Intechno Journal : Information Technology Journal
Vol. 7 No. 2 December 2025
Homepage: https://jurnal.amikom.ac.id/index.php/intechno

INTECHNO Journal
INFORMATION TECHNOLOGY JOURNAL

e-ISSN: 2655-1438
p-ISSN: 2655-1632

.

A Comparative Analysis of Decision Tree, Logistic Regression, and Support Vector Machine Algorithms in Sentiment Analysis of Threads App Reviews

Rahmat Hidayat[1], Farhan Aminulhaq[2]
[1]Universitas Terbuka, Jalan Cabe Raya, Pondok Cabe, Pamulang, Tangerang Selatan 15437, Indonesia
[2]TVRI Stasiun Sumatera Barat

Article Info

Article history:

Received: 22 December 2025
Revised: 24 December 2025
Accepted: 30 December 2025
Published: 31 December 2025

Keywords:

Support Vector Machine
Decision Tree
Logistic Regression
Sentimen Analysis
Threads

ABSTRACT

Purpose: This study aims to analyze user sentiment regarding the Threads application by comparing the performance of different machine learning models. As a relatively new social media platform, understanding user feedback is crucial for identifying service gaps and improving user retention. The research seeks to determine which algorithm provides the highest precision in classifying user reviews into positive and negative sentiments.
Methods: The research utilized a dataset of 3,000 user reviews scraped from the Google Play Store. The methodology followed a systematic text mining workflow, including preprocessing stages such as noise removal, tokenization, stopword removal, and stemming. Feature extraction was performed using the Term Frequency-Inverse Document Frequency (TF-IDF) method. Three machine learning algorithms—Support Vector Machine (SVM), Decision Tree, and Logistic Regression—were implemented and evaluated using K-Fold Cross Validation to ensure statistical reliability.
Result: The experimental results indicate that the Support Vector Machine (SVM) consistently outperformed the other two models. SVM achieved a superior average accuracy of 88.18%, with a peak performance reaching 92.69% during K-Fold testing. Logistic Regression and Decision Tree showed lower accuracy and less stability in handling the high-dimensional text data. These figures confirm that SVM is the most effective model for analyzing the linguistic nuances found in Threads app reviews.
Novelty/Originality/Value: This research contributes to the field of software evaluation by providing an empirical comparison of classification algorithms specifically for newly launched social media platforms like Threads. The findings offer practical value for developers to automate the monitoring of user satisfaction. The study demonstrates that integrating rigorous TF-IDF weighting with SVM significantly enhances the accuracy of sentiment detection in short-form mobile application reviews.

Corresponding Author:

Rahmat Hidayat
Email: imjustrahmat2722@gmail.com

1. INTRODUCTION

Advancements in information technology have revolutionised interpersonal interactions and opinion sharing in the digital realm, especially with the increasing use of microblogging social media platforms like as Threads [1]. Threads, a global platform enabling users to post text, photographs, and videos while participating in public discourse, has become integral to contemporary digital existence. Members of Threads offer feedback by submitting evaluations of the service. The user ratings furnish Threads with essential insights into user happiness and possible application difficulties [3]. This study aims to elucidate the distribution of neutral, negative, and positive sentiments in Threads user reviews [4]. This study will elucidate and compare the efficacy of three classifiers: Support Vector Machine (SVM), Decision Tree, and Logistic Regression. This assessment will determine whether algorithm has superior performance in the sentiment categorisation of Threads review data. [5]

Title: "Sentiment Analysis Regarding Mobile Identity Card Data on Twitter." The aim is to recognise the term "baran" and evaluate the sentiment classification of public opinion on Twitter related to this issue [6]. The model's development findings indicate that the Support Vector Machine technique exhibits the highest performance, with a f1-score of 0.81, followed by Random Forest at 0.78, IndoBERT at 0.76, and Logistic Regression at 0.74 [7]. The class imbalance and inadequate training data negatively impact IndoBERT's performance, rendering it inferior to competing approaches while being a state-of-the-art model in NLP [8]. The main aim of this study is to provide Threads with insights into user assessments of their services derived from feedback. By understanding this emotion, Threads may identify areas within their services that want improvement or extension, while also highlighting appealing elements that customers appreciate [9]. This study aims to investigate the influence of preprocessing techniques, such as character analysis, tokenisation, and stemming, on classification accuracy. Proficient text preparation is crucial as it boosts the quality of data input for machine learning algorithms, hence enhancing system performance and classification accuracy [10]. Threads can utilise the results of this sentiment analysis to inform business decisions, such as product development, content creation, or enhancements in customer support. This study conceptually advances sentiment analysis and the use of machine learning in text categorisation, providing a foundation for future research in user review analysis and other machine learning applications in the digital entertainment industry.

