SHM PUBLISHER
SURYA HIJAU MANFAAT

Journal of Information System Exploration and Research

https://joiser.shmpublisher.com

p-ISSN 2964-1160 | e-ISSN 2963-6361

Performance Comparison of Naive Bayes and Support Vector Machine Algorithms in Sentiment Analysis of TIX ID Application Reviews Using VADER Automatic Labeling

Zenia Kumala Rizka[1*], Jumanto[2]

[1,2] *Department of Computer Science, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Indonesia*

DOI: https://doi.org/10.52465/joiser.v4i2.3

Received 25 May 2026; Accepted 26 June 2026; Available online 29 June 2026

Article Info

Keywords:
Sentiment analysis;
TIX ID;
Naïve Bayes;
Support Vector Machine (SVM);

Abstract

This study compares the sentiment classification performance of Naive Bayes and Support Vector Machine (SVM). It uses 28247 user reviews for the Google Play Store app TIX ID collected from Kaggle. The reviews were first translated into English, then their sentiment was labeled using VADER. After completing text preprocessing, feature extraction via TF-IDF combined with 1-gram and 2-gram features, and class balancing through random oversampling, test results show that SVM achieved an accuracy of 93.45% and an F1-score of 93.78%, which outperforms Naive Bayes' respective scores of 90.90% accuracy and 91.72% F1-score. Experiments in this study found that the Support Vector Machine (SVM) outperformed Naive Bayes across all three evaluation metrics: precision, recall, and F1-score. This verifies that the approach consisting of VADER annotation, TF-IDF feature extraction, and SVM can effectively conduct sentiment analysis on mobile application reviews, and meets the needs of the industry.

1. Introduction

The rapid development of information technology and mobile applications has significantly changed the way people interact with digital services. In Indonesia, the increasing use of smartphones and internet access has encouraged the growth of various digital applications, including online ticket booking servicess [1], [2]. One of the most widely used movie ticket booking applications in Indonesia is TIX ID, which has received millions of downloads and a large number of user reviews on Google Play Store [3]. These reviews contain valuable information regarding user experiences, opinions, and satisfaction toward the services provided by the application.

User reviews play an important role in evaluating the quality of digital services because they reflect user perceptions directly [4]. However, the large volume of reviews and their unstructured text format

* ***Corresponding Author:***

Zenia Kumala Rizka,
Department of Computer Science, Faculty of Mathematics and Natural Sciences,
Universitas Negeri Semarang,
Sekaran, Gunungpati, Kota Semarang, Indonesia.
Email: zeniarizka12@students.unnes.ac.id

make manual analysis inefficient and time-consuming. Therefore, sentiment analysis is needed to automatically identify and classify user opinions into sentiment categories such as positive and negative.

Sentiment analysis is a computational technique for identifying the emotional polarity, whether positive or negative, expressed in a piece of text [5] . Sentiment analysis is a subfield of Natural Language Processing (NLP) that focuses on identifying and interpreting opinions, emotions, and attitudes expressed in textual data [6]. In recent years, various machine learning algorithms have been widely adopted for sentiment classification, particularly in the analysis of user-generated reviews on digital platforms. Among these methods, Naïve Bayes and SVM have consistently demonstrated strong performance in text classification tasks [7], [8]. SVM is particularly effective in handling high-dimensional textual data and constructing robust decision boundaries between classes [9], [10]. In contrast, Naïve Bayes is valued for its simple implementation, [11] computational efficiency, and competitive performance across a wide range of text mining applications.

Before classification can be performed, textual data generally undergo several preprocessing and feature extraction stages. One of the most commonly used feature extraction techniques is Term Frequency–Inverse Document Frequency (TF-IDF), which transforms text into numerical vectors by assigning weights based on the importance of words within a document and across the entire corpus [12]. Previous studies have shown that TF-IDF is effective in improving the performance of sentiment classification models [13]. Nevertheless, sentiment datasets often exhibit imbalanced class distributions, where one sentiment category contains substantially more instances than another. Such imbalance may negatively affect classification performance, particularly for minority classes [14]. To mitigate this issue, random oversampling can be employed to increase the representation of minority-class samples and improve model learning.

In addition, this study employs Valence Aware Dictionary and sEntiment Reasoner (VADER) as an automatic labeling approach. VADER is a lexicon- and rule-based sentiment analysis tool specifically designed to evaluate sentiment in social media content and online reviews [15]. Its ability to process informal language patterns, punctuation emphasis, and common expressions makes it suitable for analyzing user-generated textual data.

