Sentiment Analysis of Affiliate Video Comments on the TikTok App

Fathimah Nurul Azizah[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*: fathimahnurulazizah@student.uns.ac.id

Abstract: *Indonesia ranks among the largest TikTok user bases globally and has the highest number of TikTok Shop stores worldwide. However, affiliate program participation in Indonesia remains low at only 3% of total users, far below the 17.6% recorded in the United States. User comments on TikTok affiliate videos represent a valuable data source for assessing consumer responses to promotional content, yet their large volume and unstructured nature make manual analysis inefficient. This study aimed to develop a sentiment classification model for TikTok affiliate video comments using the Random Forest algorithm with Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance. A dataset of 5,726 comments was manually collected from 50 TikTok affiliate videos in the beauty and personal care category. After preprocessing, sentiment labels were automatically generated using IndoBERTweet as a pseudo-labeling approach, resulting in 1,534 negative and 962 positive comments used for binary classification. Term Frequency-Inverse Document Frequency (TF-IDF) was applied to transform textual data into numerical features, while SMOTE was used to balance the class distribution. Model optimization using GridSearchCV with 10-fold cross-validation yielded the best Random Forest configuration with a cross-validation F_1 $score$ of 0.8711. The results show that Random Forest combined with SMOTE achieved an accuracy of 82% and a macro-average F_1 $score$ of 0.81 in classifying Indonesian-language TikTok affiliate comments.*
Keuwords: *Sentiment Analysis, TikTok Affiliate, IndoBERTweet, SMOTE, Random Forest*

1. INTRODUCTION

Indonesia ranks second globally in the number of TikTok users, with approximately 108 million active users as of December 9, 2025 [1]. As of June 2025, there were 515,000 TikTok shops or TikTok Shop accounts in Indonesia, the highest number in the world [2]. The TikTok affiliate program enables third parties (affiliates) to promote products from TikTok Shop accounts in exchange for commissions. However, the percentage of affiliates relative to the total number of users in Indonesia remains low (3%), far smaller than in the United States (17.6%), which is the country with the most TikTok users in the world [3]. This indicates that the affiliate market potential in Indonesia remains underutilized. Affiliate marketing is often associated with commission-driven product promotion, which requires a high level of trust from the audience toward the affiliate [4], [5].

Since affiliate marketing relies heavily on audience trust, understanding audience perception toward affiliate content becomes essential. One key indicator for understanding audience response to affiliate content is user comments. Comments may reflect acceptance, skepticism, or rejection of affiliate promotions [6]. Comment data serve as a valuable source for evaluating promotional effectiveness. However, the large volume and unstructured nature of comments make manual analysis inefficient. Therefore, a computational approach is needed to automatically extract sentiment-related information from comments.

Sentiment analysis is a natural language processing technique used to identify and classify opinions in textual data into specific categories, such as positive, negative, or neutral [7]. Sentiment analysis approaches are generally divided into two categories: lexicon-based and machine learning-based [8]. Machine learning-based approaches are widely adopted due to their ability to learn sentiment patterns from data more flexibly [9], [10].

One of the machine learning algorithms used for text classification is Random Forest [11]. This algorithm was introduced as an ensemble learning method that builds many random decision trees and combines their predictions to improve stability and accuracy [12], [13]. Random Forest has been shown to perform well in various classification tasks, including sentiment analysis on social media data [14], [15].

However, the performance of a classification model can decline when a dataset exhibits an imbalance in the number of data points across classes. This condition causes the model to tend to predict the majority class [16]. The Synthetic Minority Over-sampling Technique (SMOTE) was developed to address this issue. SMOTE generates synthetic samples in the minority class, thereby making the data distribution more balanced [17], [18].

Most sentiment analysis studies on social media focus on platforms such as Twitter (X) [19]. Meanwhile, research analyzing TikTok video comments, particularly on affiliate content, remains limited. Comments on affiliate videos provide direct insight into consumer responses to product promotions. Therefore, this study aims to conduct a sentiment analysis of TikTok affiliate video comments using the Random Forest approach with data imbalance addressed via SMOTE.

