ISSN: 2581-1274

Comparison of LSTM and Naive Bayes in Google Play Store App Review Sentiment Analysis

Endar Nirmala [a,1], Andri Fahmi [a,2]

[a] *University of Pamulang, Jl. Raya Puspitek, Kec. Pamulang, Kota Tangerang Selatan, 15310, Indonesia*
[1] *dosen00216@unpam.ac.id*;* [2] *dosen02816@unpam.ac.id*
** corresponding author*

ARTICLE INFO

Article history:
P*ublished*
March 31, 2026

Keywords:
Sentiment Analysis
Deep Learning
Google Play Store
LSTM
Naive Bayes

ABSTRACT

The development of mobile application technology has driven increased user interaction through digital reviews on the Google Play Store platform. The review contains opinions that reflect the user's level of satisfaction, experience, and complaints about the app. However, the large number of reviews and variations in language expressions make manual analysis inefficient and potentially subjective. The main problem in this study is how to determine the most effective sentiment classification model to accurately identify user emotional tendencies. This study aims to compare the performance of the Naive Bayes method as a conventional machine learning model with Long Short Term Memory (LSTM) as a deep learning model based on word order in analyzing the sentiment of user reviews of Google Play Store applications. The dataset used comes from Google Play Store Reviews and goes through a pre-process process that includes text cleanup, tokenization, stopword removal, and sentiment labeling based on rating scales. The Naive Bayes model is trained using the TF-IDF representation, while the LSTM model uses an embedding sequence with standardized input padding. Evaluation uses accuracy metrics and F1-score with a ratio of 80 : 20 to train and test data distribution. The test results showed that the Naïve Bayes model achieved an accuracy of 65.78% with an F1 score of 0.5589, while the LSTM only achieved an accuracy of 45.26% with an F1-score of 0.2077. Thus, Naive Bayes was established as the best model in this study.

I. Introduction

The development of digital technology in the past decade has brought great changes in human life. This transformation has significantly encouraged an increase in the use of mobile applications in various sectors, such as faster and more practical communication through instant messaging applications, more efficient transportation with the presence of ride-hailing services, entertainment that is more accessible through streaming platforms, education that is increasingly flexible with e-learning applications, and financial services that can now be done only through mobile devices in real-time [1]. Along with the rapid growth of the digital ecosystem, the Google Play Store is here as one of the largest application distribution platforms that provides thousands to millions of cross-category applications and is used by millions of users every day around the world [2]. In this ecosystem, each available app can receive reviews from users in the form of text that reflects their experience using the app. These reviews reflect the level of satisfaction, criticism, suggestions, and complaints, which can be an important indicator of the quality of the application. According to [3] user reviews play a strategic role in helping developers understand user needs and preferences, so that they can evaluate and develop feature updates that are more relevant and responsive to the problems faced by users. Positive reviews can be a signal of an app's success, while negative reviews can be a warning for developers to improve their app's performance to stay competitive.

The main problem in the analysis of app review sentiment lies in the diversity of language styles used by users. Reviews are not always delivered in formal language, but are often written using everyday conversational language, abbreviations, a mixture of Indonesian and English (code-

mixing), and even accompanied by emoticons or other visual expressions that reflect the author's emotions. In addition, the use of slang words or popular terms in certain circles also complicates the process of interpreting the text. This variation in language styles makes the task of sentiment analysis systems not only limited to word recognition, but also to understanding the meaning implied in the sentence structure. According to [4], classification models that are unable to understand the relationships between words in a sentence often result in misinterpretations of the polarity of the sentiment to be conveyed. In many cases, words that are literally positive in tone can change their meaning to negative when used in a sarcastic or ironic context.

Therefore, a text classification method is needed that is not only able to recognize words separately, but also to understand the structure and relationships between words in a single sentence. Traditional machine learning methods, such as Naïve Bayes, are still often used because the training process is fast and computationally efficient [5]. However, this traditional approach has limitations because it relies only on the frequency of words without paying attention to the order of the words, so its accuracy tends to decrease when tested on data that has a complex linguistic structure. On the other hand, deep learning models such as Long Short-Term Memory (LSTM) can learn the context and dependency of the word sequence in a sentence more effectively [6]. Therefore, it is necessary to perform a performance comparison between the Naïve Bayes model and LSTM to determine the most optimal approach in the analysis of the sentiment of user reviews of applications.

