Indonesian Journal of Data and Science

Volume 6 Issue 3
ISSN 2715-9936
https://doi.org/10.56705/ijodas.v5i3.365

Research Article

Sentiment Analysis of BRImo Reviews on Google Play Store Using SVM and KNN

Olivia Sutriani Jelni [1]; Made Leo Radhitya [2]; Gede Wirya Wardhana [3]; Ni Wayan Jeri Kusuma Dewi[4] ; Ni Made Mila Rosa Desmayani [5]

[1] Institut Bisnis Dan Teknologi Indonesia, Denpasar, Indonesia, oliviajelni10@gmail.com
[2] Institut Bisnis Dan Teknologi Indonesia, Denpasar, Indonesia, leo.radhitya@instiki.ac.id
[3] Institut Bisnis Dan Teknologi Indonesia, Denpasar, Indonesia, wiryawardhana86@instiki.ac.id
[4] Institut Bisnis Dan Teknologi Indonesia, Denpasar, Indonesia, wayan.kusumadewi@instiki.ac.id
[5] Institut Bisnis Dan Teknologi Indonesia, Denpasar, Indonesia, milarosadesmayani@instiki.ac.id

Correspondence should be addressed to Olivia Sutriani Jelni; oliviajelni10@gmail.com

Received 10 November 2025; Accepted 15 December 2025; Published 31 December 2025

Abstract:

The rapid growth of digital banking has increased user interaction through mobile banking apps such as BRImo (Bank Rakyat Indonesia). Google Play Store reviews provide valuable insight into app quality, but their unstructured format makes manual analysis inefficient. This study analyzes user sentiment toward BRImo and compares the performance of Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) for sentiment classification. Reviews were collected using Google Play Scraper from May 2024 to May 2025, yielding 15,945 raw reviews. After cleaning (removing duplicates, symbols, links, emojis) and language filtering, 15,233 valid reviews remained. Sentiment labels were generated using two lexicon-based methods: INSET and VADER. Using INSET, the data consisted of 6,238 positive, 4,987 negative, and 4,383 neutral reviews, producing 11,225 reviews for modeling. Using VADER, 10,496 positive, 2,903 negative, and 1,834 neutral reviews were obtained, totaling 13,399 reviews. Datasets were split into 80% training and 20% testing with stratified sampling. Features were extracted using TF-IDF unigrams. Classification was performed using linear SVM and KNN, with the optimal K=3 selected via Grid Search. Models were evaluated using 5-fold cross-validation, reporting mean accuracy, precision, recall, and F1-score (macro-average for INSET; weighted-average for VADER due to class imbalance). Results show SVM consistently outperforms KNN, achieving 98.36% mean accuracy and 98.34% mean F1-score on INSET, and 95.59% mean accuracy and 95.56% mean F1-score on VADER. Overall, BRImo user sentiment is predominantly positive, and findings can guide developers in improving app stability and quality.

Keywords: Sentiment Analysis, BRImo, Application Performance, Reviews Google Play Store, SVM, KNN.

1. Introduction

Digital banking has become the primary channel for delivering financial services, including through mobile banking applications [1]. In Indonesia, the use of electronic and mobile banking services continues to increase, in line with the public's need for fast, convenient transactions via mobile devices [2]. Bank Rakyat Indonesia (BRI), one of the banking institutions with extensive national coverage, developed the BRImo application as part of its digital service [3]. According to the Google Play Store, BRImo has been downloaded more than 10 million times and has an average rating of 4.7, indicating high user adoption. User reviews of BRImo reflect their real experiences with these digital services and are an important source of information on the application's performance, including speed, stability, and technical issues [4]. However, the unstructured nature of reviews makes manual analysis inefficient and potentially leads to biased interpretations.

With the development of digital technology, user opinions and experiences with applications can now be easily disseminated through various digital platforms, including the Google Play Store [5]. However, free-text reviews make

manual analysis inefficient, requiring an automated approach. Sentiment analysis provides an automated approach for quickly and efficiently processing large-scale reviews to identify user opinions and sentiments towards digital applications [6]. Research on sentiment analysis of mobile banking applications has been conducted previously, but several limitations remain relevant for f urther study. Research by Kaur Sentiment Analysis Ofe-Banking Customer Review Susing NLP [7]. However, the study did not specifically address mobile banking applications in Indonesia nor compare multiple classification methods. This limitation highlights the need for comparative studies on sentiment analysis models applied to Indonesian mobile banking platforms. Research by Utama and Jamzuri compared SVM and KNN in an image-based classification task for verifying material orientation and found that SVM achieved slightly higher classification performance, while KNN was more efficient in terms of computation time; however, the study did not address sentiment analysis or mobile banking reviews and therefore does not represent the characteristics of BRImo user feedback [8]. In addition, the study by Ramadan et al. on BRImo compared Naïve Bayes and Support Vector Machine, providing valuable insights into model performance; however, K-Nearest Neighbors (KNN) was not included in the analysis, leaving room for further investigation with a broader range of classification methods [9].

These limitations indicate a significant research gap: the absence of a comprehensive study specifically comparing the performance of SVM and KNN in BRImo reviews, as well as an evaluation of the influence of two lexicon labeling schemes, INSET and VADER, on sentiment class distribution and classification results. To address this gap, this study uses BRImo user reviews from the Google Play Store as the primary dataset. It applies two standard classification algorithms for sentiment analysis: Support Vector Machine (SVM) and K-Nearest Neighbor (KNN). The objectives of this study are to (1) automatically identify and classify user sentiment toward the BRImo application, and (2) compare the performance of the two algorithms in predicting sentiment based on standard evaluation metrics such as accuracy, precision, recall, and F1-score. Based on these objectives, this study focuses on answering two main questions: (1) which algorithm performs better on BRImo sentiment classification and (2) how the differences in the characteristics of the two lexicons affect the sentiment class distribution and final evaluation metrics.

