P-ISSN : 2715-2448 | E-ISSN : 2715-7199
Vol.7 No.1 January 2026
Buana Information Technology and Computer Sciences (BIT and CS)

Application of XGBoost Algorithm in Sentiment Classification of MOBA Game Reviews on Google Play Store

Daffa Yauzan Tusianto[1]*, Syahroni Wahyu Iriananda[2], Istiadi[3]
[1,2,3]Widya Gama University of Malang, Malang Jl. Taman Borobudur Indah No.35, Telp. 0341-492282, Kec.Lowokwaru, Kota Malang, Jawa Timur 65128
E-mail: daffayauzan@gmail.com[1]*, syahroni@widyagama.ac.id[2], istiadi@widyagama.ac.id[3]

Received: *2025-07-27* | ***Revised:*** *2026-01-21* | ***Accepted:*** *2026-01-29*

Abstract

The popularity of Multiplayer Online Battle Arena (MOBA) games on mobile platforms has resulted in a huge number of user reviews on Google Play Store, which contain very important feedback for developers. Analyzing these reviews manually is not efficient, so an accurate automatic sentiment analysis method is needed. This study aims to build and evaluate a sentiment classification model using the XGBoost algorithm for Indonesian language reviews from three popular MOBA games: Mobile Legends: Bang-Bang, Honor of Kings, and League of Legends: Wild Rift. The results show that the optimized XGBoost model achieves a high accuracy of 92.87% and a log loss value of 0.2364. Comparative analysis of optimization methods demonstrates that BayesSearchCV offers the best balance between effectiveness and efficiency, delivering performance comparable to GridSearchCV but with significantly shorter computation time.

Keywords: Sentiment Analysis, XGBoost, Hyperparameter Optimization, MOBA Game, Game Reviews.

I. Introduction

The Multiplayer Online Battle Arena (MOBA) game genre has become a highly popular global phenomenon, especially following its transition from PC platforms to more accessible mobile devices [1], [2]. This popularity, which spans various age groups [3], has generated millions of user reviews on platforms such as the Google Play Store. These reviews are a valuable source of feedback on player experiences, but their sheer volume makes manual analysis impractical.

To solve this challenge, sentiment analysis methods are used as a computational solution to process text data and automatically classify opinions into positive or negative sentiments [4]. Previous studies have applied machine learning algorithms such as Naive Bayes and Support Vector Machine to analyse sentiment in game reviews [5], [6], [7]. On the other hand, the XGBoost algorithm has demonstrated excellent performance on various complex classification tasks and was selected for this study due to its advantages in terms of speed, scalability, and ability to handle imbalanced data [8], [9]

Although various methods have been applied, there is still room for improvement, particularly by optimising more advanced algorithms. This study aims to develop a sentiment classification model using XGBoost, with a focus on applying various hyperparameter optimisation techniques. According to [10], proper hyperparameter tuning is crucial for balancing model complexity and enhancing XGBoost's generalisation capabilities. It is hoped that the resulting model can accurately classify sentiment in MOBA game reviews and provide deep insights that developers can utilise to improve game quality based on player feedback.

II. Method

This research was conducted through a series of systematic stages to achieve the research objectives, starting from literature review to result analysis. In general, the research workflow is illustrated in the following Figure 1, which will be explained further in each sub-chapter.

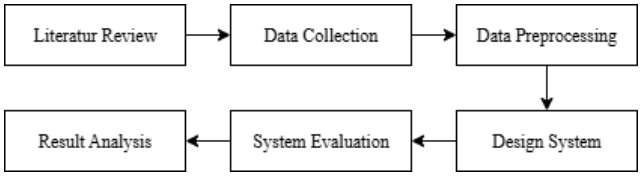

Figure 1. Result method

1. Literature Review

Literature review is a fundamental method for establishing a strong theoretical foundation in a research study [11]. Its primary purpose is to develop a conceptual framework and establish relevant research hypotheses [12]. In the context of this study, the literature review focuses on collecting references related to the XGBoost algorithm, various hyperparameter optimisation techniques, and their application in sentiment classification cases. The results of this literature review serve as a crucial foundation for designing a systematic research methodology and ensuring that the approach used aligns with the research problem being addressed.