Various previous studies have attempted to apply the classification model to the sentiment analysis of mobile app reviews. According to [7], a study comparing Artificial Neural Network (ANN), Support Vector Machine (SVM), and LSTM on 33,000 Google Play Store reviews showed that LSTM provides the highest accuracy compared to other models. Research conducted by [8] on Snapchat app reviews in Indonesia using Naïve Bayes and SVM found that traditional models are still competitive when the amount of data is limited. Furthermore, according to [9], the comparison between CNN and RNN for text sentiment analysis shows that RNN-based models, such as LSTM, are superior in understanding long and contextual sentence structures. A systematic review conducted by [10] of 180 mobile app review analysis studies concluded that the selection of the right model depends heavily on the complexity of the text and the purpose of the analysis. In addition, a study by [11] on a review of the Twitter application compared CNN and LSTM and reported that LSTM was able to achieve an accuracy above 80% after going through the optimal pre-processing stage of data.

From these various studies, it can be concluded that deep learning methods tend to produce better performance than traditional methods, especially in data that has a strong semantic context. However, according to [12], there is still limited research that directly compares the LSTM and Naïve Bayes models using the latest version of the Google Play Store dataset. In addition, some previous studies have only shown accuracy metrics without considering other metrics such as precision, recall, and F1-score, which are important, especially on unbalanced datasets. [13] also emphasized that modeling that is not accompanied by an effective parameter tuning process will result in suboptimal performance. Thus, there is a gap (GAP) in the study in terms of a comprehensive comparison between classical and deep learning models in the latest Google Play Store dataset with multi-metric reporting of evaluation and transparent tuning procedures.

Seeing this need, this study proposes a comparative approach between the Naïve Bayes model as a representation of classical machine learning methods and the Long Short-Term Memory (LSTM) model as a representation of deep learning methods based on the context of word sequences. The dataset used in this study is the Google Play Store Review Dataset (Updated 2024), which contains the latest reviews of various popular apps. The data will go through a pre-processing process that includes text cleanup, tokenization, stopword removal, and conversion of text into numerical representations using TF-IDF for Naïve Bayes and embedding for LSTM. The model will be tested using an 80:20 ratio of training and test data sharing, and hyperparameter tuning will be performed to obtain the best configuration. The purpose of this study is to compile a systematic sentiment analysis pipeline on app reviews on the Google Play Store, compare the performance of the Naïve Bayes and LSTM models based on accuracy, precision, recall, and F1-score evaluation metrics, and determine the best model that has a high generalization ability on user review data. Through this study, it is hoped that it can be found which model is superior between LSTM and Naïve Bayes in

analyzing user review sentiment. In addition, this research is expected to be a reference for application developers and researchers in the field of text analysis to choose the appropriate model based on the needs of computational accuracy and efficiency.

II. Method

This research method was systematically designed to compare the performance of two sentiment classification models, namely Naïve Bayes and Long Short-Term Memory (LSTM), in analyzing user reviews of applications taken from the Google Play Store. The following are the stages of the research used: The Following is an explanation of the steps of the method from the image above [14]:

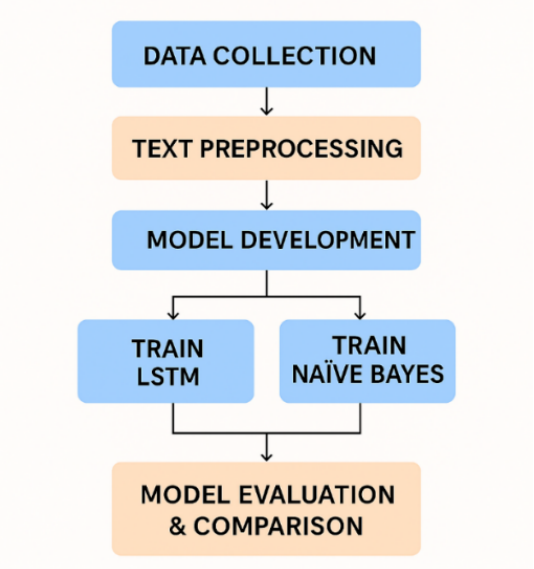

Fig.1. Research Stages

1. Data Collection
   Data collection in this study was carried out using secondary data obtained from the Google Play Store Review Dataset (Updated 2024) sourced from the Kaggle platform. The dataset consists of 12,500 user reviews, each labeled into three sentiment categories: positive, negative, and neutral, and includes main attributes such as review text and sentiment label. Quantitatively, the data represents textual user opinions with an assumed relatively balanced class distribution to support robust model training and evaluation. The selection of this dataset is based on its strong relevance to the research objective, namely sentiment analysis of mobile application reviews on the Google Play Store. To ensure data quality, a preprocessing and filtering stage was applied by removing reviews with unclear sentence structures, excessive noise and ambiguous sentiment labels. As a result, only reviews with clear linguistic structure and valid sentiment annotations were retained for further analysis, ensuring the reliability and validity of the dataset used in this study.