This research contributes by providing a systematic comparison of SVM and KNN performance on a real-world dataset of BRImo user reviews using two lexicon labeling approaches commonly employed in Indonesian-language sentiment analysis. In addition, the findings offer practical value for application developers by highlighting how algorithm and lexicon selection influence classification results, supporting data-driven improvements for application stability and user experience. This study focuses on Indonesian-language reviews collected from Google Play Store between May 2024 and May 2025 and limits the analysis to technical aspects of application performance.

2. Method

This study applies a Comparative Quantitative Research Design using Text Mining and machine learning. The review dataset is pre-processed to prepare it for analysis, then divided into training and testing data. Two algorithms were used for classification: Support Vector Machine (SVM) and K-Nearest Neighbors (K-NN) to classify each review as positive or negative. The research stages are presented in the form of a chart in **Figure 1**, as follows;

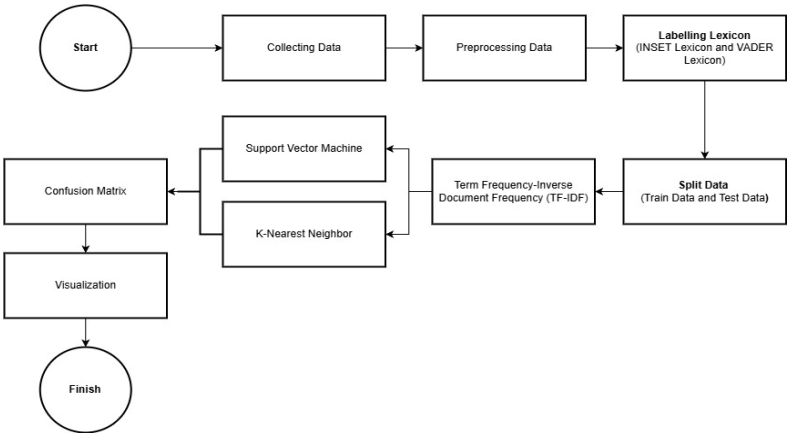

Figure 1. Research Flow

The flowchart illustrates the sentiment analysis research process, starting with the collection of BRImo app reviews, followed by preprocessing that includes cleaning, case folding, tokenization, normalization, stopword removal, and stemming. After the text is cleaned, sentiment labeling and TF-IDF weighting are performed to convert the data into numerical form. The data is then split into training and test sets before being classified using the SVM and KNN algorithms. The prediction results are evaluated using a confusion matrix, which is visualized for straightforward interpretation, and the research concludes with a summary.

Data selection

BRImo app review data was collected from the Google Play Store using the Google Play Scraper library. The reviews collected span May 2024 to May 2025, and the primary focus of this data collection is app-related, particularly login issues, transfer processes, and overall app performance. The filters used included collecting reviews in Indonesian, and the maximum data collection limit was set at 200,000 reviews to ensure that all reviews from May 2024 to May 2025 were covered without any omissions. Each collected review included the review text, rating, date, and user ID. A total of 15,945 raw reviews were obtained. After duplication removal and language filtering, the number of valid reviews was 15,233.

Table 1. Data Scraping Result

Category	Total Data
Scraping Result	15,945
After Remove Duplikat	15,233

To ensure the quality of the dataset, a gradual, structured data-cleaning process was carried out. These stages included removing irrelevant characters or symbols, removing links, filtering emojis, and detecting and removing duplicate data, ensuring each stored review was truly unique. This step not only reduced noise in the data but also prepared the dataset for optimal subsequent sentiment analysis. After the entire cleaning process was completed, the number of reviews meeting the quality standards was reduced to 15,233. A comparison of the data volume before and after cleaning is shown in **Table 1**.

Preprocessing

Preprocessing is the initial stage of text processing that converts unstructured raw data into a more organized format that is easier for the system to understand. Preprocessing is carried out systematically in the Google Colab environment, which provides access to the latest Python libraries for text processing. All stages are applied consistently to the BRImo review dataset, ensuring the text is in a structured format and ready for use in feature extraction and classification.

Before classification, the raw review data were preprocessed to improve data quality: irrelevant elements such as URLs, hashtags, user mentions, emoticons, and punctuation were removed to reduce noise and ensure uniform text representation prior to feature extraction and sentiment classification [10], [11]. Case folding was applied by converting all characters to lowercase to standardize text format and facilitate further analysis [12]. Tokenization was performed using a simple space-based approach through the tokenize() function, which split each review into individual word tokens while maintaining the natural structure of user reviews on Google Play Store [13]. Normalization was carried out using a custom dictionary in Excel format that was loaded into Python and converted into a dictionary structure; each token was checked against this dictionary and replaced with its standardized form if available, while unmatched tokens were retained, ensuring a consistent and replicable normalization process [14]. Stopword removal employed the Indonesian stopword list from NLTK, which was extended with commonly used expressions in application reviews such as "*yah*", "*oke*", "*kok*", "*nih*", and "laih" to eliminate words with minimal sentiment relevance [15]. Finally, stemming was conducted using Sastrawi, a widely used Indonesian stemming library, to convert words into their basic forms so that different variations such as "*menghubungi*", "*dihubungi*", and "*hubungan*" were treated as a single feature during classification [16].

Data Labeling

Sentiment labeling was performed using two lexicons, namely INSET for Indonesian texts and VADER for English texts. Reviews classified as neutral were removed because they were generally informative statements without explicit opinions, which could have led to ambiguity in the labeling process and reduced model consistency. In addition, the distribution of neutral classes is often unbalanced, which can introduce bias into the training process. By focusing the analysis on two main classes positive and negative, this study can assess users' evaluative tendencies more clearly and maintain the simplicity and stability of modeling, in line with standard practices in review-based sentiment analysis studies.

INSET Lexicon

In the INSET approach, each review is first converted to lowercase and split into tokens, then the label_sentiment() function counts the number of words that appear in the positive and negative lists. The difference between the number of positive and negative words yields a polarity score, which is used to determine the sentiment label: values greater than 0 indicate positive sentiment. In contrast, values less than 0 are categorized as negative [17], [18].