2. METHOD

This research applies a machine learning approach as the basis for developing and evaluating a sentiment analysis model for user reviews of the Threads app. This approach was chosen because it aligned with the research objective, namely to compare the performance of several classification algorithms in accurately and objectively identifying user sentiment. The algorithms used included Decision Tree, Logistic Regression, and Support Vector Machine, which were chosen for their representative characteristics in classifying text data. The research process was conducted in a structured and systematic manner, starting from the app review data collection stage, text preprocessing, feature extraction, model training, and performance evaluation using relevant metrics. In general, this research flow consists of five main stages arranged sequentially to ensure the consistency and validity of the comparative analysis results between algorithms.

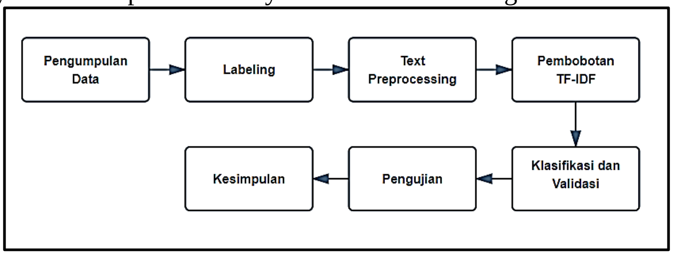

Figure 1. Research Method

2.1. Data Collection

This study uses a Threads app dataset taken from Threads app comments on the Google Play Store using Google Play Store Python scraping. The dataset consists of 3,000 entries. The following is the Threads.csv dataset.

id	content	score
0	Nonton video di Threads ngga full screen layar, jadi kurang nyaman...	1
1	Selamat sore Threads ini pengguna baru di smartphone...	5
2	😀	5
3	👍👍👍	5
4	threads masih jd apk kecintaan buat update status...	5

Figure 2. Threads dataset

.

2.2. Labeling

This study employed labelling to categorise values in the score column as positive (1), negative (-1), and neutral (0). Scores beyond 3 (>3) were classified as positive sentiment (1), scores below 3 (<3) were classified as negative sentiment (-1), while scores precisely equal to 3 (=3) were classified as neutral (0).

2.3. Text Preprocessing

This study performed text preprocessing in the form of character removal, tokenization, stopword removal, stemming, and removal of words with less than 4 characters.. The explanation is as follows:

2.3.1. Character Removal

This stage removes characters, numbers, punctuation, and so on [6].

Table 1. Character Deletion

Teks Asli	Hasil Preprocessing
sangat seru cocok untuk pengguna baru Threads/Twitter	sangat seru cocok untuk pengguna baru ThreadsTwitter
aplikasi ini keren bgt!! #threads #sosmed	aplikasi ini keren bgt threads sosmed

2.3.2. Tokenization

Tokenization is the process of breaking text into smaller parts, such as individual words or specific terms [7].

Table 2. Tokenization

Preprocessing (Kalimat)	Tokenize (List Kata)
sangat seru cocok untuk pengguna baru threads	['sangat', 'seru', 'cocok', 'untuk', 'pengguna', 'baru', 'threads']
aplikasi ini sangat ringan dan mudah digunakan	['aplikasi', 'ini', 'sangat', 'ringan', 'dan', 'mudah', 'digunakan']

2.3.3. Stopwords

This stage is carried out to remove common words that do not provide much contextual information [8].

Table 3. Stopword

Tokenize	Stopword (Result)
['sangat', 'seru', 'cocok', 'untuk', 'pencinta', 'drakor', 'anime']	['seru', 'cocok', 'pencinta', 'drakor', 'anime']
['aplikasi', 'ini', 'sangat', 'ringan', 'dan', 'mudah', 'digunakan']	['aplikasi', 'ringan', 'mudah', 'digunakan']

2.3.4. Stemming

This stage is carried out to simplify various word forms into one basic form, thus facilitating text analysis [9].

Table 4. Stemming

Stopword	Stemming (Kata Dasar)
['seru', 'cocok', 'pencinta', 'drakor', 'anime']	['seru', 'cocok', 'cinta', 'drakor', 'anime']
['aplikasi', 'ringan', 'mudah', 'digunakan']	['aplikasi', 'ringan', 'mudah', 'guna']

2.3.5. Word Length Filtering

This stage is carried out to simplify various word forms into one basic form, thus facilitating text analysis [9].