2. Data Collection

In this study, the technique used was web scraping, a method for automatically extracting data from semi-structured web pages such as Google Play Store [13]. This technique was chosen due to its ability to efficiently collect review data on a large scale, which is an essential component for reliable sentiment analysis [14]. Thus, the use of web scraping ensures the acquisition of a representative and high-quality dataset as the basis for the next stages of the research.

3. Data preprocessing

The data preprocessing stage is a crucial step that aims to clean and transform raw text data into a structured format that is ready to be processed by machine learning models. This stage consists of a series of text preprocessing steps, starting from case folding to final cleaning.The ultimate goal of this entire stage is to ensure that the review data used by the XGBoost algorithm is clean and consistent, thereby significantly improving computational efficiency and the accuracy of sentiment classification results.

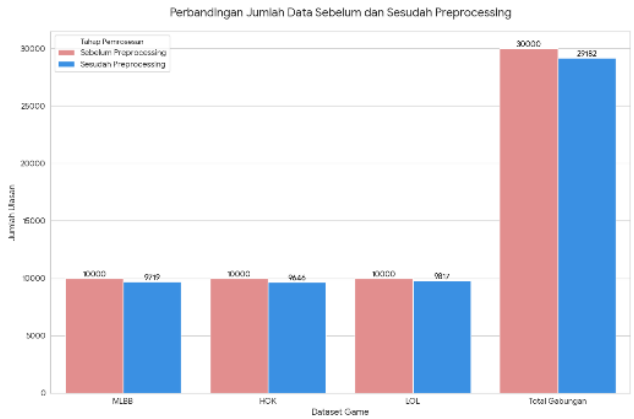

Figure 2. Comparison Total Datasets Before and After Text Preprocessing

As shown in the (Figure 2), there was a reduction in the number of reviews in each dataset. The initial dataset consisted of 30,000 reviews (10,000 reviews each games). After undergoing cleaning processes, the final dataset ready for modelling was reduced to 29.182 reviews. This reduction ensures that the data used to train the model is of high quality and relevant to the research objective, which is binary sentiment classification (positive and negative).

4. Design System

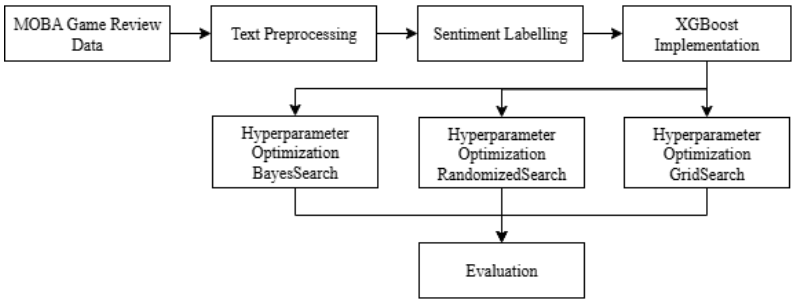

Figure 3. Design System

This system is designed to process MOBA game review data through a series of systematic stages. The process begins with text preprocessing to clean and standardise raw data, which is then translated into English before sentiment labelling. Sentiment labelling in this study uses VADER (Valence Aware Dictionary and sEntiment Reasoner) to classify each review as positive or negative.

Next, these labelled reviews are converted into numerical representations using the TF-IDF method, which weights each word based on its significance. These numerical representations are then used to train the XGBoost classification model through three hyperparameter tuning techniques: Grid Search, Randomised Search, and Bayesian Optimization, to find the best performance. The final stage is model evaluation using a series of standard metrics such as accuracy, precision, recall, F1-score, and log loss to measure its reliability and effectiveness.