Considering these aspects, this research develops a sentiment analysis framework for TIX ID user reviews by integrating VADER-based automatic labeling, TF-IDF feature extraction, random oversampling, and classification using Naïve Bayes and SVM algorithms. The primary objective is to compare the effectiveness of both classifiers in distinguishing positive and negative sentiments expressed by TIX ID users. Through this comparison, the study seeks to provide empirical evidence regarding the suitability of these approaches for sentiment analysis of mobile application reviews. Although VADER, TF-IDF, random oversampling, Naïve Bayes, and SVM have been widely applied in sentiment analysis, their use on TIX ID user reviews has been rarely explored. This study compares the performance of Naïve Bayes and SVM using sentiment labels generated by VADER and validated through user star ratings. In addition, the combination of TF-IDF feature extraction and random oversampling was employed to address class imbalance and improve classification performance. The contribution of this research lies in providing a benchmark of Naïve Bayes and SVM on TIX ID reviews, evaluating the reliability of VADER-generated labels, and demonstrating that the proposed preprocessing and balancing approach can improve the accuracy of both classification models.

2. Method

This study followed a structured sentiment analysis workflow adapted from previous research and adjusted to suit the characteristics of TIX ID user reviews. The main stages included data collection, preprocessing, sentiment labeling, feature extraction, data balancing, classification, and performance evaluation. A summary of the research workflow is shown in Figure 1.

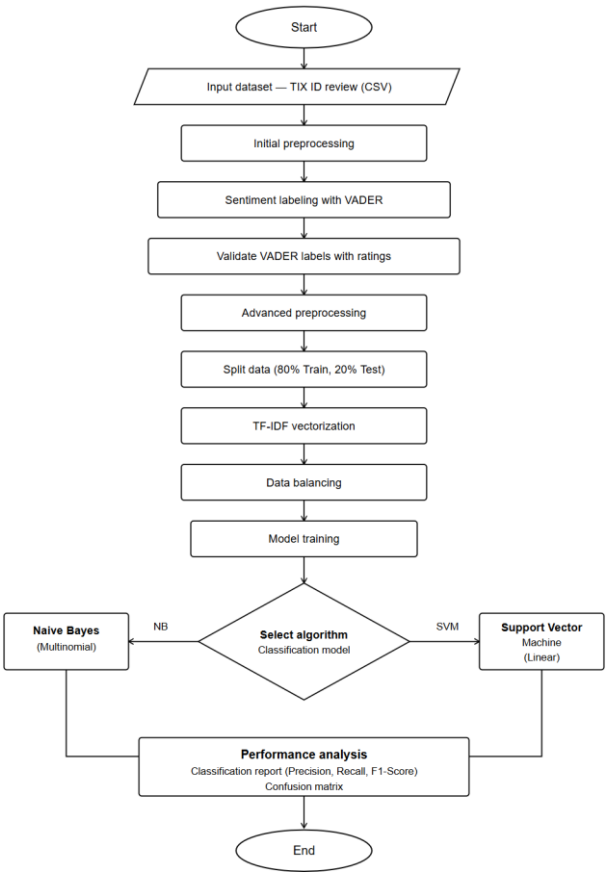

Figure 1. Research Workflow

2.1. Data Collection

The dataset used in this research is a publicly available collection of TIX ID user reviews from the Google Play Store, which was sourced from Kaggle [16]. It consists of 28,247 review entries stored in CSV format, with several attributes such as reviewId, userName, content, score, at, and appVersion. However, this study only utilized two main fields, namely the review text (English-translated content) and the rating score. The translated English reviews were specifically chosen because VADER sentiment analysis relies on an English-based lexical approach. The dataset used in this study is publicly available online and can be accessed through https://www.kaggle.com/datasets/ahmadseloabadi/tix-id-app-reviews-from-google-play-store. Table 1 presents an overview of the dataset by describing each of its attributes in detail.

Table 1. Description of Dataset Attributes

Attribute	Description
reviewId	A unique alphanumeric code automatically generated by the Google Play Store to identify each review.
userName	The display name of the Google account that submitted the review.
userImage	A URL linking to the reviewer's Google profile picture.
content	The original review text written by users, mostly in Indonesian and sometimes mixed with English (code-switching).
text	The English-translated version of the content field, used as the main input for VADER sentiment labeling and TF-IDF feature extraction since VADER relies on an English lexicon.
score	The star rating given by the user, ranging from 1 (lowest) to 5 (highest). In this study, it is used to validate the sentiment labels generated by VADER.
thumbsUpCount	The number of users who marked the review as helpful ("thumbs up").
reviewCreaatedVersion	The version of the TIX ID application installed by the user when submitting the review.
at	The timestamp indicating when the review was posted.
replyContent	The developer's response to the user review, if available; otherwise, it remains empty.
repliedAt	The timestamp showing when the developer's reply was posted.
appVersion	The application version recorded in the dataset, which may be the same as reviewCreatedVersion depending on how the data was collected.

2.2. Text Preprocessing

Preprocessing plays an important role in sentiment analysis because it converts raw textual data into a cleaner and more structured format that can be effectively processed by machine learning algorithms. The quality of preprocessing directly affects the performance of the sentiment classification model [17]. In this study, preprocessing was carried out in two stages. The first stage involved text cleaning, including the removal of URLs, non-alphabetic characters, and unnecessary whitespace, followed by case folding to convert all text into lowercase form. The second stage consisted of tokenization, stopword removal, and lemmatization. Tokenization was used to divide sentences into individual words, while stopword removal eliminated common words that carry little sentiment information, such as conjunctions and prepositions [18]. Finally, lemmatization was applied to transform words into their base forms while preserving their linguistic meaning and context.