Table 2. Sentiment Data Distribution

Sentiment Category	Amount of Data
Positive	6,238
Negative	4,987
Netral	4,383
Total Initial Data	15,233
Total Data Used	11,225

Table 3. Labeling Result INSET Lexicon

Stemming	Polarity Score	Sentiment_Label
brimo masuk coba masuk salah username kata sandi gara coba masuk kali suka pakai aplikasi rumit	-7	Negatif
pangkas dana transaksi nyaman deh pakainya	3	Positif
lancar moga amanah layan	2	Positif

Based on the data distribution **Table 2**, of the 15,233 initial reviews, 6,238 were identified as positive sentiment and 4,987 as negative. Meanwhile, 4,384 reviews were classified as neutral and excluded from the analysis, as the research design focused on two sentiment classes. Thus, the amount of data used in the modeling stage was 11,225 reviews.

VADER Lexicon

VADER labeling begins by translating all reviews into English using deep_translator through the GoogleTranslator (source='id', target='en') function, where the translation process is managed by the translate_tweet() function, which cleans up the text, converts letters to lowercase, and handles errors to ensure consistent results [19].

Table 2. Translation Results

Text	Translate Result
brimo masuk coba masuk salah username kata sandi gara coba masuk kali suka pakai aplikasi rumit	brimo try to enter wrong username password because you try to enter you like using a complicated application
pangkas dana transaksi nyaman deh pakainya	cut transaction funds its comfortable to use
lancar moga amanah layan	smoothly hopefully you can serve it safely

After translation, sentiment is calculated using the vader_sentiment() function, which utilizes analyzer.polarity_scores() to obtain compound scores. Reviews with compound values ≥ 0.05 are labeled as positive,

while values ≤ –0.05 are labeled as negative. Neutral reviews are removed because the study only uses two classes. The final results are stored in a new column, making the process easy to audit and replicate [20].

Table 5. Sentiment Data Distribution

Sentiment Category	Amount of Data
Positive	10,496
Negative	2,903
Netral	1,834
Total Initial Data	15,233
Total Data Used	13,399

Table 6. Labeling Result VADER Lexicon

Engslish Text	Compound_Score	Sentiment_Label
installing the new update failed please be kind	0.34	Positif
please brimo transfer failed balance is missing destination number is not available please send a little money please transfer its really hard to find money	-5,474	Negatif
the times you block brimo are wrong paswood likes the times you are wrong immediately block doing the customer service brimos customer service is near the name rarely opens brimo forgets paswood try blocking	-8957	Negatif

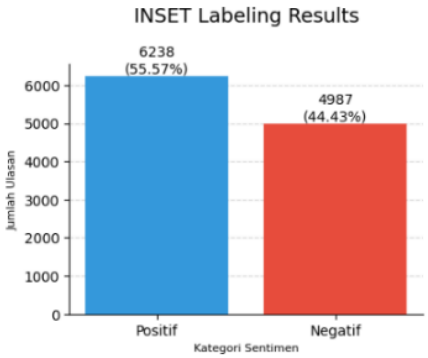

Figure 2. INSET Labeling Result

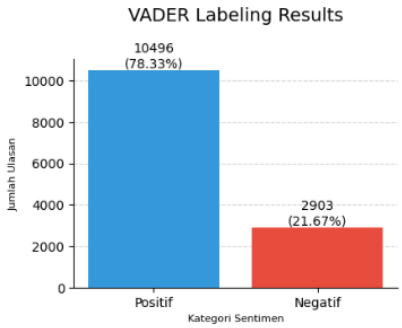

Figure 3 VADER Labeling Result

The figure shows the difference in sentiment distribution between the INSET and VADER lexicons. INSET produces more balanced labeling, with 55.57% positive reviews and 44.43% negative reviews. In contrast, VADER tends to label far more reviews as positive, at 78.33%, compared to 21.67% negative. This difference confirms that each lexicon has distinct characteristics, leading to distinct sentiment distributions even when using similar datasets.

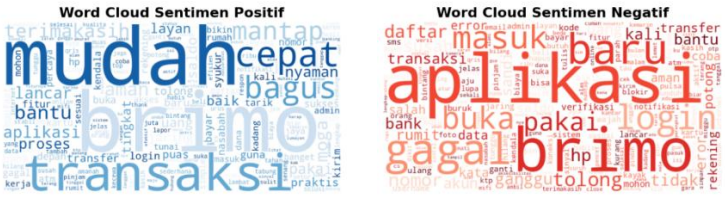

Figure 4. INSET Lexicon Word Cloud

Figure 5. VADER Lexicon Word Cloud

Split Data

Before performing classification using the Support Vector Machine and K-Nearest Neighbor methods, the review data must be split into training and test sets. The data is split using train_test_split with an 80% training and 20% test split. The division is stratified to maintain class balance, and random_state=42 is used to ensure consistent results [21].

TF-IDF

The TF-IDF weighting process in this study was carried out using the TfidfVectorizer class from the scikit-learn library. This class converts preprocessed text data into numerical representations based on word frequency and document-level importance [21]. The steps involved include initializing the vectorizer, training the model on the training data (fit_transform), and transforming the test data (transform) to ensure consistent feature representation. In this study, TF-IDF was configured using unigram (ngram_range=(1,1)), min_df=1, no feature limit (max_features=None), L2 normalization, and lowercase conversion (lowercase=True).

Support Vectore Machine

Support Vector Machine (SVM) is a supervised learning algorithm widely used for classification tasks because it performs effectively on sparse, high-dimensional text data. SVM separates data based on the best-margin between classes, and to handle different patterns, SVM can use various kernel functions. In this study, an SVM with a linear kernel is used, which is suitable for text data because the relationships between features are generally linear after TF-IDF transformation [22]. The equation in SVM is shown in equation 1 or 2.

$$f(x) = w.x + b \quad (1)$$

Description:

w : Weight vector, perpendicular to the hyperplane.

x : Feature vectors from sample data (e.g., TF-IDF values from reviews).

b : Bias (intercept) or threshold.

$$f(x) = \sum_{i=1}^{m} a_i y_i \mathrm{k(x,xi)} + \mathrm{b} \quad (2)$$

Description:

w : The hyperplane parameter sought (the perpendicular line between the hyperplane line and the support vector point)

x	:	Support Vector Machine input data points
a_i	:	Weight valueof each data point
yi	:	Class label of sample data-i.(yi €{ -1, 1 for binary classification, for example -1 for Negative sentiment and +1 for Positive).
K(x,xi)	:	Kernel function
b	:	Hyperplane parameter to be searched (bias value)