5. Evaluation

Model evaluation was performed using a set of standard metrics to provide a comprehensive performance assessment. The metrics used included Accuracy to measure the overall correctness of predictions, as well as Precision, Recall, and F1-Score, which were calculated individually for each class (positive and negative). Precision assesses the accuracy of predictions for each class, while Recall measures the model's ability to find all actual sentiments from that class. The balance between Precision and Recall is then summarised by the F1-Score, while Log Loss is used to evaluate not only the accuracy of predictions but also the confidence level of the model, where lower values indicate better performance.

6. Result Analysis

The analysis of the results provides a basis for answering the research questions, such as the extent of the XGBoost model's performance in classifying positive and negative sentiments in MOBA game reviews on Google Play Store and the effect of the data split ratio on the final performance of the sentiment analysis model.

III. Results and Discussion

This study builds a sentiment classification model for MOBA game reviews using the XGBoost algorithm. The main focus is to systematically compare three hyperparameter optimisation methods

(GridSearchCV, RandomizedSearchCV, and BayesSearchCV) and two data splitting ratios to find the most effective and efficient model configuration. Below are some of the results obtained in this study.

1. Word Cloud Before and After Text Preprocessing

Figure 4. Word Cloud Before Text Preprocessing: MLBB, HOK dan LOL

In Figure 4, before text preprocessing, the word cloud results show a predominance of common, repetitive words such as "dan", "di", "saya", and "main" in all three games. These words do not provide relevant information about the players' experiences and are more related to the sentence structures commonly used in conversations. Additionally, although there are words indicating negative sentiment such as "lag", "bug", or "dark system", they are not particularly prominent because they are still mixed with irrelevant words.

Figure 5. Word Cloud After Text Preprocessing: MLBB, HOK Dan LOL

Meanwhile, in Figure 5 after text preprocessing, the data becomes more structured and focused. Irrelevant words such as conjunctions and common words are removed, while more meaningful words such as "bagus", "dark system", "sinyal", and "matchmaking"are more prominent. This preprocessing enables clearer identification of the technical issues faced by players, as well as clarifying more specific positive and negative sentiments related to the gaming experience. Overall, preprocessing successfully improves data quality and enables more in-depth analysis of aspects that need improvement in the game.

2. Distribution of Labelling Result Using VADER

Based on the Table 1 below, the sentiment labelling results using VADER on 29.182 clean reviews show a unique profile for each game. MLBB and HOK are dominated by positive reviews, while LOL is contrastingly dominated by negative reviews. This uneven distribution, especially in the LOL dataset, is a characteristic of the data and is one of the main challenges in the classification modelling process.

Table 1. Distribution Labelling Vader

Sentiment	Dataset			
	MLBB	HOK	LOL	Mix
Positive	5.781	6.271	4.325	16.377
Negative	3.938	3.375	5.492	12.805
Total	9719	9.646	9.817	29.182

3. Results and Evaluation

a. GridSearch Ratio 75:25

Based on the results summarised in Table 2, the GridSearchCV method at a ratio of 75:25 produced excellent performance with a computation time of 47 minutes. The best performance was achieved on the HOK dataset with an accuracy of 0.9287 and the lowest log loss of 0.2364, indicating a highly accurate and confident model. Conversely, the Mix dataset was the most challenging with an accuracy of 0.8877. The balanced F1-score metric across most datasets indicates solid performance, although the LOL dataset showed some difficulty with negative sentiment recall of 0.86. Despite the specific challenges of the LOL dataset, the model's overall performance remained strong and reliable across all domains tested.