2.3. Sentiment Labeling Using VADER

VADER is a lexicon-based sentiment analysis approach developed to assess sentiment in social media and online review texts. VADER is a rule-based model capable of handling characteristics of social media text such as emoticons, abbreviations, and excessive punctuation. VADER leverages a sentiment lexicon containing words with predetermined sentiment values, and combines word-level scores to produce a sentence-level sentiment score [19]. It also accounts for intensifier words such as “very” and negation words such as “not” when determining sentiment polarity. VADER produces four output scores: positive (pos), negative (neg), neutral (neu), and a normalized compound score ranging from −1 (most negative) to +1 (most positive). Sentiment classification is determined based on the compound score: a value of ≥0.05 is classified as positive, ≤−0.05 as negative, and values in between as neutral [20]. VADER’s key advantage is that it provides sentiment scores directly without requiring labeled training data, making it practical for automatic labeling at scale before applying supervised machine learning algorithms.

2.4. VADER Validation

To verify the reliability of VADER labels, a validation was conducted by comparing VADER-generated labels against star ratings as a proxy ground truth. Star ratings 1-2 were classified as negative, ratings 4-5 as positive, and rating 3 was excluded due to ambiguity. Validation metrics included accuracy, precision, recall, and F1-score. This validation step was performed before advanced preprocessing to ensure that the labels used in classification are reliable and valid.

2.5. TF-IDF Feature Extraction

Term Frequency–Inverse Document Frequency (TF-IDF) is a text weighting technique that combines the frequency of a word in a document (Term Frequency) with the uniqueness of that word across the entire document collection (Inverse Document Frequency). This method assigns higher weights to words that are important and unique, and lower weights to common words. TF-IDF is calculated by multiplying the TF value of a word by its IDF value, where TF measures how often a word appears in a document and IDF measures the word’s importance across the corpus. TF-IDF effectively reduces the influence of overly common words, as words appearing in all documents receive an IDF close to zero [21]. Furthermore, TF-IDF helps illustrate the importance of words that appear more frequently in the document, while assigning lower weights to words that are common across the corpus[22]. This study applies TF-IDF with a unigram and bigram configuration (n-gram range of 1 to 2) to capture both individual words and contextual word pairs[23], which has been shown to improve classification performance in text-based sentiment tasks.

2.6. Random Oversampling

Random oversampling is a technique for handling class imbalance by increasing the number of minority class samples in the training dataset. Class imbalance occurs when the number of samples in one class significantly exceeds the other, causing the model to be biased toward the majority class and perform poorly on the minority class. Class imbalance can negatively affect the performance of classification models, particularly when the number of samples in one class is substantially lower than in another. To address this issue, random oversampling was applied by increasing the number of minority-class samples through duplication of existing observations until a more balanced class distribution was achieved. Unlike synthetic sampling methods, random oversampling does not create new instances, allowing the original characteristics of the data to be preserved. To ensure a fair

evaluation, the oversampling process was performed only on the training set, while the test set was left unchanged so that it continued to reflect the actual distribution of the dataset.

2.7. Classification Using Naïve Bayes

This study employed the Multinomial Naïve Bayes algorithm, a probabilistic classification method that is widely used in text mining because of its simplicity and computational efficiency [24]. The model applies Bayes' theorem to estimate the probability of a document belonging to a particular sentiment class. TF-IDF feature vectors extracted from TIX ID user reviews were used as input for the classification process.

During model training, the TF-IDF features and their corresponding sentiment labels were provided to the classifier. The Multinomial Naïve Bayes model was initialized with an alpha value of 1.0 to prevent zero-probability issues during probability estimation. The model then learned the distribution of terms associated with each sentiment category and utilized the learned probabilities to classify unseen reviews into positive or negative sentiments. The overall training workflow of the Multinomial Naïve Bayes model is presented in Figure 2.