2. Text Preprocessing
   Text Preprocessing aims to clean and prepare text for easy processing by classification models. This process includes several steps, such as cleaning (removing irrelevant punctuation, numbers, symbols, or characters), case folding (converting the entire text into lowercase), tokenization (breaking sentences into word by word), stopword removal (removing common words that do not have a significant contribution to meaning, such as "who", "and", "in"), and stemming/lemmatization (returns the word to its basic form). This stage results in a more structured text that is ready to be converted into a numerical representation for model training [15].

3. Train LSTM
   The Long Short-Term Memory (LSTM) method in this study is systematically described so that it can be replicated. The stages start from the text preprocessing including case folding, non-alphabetic character removal, and stopwords removal, then the text is converted into numerical sequences using *a* tokenizer and standardized in length through padding. The model is built with a sequential architecture consisting of an Embedding layer for word vector representation, followed by an LSTM layer that captures the contextual relationships between words. To prevent overfitting, dropouts are used, and the training process is optimized with the Adam optimizer, the loss categorical crossentropy function, and callbacks such as Early Stopping. Evaluations were conducted using accuracy, precision, recall, and F1-score metrics, allowing the model to be replicated consistently in similar studies.

4. Train Naive Bayes
   The next stage is the training of the Naive Bayes model as a traditional machine learning-based comparator model. Before the training, the preprocessed text is converted into numerical form using the Term Frequency-Inverse Document Frequency (TF-IDF) technique. Naive Bayes works based on the probability of a word appearing in a class of sentiments, assuming that each word is independent. This model was chosen as the baseline because it is simple, quick to train, and is often used in text classification tasks. The training process aims to produce a model that is able to provide sentiment predictions based on probabilistic patterns.

5. Model Evaluation and Comparison
   The last stage is Model Evaluation and Comparison, which aims to assess the performance of each model in classifying sentiment. Evaluations are conducted using metrics such as accuracy, precision, recall, and F1-score [16]. The results of the evaluation from the LSTM and Naive Bayes models were then compared to determine which model had a better performance in understanding the sentiment of user reviews of the Google Play Store application. If LSTM shows higher accuracy, then it can be concluded that deep learning models are more effective in understanding semantic context than traditional probabilistic models such as Naive Bayes.

III. Results and Discussion

The results of the analysis of the data processing process, model training, and model performance evaluation have been carried out according to the stages of the research methodology. The analysis was conducted to find out the extent to which the Naïve Bayes and Long Short-Term Memory (LSTM) models were able to classify the sentiment of user reviews on the Google Play Store into positive, neutral, and negative categories as well as the process of comparing the performance of the two models based on evaluation metrics that included accuracy, precision, recall, and F1-score.

A. Initial Sentiment Distribution

The image above shows the initial distribution of user review sentiment on the Google Play Store which has been categorized into three classes, namely Negative, Neutral, and Positive. From the graph, it is shown that reviews with Positive sentiment dominate the number of reviews with the highest number, followed by Neutral sentiment, while Negative sentiment has the lowest number. This indicates that most users give a favorable rating to the app they use, although there are still a number of neutral and negative reviews that reflect a mediocre experience or even user dissatisfaction. The distribution of sentiment in the dataset showed that the positive class dominated with about 5,700 reviews, followed by the neutral class with 4,300 reviews, and the negative class with about 2,500 reviews out of a total of about 12,500 data. This distribution indicates a class imbalance, where positive data has a larger proportion than other classes. This condition has the potential to affect the performance of the classification model, especially in increasing bias against the majority class. Therefore, the model used in this study needs to have good generalization skills in order to still be able to accurately classify minority classes such as negative sentiments.