2. METHOD

2.1 Data Source

This study employs a quantitative approach using text classification methods to analyze the sentiment of comments on TikTok affiliate videos. The data consist of primary comments collected from TikTok affiliate videos containing the TikTok Shop purchase feature. The beauty and personal care categories were selected due to their highest Gross Merchandise Value (GMV) on TikTok Shop as of June 2025. The high transaction value in these categories indicates intense promotional activity and high user interaction, making them relevant for sentiment analysis of comments [20]. The five brands with the highest sales in these categories are Glad2Glow, The Originote, Madame Gie, Wardah, and Hanasui [21]. These brands served as keywords for retrieving affiliate videos to obtain comments data representing products with high exposure.

A total of 5,726 comments were collected manually from 50 TikTok affiliate videos uploaded between January 1 and December 31, 2025. Data were collected manually because the videos were not accessible via the web interface, preventing automated scraping. Data collection began with a video search based on the five top-selling brands in the beauty and personal care categories. Next, the first 10 videos appearing in the search results and containing at least 120 comments were selected. The collected comments were then saved in Comma Separated Value (CSV) format. The dataset used in this research was collected independently from TikTok affiliate video comments. Due to platform policy and privacy considerations, the dataset is not publicly distributed. However, the data may be available from the author upon reasonable request for academic purposes.

2.2 Research Steps

The analytical stages of this study were conducted systematically in the following order.

1. Collecting TikTok affiliate video comments.
2. Preprocessing the comments (cleaning, case folding, tokenizing, stop word removal, and stemming).
3. Assigning sentiment labels using IndoBERTweet.
4. Removing neutral-labeled comments.
5. Splitting the dataset into training (80%) and testing (20%) sets using stratified sampling.
6. Applying TF-IDF vectorization.
7. Performing class balancing using SMOTE.
8. Training the classification model using Random Forest algorithm.
9. Evaluating model performance.

2.3 IndoBERTweet

The pre-trained IndoBERTweet model is used to assign positive, negative, or neutral sentiment labels to each comment. Data labeled as neutral are excluded from the dataset and not used for model training. IndoBERTweet is a transformer-based pre-trained model resulting from the pre-training of the BERT architecture, adapted using 26 million Indonesian-language tweets, enabling it to understand the characteristics of informal language, slang, and abbreviations commonly found on social media [22]. The generated sentiment labels serve as pseudo-labels and are used as target variables [23].

2.4 TF-IDF

TF-IDF is a vectorization technique used to convert text data into numerical values to enable processing by machine learning algorithms [24]. TF-IDF assigns weights based on term frequency (TF) within a document and inverse document frequency (IDF) across the corpus. Through this approach, words that frequently appear in one document but are rarely found in others receive higher weights, thereby contributing more strongly to the classification process. The formula for calculating TF-IDF weights is

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

where $w(d,t)$ is the TF-IDF weight or the weight of word t in document d, $tf(d,t)$ is the number of occurrences of word t in document d, $df(t)$ is the document frequency of word t, and N is the total number of documents.

2.5 SMOTE

SMOTE is an oversampling method designed to address the issue of class imbalance in datasets. SMOTE generates new samples by randomly interpolating between minority class samples and their nearest neighbors, thereby balancing the class distribution [17]. The formula for interpolation is

$$p_i = X + rand(0,1) \times (y_i - X)$$

where $i = 1, 2, \ldots, N$, X is a sample of minority class data, $rand(0,1)$ is a random number in the interval (0,1), and y_i is the i-th nearest neighbor of the data sample X.

2.6 *Random Forest*

Random Forest is an ensemble learning method that builds multiple decision trees simultaneously to improve predictive stability and accuracy while reducing overfitting [12]. In general, the steps of Random Forest for classification are as follows:

1. From a training dataset of size N, B bootstrap datasets are generated through random sampling with replacement.
2. For each bootstrap dataset, a decision tree is built using the Classification and Regression Tree (CART) algorithm, which involves randomly selecting m features at each node, determining the best split based on Gini impurity, and growing each tree to maximum depth without pruning.
3. Once all B trees have been formed, predictions for new samples are obtained by traversing the sample through each tree and determining the final class via majority voting.

2.7 Model Evaluation

A confusion matrix is a method for evaluating classification models by comparing predicted values with actual data values. The confusion matrix for binary classification is shown in Table 1 [25][26].

