Sentiment Analysis of TikTok Studio Application Reviews on Google Play Store

Enindya Ageng Pangesti[1*], Dewi Retno Sari Saputro[2], Sutanto[3]
[1,2,3] Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Sebelas Maret, Surakarta, 57126, Indonesia

Corresponding Author.
**Email*: enindyaagengpangesti@student.uns.ac.id

Abstract: *Sentiment analysis is a text mining approach used to identify user opinions toward an application, product, or service. User reviews of the TikTok Studio application on the Google Play Store can be utilized to determine the overall sentiment tendencies of users toward the application. This study aims to perform sentiment analysis on TikTok Studio user reviews using Logistic Regression. The research stages include data collection through web scraping, text preprocessing, sentiment labeling using Valence Aware Dictionary for sEntiment Reasoning (VADER), feature representation using Term Frequency–Inverse Document Frequency (TF-IDF), data splitting into training data (80%) and testing data (20%), class balancing using the Synthetic Minority Oversampling Technique (SMOTE), and model training and evaluation using Logistic Regression. The dataset consists of 9,247 user reviews, of which 8,717 valid reviews were obtained after preprocessing. The sentiment labeling process resulted in 6,090 positive reviews, 271 negative reviews, and 2,356 neutral reviews, where neutral reviews were excluded, yielding a final dataset of 6,361 reviews. Based on the evaluation of 1,273 testing data, the Logistic Regression model achieved an accuracy of 97.64%, with precision, recall, and F1-score (macro average) of 0.82, 0.97, and 0.88, respectively.*
Keywords: *Sentiment Analysis, Logistic Regression, TikTok Studio Reviews, TF-IDF, SMOTE*

1. INTRODUCTION

The rapid development of digital technology has led to the emergence of various applications and social media platforms that are widely used for sharing information and content. Social media platforms are not only utilized as a means of communication but also as a medium for content creators to produce, manage, and distribute content to audiences more effectively. One of the platforms experiencing rapid user growth is TikTok, which provides various features that enable creators to produce and share creative video content. Along with the increasing number of creators, supporting applications have also been developed to facilitate content management processes, one of which is TikTok Studio. This application offers various features that allow creators to upload, edit, and monitor the performance of content published on TikTok. User reviews available on Google Play Store serve as an important source of information as they reflect user satisfaction as well as opinions, suggestions, and criticisms regarding the quality of the application. However, the large volume of online reviews presents challenges in understanding public sentiment, as the data are unstructured text and continue to grow over time [1], [2]. Therefore, a computational approach is required to automatically identify user sentiment.

Sentiment analysis is a text mining technique used to identify the tendency of opinions or attitudes expressed in textual data [3], [4], [5], [6], [7]. Machine learning methods have been widely applied in sentiment classification and have demonstrated strong performance across various types of review data [8]. One commonly used model is Logistic Regression, which offers simple interpretability and competitive performance in binary sentiment classification, namely positive and negative classes [9], [10]. Logistic Regression models the relationship between features and class labels probabilistically, making it effective for text classification tasks [11], [12]. Previous studies indicate that sentiment analysis has predominantly been applied to social media platforms such as Twitter or other applications [13], while research focusing on user reviews of creator-support applications such as TikTok Studio remains limited [14]. In fact, this type of application plays an important role in supporting creator productivity and content quality, making user feedback highly valuable for feature development and application performance improvement. Without systematic analysis, developers may face difficulties in understanding overall user sentiment trends efficiently and accurately. In addition, some studies still rely on manual labeling, which is less efficient and potentially subjective [15]. Previous studies have shown that Logistic Regression demonstrates competitive performance in text sentiment classification, making it a relevant baseline method in sentiment analysis. Therefore, an approach is required that is not only automated but also capable of producing balanced and representative classifications for unstructured and imbalanced review data.

Based on these issues, this study aims to analyze sentiment in TikTok Studio user reviews on the Google Play Store using Logistic Regression. This study employs VADER for sentiment labeling, TF–IDF for feature representation, and SMOTE to address class imbalance, in order to develop a more effective sentiment classification model. Logistic Regression was selected because it offers straightforward interpretability and demonstrates good performance in binary text classification, making it suitable for efficiently identifying positive and negative sentiment.