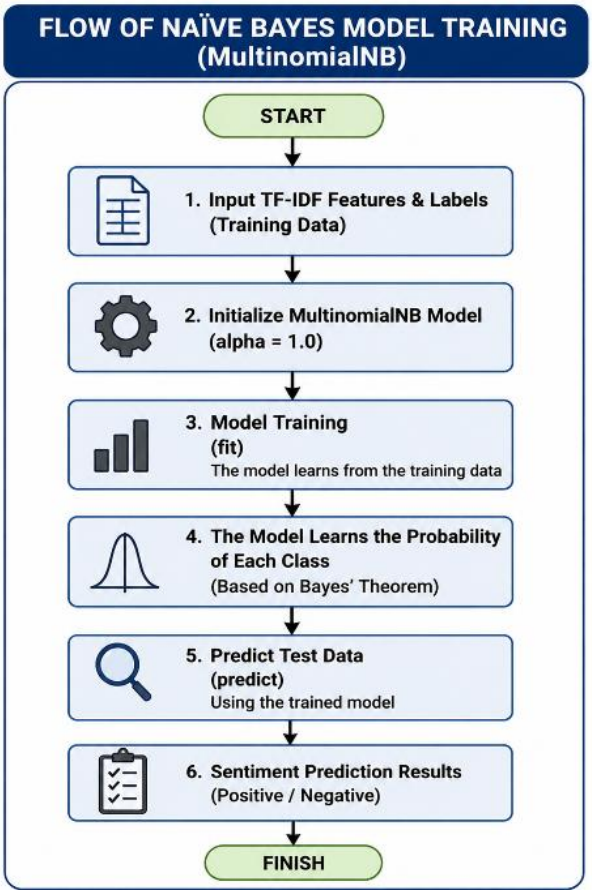

Figure 2. Naïve Bayes Model Training Workflow

Figure 2 illustrates the main stages of the training process, including TF-IDF feature input, model initialization, model training, probability estimation based on Bayes' theorem, prediction of testing data, and generation of sentiment classification results.

2.8. Classification Using SVM

In addition to Naïve Bayes, this study employed Support Vector Machine (SVM) using the LinearSVC implementation for sentiment classification. SVM is a supervised learning algorithm that has been widely adopted in text classification tasks due to its ability to handle high-dimensional feature spaces and effectively separate classes. The TF-IDF feature vectors extracted from user reviews served as the input representation for the SVM model.

The training process involved feeding the model with TF-IDF features and their corresponding sentiment labels from the training dataset. The LinearSVC classifier was configured with a regularization parameter of C = 1.0. During training, the model identified an optimal hyperplane that maximized the separation between positive and negative sentiment classes. Once trained, the model was applied to classify unseen review data and predict their sentiment labels. The overall workflow of the SVM classification process is presented in Figure 3.

.

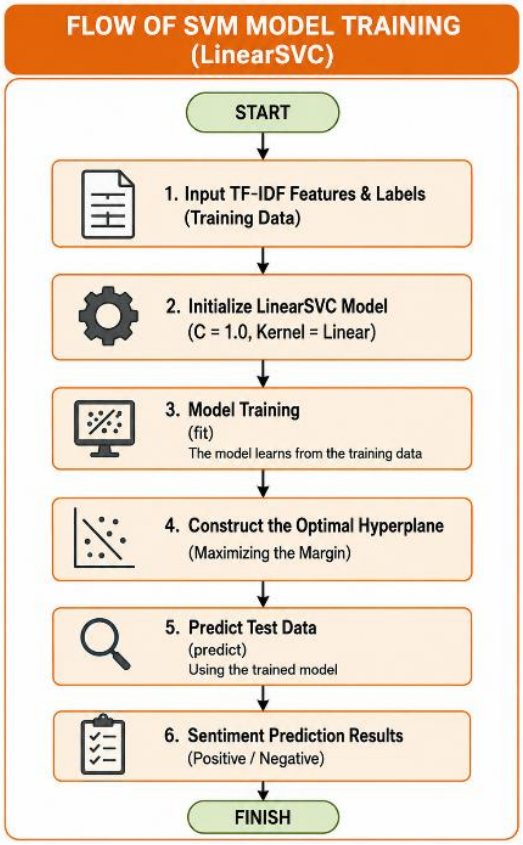

Figure 3. SVM Model Training Workflow

Figure 3 illustrates the main stages of the SVM training workflow, including feature input, model initialization, training, hyperplane optimization, sentiment prediction, and generation of classification results.

3. Results and Discussion

3.1. Dataset Collection Results

The dataset used in this study was obtained from a CSV file sourced via Kaggle and successfully loaded for analysis. It consists of 28,247 user reviews of the TIX ID application collected from the Google Play Store. The dataset includes 12 attributes, namely reviewId, userName, userImage, content, score, thumbsUpCount, reviewCreatedVersion, at, replyContent, repliedAt, appVersion, and text. An example of the dataset is shown in Figure 4.

As illustrated in Figure 4, the data contains reviews written in both Indonesian and English, accompanied by star ratings (score) on a scale of 1 to 5, as well as additional metadata such as the review submission date (at), application version (appVersion), and developer responses (replyContent). The text column, which represents the English-translated version of user reviews, was selected as the main input for this study because the VADER sentiment analysis lexicon is designed for English language processing. Meanwhile, the score column was utilized only as a reference for validating VADER sentiment labels.

The sample data also demonstrates a wide range of linguistic variations, including informal language (e.g., “Ga bisa bayar coyyyy”), non-standard spelling, and concise positive remarks such as

"Great apps!". This reflects the inherently unstructured and diverse nature of real-world user-generated content, which emphasizes the importance of implementing a structured preprocessing pipeline prior to the classification stage.

3.2. Initial Preprocessing Results

After the `dropna()` process was applied, the number of records in the dataset decreased from 28,247 to 12,007 entries. During the data checking stage, no duplicate records were detected. The `cleaning_text` function was then used to clean the text data by removing unnecessary elements such as special characters, emojis, URLs, and extra spaces. After that, case folding was carried out to convert all text into lowercase format for consistency. For example, the review "Can't pay coyyyy, what can I do?" was transformed into "can t pay coyyyy what can i do" after going through the cleaning and normalization steps. A summary of the initial preprocessing results is provided in Table 2.