Table 5. Delete Words

Stemming	Hapus Kata (Final)
['seru', 'cocok', 'cinta', 'drakor', 'anime']	['seru', 'cocok', 'cinta', 'drakor', 'anime']
['s', 'seru', 'cocok', 'cinta', 'drakor', 'anime']	['seru', 'cocok', 'cinta', 'drakor', 'anime']
['aplikasi', 'ringan', 'mudah', 'guna', 'y']	['aplikasi', 'ringan', 'mudah', 'guna']
['fitur', 'baru', 'bantu', 'produktivitas', 'k']	['fitur', 'baru', 'bantu', 'produktivitas']
['a', 'sering', 'error', 'login', 'akun']	['sering', 'error', 'login', 'akun']

In Table 5, the wording remains unchanged because there are no words with less than 4 letters.

2.4. TF-IDF Weighting

At this juncture, TF-IDF weighting is executed to assess the pertinence of terms within the context of a specific document [11]. The TF-IDF weighting is executed with a maximum of 2,500 characteristics. This is a compilation of TFIDF features.

Table 6. TF-IDF

Term	TF-IDF
bagus	242,960,541
aplikasi	133,937,176
good	116,093,599
sangat	115,471,402
film	106,958,993
mantap	103,364,437
bisa	91,602,613
keren	76,110,831
Threads	67,998,000
nonton	63,539,290

After that, the SVM and Decision Tree algorithms were tested, which will be discussed in Chapter 3.

3. RESULTS AND DISCUSSIONS

At this stage, testing of the Support Vector Machine, Decision Tree, and Logistic Regression algorithms was conducted, as well as data validity testing using Cross Validation.

3.1. Support Vector Machine Testing

SVM (Support Vector Machine) is a classification algorithm that finds the best hyperplane to separate data into different classes by maximizing the margin [2].

Table 7. Klasifikasi Sentimen Suppoer Vector Machine

DL : DU	Accuracy	Precision	Recall	F1-Measure
90% : 10%	89.14%	89.18%	89.14%	89.13%
80% : 20%	88.14%	88.43%	88.14%	88.13%
70% : 30%	88.29%	88.43%	88.29%	88.27%
60% : 40%	87.93%	88.16%	87.93%	87.91%

Support Vector Machine testing exhibited optimal performance utilising 90% of the training data and 10% of the testing data, attaining an accuracy of 89.14%, with precision, recall, and F-Measure values of roughly 89.18%, 89.14%, and 89.13%, respectively. Prior studies [12] indicate that augmenting the volume of training data generally enhances the efficacy of text classification algorithms. Figure 3 delineates the test findings, demonstrating the effects of altering the quantities of training and testing data.

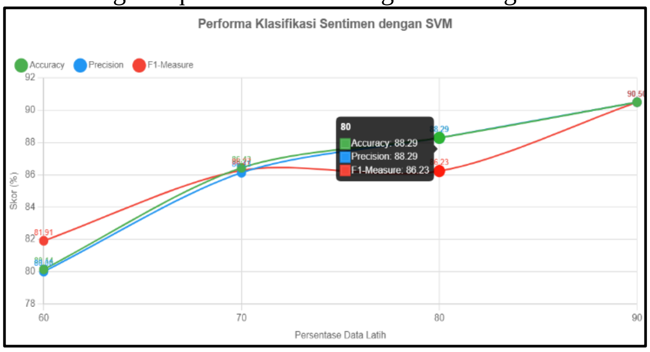

Figure 3. Test Graph of Training Data and Support Vector Test Data Machine

3.2. Decision Tree Testing

A decision tree is an algorithm used to separate data into classes based on a series of hierarchical decision rules [13].

.

Table 8. Decision Tree Sentiment Classification

Rasio DL:DU	Accuration	Precision	Recall	F1-Measure
90%:10%	86.57%	86.81%	86.57%	86.57%
80%:20%	84.00%	84.00%	84.00%	83.98%
70%:30%	85.81%	86.09%	85.81%	85.79%
60%:40%	85.07%	85.21%	85.07%	85.06%

The Decision Tree test exhibited optimal performance utilising 90% of the training data and 10% of the testing data, attaining an accuracy of 86.57% and precision, recall, and F-Measure values of roughly 86.81%, 86.57%, and 86.57%, respectively. This observation aligns with the classification report results, as indicated by [12], which asserts that augmenting the training data typically enhances the efficacy of text classification models. Figure 4 illustrates the effect of altering the quantity of training and testing data.