2. METHOD

This study employs a quantitative approach by utilizing secondary data in the form of user reviews of the TikTok Studio application obtained from the Google Play Store through web scraping techniques [16], [17]. A quantitative approach is applied because this study focuses on processing data in numerical form to develop a sentiment classification model that is objective and measurable. In general, the research stages are systematically conducted as follows:

1. Collecting TikTok Studio application reviews from the Google Play Store using web scraping techniques.
2. Performing preprocessing on the review data, including cleaning, case folding, stopword removal, tokenization, and stemming.
3. Conducting sentiment labeling on the preprocessed data using the VADER lexicon available in a Python package and classifying the reviews into three sentiment categories: positive, negative, and neutral.
4. Removing reviews labeled as neutral, so that only positive and negative sentiments are used in the modeling process.
5. Splitting the dataset into training data (80%) and testing data (20%).
6. Representing text features in the training data using the TF–IDF method.
7. Evaluating the class distribution in the training data to identify class imbalance. If imbalance is detected, class balancing is performed using the SMOTE method.
8. Training the sentiment classification model using the Logistic Regression algorithm on the training data.
9. Testing the trained model using the testing data.
10. Evaluating the model using accuracy and F1-score metrics.
11. Analyzing the evaluation results of the model.

Based on the research stages described above, a detailed explanation of each stage is presented in the following subsections.

2.1 Data Source

The collected data consist of TikTok Studio application reviews from users in the United States during the period from July 1 to November 30, 2025, with a total of 9,247 user reviews. The selection of the United States as the study region is based on reports indicating that it is one of the largest TikTok user markets in the world in 2025 [18]. The high number of TikTok users in this region reflects intensive creator activity, which in turn increases the use of supporting applications such as TikTok Studio. In addition, previous studies suggest that exposure to global social media can influence users' digital behavior in interacting with various online platforms [19]. Therefore, reviews from this region are considered sufficiently representative to reflect user opinions toward the TikTok Studio application. The collected review data are in the form of unstructured text, requiring further preprocessing before conducting sentiment analysis.

2.2 Sentiment Analysis Approach

The approach used in this study is fine-grained sentiment analysis because the analysis is conducted at the level of individual user reviews [20]. The objective of sentiment analysis is to classify user reviews into two sentiment classes, namely positive and negative. Sentiment labeling was performed using VADER, a lexicon-based method that classifies text into positive, negative, and neutral sentiment categories based on polarity scores derived from a sentiment lexicon [22], [23]. VADER was selected because it enables automatic and consistent sentiment labeling for large-scale data. Lexicon-based approaches are widely used in social media sentiment analysis as they can efficiently identify text polarity without requiring time-consuming manual labeling. In addition, automatic labeling helps reduce subjectivity that may arise from differences in human interpretation during manual annotation. In this study, reviews labeled as neutral were excluded because they do not indicate a clear opinion regarding the application and may introduce ambiguity in the classification process. Therefore, only reviews with positive and negative labels were used in the modeling stage. The use of binary classification allows the model to learn the differences between positive and negative sentiment characteristics more clearly, resulting in a more focused interpretation of the results.

2.3 Preprocessing

Preprocessing is conducted to improve data quality and ensure its suitability for the modeling process [21]. This stage consists of several steps, namely cleaning, case folding, tokenization, stopword removal, and stemming. Cleaning is performed to remove unnecessary characters as well as duplicate and invalid data. Case folding is applied to convert all text into lowercase to maintain consistency. Tokenization is used to split the text into individual word units for computational processing. Furthermore, stopword removal is carried out to eliminate common words with little semantic value. Finally, stemming is performed to reduce words to their base form.

2.4 Feature Extraction

The data that have undergone sentiment labeling are subsequently divided into training and testing sets with a ratio of 80:20. The training data are used for model training, while the testing data are used to evaluate the model's performance in classifying sentiment. Furthermore, the text data are transformed into numerical representations using the TF–IDF method, which assigns weights to each term based on its frequency within a document and its distribution across the entire corpus [24]. The weight of each term is determined based on its frequency within a document and its distribution across the entire corpus, which is defined as

$$W(d,t) = tf(d,t) \times \ln\left(\frac{N}{df(t)}\right)$$

where $tf(d,t)$ denotes the frequency of term t in document d, N is the total number of documents, and $df(t)$ represents the number of documents containing term t.

2.5 Handling Imbalanced Data

Class imbalance in the training data is addressed using the SMOTE. This method is implemented using the Python programming language and generates synthetic samples for the minority class through interpolation between existing minority instances and their nearest neighbors [25].