Table 2. Initial Preprocessing Summary

Stage	Column	Example Result
Text Cleaning	clean_text	Can t pay coyyyy what can I do
Case Folding	case_folded	can t pay coyyyy what can i do

3.3. VADER Labeling Results

VADER sentiment labeling was conducted on 12,007 cleaned review records. Based on the results presented in Table 3, the dataset was classified into 8,196 positive reviews (68.26%), 1,197 negative reviews (9.97%), and 2,614 neutral reviews (21.77%). The neutral class was subsequently excluded from the dataset, resulting in a final total of 9,393 records used for further analysis. The predominance of positive sentiment indicates that users generally tend to express favorable opinions when they are satisfied with the application. Although negative reviews account for a relatively small portion of the data (9.97%), they remain valuable as they provide important insights for identifying areas that require improvement.

Table 3. VADER Labeling Results Distribution

Sentiment	Count	Percentage	Status
Positive	8,196	68.26%	Retained
Neutral	2,614	21.77%	Removed
Negative	1,197	9.97%	Retained
Total	12,007	100%	-

3.4. VADER Validation Results

Validation of the VADER-generated sentiment labels was carried out using 9,206 samples and compared against star ratings, as summarized in Table 4. The evaluation results show an overall accuracy of 92.68% with a weighted F1-score of 92.46%, indicating that the labels produced by VADER are generally reliable and appropriate for use in the training process.

In terms of class-wise performance, VADER achieved its strongest results on the negative class, with a precision of 94.77%, recall of 96.83%, and an F1-score of 95.79%. This strong performance is likely due to the fact that negative reviews often contain clear and explicit negative terms that are easily recognized by the VADER lexicon. For the positive class, the performance was comparatively lower, with a precision of 77.55%, recall of 67.21%, and an F1-score of 72.01%. This difference can be attributed to the nature of the data, where many positive reviews are expressed using informal Indonesian language, slang, abbreviations, and code-switching between Indonesian and English, which are not fully captured by VADER's English-based lexicon. In addition, positive sentiments are sometimes conveyed implicitly, making them more difficult to detect accurately. Despite these limitations, the overall accuracy of 92.68% still indicates that VADER-based labeling is sufficiently dependable for use in this research.

It should be noted that star ratings were used only as a practical substitute for ground-truth labels because manually annotated sentiment data were not available. Although ratings generally reflect user satisfaction, they do not always match the sentiment expressed in the review text. For example, users may provide a high rating while mentioning specific complaints, or their ratings may be influenced by factors unrelated to the written review. To reduce ambiguity, reviews with a rating of 3 were excluded from the validation process. Therefore, the validation accuracy of 92.68% should be interpreted as the level of agreement between VADER labels and user ratings rather than a direct measure of true

sentiment accuracy. Future studies could improve this validation by using manually annotated review data.

Table 4. VADER Validation Classification Report

Class	Precision	Recall	F1-Score	Support
Positive (VADER)	77.55%	67.21%	72.01%	1,290
Negative (VADER)	94.77%	96.83%	95.79%	7,916
Accuracy	92.68%	-	-	9,206
Weighted Average	92.36%	92.68%	92.46%	9,206

3.5. Advanced Preprocessing Results

The advanced preprocessing stage further refined the text into a cleaner and more structured representation. Table 5 presents a step-by-step illustration of how a sample review was transformed throughout this process. Initially, tokenization was applied to divide the case-folded text into individual words or tokens. This was followed by stopword removal using the NLTK English stopword list, which eliminated frequently occurring words such as “can,” “t,” “what,” “i,” and “do,” as they do not carry significant sentiment information.Next, lemmatization was performed to reduce words to their base or normalized forms. For instance, elongated or repeated character expressions such as “coyyyy” were standardized into “coy.” After completing all preprocessing steps, the resulting `text_ml` representation of the example review became “pay coy.” This condensed form preserves the essential meaning of the original text while making it more suitable for TF-IDF feature extraction and subsequent modeling.

Table 5. Advanced Preprocessing Step Summary

Stage	Column	Result (Example)
Tokenization	tokens	[can, t, pay, coyyyy, what, can, i, do]
Stopword Removal	tokens_sw	[pay, coyyyy]
Lemmatization	tokens_lem	[pay, coy]
Join (final)	text_ml	pay coy

3.6. Data Splitting Results

The 9,393 labeled records were divided into training and testing sets using an 80:20 split, resulting in 7,514 training samples and 1,879 test samples. As presented in Table 6, a stratified sampling approach was applied to preserve the original class distribution across both subsets. The results show that the proportion of classes remained highly consistent between the two sets, with the training data consisting of 87.26% positive and 12.74% negative samples, while the test data contained 87.28% positive and 12.72% negative samples. This near-identical distribution indicates that the stratified splitting method was successfully implemented, ensuring that both training and testing datasets are representative of the overall data distribution and suitable for reliable model evaluation.