Table 2. Evaluation Of The Results GridSearch With Ratio 75:25

No.	Dataset	Sentiment	Precision	Recall	F1-Score	Accuracy	Log Loss
1.	MLBB	Negative	0.91	0.89	0.90	0.9189	0.2531
		Positive	0.93	0.94	0.93		
2.	HOK	Negative	0.89	0.91	0.90	0.9287	0.2364
		Positive	0.95	0.94	0.94		
3.	LOL	Negative	0.89	0.92	0.91	0.8941	0.3260
		Positive	0.90	0.86	0.88		
4.	Mix	Negative	0.87	0.88	0.87	0.8877	0.3084
		Positive	0.90	0.90	0.90		

b. GridSearch Ratio 60:40

The GridSearchCV method at a 60:40 ratio required 39 minutes of computation time. Table 3 shows that reducing the training data at a 60:40 ratio caused a slight decrease in overall performance, but the model still showed strong performance. The HOK dataset once again emerged as the best performer with an accuracy of 0.9202. The Mix dataset recorded the lowest accuracy 0.8792. In the LOL dataset, the challenge was evident in the lower positive sentiment recall 0.85, indicating the model's difficulty in identifying all positive reviews with less training data.

Table 3. Evaluation Of The Results GridSearch With Ratio 60:40

No.	Dataset	Sentiment	Precision	Recall	F1-Score	Accuracy	Log Loss
1.	MLBB	Negative	0.88	0.89	0.89	0.9072	0.2598
		Positive	0.92	0.92	0.92		
2.	HOK	Negative	0.88	0.89	0.89	0.9202	0.2509
		Positive	0.94	0.93	0.94		
3.	LOL	Negative	0.88	0.91	0.90	0.8824	0.3411
		Positive	0.88	0.85	0.86		
4.	Mix	Negative	0.86	0.87	0.86	0.8792	0.3151
		Positive	0.90	0.89	0.89		

c. RandomizedSearch Ratio 75:25

Referring to Table 4, the RandomizedSearchCV method provides competitive results with highly efficient computation time of 12 minutes. The highest performance was achieved on the HOK dataset with an accuracy of 0.9138. Although its accuracy is slightly below that of GridSearchCV, this method offers an excellent trade off between speed and quality. The LOL dataset was the most challenging for this model, with the lowest accuracy of 0.8664 and the highest log loss of 0.3658, primarily due to a significant drop in positive sentiment recall to 0.82.

Table 4. Evaluation Of The Results RandomizedSearch With Ratio 75:25

No.	Dataset	Sentiment	Precision	Recall	F1-Score	Accuracy	Log Loss
1.	MLBB	Negative	0.88	0.89	0.89	0.9082	0.2820
		Positive	0.92	0.92	0.92		
2.	HOK	Negative	0.87	0.88	0.88	0.9138	0.2676
		Positive	0.94	0.93	0.93		
3.	LOL	Negative	0.86	0.90	0.88	0.8664	0.3658
		Positive	0.87	0.82	0.84		
4.	Mix	Negative	0.85	0.88	0.87	0.8816	0.3266
		Positive	0.90	0.88	0.89		

d. RandomizedSearch Ratio 60:40

As shown in Table 5, this method was the fastest, taking only 9 minutes. Even under the most limited data conditions, the model still delivers solid performance. HOK remains the dataset with the best performance, achieving an accuracy of 0.9088, while LOL once again proves to be the most challenging, with an accuracy of 0.8658. These results demonstrate that Randomised Search CV is highly effective for quickly obtaining a good baseline model, despite a slight trade-off in accuracy.

Table 5. Evaluation Of The Results Randomizedsearch With Ratio 60:40

No.	Dataset	Sentiment	Precision	Recall	F1-Score	Accuracy	Log Loss
1.	MLBB	Negative	0.88	0.88	0.88	0.9033	0.2891

		Positive	0.92	0.92	0.92		
2.	HOK	Negative	0.86	0.88	0.87	0.9088	0.2789
		Positive	0.93	0.92	0.93		
3.	LOL	Negative	0.87	0.90	0.88	0.8658	0.3734
		Positive	0.86	0.83	0.84		
4.	Mix	Negative	0.85	0.86	0.86	0.8753	0.3340
		Positive	0.89	0.89	0.89		

e. BayesSearch Ratio 75:25

The results in Table 6 show that BayesSearchCV emerged as the most balanced method, with a computation time of 21 minutes. This method achieved accuracy nearly matching GridSearchCV, with peak performance on the HOK dataset yielding an accuracy of 0.9233 and the lowest log loss of 0.2395. These results demonstrate that BayesSearch's intelligent search is capable of identifying highly optimal parameters with significantly better time efficiency than exhaustive search.