2.6 Classification Model

The sentiment analysis model used in this study is Logistic Regression, a binary classification method that predicts the probability of a data instance belonging to a particular class using a logistic function [11]. The Logistic Regression model is formulated as

$$y = \frac{1}{1 + e^{-z}}$$

where

$$z = b_0 + b_1x_1 + b_2x_2 + \cdots + b_ix_i.$$

Here, z represents the linear combination of all TF–IDF features in a review, b_0 is the intercept, and b_i denotes the coefficient representing the contribution of the i-th feature. The probability value y ranges from 0 to 1, where values closer to 1 are classified as positive sentiment and values closer to 0 are classified as negative sentiment, with the decision boundary set at $z = 0$.

2.7 Model Evaluation

Model evaluation is conducted to assess the performance of Logistic Regression in classifying positive and negative sentiments on the testing data using a confusion matrix consisting of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [26], [27]. Based on these values, evaluation metrics such as accuracy and F1-score are calculated. Accuracy is used to measure the overall correctness of the model, which is formulated as

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}.$$

Meanwhile, the F1-score is used to evaluate the balance between precision and recall, which is defined as

$$F1-score = \frac{2 \times Presisi \times Recall}{Presisi + Recall}$$

where

$$Precision = \frac{TP}{TP+FP} \text{ and } Recall = \frac{TP}{TP+FN}.$$

3. RESULTS AND DISCUSSION

The results and discussion in this study are divided into seven sections, namely preprocessing, sentiment labeling, training and testing data splitting, text representation into numerical features, class balancing using SMOTE, Logistic Regression model training, and model performance evaluation. Each stage is described in detail as follows.

3.1 Preprocessing

Preprocessing is the initial stage in processing TikTok Studio application review data before performing sentiment labeling and building the sentiment analysis model. This process consists of five steps, namely cleaning, case folding, tokenization, stopword removal, and stemming. Examples of the results from each preprocessing stage are presented in Table 1.

Table 1. Preprocessing Simulation

Preprocessing Stage	Review 1	Review 2	Review 3
Original Review	Tiktok studio always crashing on my infinix hot 30 can you fix bugs and crashes tiktok i need to edit my videos and please	this app is some else alot to learn but im teady	Very good working very good app
Cleaning	Tiktok studio always crashing on my infinix hot can you fix bugs and crashes tiktok i need to edit my videos and please	this app is some else alot to learn but im teady	Very good working very good app
Case Folding	tiktok studio always crashing on my infinix hot can you fix bugs and crashes tiktok i need to edit my videos and please	this app is some else alot to learn but im teady	very good working very good app
Tokenizing	tiktok, studio, always, crashing, on, my, infinix, hot, can, you, fix, bugs, and, crashes, tiktok, i, need, to, edit, my, videos, and, please	this, app, is, some, else, alot, to, learn, but, im, teady	very, good, working, very, good, app
Stopword Removal	tiktok, studio, always, crashing, infinix, hot, fix, bugs, crashes, tiktok, need, edit, videos, please	app, else, alot, learn, im, teady	good, working, good, app
Stemming	tiktok studio alway crash infinix hot fix bug crash tiktok need edit video pleas	app else alot learn im teady	good work good app

After applying all preprocessing steps, the number of review data decreased from 9,247 to 8,717 due to the removal of empty and invalid entries. Thus, a total of 530 reviews were eliminated for not meeting the data quality criteria. This result indicates that the preprocessing stage effectively improved the dataset quality by retaining only relevant reviews for sentiment analysis.

3.2 Sentiment Labeling

Sentiment labeling was performed on 8,717 preprocessed reviews using the VADER method, which classifies reviews into three sentiment categories: positive, negative, and neutral. The labeling results show that there are 6,090 positive reviews, 271 negative reviews, and 2,356 neutral reviews. In this study, neutral reviews were excluded as they do not represent a clear opinion polarity. Therefore, all neutral reviews were removed from the dataset. After this filtering process, the dataset used in the subsequent stages consisted of 6,361 reviews, including 6,090 positive reviews and 271 negative reviews. This distribution indicates a significant class imbalance, where positive reviews are substantially more dominant than negative reviews.