In making decisions, SVM uses a kernel function K(xi, xd). In this research, sentiment classification uses the Radial Basis Function kernel equation shown in equation 3.

$$K(xi, x) = \exp(-Y\ |Xi - X\ |^2), \text{ Y} > 0 \quad (3)$$

The classification model was built using the Support Vector Machine (SVM) algorithm with a linear kernel through the SVC class from scikit-learn, with kernel='linear', C=1.0, and random_state=42 configurations to ensure reproducibility of results. Performance evaluation was performed using five-fold cross-validation through the cross_validate function, with four main metrics measured, namely accuracy, macro precision, macro recall, and macro F1-score, which were defined using make_scorer to ensure consistent calculations. The cross-validation process was run on the TF-IDF training data, and all evaluation results were stored in a dictionary structure for easy analysis. After the training process was complete, model parameters such as kernel type, C value, and number of support vectors were displayed to ensure configuration transparency and facilitate replication in future research.

K-Nearest Neighbor

The K-nearest neighbors (KNN) algorithm is a commonly 63 used nonparametric method that was proposed by Cover and 64 Hart in 1968 [23]. The KNN algorithm does not need to train a model in advance and is often selected to solve regression and classification problems. [24] KNN classifies new data points based on the majority class of its K nearest neighbors8. The distance between data points is calculated using a distance metric; in this study, the Euclidean distance. The equation in KNN is shown in equation 4.

Euclidean Distance Equation:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (xi - yi)^2} \quad (4)$$

Description:

$d(x, y)$	:	The Euclidean distance between two data points, x and y.
d_i	:	Euclidean distance between data i,
X_i	:	Feature value i of the first data (usually test data),
Y_i	:	Feature value i of the second data (usually training data),
N	:	Total number of features (attributes) in the data,
∑	:	Summation symbol.

The K-Nearest Neighbor (KNN) model was built using the KNeighborsClassifier class from scikit-learn with the Euclidean distance metric. Before training, feature data was normalized using StandardScaler to ensure that all feature values were on a comparable scale, as KNN is sensitive to differences in feature scales. The best K value was selected via Grid Search over 1 to 21, with 5-fold cross-validat ion to obtain the most stable performance. This process was performed using GridSearchCV with the accuracy scoring parameter (scoring='accuracy'). The search results showed

that the best K value was K=3, which produced the highest average accuracy. The best model was then used to make predictions on the normalized test data.

Confusion Matrix

A confusion matrix is a tool used to measure the performance of a classification method in estimating objects that are true or false. Confusion matrices are commonly used in two-class supervised learning to assess the accuracy of predictions [25]. A single confusion matrix metric is difficult to measure the merit of the model. Therefore, Precision, Recall, and F1-Score were used as the evaluation metrics for model performance in the research setting of this paper [26].

Table7. Confusion Matrix

		Prediceted Values	
Actual Values	*Positive*	*Positive*	*Negative*
		True Positive (TP)	*False Positive* (FP)
	Negative	*False Negative* (FN)	*True Negative* (TN)

Based on **Table 7**, performance measurement using a confusion matrix consists of four parts to identify a prediction, including True Positive (TP), which occurs when the model predicts a sample as positive and the prediction is correct. True Negative (TN) occurs when the model correctly predicts a sample as negative. A false positive (FP) occurs when the model predicts a sample as positive, even though it is actually negative. A false negative (FN) occurs when the model predicts a sample as negative, but the sample is actually positive. Using the confusion matrix, we can find out how often the model makes correct or incorrect predictions for each class.

3. Result and Discussion

Result

The results of sentiment prediction for BRImo app reviews show that both the SVM and KNN algorithms tend to produce classifications dominated by positive sentiment across both datasets, namely the INSET Lexicon and the VADER Lexicon. In the INSET Lexicon dataset, the SVM algorithm predicted 1,232 positive reviews (54.88%) and 1,013 negative reviews (45.12%), while the KNN algorithm produced 1,386 positive reviews (61.74%) and 859 negative reviews (38.26%). The dominance of positive sentiment was even more apparent in the VADER Lexicon dataset, where SVM predicted 2,109 positive reviews (78.69%) and 571 negative reviews (21.31%). The KNN algorithm even showed the highest positive sentiment prediction with 2,291 reviews (85.49%) and 389 negative reviews (14.51%). The graphs of the prediction results are shown in **Figures 6** and **7**.

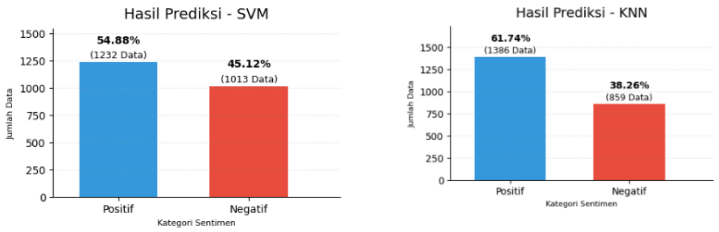

Figure 6. SVM and KNN Sentiment Prediction Results on INSET

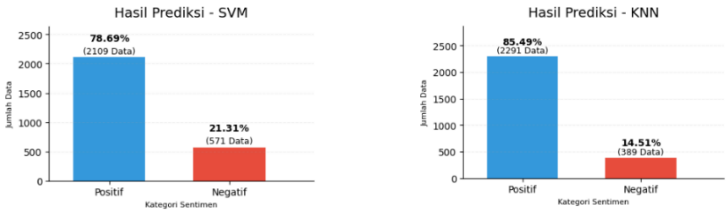

Figure 7. SVM and KNN Sentiment Prediction Results on VADER

The sentiment classification results based on the INSET and VADER Lexicons indicate that positive sentiment dominates user reviews of the BRImo application. In the INSET dataset, SVM produced 1,232 positive reviews (54.88%) and 1,013 negative reviews (45.12%), while KNN produced 1,386 positive reviews (61.74%) and 859 negative reviews (38.26%). On the VADER dataset, SVM predicted 2,109 positive reviews (78.69%) and 571 negative reviews (21.31%), while KNN produced 2,291 positive reviews (85.49%) and 389 negative reviews (14.51%). When comparing the two lexicons, SVM showed a more balanced distribution in the INSET data, while KNN tended to produce a higher proportion of positive results. In the VADER data, both models showed a dominance of positive sentiment, reflecting the lexicon's positive polarity. Overall, differences in lexicon characteristics affect prediction proportions, but both still illustrate that users generally have a positive assessment of the BRImo application.