Table 6. Dataset Split Distribution

Subset	Total	Positive	Negative	Proportion
Training set	7,514	6,556 (87.26%)	958 (12.74%)	80%
Testing set	1,879	1,640 (87.28%)	239 (12.72%)	20%
Total	9,393	8,196	1,197	100%

3.7. TF-IDF Feature Extraction Results

The TF-IDF vectorization was configured using an n-gram range of (1,2), with `min_df=5` and `max_df=0.8`, resulting in a total of 1,380 features extracted from the training data. These features consist of both single-word terms (unigrams) and two-word combinations (bigrams), such as “able,” “access,” “add payment,” “admin fee,” and “alfamart.” The final training representation formed a matrix with dimensions (7,514 × 1,380), indicating that each of the 7,514 training documents was converted into a 1,380-dimensional feature vector based on TF-IDF weighting. An analysis of feature importance shows that the highest-weighted term was “good” (0.185), followed by other frequently influential terms such as “nice,” “ok,” “easy,” “great,” “app,” and “helpful.” The dominance of positively oriented terms among the top features is consistent with the overall sentiment distribution of the dataset, which is largely positive.

3.8. Data Balancing Results

The original training dataset consisted of 6,556 positive samples and 958 negative samples, producing a highly imbalanced class distribution with a ratio of approximately 6.8:1. This imbalance could potentially lead the model to become biased toward the majority (positive) class during training. To address this issue, RandomOverSampler was applied to the training data by synthetically increasing the minority class through random duplication. As a result, 5,598 additional negative samples were generated, bringing the negative class count to 6,556—equal to the number of positive samples. After resampling, the training dataset became fully balanced, with 6,556 samples for each class and a total of 13,112 records, as summarized in Table 7.

Table 7. Class Distribution Before and After Oversampling

Class	Before Oversampling	After Oversampling
Positive	6,556	6,556
Negative	958	6,556 (+5,598 duplicated)
Total	7,514	13,112

3.9. Naïve Bayes Classification Results

The Multinomial Naive Bayes (MNB) model was trained using the balanced training dataset and subsequently evaluated on the original (unmodified) test set. The complete performance evaluation is reported in Table 8.

Table 8. Naive Bayes Classification Report

Class	Precision	Recall	F1-Score	Support
Negative	58.99%	93.31%	72.29%	239
Positive	98.93%	90.55%	94.56%	1,640
Accuracy	90.90%	-	-	1,879
Macro Average	78.96%	91.93%	83.42%	1,879
Weighted Average	93.85%	90.90%	91.72%	1,879

As presented in Table 7, the Multinomial Naive Bayes model obtained an overall accuracy of 90.90% and a weighted F1-score of 91.72%, indicating a strong overall classification performance.

For the positive class, the model achieved excellent results, with a precision of 98.93%, recall of 90.55%, and an F1-score of 94.56%. This strong performance can be attributed to the model's effective learning of dominant positive word patterns present in the training data.

In contrast, performance on the negative class was comparatively weaker, with a precision of 58.99%, recall of 93.31%, and an F1-score of 72.29%. Although the high recall value shows that most negative reviews were successfully identified, the relatively low precision indicates that a number of positive reviews were mistakenly classified as negative (false positives).

This is also supported by the confusion matrix results, where 1,485 instances were correctly predicted as positive (true positives), 155 positive samples were misclassified as negative (false negatives), 223 negative samples were correctly classified (true negatives), and 16 positive samples were incorrectly predicted as negative (false positives). The lower precision in the negative class is likely due to the independence assumption inherent in Naive Bayes, which limits its ability to capture more complex contextual relationships between words.

3.10. SVM Classification Results

The LinearSVC model demonstrated better performance compared to Multinomial Naive Bayes, as summarized in Table 9.

Table 9. SVM Classification Report

Class	Precision	Recall	F1-Score	Support
Negative	69.08%	87.87%	77.35%	239
Positive	98.16%	94.27%	96.17%	1,640
Accuracy	93.45%	-	-	1,879
Macro Average	83.62%	91.07%	86.76%	1,879
Weighted Average	94.46%	93.45%	93.78%	1,879

As shown in Table 9, the model achieved an overall accuracy of 93.45% with a weighted F1-score of 93.78%, indicating strong classification performance. For the positive class, the model produced a

precision of 98.16%, recall of 94.27%, and an F1-score of 96.17%, reflecting highly accurate identification of positive reviews.

For the negative class, performance was lower than the positive class but still improved compared to Naive Bayes, with a precision of 69.08%, recall of 87.87%, and an F1-score of 77.35%. Although precision remained relatively modest, it was notably higher than the Naive Bayes result (58.99%), indicating fewer false positive predictions.

The confusion matrix further shows: 1,546 true positives, 94 false negatives, 207 true negatives, and 32 false positives. Compared to Naive Bayes, SVM increased the number of correctly classified positive samples (+61) and reduced false negatives (−61), demonstrating better classification reliability for the majority class.