3.3 Training and Testing Data Split

The dataset was divided into training and testing sets with a proportion of 80% and 20%, respectively. From a total of 6,361 reviews, the training set consists of 4,871 positive reviews and 217 negative reviews, while the testing set consists of 1,219 positive reviews and 54 negative reviews. This distribution preserves the class proportion of the original dataset, where positive reviews are significantly more dominant than negative reviews, indicating the presence of class imbalance. The training data are used to construct feature representations and train the Logistic Regression model, while the testing data are used to evaluate the model's performance on previously unseen data,

3.4 Text Representation into Numerical Features

The review data that have undergone preprocessing and sentiment labeling are subsequently transformed into numerical representations using the TF–IDF method. The IDF weights are computed only on the training data to prevent data leakage and to maintain objectivity in the model evaluation process. Based on the TF–IDF transformation, the training data produce a feature matrix with dimensions of 5,088 × 2,469, where each row represents a review and each column represents a word feature from the training vocabulary. This representation is used as input for training the Logistic Regression model. Table 2 presents an example of TF–IDF representation in the training data, illustrating the weight of each term in the form of numerical vectors.

Table 2. Example of TF–IDF Feature Representation in the Training Data

Term	TF	IDF	TF-IDF	Term	TF	IDF	TF-IDF
love	0.50	2.850	1.425	hot	0.071	7.464	0.533
app	0.25	1.712	0.428	fix	0.071	5.312	0.379
tiktok	0.143	2.392	0.342	bug	0.071	6.240	0.446
studio	0.071	3.283	0.235	need	0.071	4.930	0.352
alway	0.071	5.518	0.394	edit	0.071	4.478	0.320
crash	0.143	7.127	1.018	video	0.071	3.033	0.217
infinix	0.071	9.073	0.648	pleas	0.071	3.679	0.263

Subsequently, the testing data are transformed using the TF–IDF vectorizer that has been fitted on the training data, so only the transformation process is applied without recalculating the weights. The transformation results in a feature matrix with dimensions of 1,273 × 2,469, indicating that the number of word features remains consistent with the training data. This consistency in the feature space allows the testing data to be directly used in the testing and evaluation process of the Logistic Regression model. An example of the TF–IDF representation for the testing data is presented in Table 3.

Table 3. Example of TF–IDF Feature Representation in the Testing Data

Term	$TF_{testing}$	$IDF_{training}$	TF-IDF	Term	$TF_{testing}$	$IDF_{training}$	TF-IDF
job	0.5000	6.0085	0.8498	point	0.0909	7.9254	0.3309
great	0.5000	3.7267	0.5271	phone	0.0909	6.7622	0.2824
innov	0.5000	8.8417	0.8429	alway	0.0909	6.3160	0.2637
cool	0.5000	5.6430	0.5380	still	0.0909	6.1336	0.2561
stuck	0.1818	6.8268	0.5701	even	0.0909	6.1008	0.2548
switch	0.0909	8.8417	0.3692	upload	0.0909	5.5459	0.2316

3.5 Class Balancing using SMOTE

After data splitting and TF-IDF representation, the class distribution in the training data was analyzed. The results show a significant class imbalance, with 4,871 positive reviews and 217 negative reviews, indicating that the negative class is the minority. To address this issue, the SMOTE method was applied to the training data. This method generates synthetic samples for the minority class, resulting in a more balanced class distribution. The application of SMOTE increased the number of training instances from 5,088 to 9,742 while maintaining the same number of features (2,469), producing a balanced distribution of 4,871 positive and 4,871 negative reviews. A more balanced class distribution reduces the tendency of the model to be biased toward the majority class, allowing Logistic Regression to learn the characteristics of both sentiment classes in a more proportional manner.

3.6 Logistic Regression Model Training

The training process was conducted using the SMOTE-balanced training data with a feature matrix dimension of 9,742 × 2,469, where each row represents a review and each column represents a word feature. During training, the model learns parameter weights that reflect the contribution of each feature in determining the sentiment tendency of a review. The trained model is then stored and used in the testing phase with the testing dataset.

3.7 Logistic Regression Performance Evaluation

The evaluation process was carried out by calculating accuracy, precision, recall, and F1-score, as well as presenting a confusion matrix to illustrate the comparison between the model predictions and the actual labels. The confusion matrix results of the Logistic Regression model on the testing data are shown in Figure 1.