Table 1. Confusion matrix for binary classification.

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

TP is the number of positive labels correctly predicted by the model, TN is the number of negative labels correctly predicted by the model, FP is the number of positive labels incorrectly predicted by the model, and FN is the number of negative labels incorrectly predicted by the model. Based on these four values, the accuracy and F_1 $score$ can be calculated using the formulas expressed as

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

$$F_1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

where

$$Precision = \frac{TP}{TP + FP}$$

and

$$Recall = \frac{TP}{TP + FN}$$

Accuracy measures the proportion of correctly classified instances, while the $F_1\ score$ reflects the balance between precision and recall.

3. RESULTS AND DISCUSSION

The results and discussion of this study are divided into six sections: preprocessing, sentiment labeling and data splitting, TF-IDF, SMOTE, classification model training, and model performance evaluation. Each section is described as follows.

3.1 *Preprocessing*

Preprocessing aims to transform unstructured text into a structured representation. The preprocessing steps performed in this study consist of five stages. The first is cleaning, which removes unnecessary elements such as punctuation marks, numbers, symbols, emojis, HTML tags, URLs, or other symbols that do not contribute to the actual meaning. Next is case folding, which standardizes the capitalization of all letters to avoid inconsistencies caused by capitalization differences. Following that is tokenization, the process of breaking down sentences or documents into smaller units, typically words or tokens (the smallest units of analysis), that can be analyzed separately. The fourth step is stopword removal, which is the process of removing common words that contribute little semantic information for classification, such as “di”, “ke”, “dan”, “yang”, “adalah”, etc. The final step is stemming, which is the process of converting words to their base forms so that variations of words with the same meaning can be grouped together. For example, the words “bermain”, “permainan”, and “dimainkan” are converted to the word “main” [26]. Table 2 shows the results of each preprocessing step on two comments obtained.

Table 2. Results of preprocessing on two comments obtained

No.	Preprocessing stage	First comment	Second comment
1	Original comment	menurut ku yang paling bagus untuk bruntusan sama jerawat itu pratista calming spray gak ad dari itu 😘 😘	Sumpah bagus wajah jdi cerahan . Tp knp cpet hbissss 😌 😌 😌
2	Cleaning	menurut ku yang paling bagus untuk bruntusan sama jerawat itu pratista calming spray gak ad dari itu	Sumpah bagus wajah jdi cerahan Tp knp cpet hbissss
3	Case folding	menurut ku yang paling bagus untuk bruntusan sama jerawat itu pratista calming spray gak ad dari itu	sumpah bagus wajah jdi cerahan tp knp cpet hbissss
4	Tokenizing	‘menurut’, ‘ku’, ‘yang’, ‘paling’, ‘bagus’, ‘untuk’, ‘bruntusan’, ‘sama’, ‘jerawat’, ‘itu’, ‘pratista’, ‘calming’, ‘spray’, ‘gak’, ‘ad’, ‘dari’, ‘itu’	‘sumpah’, ‘bagus’, ‘wajah’, ‘jdi’, ‘cerahan’, ‘tp’, ‘knp’, ‘cpet’, ‘hbissss
5	Stopword removal	‘ku’, ‘paling’, ‘bagus’, ‘bruntusan’, ‘sama’, ‘jerawat’, ‘pratista’, ‘calming’, ‘spray’, ‘gak’, ‘ad’	‘sumpah’, ‘bagus’, ‘wajah’, ‘jdi’, ‘cerahan’, ‘tp’, ‘knp’, ‘cpet’, ‘hbissss’
6	Stemming	‘ku’, ‘paling’, ‘bagus’, ‘bruntusan’, ‘sama’, ‘jerawat’, ‘pratista’, ‘calming’, ‘spray’, ‘gak’, ‘ad’	‘sumpah’, ‘bagus’, ‘wajah’, ‘jdi’, ‘cerah’, ‘tp’, ‘knp’, ‘cpet’, ‘hbissss’

3.2 Sentiment Labeling and Data Splitting

After the preprocessing stage, a total of 4,904 comment lines were obtained for sentiment labeling. Next, all comments were sentiment-labeled using the pre-trained IndoBERTweet model. The model classified the comments into three sentiment categories: positive, negative, and neutral. Based on the sentiment labeling using IndoBERTweet, 2,408 comments were labeled as neutral, 1,534 as negative, and 962 as positive. However, no manual validation was conducted to quantitatively assess labeling accuracy, which may introduce bias due to potential misclassification by the model. This limitation should be addressed in future work through human-annotated evaluation.

