6. Results of SVM Model Evaluation on Test Data

The classification report shows high and consistent performance across all classes, with relatively balanced precision and recall values. This indicates that the model is capable of accurately classifying negative, neutral, and positive sentiments with minimal misclassification. Furthermore, these results also demonstrate that the text representation technique using TF-IDF is effective in supporting sentiment classification with the SVM algorithm.

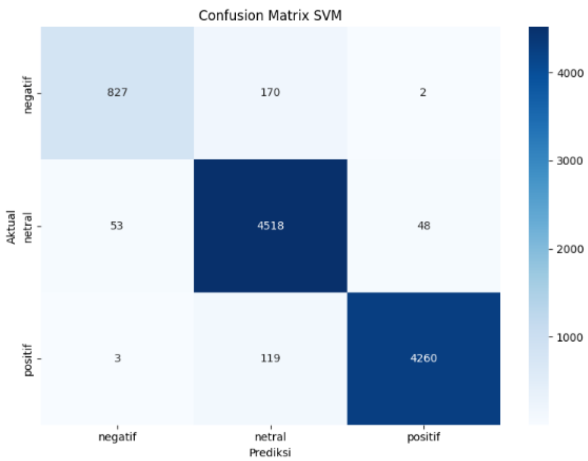

Figure 7. Confusion Matrix SVM

c. RF Model Evaluation

After training, the Random Forest model was evaluated on the test data using accuracy metrics and classification reports (precision, recall, f1-score) for three sentiment classes: negative, neutral, and positive. The model achieved a training accuracy of 94.92% and a testing accuracy of 91.69%, demonstrating good generalization performance on previously unseen data.

```
Evaluasi Model Random Forest dengan TF-IDF
Akurasi Latih: 0.9492
Akurasi Uji: 0.9169

Laporan Klasifikasi:
              precision    recall  f1-score   support

     negatif       0.86      0.73      0.79      1475
      netral       0.92      0.91      0.92      6925
     positif       0.92      0.96      0.94      6600

    accuracy                           0.92     15000
   macro avg       0.90      0.87      0.88     15000
weighted avg       0.92      0.92      0.92     15000
```

Figure 8. Results of RF Model Evaluation on Test Data

The classification report shows good and consistent performance across all classes, with relatively balanced precision and recall values. This indicates that the model is capable of accurately classifying negative, neutral, and positive sentiments while minimizing misclassification errors. These results also demonstrate that the use of TF-IDF feature representation effectively improves the performance of the Random Forest model on sentiment classification tasks.

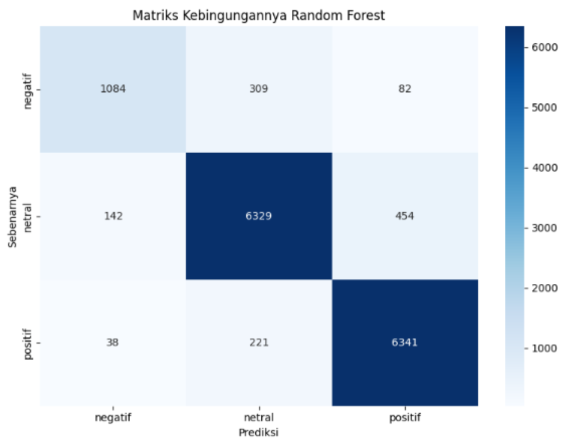

Figure 9. Confusion Matrix Random Forest

d. Model Comparison

Comparison of the performance of the three machine learning models, namely LSTM with Word Embedding, SVM with TF-IDF, and Random Forest (RF) with TF-IDF, was carried out based on the accuracy value on the test data.

Summary of Results

	Model	Accuracy	Split Ratio	Feature Type
0	LSTM + Embedding (80/20)	97.74%	80/20	Word Embedding
1	SVM + TF-IDF (80/20)	96.05%	80/20	TF-IDF
2	RF + TF-IDF (70/30)	91.69%	70/30	TF-IDF

Figure 10. Summary of Comparison Results of 3 Models

The LSTM model performed best with an accuracy of 97.29%, followed by the SVM model with an accuracy of 96.05%. The Random Forest model came in third with an accuracy of 91.69%. Differences in data split ratio and the type of features used also influenced these results. Overall, all three models were capable of sentiment classification with good accuracy, but the LSTM with Word Embedding provided the most optimal results in the context of this dataset.

Saving Model

`After the training and evaluation process is complete, the created models are saved for reuse without the need for retraining. LSTM models are saved in the HDF5 file format with the .h5 extension, which supports storing the complete model architecture and weights. SVM and Random Forest models are saved in the pickle (.pkl) format, which allows for efficient serialization of Python objects.