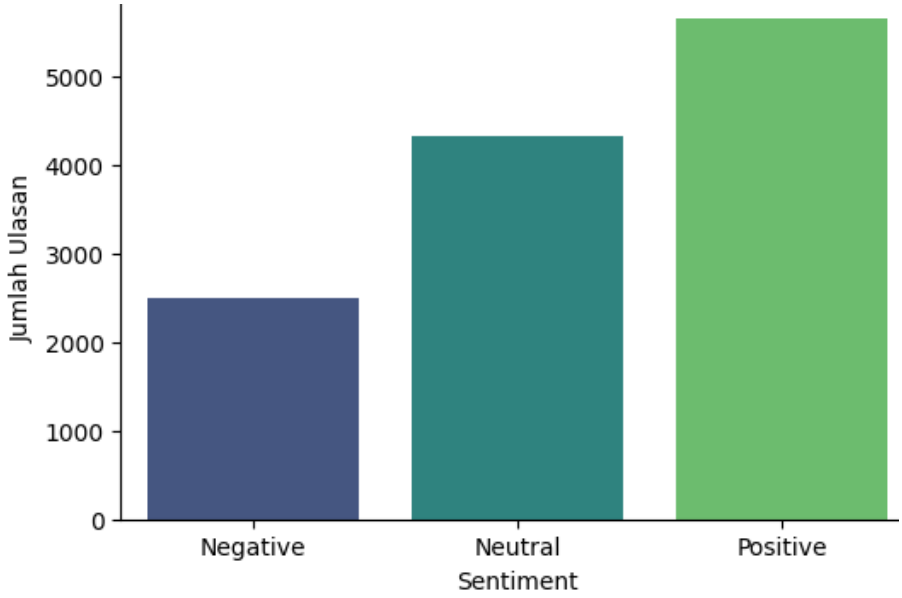

Fig.2. Initial Sentiment Distribution

B. Visualisasi Confusion Matrix Naïve Bayes and LSTM

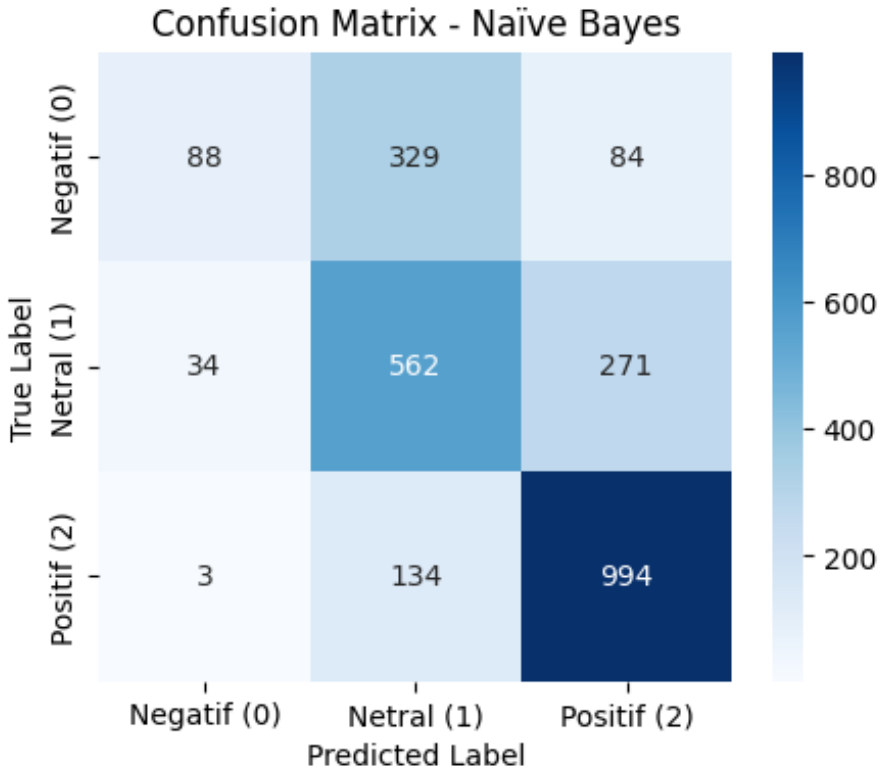

Fig.3. Visualizes Confusion Matrix Naïve bayes

The Confusion Matrix in the Naïve Bayes model shows that the model has a fairly good performance in classifying reviews with positive sentiment, as seen from the number of correct predictions in the positive class which reached 994 reviews, much higher than the other classes. However, in the neutral class, the model was still confused, with 562 reviews correctly classified, but 271 neutral reviews were classified as positive and 34 reviews predicted as negative. The same thing happened in the negative class, where only 88 reviews were correctly classified, while 329 negatives.

reviews were predicted as neutral and 84 were predicted as positive. These results indicate that the Naïve Bayes model tends to be biased towards positive and neutral sentiments, as well as less sensitive in distinguishing negative sentiments, which is likely due to the dominance of common words that appear more often in non-negative reviews and the model's limitations in understanding the emotional context of the text.