Neutral comments were excluded to simplify the classification task into a binary problem, allowing the model to focus on distinguishing clearly positive and negative sentiments. This approach is commonly used to improve classification performance when the primary objective is to identify polarity extremes. However, this simplification reduces the realism of sentiment representation, as neutral opinions are prevalent in real-world data. Therefore, future research should consider multi-class classification to better reflect natural sentiment distributions.

The dataset was then split into training and testing data in an 80:20 ratio using stratified sampling to preserve the sentiment distribution across both datasets. Based on this data split, 1,996 comments were allocated to the training set and 500 comments to the testing set. The sentiment in each dataset was distributed as 61.5% negative comments and 38.5% positive comments. Thus, 1,227 negative comments and 769 positive comments were obtained for the training data, and 307 negative comments and 193 positive comments for the testing data.

3.3 TF-IDF

TF-IDF vectorization on the training data produces a 1,996 × 2,559 matrix, while the testing data produces a 500 × 2,559 matrix. The identical number of features indicates that the testing data uses a vocabulary derived from the training data, allowing it to be directly applied during the model evaluation phase. An example of the TF-IDF vectorization results for the training data is shown in Table 3.

Table 3. Example of TF-IDF vectorization results on the training data

Word	TF	IDF	TF-IDF
‘aha’	0.0714	7.5008	0.3286
‘aku’	0.0714	2.3360	0.1023
‘apa’	0.0714	3.9942	0.1750
‘bha’	0.0714	7.9063	0.3463
‘fw’	0.0714	6.2015	0.2716
‘gimana’	0.0714	6.2015	0.2716
‘iiih’	0.0714	7.9063	0.3463
‘kalo’	0.0714	4.3947	0.1925
‘misal’	0.0714	7.2131	0.3160
‘ndak’	0.0714	7.9063	0.3463
‘nya’	0.0714	3.3899	0.1485
‘pake’	0.0714	3.0159	0.1321
‘retinol’	0.0714	4.9618	0.2173
‘tah’	0.0714	7.9063	0.3463

The TF-IDF vectorization converts textual data into numerical feature vectors that represent the relative importance of words within the dataset. Words that appear frequently in specific documents but rarely across other documents receive higher TF-IDF weights, allowing the model to identify discriminative linguistic features associated with sentiment polarity. The consistent vocabulary between training and testing data ensures that the model evaluates unseen comments within the same feature space, enabling reliable classification performance. The testing data was transformed using the TF-IDF model fitted on the training data to maintain consistency in the feature space. An example of TF-IDF values for the testing data is presented in Table 4.

Table 4. Example of TF-IDF vectorization results on the testing data

Word	TF	IDF	TF-IDF
‘aja’	0.1	4.5390	0.3637
‘aku’	0.2	2.3360	0.3744
‘cocok’	0.1	3.8119	0.3055

'jerawat'	0.1	3.6578	0.2931
'kulit'	0.1	3.3419	0.2678
'kuning'	0.1	5.3036	0.4250
'minyak'	0.1	3.9744	0.3185
'pke'	0.1	4.5564	0.3651
'yg'	0.1	3.0860	0.2473

3.4 SMOTE

After performing TF-IDF vectorization, the balance of the sentiment class distribution in the training data was identified. The results indicate the presence of class imbalance prior to SMOTE, with 1,227 negative and 769 positive comments. To address this issue, SMOTE was applied to the training data. The SMOTE results showed that the sentiment distribution became balanced, with 1,227 comments in each class. Consequently, the amount of training data used also increased from 1,996 comments to 2,454 comments.

3.5 Classification Model Training

After class balancing was performed using SMOTE, the training dataset consisted of 1,227 positive-labeled comments and 1,227 negative-labeled comments. This balanced dataset was then used to train a classification model using the Random Forest algorithm. The Random Forest model builds multiple decision trees based on bootstrap samples using the CART algorithm. Final predictions are obtained through majority voting across all trees.