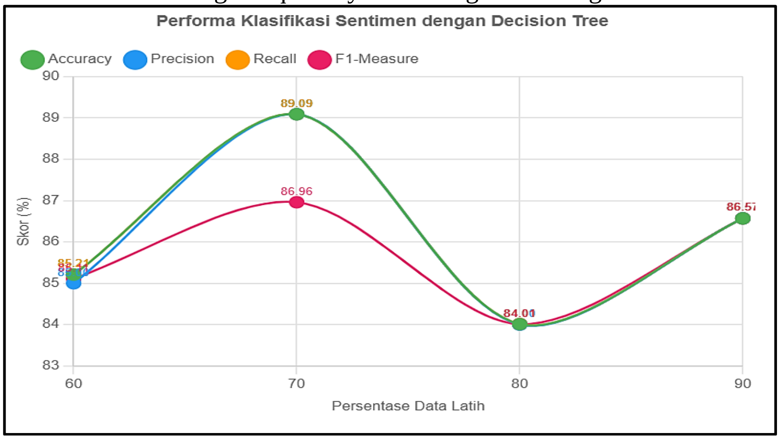

Figure 4. Training Data and Decision Tree Test Data

3.3. Logistic Regression Testing

Logistic regression is a common method used to predict the class or label of data based on given input features [14].

Table 9. Logistic Regression Sentiment Classification

Rasio DL:DU	Accuration	Precision	Recall	F1-Measure
90%:10%	86.57%	88.39%	86.57%	86.49%
80%:20%	84.14%	85.52%	84.14%	84.04%
70%:30%	85.43%	86.44%	85.43%	85.33%
60%:40%	85.36%	86.59%	85.36%	85.25%

Logistic Regression testing exhibited optimal performance with 90% of the training data and 10% of the testing data, attaining an accuracy of 86.57% and precision, recall, and F-Measure values of approximately 88.39%, 86.57%, and 86.49%, respectively. This observation aligns with the classification report detailed in [15], which indicated that augmenting the training data typically enhances the efficacy of text classification models. Figure 5 illustrates the effect of altering the quantity of training and testing data.

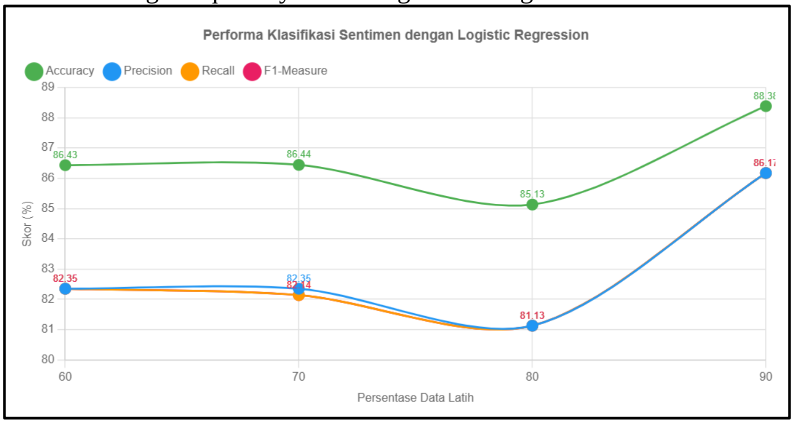

Figure 5. Training Data and Logistic Regression Test Data

3.4. Cross-Validation Testing

Cross-validation testing on the Support Vector Machine algorithm is performed to validate data based on the best results from previous testing [16], which are presented in

Table 10. Support Vector Machine Cross Validation Testing

Fold	Accuration	Precision	Recall	F1-Measure
1	90.38%	90.47%	90.38%	90.25%
2	87.31%	87.56%	87.31%	87.21%
3	85.77%	85.71%	85.77%	85.73%
4	87.69%	87.67%	87.69%	87.54%
5	89.62%	89.99%	89.62%	89.48%
6	85.38%	85.44%	85.38%	85.22%
7	87.69%	88.10%	87.69%	87.43%
8	89.62%	89.60%	89.62%	89.57%
9	92.69%	92.73%	92.69%	92.64%
10	89.58%	89.57%	89.58%	89.50%
Average	88.18%	88.68%	88.68%	88.46%*

Based on Table 10, the highest performance was achieved in the ninth iteration, with an accuracy of 92.69%, a precision of 92.73%, a recall of 92.69%, and an F-measure of 92.64%. Detailed results of the K-Fold Cross Validation test using the Support Vector Machine algorithm can be seen in Figure 5 below.