Model Evaluation

After labeling both datasets, the two sentiment-class datasets were used to train and test the SVM and KNN models. The performance of each algorithm was evaluated using standard metrics such as accuracy, precision, recall, and F1-score to assess the consistency of the models in recognizing sentiment patterns. The evaluation results are presented in a table to provide a structured overview of each model's performance on both datasets.

Evaluation Results on the INSET Dataset

Evaluation of the model on the INSET dataset using the macro average metric without a weighted scheme, because the class distribution is relatively balanced: 6,238 positive reviews (55.5%) and 4,987 negative reviews (44.5%). A difference of around 11% is still commonly considered a balanced dataset, so using a weighted average is not necessary. Under these conditions, the macro metric is considered more appropriate because it gives equal weight to each class. The evaluation results are presented in the following **Table 8**.

Table 8. Model Evaluation Results on the INSET Dataset (Macro Metrics)

Model	Accuracy (Mean ± SD)	Precision (Macro Mean ± SD)	Recall (Macro Mean ± SD)	F1-Score (Macro)
SVM	93.36% ± 0.36%	98.31% ± 0.37%	98.38% ± 0.36%	98.34% (95% CI: 97.89% - 98.80%)
KNN (K=3)	79.15% ± 0.98%	80.43% ± 0.96%	77.74% ± 1.05%	78.16% (95% CI: 76.82% - 79.49%)

The SVM model showed excellent performance on the INSET dataset, with an average accuracy (Mean ± SD) of 98.36% ± 0.36% and a Macro F1-score of 98.34% ± 0.37%. The very narrow 95% Confidence Interval (CI) for the F1-Score, namely (97.89% — 98.80%), indicates that this model is very stable. The high precision and recall values for both classes also indicate that the SVM is capable of handling text patterns in a relatively clean, stable dataset. In contrast, KNN showed much lower performance, with an average accuracy of 79.15% ± 0.98% and a Macro F1-score of 78.16% ± 1.07%. The 95% CI width for the KNN F1-Score is (76.82% — 79.49%), which is wider (2.67 percentage points) than the SVM CI (0.91 percentage points). This is consistent with KNN's tendency to perform poorly on high-dimensional data, such as TF-IDF representations, and indicates lower generalization stability than SVM.

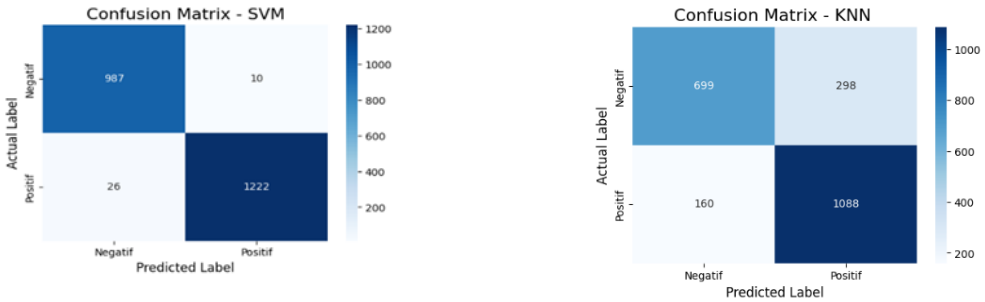

Figure 8. Results of the Confusion Matrix Evaluation of SVM

Figure 9. Results of the Confusion Matrix Evaluation of KNN

The classification performance evaluation, shown in the Confusion Matrices in **Figures 8** and **9**, indicates that the Support Vector Machine (SVM) algorithm outperforms the K-Nearest Neighbor (KNN) algorithm on the INSET lexicon dataset. The SVM model achieves high accuracy, as evidenced by True Positive (TP) and True Negative (TN) values of 1222 and 987, respectively. In addition, the number of classification errors is relatively small, with 10 False Positives (FP) and 26 False Negatives (FN), reflecting good class separation. In contrast, the KNN model showed a decline in performance, with a lower TN value (699) and a significant increase in classification errors, particularly in FP (298) and FN (160). These findings indicate that SVM is more effective at forming an optimal separating hyperplane in high-dimensional feature space. In contrast, the distance-based method in KNN is more sensitive to noise and less efficient at handling complex data distributions.

Table 9. Misclassification Examples for SVM on the INSET Dataset

No	Text	Sentiment	Prediction	Types of Errors
1	*aplikasi bank buruk brimo bikin kesal jaring bagus akun brimo foto terik matahari error nomor hp aktif pasang hp internet hidup tolong baik*	Positif	Negatif	FN
2	*aplikasi bagus mudah bolak balik uang transfer muka makasih brimo*	Negatif	Positif	FP

Table 3. Misclassification Examples for KNN on the INSET Dataset

No	Text	Sentiment	Prediction	Types of Errors
1	*aplikasi guna aman cuman kadang macet*	Positif	Negatif	FN
2	*aplikasi susah daftar kemarin gagal gagal pahal sinyal bagus sungguh bikin kecewa*	Negatif	Positif	FP

Based on the classification error examples in the **Table 9**, prediction errors generally occur in reviews that contain multiple polarities within a sentence, use subjective words, or have sentence structures that are not explicit. This shows that lexicon-based models tend to be more sensitive to keywords than to the overall sentence context. Therefore, classification errors appear more frequently in reviews that are ambiguous, emotional, or contain conflicting meanings.