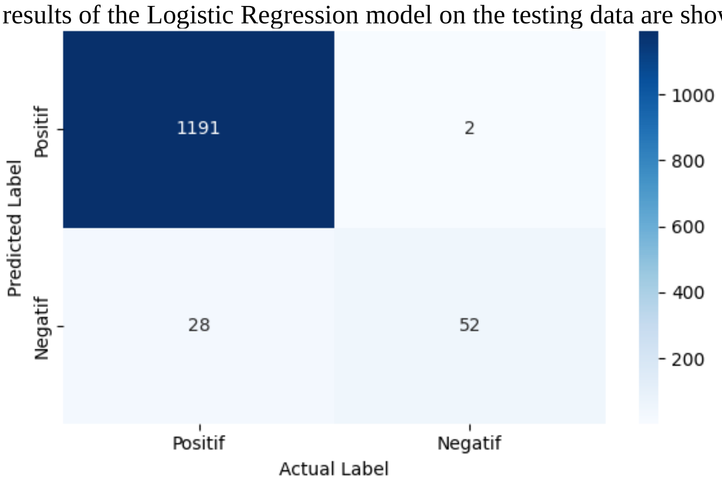

Figure 1. Confusion matrix of the logistic regression model

Based on Figure 1, the model correctly classified 1,191 positive reviews and 52 negative reviews. Misclassifications consist of 28 positive reviews that were predicted as negative and 2 negative reviews that were predicted as positive. Based on the values in the confusion matrix, the evaluation metrics are subsequently calculated, as presented in Table 4.

Table 4. Evaluation metrix of Logistic Regression model

Class	Precision	Recall	F1-score	Support
Negatif	0.65	0.96	0.78	54
Positif	1.00	0.98	0.99	1,219
Macro Average	0.82	0.97	0.88	1,273
Accuracy		0.9764		

Based on Table 4, the Logistic Regression model achieved an accuracy of 0.9764, while the precision, recall, and F1-score based on the macro average were 0.82, 0.97, and 0.88, respectively. The macro average value is used to provide a more balanced evaluation of model performance across both sentiment classes, as each class is weighted equally regardless of differences in the number of data instances. The high accuracy indicates that Logistic Regression is able to identify sentiment patterns effectively in TikTok Studio user reviews. These results suggest that the model is capable of learning the characteristics of both positive and negative sentiments from the test data. Although the precision value for the negative class is lower than that of the positive class, the high recall indicates that most negative reviews are still correctly identified. The difference in performance between the two classes is influenced by the smaller number of negative reviews compared to positive reviews, resulting in more limited variation in negative sentiment patterns. Overall, the evaluation results demonstrate that Logistic Regression performs well in classifying sentiment in TikTok Studio user reviews.

4. CONCLUSION

Based on the results of sentiment analysis conducted on 1,273 testing data from TikTok Studio application reviews using Logistic Regression, the evaluation results indicate that the model achieved an accuracy of 97.64%. In addition, the precision, recall, and F1-score based on the macro average are 0.82, 0.97, and 0.88, respectively. These results indicate that the Logistic Regression model is capable of classifying positive and negative sentiments in TikTok Studio user reviews.

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to all parties who contributed to this study, particularly in the development of the mathematical formulation, the selection of appropriate statistical models, as well as the estimation and validation procedures. Intensive scientific discussions regarding model assumptions, inferential consistency, and predictive performance evaluation have significantly strengthened the methodological validity of this research.

The authors also acknowledge the support in data provision and computational resources, which enabled the efficient implementation of algorithms and numerical simulations. Such contributions have ensured that the results obtained are both reliable and reproducible.

REFERENCES

[1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.

[2] J. Sadowski, "When data is capital: Datafication, accumulation, and extraction," *Big Data Soc.*, vol. 6, no. 1, pp. 1–12, 2019.

[3] B. Liu, *Sentiment Analysis and Opinion Mining*. Chicago, IL, USA: Morgan and Claypool, 2012.

[4] Y. Wu et al., "Large scale incremental learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 374–382.

[5] N. Stylos and J. Zwiegelaar, *Big Data as a Game Changer: How Does It Shape Business Intelligence Within a Tourism and Hospitality Industry Context?* 2019.

[6] A. Giachanou and F. Crestani, "Like it or not: A survey of Twitter sentiment analysis methods," ACM Computing Surveys, vol. 49, no. 2, pp. 1–41, 2016.

[7] M. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," Ain Shams Engineering Journal, vol. 5, no. 4, pp. 1093–1113, 2014.

[8] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2002, pp. 79–86.

[9] H. Chen, Y. Liu, and H. Teo, "Regularised text logistic regression: Key word detection and sentiment classification for online reviews," *IEEE Trans. Comput. Social Syst.*, 2020.