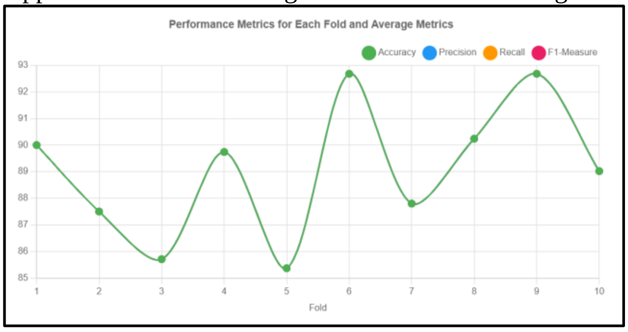

Figure 6. Support Vector Machine Cross Validation Graph

Next is the Decision Tree cross-validation test, as shown in Table 11.

Table 11. Decision Tree Cross Validation Testing

Fold	Accuracy	Precision	Recall	F1-Measure
1	84.62%	84.50%	84.62%	84.42%
2	83.08%	83.29%	83.08%	82.93%
3	85.00%	84.99%	85.00%	84.76%
4	85.00%	84.90%	85.00%	84.91%
5	81.92%	82.04%	81.92%	81.69%
6	83.46%	83.41%	83.46%	83.33%
7	85.77%	86.13%	85.77%	85.44%
8	83.46%	83.42%	83.46%	83.33%
9	88.85%	88.93%	88.85%	88.72%
10	86.87%	87.01%	86.87%	86.66%
Average	84.70%	84.86%	84.76%	84.52%

According to Table 11, the K-Fold Cross Validation findings for the Decision Tree method indicate that the optimal performance occurred in the ninth iteration, yielding an accuracy of 88.85%, precision of 88.93%,

.

recall of 88.85%, and F-measure of 88.72%. The comprehensive outcomes of the K-Fold Cross Validation assessment on the Decision Tree method are illustrated in Figure 7 below.

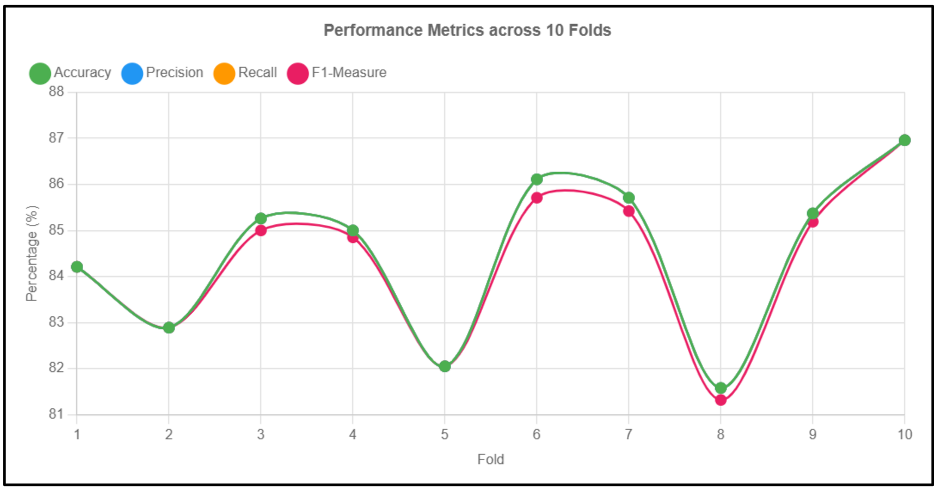

Figure 7. Cross Validation Decision Tree Graph

Next is the Logistic Regression cross-validation test, as shown in Table 12.

Table 12. Cross Validation Logistic Regression Test

Fold	Accuration	Precision	Recall	F1-Measure
1	89.62%	90.42%	89.62%	89.36%
2	86.92%	87.37%	86.92%	86.56%
3	85.77%	86.69%	85.77%	85.51%
4	87.31%	87.40%	87.31%	87.02%
5	88.85%	89.65%	88.85%	88.61%
6	82.69%	82.74%	82.69%	82.22%
7	89.62%	90.04%	89.62%	89.40%
8	87.69%	87.69%	87.69%	87.55%
9	89.58%	89.52%	89.58%	89.53%
10	89.19%	89.71%	89.19%	89.10%
Average	87.74%	88.28%	87.92%	87.58%

Based on Table 12, the results of the K-Fold Cross Validation test on the Logistic Regression algorithm show that the highest performance was achieved in the first iteration, with an accuracy of 89.62%, precision of 90.42%, recall of 89.62%, and F-measure of 89.36%. Detailed results of the K-Fold Cross Validation test on the Logistic Regression algorithm can be seen in Figure 8 below.