Evaluation Results on the VADER Dataset

Evaluation of the VADER dataset uses the weighted average metric because its class distribution is highly unbalanced. Of the total data, 10,496 (78.3%) are positive reviews, and 2,903 (21.7%) are negative reviews. This imbalance of more than 50 percent places the VADER dataset in the heavy class imbalance category, so using the macro metric alone is insufficient to capture the model's overall performance. In such conditions, the weighted average provides a more realistic picture because each class is weighted by the amount of data it contains. This metric prevents evaluation results from being biased by the dominance of the majority class. The results of the model evaluation on the VADER dataset are shown in the following table:

Table 4. Model Evaluation Results on the VADER Dataset (Weighted Metrics)

Model	Accuracy (Mean ± SD)	Precision (Macro Mean ± SD)	Recall (Macro Mean ± SD)	F1-Score (Macro)
SVM	95.59% ± 0.51%	95.55% ± 0.52%	95.59% ± 0.51%	95.56% (95% CI: 94.92% - 96.20%)
KNN (K=3)	82.42% ± 0.39%	80.79% ± 0.54%	82.42% ± 0.39%	80.03% (95% CI: 79.36% - 80.69%)

The SVM model showed very stable performance on the unbalanced VADER dataset, achieving an average accuracy (Mean ± SD) of 95.59% ± 0.51% and a Weighted F1-score of 95.56% ± 0.51%. The 95% Confidence Interval (CI) value for the F1-Score is (94.92% - 96.20%). The nearly symmetrical figures for precision and recall indicate that SVM can handle positive class dominance without sacrificing its ability to recognize minority (negative) classes. This is reasonable because SVMs are generally stronger in high-dimensional feature spaces and are not easily affected by imbalanced class distributions. In contrast, KNN performance declined again under extreme imbalance conditions, as seen in its average accuracy of 82.42% ± 0.39% and Weighted F1-score of 80.03% ± 0.54%. The 95% CI range for KNN's F1-Score is (79.36% - 80.69%). The CI ranges of the two models do not overlap, providing statistical evidence of SVM's superiority. These results confirm that the weighted metric is the appropriate evaluation metric for the VADER dataset and, at the same time, show that SVM is far better at maintaining prediction consistency (stability) than KNN on uneven data.

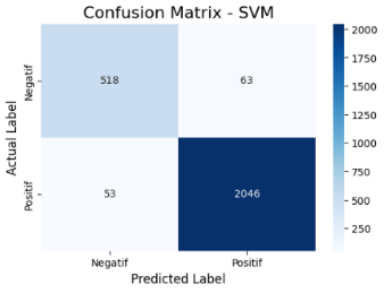

Figure 10. Results of the Confusion Matrix Evaluation of SVM

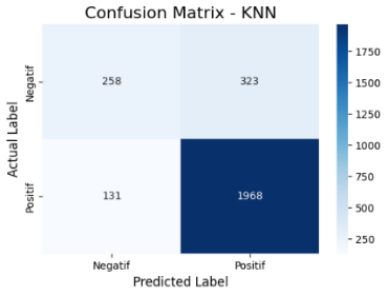

Figure 11. Results of the Confusion Matrix Evaluation of KNN

Performance evaluation using the Confusion Matrix shows a striking difference between the SVM and KNN algorithms. SVM shows far superior results, reflected in the high number of True Positives (TP) (2046) and True Negatives (TN) (518), as well as low False Positives (FP) (63) and False Negatives (FN) (53) values. In contrast, KNN experienced a significant decline in performance, particularly evident in the increase in errors, with FP reaching 323 and FN reaching 131. This condition confirms that SVM is capable of finding a more optimal separating hyperplane, resulting in stronger generalization. On the other hand, the proximity-based method in KNN is more susceptible to noise and complex feature distributions, ultimately reducing the accuracy of the algorithm's predictions.

Table 5. Misclassification Examples for SVM on the VADER Dataset

No	Text	Sentiment	Prediction	Types of Errors
1	brimo is ready to borrow using brimo minimum balance for borrowing and adding trading capital	Positif	Negatif	FN
2	believe in saving brimo money you can pay in installments by bank pay legally deduct status auto grab fund deduct balance pay month in arrears	Negatif	Positif	FP

Table 6. Misclassification Examples for KNN on the VADER Dataset

No	Text	Sentiment	Prediction	Types of Errors
1	good application easy transactions be careful not to get cheated	Positif	Negatif	FN
2	good complicated difficult to login login arises usernem wrong password password confusing	Negatif	Positif	FP

Comparison of Model Stability and Performance

The evaluation of both models was based on Mean ± SD metrics, as well as 95% Confidence Interval (CI) ranges reported in **Table 8** and **Table 9**, to measure the stability of model generalization. In both datasets (INSET and VADER), the SVM model statistically significantly outperformed the KNN model

1) Average Performance (Mean F1-Score):

- INSET (Balanced Data) : SVM (98.34%) is far above KNN (78.16%)
- VADER (Unbalanced Data) : SVM (95.56%) is far above KNN (80.03%)

2) Stabilitas (95% Confidence Interval):

- INSET : The CI range of F1-Macro SVM (97.89% - 98.80%) has a width of only 0.91 percentage points. The CI range of KNN (76.82% - 79.49%) is much wider, at 2.67 percentage points.
- VADER : The CI range of F1-Weighted SVM (94.92% - 96.20%) has a width of 1.28 percentage points. The CI range of KNN (79.36% - 80.69%) has a width of 1.33 percentage point.

The CI ranges of both models (SVM and KNN) on both datasetsdo not overlap, which statistically proves the superiority of SVM. The consistent and narrow CI widthin SVM shows that this model has much stronger generalization powerand is more resistant to variations in the division oftraining and testing data, compared to KNN.