In addition to the model, supporting objects such as the TF-IDF vectorizer and tokenizer are also stored using pickle to ensure consistent reproducibility of the data preprocessing process during subsequent inference or testing. This storage ensures the integrity and consistency of the sentiment classification pipeline and facilitates the model's use in production applications or further testing.

- lstm_sentiment_model.h5
- rf_sentiment_model.pkl
- svm_sentiment_model.pkl
- tokenizer.pkl
- vectorizer.pkl

Figure 10. *Trained Model Form*

Structured storage of models and supporting objects enables more efficient project management and supports model deployment into real applications and subsequent testing without the need for time-consuming and resource-intensive retraining processes.

Dashboard General View

This dashboard is designed to display sentiment analysis results for Shopee app user reviews. Built using Streamlit, it provides an interactive interface that makes it easy for users to understand consumer perceptions through various visualizations and features.

Figure 33 General Dashboard View

- **Tab Overview**

  The Overview tab displays a summary of the sentiment analysis results for reviews, divided into three categories: positive, neutral, and negative. The visualization is displayed as a pie chart or bar chart, depending on the user's selection in the sidebar. The majority of reviews are positive, reflecting a high level of user satisfaction with Shopee.

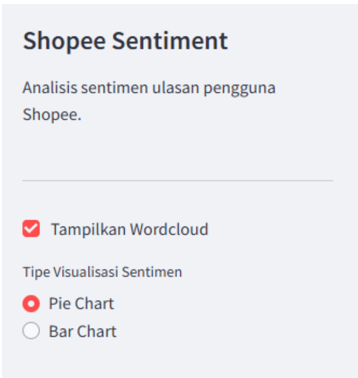

Figure 11. Sidebar Dashboard

Additionally, a word cloud displays the most frequently occurring words in reviews. These words are dominant, indicating users' primary focus on service speed and product price.

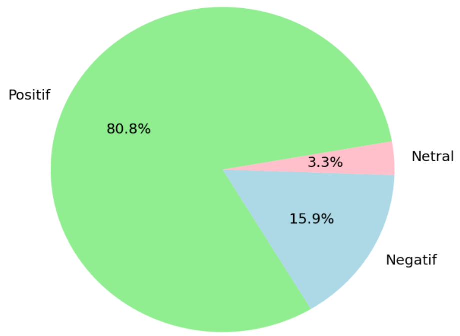

Figure 12. Visualization of Sentiment Distribution

Figure 13. Wordcloud Visualization

- **Data Exploration Tab**

  The second tab, Data Exploration, aims to provide a more in-depth analysis of the numerical and textual data in reviews. The first section of this tab displays the distribution of review scores, ranging from 1 to 5. The bar chart visualization shows that the majority of users gave the highest score (5), followed by those with a score of 4. Low scores (1–2) are relatively fewer, but still important for further analysis as they can provide insights into poor user experiences.

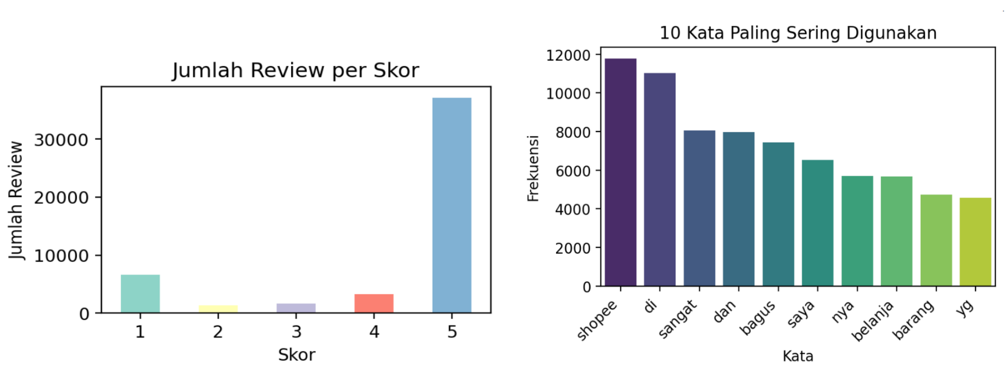

Figure 14. Visualization of Review Score Distribution and the 10 Most Common Words

- **Sentiment Test Tab**

  The third tab serves as a sentiment testing or inference feature. Users can type a new review sentence into the provided text field. After pressing the "Predict Sentiment" button, the system will process the text using a pre-trained Long Short-Term Memory (LSTM) model and then issue a prediction in the form of a category: Positive, Neutral, or Negative.

Figure 15. Sentiment Test

- **Tab Data Viewer**

  The last tab titled Data Viewer allows users to directly view Shopee review data in tabular form.