The above image of the confusion matrix in the LSTM model shows that all reviews of the three sentiment classes are Negative, Neutral, and Positive. This can be seen from the absence of predictions for the negative or neutral classes, while all prediction values were concentrated in the Positive column, each with 501 reviews for the negative class, 867 reviews for the neutral class, and 1131 reviews for the positive class. This condition indicates that the model experiences overfitting or imbalance in the learning process so that it tends to generalize the entire text as a positive sentiment, possibly due to the distribution of training data dominated by positive classes. Although the model produces a high accuracy value overall, these results suggest that the LSTM is unable to effectively distinguish between negative, neutral, and positive reviews, requiring adjustments such as class balancing, hyperparameter tuning, or model architecture improvements to make predictive performance more proportional to the entire sentiment class.

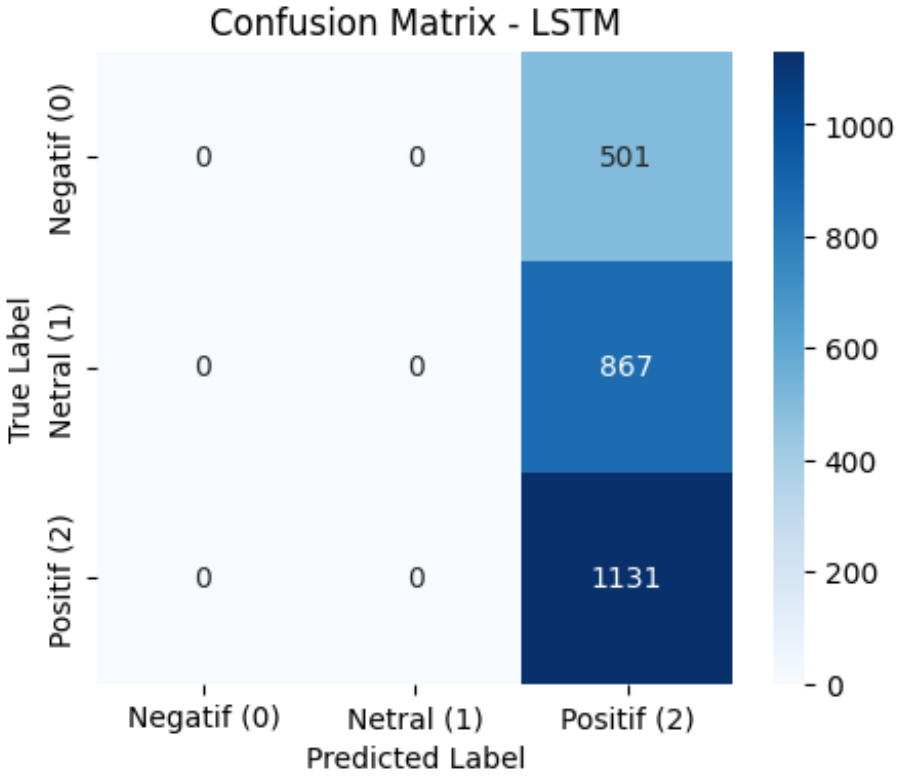

Fig.4. Visualizes Confusion Matrix LSTM

C. Visualisasi Training LSTM

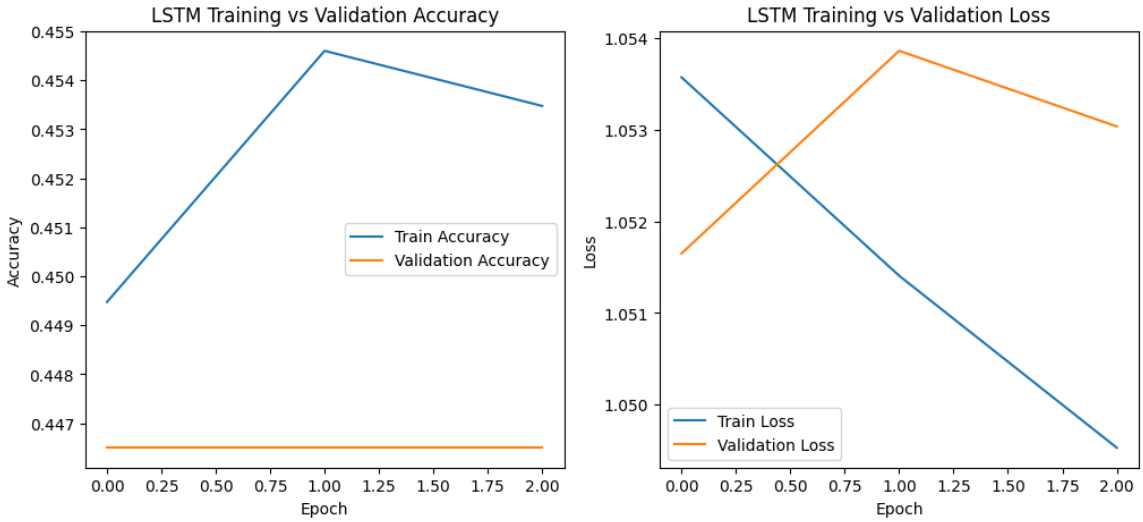

Fig.5 Visualizes Training LSTM

The training performance of the LSTM model based on accuracy and loss graphs on training data and validation data during the training process. In the LSTM Training vs Validation Accuracy graph, it can be seen that the accuracy of the training data has increased slightly from the first epoch to the second epoch, but the accuracy of the validation data tends to stagnate at the same point without any significant improvement. This indicates that the model understands only a little of the patterns in the training data but does not show a good generalization of the validation data. In the LSTM Training vs

Validation Loss chart, the loss value in the training data has consistently decreased, which shows that the model is getting better at minimizing errors during training. However, the loss in validation data actually increased in the second epoch before declining slightly again, signaling the possibility of the beginning of overfitting, where the model focuses too much on the training data and begins to lose the ability to generalize to new data.