The accuracy improvement of 2.55 percentage points over Naive Bayes (93.45% vs. 90.90%) is meaningful in text classification tasks, as it reflects more consistent performance across both classes. This improvement can be explained by the fundamental difference in the models. Unlike Naive Bayes, which relies on conditional probability assumptions and treats features independently, SVM identifies an optimal separating hyperplane in the high-dimensional TF-IDF space. This margin-maximization strategy allows SVM to better capture complex relationships between features, making it particularly effective for high-dimensional text data.

To provide additional insights beyond the quantitative results, some misclassified reviews were manually analyzed. Three common patterns were identified. First, very short negative reviews containing few explicit negative words were frequently misclassified by Naïve Bayes. This is likely because the model relies heavily on the presence of sentiment-related terms, making it less effective when reviews provide limited textual information. In contrast, SVM was able to classify some of these reviews more accurately, likely due to its ability to better utilize TF-IDF features. Second, reviews containing sarcasm or indirect criticism presented challenges for both models. Although some reviews included positive words, their actual meaning reflected dissatisfaction, making them difficult to correctly classify using lexicon-based labeling and TF-IDF representation. Third, reviews containing informal abbreviations or slang reduced classification performance. These expressions are often poorly represented in the vocabularies used by VADER and TF-IDF, limiting the model's ability to capture sentiment. These findings suggest that, while random oversampling helps address class imbalance, it cannot fully address issues related to language variation and contextual meaning. Future studies might benefit from using more sophisticated text representations or sentiment resources specifically designed for Indonesian-language reviews.

4. Conclusion

This study successfully applied both Naive Bayes and SVM algorithms for sentiment classification of TIX ID application user reviews using VADER-based automatic labeling and TF-IDF feature extraction. The overall pipeline included systematic text preprocessing, sentiment labeling using VADER, validation against star ratings, advanced preprocessing steps (tokenization, stopword removal, and lemmatization), TF-IDF feature extraction with unigram–bigram configuration, and RandomOverSampler to address class imbalance. The Naive Bayes model achieved an accuracy of 90.90% with a weighted F1-score of 91.72%, showing good overall performance, particularly for the positive class (F1-score = 94.56%). However, its performance on the negative class was weaker (precision = 58.99%), mainly due to the independence assumption, which limits its ability to capture contextual relationships between words. Meanwhile, the SVM model outperformed Naive Bayes with an accuracy of 93.45% and a weighted F1-score of 93.78%, showing better performance across all evaluation metrics. The superiority of SVM is attributed to its margin-maximization approach, which identifies an optimal separating hyperplane in a high-dimensional TF-IDF feature space, making it more effective for text classification tasks. VADER validation against star ratings also produced strong results, with an accuracy of 92.68% and a weighted F1-score of 92.46%, confirming that VADER-generated labels are reliable for training purposes. In addition, because star ratings served only as a proxy ground truth rather than human-annotated sentiment labels, future studies should incorporate manually labeled samples to more rigorously validate automatic labeling approaches such as VADER. Based on these findings, SVM is recommended as the most effective method for sentiment classification of TIX ID reviews, while VADER-based labeling provides a practical alternative to manual annotation for large-scale datasets. Future work is recommended to incorporate Indonesian-specific sentiment lexicons,

word embedding techniques such as Word2Vec, GloVe, or FastText, and transformer-based models like IndoBERT to further improve performance on mixed-language (Indonesian–English) review data.

References

[1] Statista, "Number of Social Network Users in Selected Countries in 2022 and 2027 (in millions) [Graph]," 2025. https://www.statista.com/statistics/278341/number-of-social-network-users-in-selected-countries/

[2] Tikno, Y. S. Dharmawan, and Ngatini, "Investigating Consumer Acceptance of Mobile Payment Services in Indonesia," *Procedia Comput. Sci.*, vol. 234, no. 2023, pp. 1095–1102, 2024, doi: 10.1016/j.procs.2024.03.104.

[3] D. Yolanovia and A. D. Indriyanti, "Evaluasi User Experience Aplikasi TIX ID Menggunakan Metode Heuristic Evaluation," *Emerg. Inf. Syst. Bus. Intell.*, vol. 02, no. 03, pp. 8–13, 2021.

[4] B. R. Prasetyo, I. Tazkiyah, R. C. S. Fadillah, Ainun Rizkyani Indonesiawan, and M. Alroy, "Evaluasi Aplikasi E-Ticketing TIX ID dengan Menggunakan Metode Usability Testing Evaluation of TIX ID E-Ticketing Application," *Semin. Nas. Teknol. dan Sist. Inf.*, no. September, pp. 10–11, 2022.

[5] I. L. Keksi, Jumanto, and A. P. A. Masa, "Sentiment analysis of youtube comments on the palestine-israel conflict: Performance comparison of SVM, KNN, and RFC," *J. Student Res. Explor.*, vol. 3, no. 1, pp. 23–37, 2025.

[6] D. Wang, Z. Gong, G. Wei, M. A. Martinez, and E. Herrera-Viedma, "Using sentiment analysis and CEEMDAN to learn the preferences of consumer groups : A case study of online hotel reviews," *Appl. Soft Comput.*, vol. 191, no. January, 2026, doi: 10.1016/j.asoc.2026.114625.