During the model development phase, 10-fold cross-validation was performed to evaluate the model's performance stability across various subsets of the training data [27]. Additionally, parameter optimization was conducted using GridSearchCV to identify the parameter combination that yields the best performance [28]. The parameters tested included the number of trees (n_estimators) set to 100 and 200, tree depth (max_depth) set to None, 20, and 40, the minimum number of samples required to split a node (min_samples_split) set to 2 and 5, the minimum number of samples at terminal nodes (min_samples_leaf) of 1 and 2, and the number of features selected at each split (max_features) using the sqrt value. Each parameter combination was evaluated using the $F_1\,score$ metric.

Based on the results of the parameter search using GridSearchCV, the best combination was obtained with n_estimators = 200, max_depth = None, min_samples_split = 5, min_samples_leaf = 1, and max_features = sqrt. This combination yielded a cross-validation score of 0.8711, which was the highest score compared to other parameter combinations. The value n_estimators = 200 indicates that the model builds 200 decision trees, which are then combined using majority voting, resulting in a more stable final prediction. The parameter max_depth = None indicates that each CART tree is grown without a specific depth limit until no better split is found based on Gini impurity or the minimum number of samples at a node.

The parameter min_samples_split = 5 indicates that a node is split only if it contains at least five samples, thereby reducing the formation of overly specific branches and improving the model's generalization ability. Meanwhile, min_samples_leaf = 1 ensures that each terminal node contains at least one sample, allowing the model to continue capturing detailed patterns in the data. The parameter max_features = sqrt indicates that, at each node, the model considers only a number of features equal to the square root of the total number of available features to determine the best split. With 2,559 TF-IDF features, approximately 50 features are randomly selected during each node splitting process. This approach increases the diversity among trees in the forest, thereby reducing correlation between trees and mitigating the risk of overfitting.

3.6 Model Performance Evaluation

The model with the best parameters was then tested using the testing data to assess its classification performance on data it had not previously been trained on. The results of the testing for the best classification model are shown in Table 5 below.

Table 5. Results of testing the best classification model

	Precision	Recall	$F_1 score$	Support
Negative	0.82	0.91	0.86	307
Positive	0.83	0.68	0.75	193
Accuracy			0.82	500
Macro Avg	0.83	0.80	0.81	500
Weighted Avg	0.82	0.82	0.82	500

The results in Table 5 show that the model achieved an accuracy of 0.82, indicating that 82% of testing samples were correctly classified. Table 6 presents the confusion matrix of the classification results.

Table 6. Confusion matrix results

	Predicted Negative	Predicted Positive
Actual Negative	280	27
Actual Positive	61	132

The confusion matrix shows that out of 307 actual negative data points, 280 were correctly predicted as negative, while 27 were misclassified as positive. In the positive class, out of 193 actual positive data points, 132 were correctly predicted, while 61 were predicted as negative. These results indicate that the model performs better in identifying the negative class than the positive class, as reflected by higher recall for negative comments (0.91) compared to positive comments (0.68). The higher recall for the negative class indicates that the model identified negative sentiment instances more consistently than positive sentiment instances in the testing data.

3.7 Discussion

The results indicate that the proposed model achieved satisfactory performance, with an accuracy of 0.82 and a macro-average $F_1\ score$ of 0.81. However, a deeper analysis reveals several important patterns related to model behavior and dataset characteristics.

First, the model demonstrates higher recall for the negative class (0.91) than for the positive class (0.68), indicating greater sensitivity in detecting negative sentiment. One possible explanation is that negative comments tend to contain more explicit and distinctive lexical patterns, such as complaints, dissatisfaction, or strong emotional expressions, making them easier to classify. In contrast, positive comments are often shorter, more ambiguous, or expressed using informal slang, which reduces their separability in the feature space. Second, although SMOTE successfully balanced the class distribution, it may also introduce synthetic samples that do not fully represent real linguistic variations. This can affect the model's generalization ability, particularly for the positive class. Third, the use of TF-IDF as a feature representation captures term importance but does not preserve contextual or semantic relationships between words. This limitation may reduce the model's ability to correctly interpret nuanced or context-dependent sentiment expressions commonly found in social media text.