D. Comparison of the Accuracy of the Naïve Bayes and LSTM Models

The results of the comparative evaluation of the model's performance showed that Naive Bayes had a much better performance than the LSTM in this dataset. Based on the comparison table, Naïve Bayes obtained an accuracy of 0.6578, a precision macro of 0.6630, a recall macro of 0.5675, and an F1-Score macro of 0.5589. In contrast, the LSTM model only achieved an accuracy of 0.4526, a precision macro of 0.1508, a recall macro of 0.3333, and an F1-Score macro of 0.2077. The F1-score comparison chart shows a significant difference, where Naïve Bayes notes an F1-score value of around 0.56, while the LSTM is only around 0.21. The low performance of LSTM was due to its failure to recognize negative and neutral classes, which was seen in the classification report with a precision and recall value of 0.00 for both classes. This indicates that LSTM is biased towards positive classes due to data distribution imbalances as well as the possible lack of hyperparameter tuning and an adequate number of training epochs. In contrast, Naïve Bayes shows more stable generalizations due to their simple nature and being able to handle word-frequency-based datasets. sentiment analysis of user reviews on the Google Play Store. Thus, under the conditions of this dataset, Naïve Bayes is more effectively used than LSTM.

```
=== HASIL EVALUASI LSTM ===
Akurasi: 0.4526

Classification Report (LSTM):
              precision    recall  f1-score   support

Negative (0)       0.00      0.00      0.00       501
 Neutral (1)       0.00      0.00      0.00       867
Positive (2)       0.45      1.00      0.62      1131

    accuracy                           0.45      2499
   macro avg       0.15      0.33      0.21      2499
weighted avg       0.20      0.45      0.28      2499
```

```
=== PERBANDINGAN KINERJA MODEL ===
```

	Model	Accuracy	Precision (Macro)	Recall (Macro)	F1-Score (Macro)
0	Naïve Bayes	0.657863	0.663045	0.567576	0.558948
1	LSTM	0.452581	0.150860	0.333333	0.207713