[10] P. Pareek, N. Sharma, R. Raj, and C. Dalwadi, "Sentiment analysis for Amazon product reviews using logistic regression model," *J. Develop. Econ. Manage. Res. Stud.*, vol. 9, no. 11, pp. 29–42, 2022.

[11] B. Ramadhani and R. R. Suryono, "Komparasi algoritma Naïve Bayes dan logistic regression untuk analisis sentimen metaverse," *Jurnal Media Informatika Budidarma*, vol. 8, no. 2, pp. 714–725, 2024.

[12] S. Leonelli and N. Tempini, *Data Journeys in the Sciences*. Cham, Switzerland: Springer, 2020.

[13] N. M. G. S. Nirmala and N. A. Sanjaya, "Analisis sentimen dengan logistic regression untuk deteksi kata pada Livin' by Mandiri," *Jurnal Nasional Teknologi Informasi dan Aplikasinya*, vol. 2, pp. 869–878, 2024.

[14] P. A. Permatasari, L. Linawati, and L. Jasa, "Survei tentang analisis sentimen pada media sosial," *Majalah Ilmiah Teknologi Elektro*, vol. 20, no. 2, pp. 177–184, 2021.

[15] M. Haerunnissa, A. Priyanto, C. Asnawi, and N. A. Sa'diya, "Analisis sentimen kepuasan pelanggan perusahaan telekomunikasi seluler Telkomsel di Twitter," *Jurnal Teknomatika*, vol. 15, no. 2, pp. 68–75, 2022.

[16] TikTok Pte. Ltd., "TikTok Studio." [Online]. Available: https://play.google.com/store/apps/details?id=com.ss.android.tt.creator. Accessed: Oct. 1, 2025.

[17] D. D. A. Yani, H. S. Pratiwi, and H. Muhardi, "Implementasi web scraping untuk pengambilan data pada situs marketplace," *JUSTIN: Jurnal Sistem dan Teknologi Informasi*, vol. 7, no. 4, pp. 257–262, 2019.

[18] DataReportal, "TikTok users: Global digital overview 2025." [Online]. Available: https://datareportal.com/essential-tiktok-stats. Accessed: Oct. 1, 2025.

[19] M. Pangesti et al., "The influence of social media on the cultural identity of the millennial generation: Indonesian case study," *Pubmedia Social Sciences and Humanities*, vol. 2, no. 1, pp. 1–7, 2024.

[20] C. R. Fink, D. S. Chou, J. J. Kopecky, and A. J. Llorens, "Coarse- and fine-grained sentiment analysis of social media text," *Johns Hopkins APL Tech. Dig.*, vol. 30, no. 1, pp. 22–30, 2011.

[21] Noryasminda, T. H. Saragih, R. Herteno, M. R. Faisal, and A. Farmadi, "Analysis of the effect of feature extraction on sentiment analysis using BiLSTM: Monkeypox case study on X/Twitter," *Indonesian J. Electron., Electromed. Eng., Med. Informatics*, vol. 7, no. 2, pp. 344–357, 2025.

[22] C. J. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in *Proc. Int. AAAI Conf. Web Soc. Media*, 2014, pp. 216–225.

[23] F. A. Kusumo, D. R. S. Saputro, and P. Widyaningsih, "Sentiment analysis of reviews on X apps on Google Play Store using support vector machine and N-gram feature selection," *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, vol. 19, no. 2, pp. 1037–1046, 2025.

[24] Q. Song, H. Ge, J. Caverlee, and X. Hu, "Tensor completion algorithms in big data analytics," *Big Data Min. Anal.*, vol. 13, no. 1, 2017.

[25] X. Tan, S. Su, Z. Huang, X. Guo, Z. Zuo, X. Sun, and L. Li, "Wireless sensor networks intrusion detection based on SMOTE and the random forest algorithm," *Sensors*, vol. 19, no. 1, pp. 1–15, 2019.

[26] G. Mutanov, V. Karyukin, and Z. Mamykova, "Multi-class sentiment analysis of social media data with machine learning algorithms," *Comput., Mater. Continua*, vol. 69, no. 1, pp. 914–930, 2021.

[27] Y. Yu, M. Li, L. Liu, Y. Li, and J. Wang, "Clinical big data and deep learning: Applications, challenges, and future outlooks," *Big Data Min. Anal.*, vol. 2, no. 4, pp. 288–305, 2019.