Previous studies on sentiment analysis using machine learning methods have reported varying performance depending on dataset characteristics and feature representation. For example, studies using SVM and Random Forest on social media data have shown comparable performance, with $F_1\ score$ generally ranging between 0.75 and 0.85 [14], [15]. The results obtained in this study ($F_1\ score$ of 0.81) fall within this range, indicating that the proposed approach performs consistently with existing literature. However, differences in dataset domain, language style, and preprocessing techniques may significantly influence performance outcomes. Furthermore, the absence of direct benchmarking limits the ability to determine whether Random Forest is the optimal model.

Additionally, the reliance on pseudo-labeling using IndoBERTweet introduces potential labeling noise. Without manual validation, misclassified labels may propagate into the training process and affect model performance. This limitation highlights the importance of combining automated labeling with human annotation in future research. Finally, the exclusion of neutral sentiment simplifies the classification problem but reduces the realism of sentiment distribution. In real-world scenarios, neutral opinions are common and play an important role in understanding audience perception. Therefore, future studies should consider multi-class classification to better capture the full spectrum of sentiment.

4. CONCLUSION

This study demonstrates that Random Forest combined with TF-IDF and SMOTE can effectively classify sentiment in TikTok affiliate comments. The best model achieved an accuracy of 82% and a macro-average $F_1\ score$ of 0.81. The model showed stronger performance in detecting negative sentiment than positive sentiment, as reflected in the class-wise F_1 scores. SMOTE balanced the training data distribution from 1,227 negative and 769 positive comments to 1,227 comments in each class, enabling balanced learning across classes. However, several limitations remain, including the lack of pseudo-label validation, the exclusion of neutral sentiment, and the absence of baseline model comparison. These limitations may affect the robustness and generalizability of the findings. Future research should address these issues by incorporating manual annotation, multi-class sentiment classification, and comparative evaluation with other machine learning and deep learning models.

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] "TikTok Users by Country 2025," World Population Review, 2025. [Online]. Available: https://worldpopulationreview.com/country-rankings/tiktok-usersby-country. [Accessed: Dec. 9, 2025].

[2] "TikTok Shop Statistics (2025) Global GMV, Sales, and More," Resourcera, 2025. [Online]. Available: https://resourcera.com/data/social/tiktok-shop-statistics/. [Accessed: Dec. 9, 2025].

[3] "Affiliate Marketing Semakin Meningkat di Indonesia Tapi Jumlah Affiliator Masih Rendah, Ini Penjelasannya!," KOL.id, 2025. [Online]. Available: https://kol.id/blog/strategiaffiliate-marketing-semakin-meningkat-di-indonesia-tapi-jumlahaffiliator-rendah. [Accessed: Dec. 9, 2025].

[4] N. P. C. Piliana and A. F. Maradona, "Strategy to increase consumer trust in affiliate links on social media," *Quantitative Economics and Management Studies*, vol. 5, no. 1, pp. 1–10, 2024, doi: 10.35877/454RI.qems2800.

[5] A. Mathur, A. Narayanan, and M. Chetty, "Endorsements on social media: An empirical study of affiliate marketing disclosures on YouTube and Pinterest," *Proc. ACM Hum.-Comput. Interact.*, 2018. [Online]. Available: https://arxiv.org/abs/1809.00620. [Accessed: Mar. 18, 2026].

[6] T. U. Yugita and A. B. P. Kasmo, "The role of trust in moderating the impact of affiliate and social media marketing on financial service purchase decisions," *Formosa Journal of Applied Sciences*, vol. 3, no. 8, 2024, doi: 10.55927/fjas.v3i8.10726.

[7] M. Umar, H. Binji, and A. Balarabe, "Corpus-based approaches for sentiment analysis: A review," Asian J. Res. Comput. Sci., vol. 17, 2024, doi: 10.9734/AJRCOS/2024/v17i7481.

[8] Q. Ain *et al.*, "Sentiment analysis using deep learning techniques: A review," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 6, 2017, doi: 10.14569/IJACSA.2017.080657.