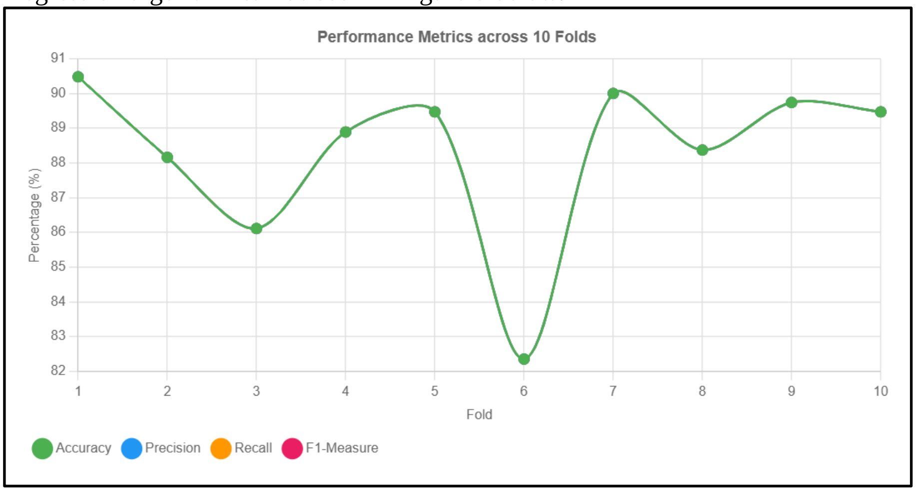

Figure 8. Cross Validation Logistic Regression Graph

3.5. Comparison of Support Vector Machine, Decision Tree, and Logistic Regression Testing

This stage involves comparing the results obtained from the three methods used in the study: Support Vector Machine, Decision Tree, and Logistic Regression. This comparison is based on the results of the final test using Cross Validation.

Table 13. Algorithm Comparison

Metrik Evaluasi	Support Vector Machine (SVM)	Decision Tree	Logistic Regression
Accuracy	88.18%	84.70%	87.74%
Precision	88.68%	84.86%	88.28%
Recall	88.68%	84.76%	87.92%
F1-Measure	88.51%	84.52%	87.58%

According to Table 11, the Support Vector Machine method exhibited superior test results in this study review paper relative to the Decision Tree and Logistic Regression techniques. The Support Vector Machine method attained an accuracy of 88.18%, precision of 88.68%, recall of 88.68%, and F-measure of 88.51%. The Decision Tree method attained an accuracy of 84.70%, precision of 84.86%, recall of 84.76%, and F-measure of 84.52%. The Logistic Regression technique attained an accuracy of 87.74%, precision of 88.28%, recall of 87.92%, and an F1-Measure of 87.58%. Figure 7 below presents a comprehensive comparison of the outcomes from the Support Vector Machine, Decision Tree, and Logistic Regression techniques.