Fig.6. Model Performance Comparison

E. Word Cloud Analysis

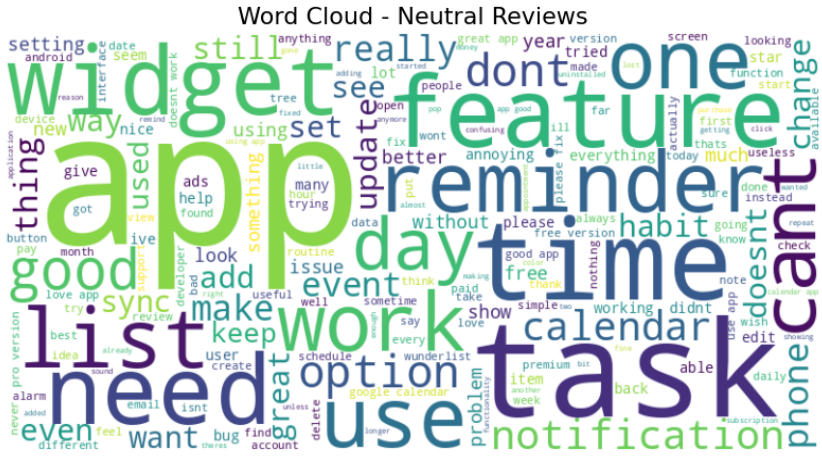

Fig.7. Neutral Reviews

Fig.8. Positive Reviews

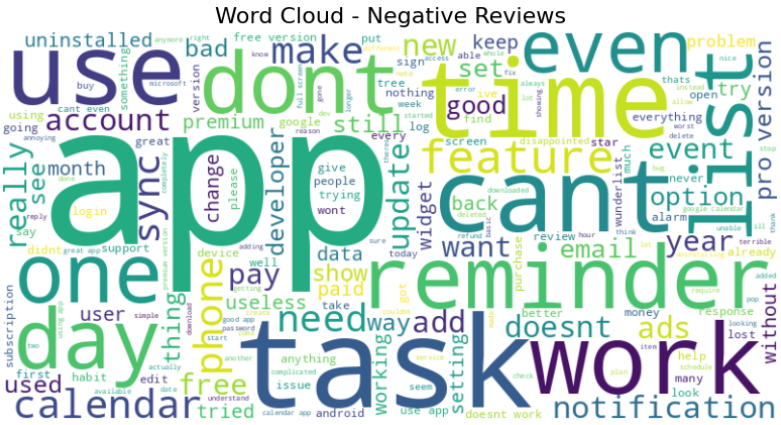

Fig.9. Negative Reviews

The three images above illustrate the words that appear most often in user reviews based on sentiment categories, namely negative, neutral, and positive. In negative reviews, words such as can't, don't, bad, crash, issue, and useless appear dominantly, indicating complaints related to the app's performance, inability to run features, and disappointing user experience. This indicates that users who leave negative reviews tend to express their dissatisfaction explicitly through frustrated words. In neutral reviews, words such as app, feature, task, widget, and free appear more often without being accompanied by strong emotional expressions, reflecting more informative or descriptive opinions, such as conveying the functionality of the app without giving extreme judgments. Meanwhile, Word Cloud's positive sentiment shows the dominance of words such as love, great, good, useful, and easy, which indicates users' appreciation for the quality of the application, ease of use, and the availability of features that help their daily activities. The difference in dominant words in each Word Cloud reinforces that linguistic expressions in review text are closely related to sentiment tendencies, thus supporting the relevance of research in comparing the performance of the Naïve Bayes and LSTM models in understanding language patterns in app user reviews on the Google Play Store.

IV. Conclusion

Based on the results of the evaluation, the Naïve Bayes model showed better performance than LSTM in the experimental configuration used, with an accuracy of 65.78% and an F1-score macro of 0.5589, while the LSTM only achieved an accuracy of 45.26% and an F1-score of 0.2077. The confusion matrix shows that Naïve Bayes is able to classify the three classes of sentiments more evenly, while LSTM tends to be biased towards positive classes and fails to recognize negative and neutral sentiments. However, these results need to be interpreted with caution because the low performance of LSTMs is likely due to suboptimal experimental conditions, such as a limited number of epochs. The number of epochs used in the LSTM model training in this study was relatively low, so the model did not have enough opportunities to optimally study patterns and contextual relationships from the data. This can be seen from the model's performance which is still low and has not been able to generalize well on the test data. Therefore, further experimentation is needed by increasing the number of epochs as well as making adjustments to other parameters such as learning rate, batch size, and model architecture to improve LSTM performance. Additionally, the use of techniques such as early stopping is still recommended to prevent overfitting during a longer training process. Therefore, it can be concluded that Naive Bayes is superior in this study, but only in the conditions and configurations used. These findings confirm that model selection and arrangement have a significant impact on the results of sentiment analysis, and show that simple methods can still provide competitive performance, while LSTMs have the potential to be better if further optimized.

References

[1] P. H. C. Samanmali and R. A. H. M. Rupasingha, “Sentiment analysis on google play store app users’ reviews based on deep learning approach,” *Multimed. Tools Appl.*, vol. 83, no. 36,

pp. 84425–84453, 2024, doi: 10.1007/s11042-024-19185-w.

[2] N. Singh and U. C. Jaiswal, "Sentiment analysis on playstore user reviews of healthcare apps using deep learning techniques," *Heal. Serv. Outcomes Res. Methodol.*, 2025, doi: 10.1007/s10742-025-00354-9.

[3] S. Sadiq, M. Umer, S. Ullah, S. Mirjalili, V. Rupapara, and M. Nappi, "Discrepancy detection between actual user reviews and numeric ratings of Google App store using deep learning," *Expert Syst. Appl.*, vol. 181, p. 115111, 2021, doi: https://doi.org/10.1016/j.eswa.2021.115111.