[9] T. Khan, R. Sadiq, Z. Shahid, M. Alam, and M. Su'ud, "Sentiment analysis using support vector machine and random forest," J. Informatics Web Eng., vol. 3, no. 1, 2024, doi: 10.33093/JIWE.2024.3.1.5.

[10] M. Kumar, L. Khan, and H. Chang, "Evolving techniques in sentiment analysis: A comprehensive review," PeerJ Comput. Sci., vol. 11, 2025, doi: 10.7717/peerj-cs.2592.

[11] [11] J. Hu and S. Szymczak, "A review on longitudinal data analysis with random forest," Brief. Bioinform., vol. 24, 2022, doi: 10.1093/bib/bbad002.

[12] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[13] R. Genuer, J. Poggi, C. Tuleau-Malot, and N. Villa-Vialaneix, "Random forests for big data," Big Data Res., vol. 9, pp. 28–46, 2017, doi: 10.1016/j.bdr.2017.07.003.

[14] J. Hartmann, J. Huppertz, C. Schamp, and M. Heitmann, "Comparing automated text classification methods," Int. J. Res. Mark., vol. 36, no. 1, pp. 20–38, 2019, doi: 10.1016/j.ijresmar.2018.09.009.

[15] R. Naseem and S. Sharma, "Benchmarking machine learning methods for sentiment analysis in social media:

A comprehensive investigation," in Proc. IEEE Students Conference on Engineering and Systems (SCES), 2024, pp. 1–5, doi: 10.1109/SCES61914.2024.10652327.
[16] M. Azwar, P. Hariyadi, and R. Azhar, "Assessing Twitter user sentiment regarding divorce issues using the random forest method," Int. J. Eng. Comput. Sci. Appl., vol. 4, no. 2, 2025, doi: 10.30812/IJECSA.v4i2.4980.
[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.
[18] A. Maxwell, T. Warner, and F. Fang, "Implementation of machine-learning classification in remote sensing: An applied review," Int. J. Remote Sens., vol. 39, pp. 2784–2817, 2018, doi: 10.1080/01431161.2018.1433343.
[19] J. Cui, Z. Wang, S.-B. Ho, and E. Cambria, "Survey on sentiment analysis: Evolution of research methods and topics," Artif. Intell. Rev., vol. 56, no. 8, pp. 8469–8510, 2023, doi: 10.1007/s10462-022-10386-z.
[20] "TikTok shop statistics (2025): Global GMV, sales, and more," Resourcera, 2025. [Online]. Available: https://resourcera.com/data/social/tiktok-shop-statistics/. [Accessed: Dec. 9, 2025].
[21] "TikTok beauty & personal care shop in Indonesia 2025," FindNiche, 2025. [Online]. Available: https://findniche.com/tiktok/top-sellers-beauty-and-personal-careid. [Accessed: Dec. 9, 2025].
[22] F. Koto, J. H. Lau, and T. Baldwin, "IndoBERTweet: A pretrained language model for Indonesian Twitter with effective domain-specific vocabulary initialization," in *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, Nov. 2021, pp. 10660–10668.
[23] A. Aribowo, S. Khomsah, and S. Saifullah, "Semi-supervised sentiment classification using self-learning and enhanced co-training," J. Infotel, vol. 17, no. 3, pp. 472–489, Aug. 2025, doi: 10.20895/INFOTEL.v17i3.1344.
[24] J. Ramos, "Using TF-IDF to determine word relevance in document queries," in *Proc. First Instructional Conf. Mach. Learn.*, 2003, pp. 133–142.
[25] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Waltham, MA, USA: Morgan Kaufmann, 2011.
[26] 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 J. Ilmu Mat. Terap., vol. 19, no. 2, pp. 1037–1046, 2025.
[27] M. R. Ramadhan and K. Budiman, "Sentiment analysis of presidential candidates in 2024: A comparison of the performance of support vector machine and random forest with N-gram method," Recursive J. Informatics, vol. 3, no. 1, pp. 34–42, 2025.
[28] G. Anyanwu, C. Nwakanma, J. Lee, and D. Kim, "Optimization of RBF-SVM kernel using grid search algorithm for DDoS attack detection in SDN-based VANET," IEEE Internet of Things Journal, vol. 10, pp. 8477–8490, 2023, doi: 10.1109/JIOT.2022.3199712.