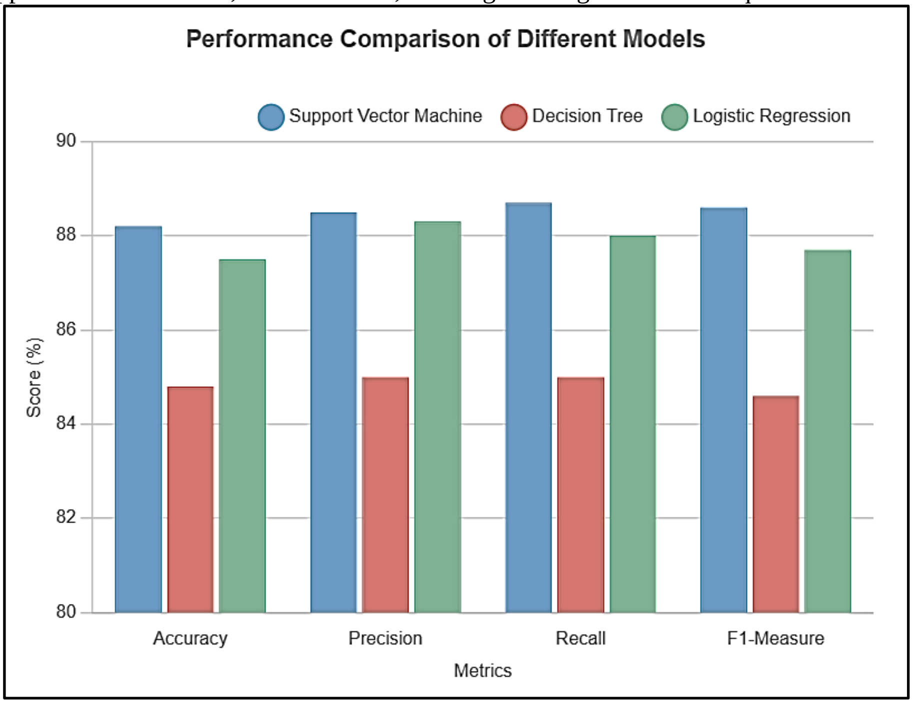

Figure 9. Algorithm Comparison Diagram

4. DISCUSSION

The aforementioned research findings indicate that the Support Vector Machine (SVM) regularly outperforms the Decision Tree in sentiment classification, with SVM attaining an average accuracy of 88.18% and a peak score of 92.69% in K-Fold Cross Validation. The Decision Tree demonstrated an average accuracy of 84.70%, while remaining effective. For a comprehensive examination of sentiment classification within the Threads application, refer to the following:

4.1. Distribution and Analysis of Words in Sentiment Classes

Sentiment classes are divided into two categories, positive and negative, in Threads app user reviews. There are a total of 1,749 positive reviews and 1,138 negative reviews.

.

4.1.1. Negative Sentiment

The negative sentiment data from Threads app user reviews totaled 1,138. Next, the word distribution was visualized using a Word Cloud. The results of this visualization are shown in Figure 10.

Figure 10. Negative Wordcloud

These words provide specific information about Threads user reviews. Several words appear frequently and indicate negative sentiment, such as "can," "pay," "me," "app," and "login." While these dominant words do not directly represent a negative class, it is important to pay attention to how these words are used in Threads user reviews. The following is an example of negative sentiment data containing these dominant words, where their use indicates a negative review:

- can't use, pay via funds, poor system always has an error
- Threads is getting worse, always having an error, I can log in, I'm really disappointed
- Why can't I install Threads? I've tried, but I still can't uninstall it. I've installed it and reinstalled it because of an error, but I can't uninstall it. Please help me.
- Error keeps logging in, always showing a try again screen, device is disturbing

Based on the examples of dominant words indicating negative sentiment in Threads user reviews, it can be concluded that negative sentiment is largely related to complaints about the Threads app's performance. These reviews include several features that are considered unsatisfactory and inconvenient for users.

4.1.2. Positive Sentiment

The total of positive sentiment data from user reviews of the Threads app amounted to 1,749. The subsequent step involved visualising word distribution with a Word Cloud. The outcomes of this visualisation are depicted in Figure 11.

Figure 11. Positive Wordcloud

This text offers detailed insights about Threads user evaluations. Certain terms commonly manifest and denote positive sentiment, including "good," "movie," "app," "very," and "the." The prevalent terms do not explicitly denote a positive category, hence it is crucial to examine instances of their application in Threads user evaluations. This is an illustration of positive sentiment data featuring predominant words that contribute to favourable reviews:

- I am capable of seeing quality films.

- Excellent selection. Numerous options available. Please update with further new films.
- exceptionally proficient

Analysis of prevalent terms reflecting good sentiment in Threads user reviews suggests that such sentiment is predominantly associated with the app's simplicity and user-friendliness. Users consider this application really beneficial for viewing films.

5. CONCLUSION

This study concluded that the Support Vector Machine (SVM) method exhibited greater performance in classifying the sentiment of Threads app reviews compared to Decision Tree and Logistic Regression, based on the test data. The Support Vector Machine (SVM) achieved an average accuracy of 88.18%, with a maximum performance of 92.69% during the ninth iteration of the K-Fold Cross Validation test. Simultaneously, the Logistic Regression technique attained an average accuracy of 87.74%, and the Decision Tree exhibited the lowest performance at 84.70%. Based on the sentiment analysis, Threads users exhibited a predominantly positive response, with 1,749 favourable reviews contrasted against 1,138 unfavourable reviews. This favourable emotion was primarily associated with user satisfaction over the app's usability and functionalities. In contrast, negative attitude was primarily characterised by grievances over technical problems, including login failures, application malfunctions, and installation challenges. This research yields strategic insights for developers to prioritise technological enhancements to improve future user experience.

REFERENCES

[1] O. Hondro, "Analisis Sentimen Pengguna Aplikasi E-Wallet Dana Melalui Postingan di Media Sosial Twitter Menggunakan Naïve Bayes," KETIK J. Inform., 2023, [Online]. Available: https://jurnal.faatuatua.com/index.php/KETIK/arti cle/view/18.