Table 6. Evaluation Of The Results Bayessearch With Ratio 75:25

No.	Dataset	Sentiment	Precision	Recall	F1-Score	Accuracy	Log Loss
1.	MLBB	Negative	0.90	0.90	0.90	0.9210	0.2502
		Positive	0.93	0.93	0.93		
2.	HOK	Negative	0.89	0.89	0.89	0.9233	0.2395
		Positive	0.94	0.94	0.94		
3.	LOL	Negative	0.89	0.92	0.90	0.8908	0.3254
		Positive	0.89	0.86	0.87		
4.	Mix	Negative	0.87	0.88	0.87	0.8872	0.3057
		Positive	0.90	0.89	0.90		

f. BayesSearch Ratio 60:40

The BayesSearch method with a 60:40 ratio requires 16 minutes of computation time. Based on the summary in Table 7, the results from BayesSearchCV are almost identical to those from GridSearchCV. The HOK dataset once again leads with an accuracy of 0.9202. This phenomenon indicates that under limited data conditions, both intelligent search methods and exhaustive search methods tend to converge on the same optimal solution. This further solidifies BayesSearch's position as the most efficient method, as it can achieve peak performance in significantly less time.

Table 7 Evaluation Of The Results Bayessearch With Ratio 60:40

No.	Dataset	Sentiment	Precision	Recall	F1-Score	Accuracy	Log Loss
1.	MLBB	Negative	0.88	0.89	0.89	0.9072	0.2598
		Positive	0.92	0.92	0.92		
2.	HOK	Negative	0.88	0.89	0.89	0.9202	0.2509
		Positive	0.94	0.93	0.94		
3.	LOL	Negative	0.88	0.91	0.90	0.8824	0.3411

4.	Mix	Positive	0.88	0.85	0.86	0.8792	0.3151
		Negative	0.86	0.87	0.86		
		Positive	0.90	0.89	0.89		

4. Comparative Visualisation

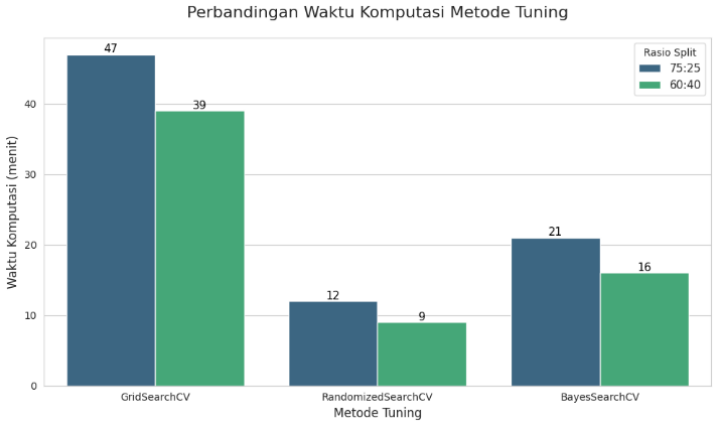

Figure 6. Comparison Of Computation Time

Figure 6 highlights the significant differences in computation efficiency between optimisation methods. GridSearchCV is the slowest method (up to 47 minutes) due to its exhaustive nature. In contrast, RandomisedSearchCV and BayesSearchCV, whose searches are limited to 20 iterations, demonstrate much higher efficiency. With this configuration, RandomizedSearchCV is the fastest (9–12 minutes), while BayesSearchCV (16–21 minutes) positions itself as an efficient middle ground, demonstrating that limited and intelligent searches can drastically reduce computational time.