Discussion

The comparative results show that the SVM model provides superior performance compared to KNN on both datasets, as demonstrated not only by higher mean F1-scores, namely 98.34% on INSET and 95.56% on VADER, but also by better stability (narrower Standard Deviation and Confidence Interval values). The mean Precision, Recall, and F1-score values for SVM are consistently better, while KNN shows more varied performance and is sensitive to data distribution, especially in the unbalanced VADER dataset. The results of this study indicate that the Support Vector Machine (SVM) algorithm outperforms the K-Nearest Neighbors (KNN) algorithm in sentiment analysis of BRImo app reviews. These findings are consistent with the patterns reported in several previous studies. A study by Aprilianti *et al*. reported that SVM achieved the highest accuracy of 81.6%, outperforming Naïve Bayes (73.2%) and KNN (66.9%) in sentiment analysis of religious application reviews [27]. Research by Ichwani et al. also demonstrated the superiority of SVM, achieving approximately 96% accuracy, outperforming KNN, which obtained only about 77% accuracy, in Google Play Store–based Shopee application reviews [28]. In addition, Deo et al. found that SVM achieved accuracies of 85% on the *all_tweets* dataset and 70% on the *Financial Phrase Bank* dataset, outperforming KNN which obtained only 65% and 59%, respectively, in sentiment classification tasks [29]. The consistency of these results reinforces the findings of this study: SVM tends to provide more stable, accurate performance in high-dimensional text data processing, while KNN is more sensitive to variations in the distance metric.

The consistency of these results is also reinforced by several international studies that report that SVM generally outperforms KNN in sentiment analysis and text classification tasks [30], [31]. In this study, the performance of SVM was again higher because it can separate sparse, high-dimensional text features more effectively. In addition, using the INSET lexicon as the basis for labeling can introduce label noise, especially in reviews that use informal language or ambiguous contexts. In such conditions, SVM tends to be more resistant to label inaccuracies than KNN, which is sensitive to errors in its nearest samples, leading to more stable performance. Therefore, further research is recommended to add sentiment categories, expand the data range, utilize advanced models such as LSTM or BERT, and apply aspect-based sentiment analysis to ensure results are more comprehensive and relevant to the development of the BRImo application.

4. Conclusion

This study compares the performance of Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) algorithms for sentiment classification in user reviews of the BRImo application, using the INSET and VADER labeling methods. Based on the test results, SVM showed superior, consistent performance compared to KNN across both labeling approaches, making it a suitable recommendation as a basic model for sentiment analysis in digital banking services. The main limitations of this study are the exclusion of the neutral sentiment class and the reliance on lexicon-based labeling, which may lead to inaccurate labels in complex sentence contexts. In terms of implementation, the results of this study can serve as a reference for BRImo managers to monitor user opinions and support more accurate improvement of service quality. In the future, further research is recommended to integrate

transformer-based models, compile a special banking lexicon in Indonesian, and apply aspect-based sentiment analysis to uncover specific problems, such as login, transaction, and balance issues.

References

[1] R. Ranjan, "The Evolution Of Digital Banking: Impacts On Traditional Financial Institutions International Journal Of Progressive Research In Engineering Management And Science (Ijprems) (Int Peer Reviewed The Evolution Of Digital Banking: Impacts On Traditional Financial Institutions," *Artic. Int. J. Progress. Res. Eng. Manag. Sci.*, vol. 04, no. 09, pp. 753–763, 2024.

[2] T. E. Sebayang, D. B. Hakim, T. Bakhtiar, and D. Indrawan, "What Accelerates the Choice of Mobile Banking for Digital Banks in Indonesia?," *J. Risk Financ. Manag.*, vol. 17, no. 1, 2024, doi: 10.3390/jrfm17010006.

[3] H. Yohanes Jefrinus Bessy, Yenny, "Communication Managerial Skill For Business Communication On Employee Relations Studies," *J. Ekon.*, vol. 13, no. 2, pp. 541–554, 2024, doi: 10.54209/ekonomi.v13i02.

[4] F. A. Willa Fatika Sari, Rida Rahim, "Sentiment Analysis Using Big Data User Reviews on Mobile Banking Performance in Indonesia," *IAR J. Bus. Manag.*, vol. 4, no. 3, pp. 27–35, 2023, doi: 10.47310/iarjbm.2023.v04i01.028.

[5] C. Yang, L. Wu, C. Yu, and Y. Zhou, "A phrase-level user requests mining approach in mobile application reviews: Concept, framework, and operation," *Inf.*, vol. 12, no. 5, pp. 1–24, 2021, doi: 10.3390/info12050177.

[6] Y. Mao, Q. Liu, and Y. Zhang, "Sentiment analysis methods, applications, and challenges: A systematic literature review," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 36, no. 4, p. 102048, 2024, doi: 10.1016/j.jksuci.2024.102048.

[7] N. Kaur, "Sentiment Analysis of E-Banking Customer Reviews Using Nlp," *ShodhKosh J. Vis. Perform. Arts*, vol. 2, no. 2, pp. 458–465, 2021, doi: 10.29121/shodhkosh.v2.i2.2021.5743.

[8] E. Utama and E. Rudiawan Jamzuri, "Performance Comparison of Support Vector Machine (SVM) and k-Nearest Neighbors (kNN) in Verifying Material Orientation," *J. Appl. Comput. Sci. Technol.*, vol. 6, no. 1, pp. 17–22, 2025, doi: 10.52158/jacost.v6i1.1037.

[9] M. F. Ramadan, Martanto, A. R. Dikananda, and A. Rifa'i, "Comparison of Sentiment Analysis Models Enhanced by Naïve Bayes and Support Vector Machine Algorithms on Mobile Banking BRImo Reviews," *J. Artif. Intell. Eng. Appl.*, vol. 4, no. 2, pp. 677–686, 2025, doi: 10.59934/jaiea.v4i2.732.

[10] M. A. Palomino and F. Aider, "Evaluating-the-Effectiveness-of-Text-PreProcessing-in-Sentiment-AnalysisApplied-Sciences-Switzerland.pdf," *Mdpi*, vol. 12, p. 8765, 2022.

[11] B. S. Ainapure *et al.*, "Sentiment Analysis of COVID-19 Tweets Using Deep Learning and Lexicon-Based Approaches," *Sustain.*, vol. 15, no. 3, 2023, doi: 10.3390/su15032573.