[2] K. P. J. Sitompul, A. R. Pratama, and K. A. Baihaqi, "Komparasi Algoritma Naive Bayes, Support Vector Machine, dan Logistic Regression Pada Analisis Sentimen Pengguna Aplikasi Transportasi Online," Kumpul. J. Ilmu Komput., vol. 10, no. 01, pp. 27–38, 2023.

[3] A. Zamsuri, E. Asril, and M. Sadar, "Analisis Sentimen Opini Terhadap Vaksin Covid-19 Pada Media Sosial Twitter," vol. 5, no. 1, pp. 100–110, 2015.

[4] M. I. Amal, E. S. Rahmasita, E. Suryaputra, and N. A. Rakhmawati, "Analisis Klasifikasi Sentimen Terhadap Isu Kebocoran Data Kartu Identitas Ponsel di Twitter," J. Tek. Inform. dan Sist. Inf., vol. 8, no. 3, 2022, doi: 10.28932/jutisi.v8i3.5483.

[5] F. A. Larasati, D. E. Ratnawati, and ..., "Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest," J. Pengemb. …, 2022, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/jptiik/article/view/11562.

[6] Sepriadi, N. N. (2023). *Analisis Sentimen Review Aplikasi Mypertamina Menggunakan Word Embedding Fasttext Dan Algoritma K-Nearest Neighbor* [Skripsi/Tugas Akhir, Universitas Islam Negeri Sultan Syarif Kasim Riau]. Repository UIN Suska. https://repository.uin-suska.ac.id/73902/

[7] Achmad, A. A., Iin, K., & Iska, Y. (2023). Analisis Klasifikasi Sentimen Berbasis Topik pada Ulasan Layanan Dana dan Sakuku dengan Convolutional Neural Network. *INFORMASI (Jurnal Informatika dan Sistem Informasi)*, *5*(2). http://ojs.stmik-im.ac.id/index.php/INFORMASI/article/view/267

[8] Oktavianus, & Hondro, M. (2023). Analisis Sentimen Pengguna Aplikasi E-Wallet Dana Melalui Postingan di Media Sosial Twitter Menggunakan Naïve Bayes. *Jurnal Informatika*, *1*(1), 27–31.

[9] Kusuma, I. N., & Ali, I. (2024). Analisis Sentimen Pada Pengguna Aplikasi Dana Menggunakan Algoritma Naive Bayes. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(1). https://ejournal.itn.ac.id/index.php/jati/article/view/9041

[10] Saputro, W. E., & Yuana, H. (2023). Analisis Sentimen Pengguna Dompet Digital Dana Pada Kolom Komentar Google Play Store Dengan Metode Klasifikasi Support Vector Machine. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(1). https://ejournal.itn.ac.id/index.php/jati/article/view/6842

[11] Saputra, A. (2023). *Analisis Sentimen Aplikasi Investasi Reksa Dana (Bibit) Menggunakan Metode Support Vector Machine* [Skripsi/Tugas Akhir, UPN "Veteran" Yogyakarta]. E-Prints UPNYK. http://eprints.upnyk.ac.id/

[12] Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

[13] Kiedrowsky, F. F., & Andrianingsih. (2023). Sentiment Analysis Marketplaces Digital menggunakan Machine Learning. *J. JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, *7*(3), 493–499. https://doi.org/10.35870/jtik.v7i3.1002

[14] Imamah, & Rachman, F. H. (2020). Twitter sentiment analysis of Covid-19 using term weighting TF-IDF and logistic regression. *Proceeding - 6th Information Technology International Seminar (ITIS) 2020*, 238–242. https://doi.org/10.1109/ITIS50118.2020.9320958

[15] Wulandari, H. D., & Rahman, I. A. (2022). Cryptocurrency Price Volatility Analysis On Bitcoin And Altcoins Before And During The Covid-19 Pandemic In Indonesia. *Journal of Applied Business, Taxation and Economics Research*, *1*(6), 612-624.

[16] F. M. Delta Maharani, A. Lia Hananto, S. Shofia Hilabi, F. Nur Apriani, A. Hananto, and B. Huda, “Comparison of Sentiment Classification Methods Analysis of E-Wallet Usage Using Naive Bayes and K-Nearest Neighbor Algorithms,” Metik J., vol. 6, no. 2, pp. 97–103, 2022, doi: 10.47002/metik.v6i2.372.