[4] I. Irawanto, C. Widodo, A. Hasanah, P. A. Dharma Kusumah, K. Kusrini, and K. Kusnawi, "Sentiment Analysis and Classification of Forest Fires in Indonesia," *Ilk. J. Ilm.*, vol. 15, no. 1, pp. 175–185, 2023, doi: 10.33096/ilkom.v15i1.1337.175-185.

[5] M. Saymon Ahammad, S. A. Sinthia, M. Muaj Chowdhury, N.-A.-A. Asif, and M. Nurul AfsarIkram, "Sentiment Analysis of Various Ride Sharing Applications Reviews: A Comparative Analysis Between Deep Learning and Machine Learning Algorithms," in *Computational Intelligence in Data Science*, M. L. Owoc, F. E. Varghese Sicily, K. Rajaram, and P. Balasundaram, Eds., Cham: Springer Nature Switzerland, 2024, pp. 434–448.

[6] D. Khoirunnisa and E. W. Pamungkas, "Sentiment Analysis on Shopee App User Feedback: A Comparison of LSTM and BiLSTM Algorithms," in *2025 International Conference on Smart Computing, IoT and Machine Learning (SIML)*, 2025, pp. 1–5. doi: 10.1109/SIML65326.2025.11081164.

[7] P. D. Rinanda and Mustakim, "Implementation of PNN, ANN And K-NN Algorithms on Indonesian Marketplace Reviews on Google Play Store," in *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)*, 2024, pp. 1070–1074. doi: 10.1109/ICETSIS61505.2024.10459477.

[8] A. B. Muzayyanah, R. E. Pawening, and Z. Arifin, "Analisis Sentimen Pada Ulasan Aplikasi Ehadrah Di Google Playstore Menggunakan Support Vector Machine (Svm)," *IDEALIS Indones. J. Inf. Syst.*, vol. 7, no. 2, pp. 258–266, 2024, doi: 10.36080/idealis.v7i2.3250.

[9] S. Bodapati, H. Bandarupally, R. N. Shaw, and A. Ghosh, "Comparison and Analysis of RNN-LSTMs and CNNs for Social Reviews Classification," in *Advances in Applications of Data-Driven Computing*, J. C. Bansal, L. C. C. Fung, M. Simic, and A. Ghosh, Eds., Singapore: Springer Singapore, 2021, pp. 49–59. doi: 10.1007/978-981-33-6919-1_4.

[10] A. Chader, L. Hamdad, and A. Belkhiri, "Sentiment Analysis in Google Play Store: Algerian Reviews Case," in *Modelling and Implementation of Complex Systems*, S. Chikhi, A. Amine, A. Chaoui, D. E. Saidouni, and M. K. Kholladi, Eds., Cham: Springer International Publishing, 2021, pp. 107–121.

[11] U. D. Gandhi, P. Malarvizhi Kumar, G. Chandra Babu, and G. Karthick, "Sentiment Analysis on Twitter Data by Using Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM)," *Wirel. Pers. Commun.*, 2021, doi: 10.1007/s11277-021-08580-3.

[12] E. M. C. Trecca, A. Lonigro, M. Gelardi, B. Kim, and M. Cassano, "Mobile Applications in Otolaryngology: A Systematic Review of the Literature, Apple App Store and the Google Play Store," *Ann. Otol. Rhinol. \& Laryngol.*, vol. 130, no. 1, pp. 78–91, 2021, doi: 10.1177/0003489420940350.

[13] P. Gupta and S. Srivastava, "Exploring customer attitude towards neo-banking apps: a thematic analysis using Google Play and Apple App Store reviews," *Qual. Res. Financ. Mark.*, 2025, doi: 10.1108/QRFM-07-2024-0198.

[14] M. Hadwan, M. Al-Sarem, F. Saeed, and M. A. Al-Hagery, "An Improved Sentiment Classification Approach for Measuring User Satisfaction toward Governmental Services' Mobile Apps Using Machine Learning Methods with Feature Engineering and SMOTE Technique," *Appl. Sci.*, vol. 12, no. 11, 2022, doi: 10.3390/app12115547.

[15] A. A. Qureshi, M. Ahmad, S. Ullah, M. N. Yasir, F. Rustam, and I. Ashraf, "Performance evaluation of machine learning models on large dataset of android applications reviews,"

Multimed. Tools Appl., vol. 82, no. 24, pp. 37197–37219, 2023, doi: 10.1007/s11042-023-14713-6.

[16] S. Lina, M. Sitio, and N. Rofiq, "Classification of Creditworthy Customer Using Support Vector Machine Algorithm," vol. 10, no. 2, pp. 339–345, 2025, doi: 10.31572/inotera.Vol10.Iss2.2025.ID502.