[7] A. Borg and M. Boldt, "Using VADER sentiment and SVM for predicting customer response sentiment," *Expert Syst. Appl.*, vol. 162, 2020, doi: 10.1016/j.eswa.2020.113746.

[8] Z. Li, R. Li, and G. Jin, "Sentiment Analysis of Danmaku Videos Based on Naïve Bayes and Sentiment Dictionary," *IEEE Access*, vol. 8, pp. 75073–75084, 2020, doi: 10.1109/ACCESS.2020.2986582.

[9] T. H. J. Hidayat, Y. Ruldeviyani, A. R. Aditama, R. G. Madya, A. W. Nugraha, and M. W. Adisaputra, "Sentiment analysis of twitter data related to Rinca Island development using Doc2Vec and SVM and logistic regression as development using classifier," *Procedia Comput. Sci.*, vol. 197, no. 2021, pp. 660–667, 2022, doi: 10.1016/j.procs.2021.12.187.

[10] V. Gupta and P. Rattan, "Improving Twitter Sentiment Analysis Efficiency with SVM- PSO Classification and EFWS Heuristic," *Procedia Comput. Sci.*, vol. 230, pp. 698–715, 2023, doi: 10.1016/j.procs.2023.12.125.

[11] A. Serlina, A. Rahim, and Arbansyah, "Comparative Analysis of Naïve Bayes Algorithm Performance in English and Indonesian Text Sentiment Classification on Duolingo Application in Playstore," *Teknika*, vol. 14, no. March, pp. 165–171, 2025, doi: 10.34148/teknika.v14i1.1207.

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

[13] M. Isnan, N. G. Elwirehardja, and B. Pardamean, "Sentiment Analysis for TikTok Review Using VADER Sentiment and SVM Model," *Procedia Comput. Sci.*, vol. 227, pp. 168–175, 2023, doi: 10.1016/j.procs.2023.10.514.

[14] Z. Song, C. Zhang, and Y. Lu, "The methodology for evaluating the fire resistance performance of concrete-filled steel tube columns by integrating conditional tabular generative adversarial networks and random oversampling," *J. Build. Eng.*, vol. 97, no. July, p. 110824, 2024, doi: 10.1016/j.jobe.2024.110824.

[15] E. Rosenberg *et al.*, "Sentiment analysis on Twitter data towards climate action," *Results Eng.*, vol. 19, no. July, p. 101287, 2023, doi: 10.1016/j.rineng.2023.101287.

[16] A. S. Abadi, "TIX ID app reviews from google play store," 2024, doi: 10.34740/kaggle/dsv/10194275.

[17] R. Kusumaningrum, I. Z. Nisa, R. P. Nawangsari, and A. Wibowo, "Sentiment analysis of Indonesian hotel reviews : from classical machine learning to deep learning," *Int. J. Adv. Intell. Informatics*, vol. 7, no. 3, pp. 292–303, 2021.

[18] M. Siino, I. Tinnirello, and M. La Cascia, "Is text preprocessing still worth the time ? A

comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers," *Inf. Syst.*, vol. 121, no. December 2023, p. 102342, 2024, doi: 10.1016/j.is.2023.102342.

[19] D. C. Youvan, "Understanding Sentiment Analysis with VADER: A Comprehensive Overview and Application," *AI Data Sci.*, pp. 1–27, 2024.

[20] Jumanto, M. A. Muslim, Y. Dasril, and T. Mustaqim, "Accuracy of Malaysia Public Response to Economic Factors During the Covid-19 Pandemic Using Vader and Random Forest," *J. Inf. Syst. Explor. Res.*, vol. 1, no. 1, pp. 49–70, 2023.

[21] T. U. Sen, M. C. Yakit, M. S. Gumus, O. Abar, and G. Bakal, "Combining N-grams and graph convolution for text classification," *Appl. Soft Comput.*, vol. 175, no. April, p. 113092, 2025, doi: 10.1016/j.asoc.2025.113092.

[22] Rofik, R. A. Hakim, Jumanto, B. Prasetiyo, and M. A. Muslim, "Optimization of SVM and Gradient Boosting Models Using GridSearchCV in Detecting Fake Job Postings," *J. Manajemen, Tek. Inform. dan Rekayasa Komput. Vol.*, vol. 23, no. 2, pp. 419–430, 2024, doi: 10.30812/matrik.v23i2.3566.

[23] L. Chen, "An extended TF-IDF method for improving keyword extraction in traditional corpus-based research : An example of a climate change corpus," *Data Knowl. Eng.*, vol. 153, p. 102322, 2024, doi: 10.1016/j.datak.2024.102322.

[24] P. Utami, Y. M. P. Tangai, J. Unjung, and M. A. Muslim, "Enhancing Abusive Language Detection on Twitter Using Stacking Ensemble Learning," *J. Inf. Syst. Explor. Res.*, vol. 3, no. 2, pp. 53–62, 2025.