[12] A. Arwan Sulaeman, M. Danny, S. Butsianto, and S. Pratama, "Sentiment Analysis on Social Media X (Twitter) Against ChatGBT Using the K-Nearest Neighbors Algorithm," *Brill. Res. Artif. Intell.*, vol. 4, no. 1, pp. 265–275, 2024, doi: 10.47709/brilliance.v4i1.4105.

[13] A. H. Sweidan, N. El-Bendary, and H. Al-Feel, "Sentence-Level Aspect-Based Sentiment Analysis for Classifying Adverse Drug Reactions (ADRs) Using Hybrid Ontology-XLNet Transfer Learning," *IEEE Access*, vol. 9, pp. 90828–90846, 2021, doi: 10.1109/ACCESS.2021.3091394.

[14] S. Nazir, M. Asif, M. Rehman, and S. Ahmad, "Machine learning based framework for fine-grained word segmentation and enhanced text normalization for low resourced language," *PeerJ Comput. Sci.*, vol. 10, no. 1, pp. 1–19, 2024, doi: 10.7717/peerj-cs.1704.

[15] D. Fatharani, E. Syahrul, and U. Gunadarma, "hybrid sentiment analysis of maxim app users using support vector machine and lexicon-," vol. 13, no. 3.

[16] Rianto, A. B. Mutiara, E. P. Wibowo, and P. I. Santosa, "Improving the accuracy of text classification using stemming method, a case of non-formal Indonesian conversation," *J. Big Data*, vol. 8, no. 1, pp. 1–16, 2021, doi: 10.1186/s40537-021-00413-1.

[17] R. Firdaus, I. Asror, and A. Herdiani, "Lexicon-Based Sentiment Analysis of Indonesian Language Student Feedback Evaluation," *Indones. J. Comput.*, vol. 6, no. 1, pp. 1–12, 2021, doi: 10.34818/indojc.2021.6.1.408.

[18] V. Bonta, N. Kumaresh, and N. Janardhan, "A Comprehensive Study on Lexicon Based Approaches for Sentiment Analysis," *Asian J. Comput. Sci. Technol.*, vol. 8, no. S2, pp. 1–6, 2019, doi: 10.51983/ajcst-2019.8.s2.2037.

[19] E. Elinda, H. Yuliansyah, and M. I. A. Latiffi, "Sentiment Analysis of the Sheikh Zayed Grand Mosque's Visitor Reviews on Google Maps Using the VADER Method," *Int. J. Adv. Data Inf. Syst.*, vol. 5, no. 1, pp. 71–84, 2024, doi: 10.59395/ijadis.v5i1.1320.

[20] A. Saoualih *et al.*, "Exploring the Tourist Experience of the Majorelle Garden Using VADER-Based Sentiment Analysis and the Latent Dirichlet Allocation Algorithm: The Case of TripAdvisor Reviews," *Sustain.*, vol. 16, no. 15, 2024, doi: 10.3390/su16156378.

[21] A. Febriani, Khotibul Umam, and Mokhammad Iklil Mustofa, "Implementation of Support Vector Machine for Classifying User Reviews on the Sentuh Tanahku Application," *J. Appl. Informatics Comput.*, vol. 9, no. 4, pp. 1551–1558, 2025, doi: 10.30871/jaic.v9i4.9832.

[22] R. Obiedat *et al.*, "Sentiment Analysis of Customers' Reviews Using a Hybrid Evolutionary SVM-Based Approach in an Imbalanced Data Distribution," *IEEE Access*, vol. 10, pp. 22260–22273, 2022, doi: 10.1109/ACCESS.2022.3149482.

[23] R. K. Halder, M. N. Uddin, M. A. Uddin, S. Aryal, and A. Khraisat, "Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications," *J. Big Data*, vol. 11, no. 1, 2024, doi: 10.1186/s40537-024-00973-y.

[24] H. Zhang, H. Niu, Z. Ma, and S. Zhang, "Wind Turbine Condition Monitoring Based on Bagging Ensemble Strategy and KNN Algorithm," *IEEE Access*, vol. 10, no. April, pp. 93412–93420, 2022, doi: 10.1109/ACCESS.2022.3164717.

[25] R. Setiyawan and Z. Mustofa, "Comparison of the performance of naive bayes and support vector machine in sirekap sentiment analysis with the lexicon-based approach," *J. Soft Comput. Explor.*, vol. 5, no. 2, pp. 122–132, 2024, doi: 10.52465/joscex.v5i2.367.

[26] L. Li, L. Yang, and Y. Zeng, "Improving sentiment classification of restaurant reviews with attention-based bi-gru neural network," *Symmetry (Basel).*, vol. 13, no. 8, 2021, doi: 10.3390/sym13081517.

[27] Heti Aprilianti, Khothibul Umam, and Maya Rini Handayani, "Comparative Study of SVM, KNN, and Naïve Bayes for Sentiment Analysis of Religious Application Reviews," *J. Appl. Informatics Comput.*, vol. 9, no. 3, pp. 920–927, 2025, doi: 10.30871/jaic.v9i3.9482.

[28] A. Ichwani and R. Gantino, "Sentiment Analysis of Marketplace Application Reviews Using Support Vector Machine (SVM) and K-Nearest Neighbors (KNN)," vol. 8, no. 2, 2025.

[29] T. K. Deo, R. K. Deshmukh, and G. Sharma, "Comparative Study of Performance of K Nearest Neighbor and Support Vector Machine Classifiers in Sentiment Analysis," *Int. Res. J. Eng. Technol.*, vol. 11, no. 2, pp. 501–506, 2024.

[30] M. K. Anam, T. A. Fitri, A. Agustin, L. Lusiana, M. B. Firdaus, and A. T. Nurhuda, "Sentiment Analysis for Online Learning using The Lexicon-Based Method and The Support Vector Machine Algorithm," *Ilk. J. Ilm.*, vol. 15, no. 2, pp. 290–302, 2023, doi: 10.33096/ilkom.v15i2.1590.290-302.

[31] V. Nurcahyawati and Z. Mustaffa, "Vader Lexicon and Support Vector Machine Algorithm to Detect

Customer Sentiment Orientation," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 9, no. 1, pp. 108–118, 2023, doi: 10.20473/jisebi.9.1.108-118.