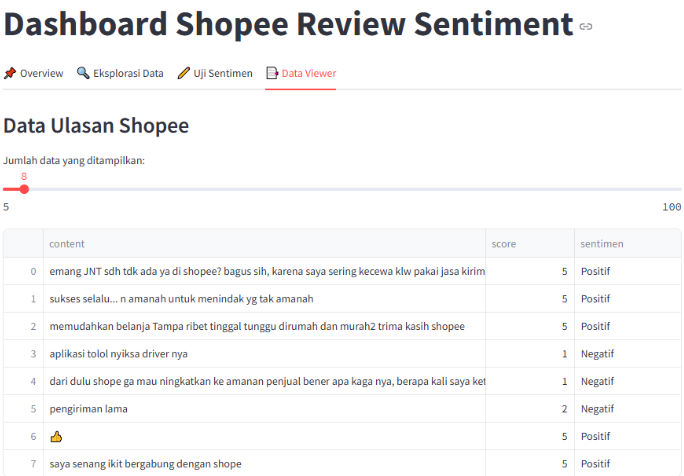

Figure 16. Data Viewer

The dashboard also features an interactive slider that allows users to adjust the number of rows of data displayed, from 5 to 100. This feature is useful for data validation and ensures that visualizations and sentiment classifications are based on accurate and clean data.

CONCLUSION

This study aims to compare the performance of Machine Learning algorithms (Random Forest and Support Vector Machine) with Deep Learning algorithms (Long Short-Term Memory) in sentiment analysis of Shopee app user reviews. Based on the evaluation results, the LSTM algorithm proved to have the best performance in classifying sentiment into three categories: positive, neutral, and negative. From the analysis of 50,000 reviews, it was found that the majority were positive sentiment (80.8%), while negative and neutral reviews were 15.9% and 3.3%, respectively. This indicates the dominance of user satisfaction with Shopee services. The LSTM model was then implemented into a Streamlit-based interactive dashboard that allows data visualization and direct sentiment testing. These results prove that the Deep Learning approach is not only technically superior, but can also be practically applied to support data-driven decision-making in the e-commerce sector.

The developed sentiment analysis model is capable of effectively classifying user opinions. For further development, manual or semi-supervised data labeling is recommended to improve the quality of the training data. Furthermore, the use of the latest Deep Learning algorithms such as BERT can be implemented to better capture language context and improve accuracy. Dashboard development can also focus on aspect-based sentiment analysis, allowing for more detailed analysis of user opinions on price, delivery, product quality, and customer service. Finally, model testing on datasets from other e-commerce platforms can be conducted to test the model's generalizability and adaptability more broadly.

REFERENCE

Asmiatun, A., Aryani, N. P., & Yuniarti, A. (2020). Interactive dashboard for sentiment data analysis. *Jurnal Teknologi Informasi dan Ilmu Komputer, 7*(2), 145–153. https://doi.org/10.25126/jtiik.2020721091

Bisong, E. (2019). Google Colaboratory. In *Building machine learning and deep learning models on Google Cloud Platform* (pp. 59–64). Apress. https://doi.org/10.1007/978-1-4842-4470-8_7

Chollet, F. (2019). *Deep learning with Python* (2nd ed.). Manning Publications. https://www.manning.com/books/deep-learning-with-python-second-edition

Das, A. K., & Behera, H. S. (2021). Exploratory data analysis and its importance in data science. *Journal of Emerging Technologies and Innovative Research (JETIR), 8*(6), 229–234. https://www.jetir.org/view?paper=JETIR2106173

GitHub Inc. (2023). *GitHub documentation*. https://docs.github.com

Keras Team. (2023). *Keras: Deep learning for humans*. https://keras.io

NLTK Project. (2023). *Natural Language Toolkit documentation*. https://www.nltk.org

Rizki, A. (2020). *Visualisasi data dalam analisis data besar: Konsep dan aplikasi*. Jakarta: Mitra Cendekia Press.

Scikit-learn Developers. (2022). *Scikit-learn: Machine learning in Python*. https://scikit-learn.org/stable/

Streamlit Inc. (2023). *Streamlit: The fastest way to build data apps*. https://streamlit.io

Sugiyono. (2020). *Metode penelitian kuantitatif, kualitatif, dan R&D*. Bandung: Alfabeta.

Sundararajan, R., & Ramesh, M. (2021). Real-time sentiment analysis for public policy evaluation using social media. *International Journal of Data Science and Analytics, 12*(4), 375–387. https://doi.org/10.1007/s41060-021-00262-9

TensorFlow. (2023). *TensorFlow: An end-to-end open source machine learning platform*. https://www.tensorflow.org

Wang, Y., Li, C., & Zhang, H. (2023). Deep learning-based sentiment analysis in e-commerce: A review. *Journal of Information and Data Management, 15*(1), 101–115. https://doi.org/10.1007/s42488-023-00060-1