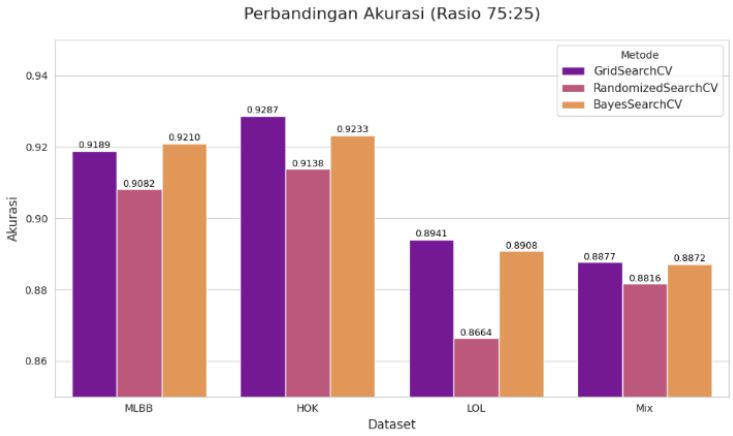

Figure 7. Comparison Of Accuracy Results For 75:25 Ratio

At a ratio of 75:25 (Figure 7), GridSearchCV and BayesSearchCV consistently provided the highest and nearly identical accuracy, proving that Bayes intelligent search is capable of matching the quality of comprehensive search. BayesSearchCV even slightly outperformed on the MLBB dataset. Conversely, RandomisedSearchCV ranks lowest, though with an insignificant margin, reinforcing its role as a robust baseline method.

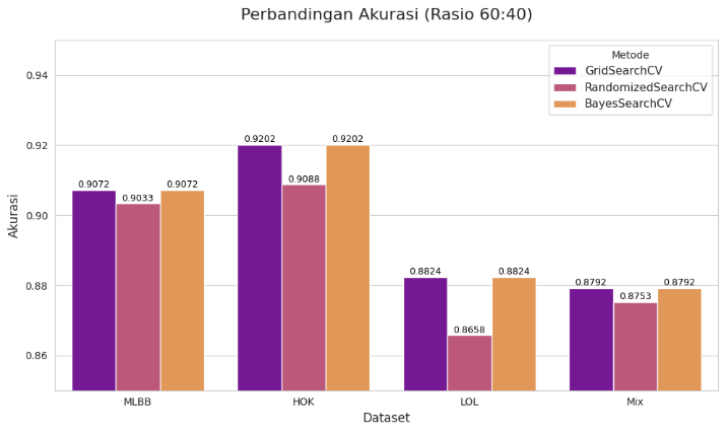

Figure 8. Comparison Of Accuracy Results For 75:25 Ratio

At a ratio of 60:40 (Figure 8), the accuracy performance of GridSearchCV and BayesSearchCV becomes nearly identical. This indicates that both methods (exhaustive search and intelligent search) converge on the same optimal solution when the training data is more limited. As expected, all methods experience a slight decrease in accuracy compared to the 75:25 ratio, but the performance hierarchy across datasets remains consistent, with HOK being the best and LOL the most challenging.

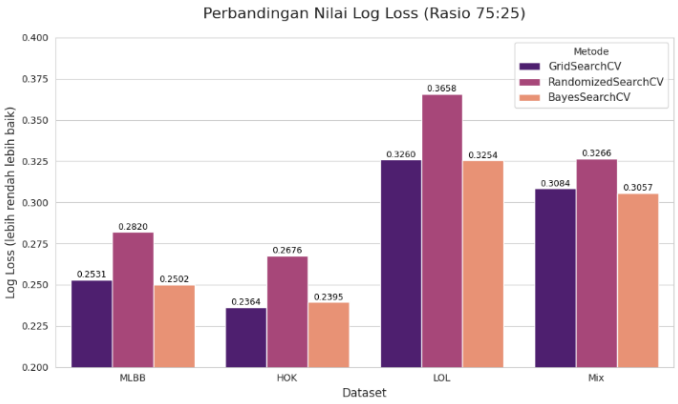

Figure 9. Comparison Of Log Loss Values 75:25 Ratio

The log loss visualization for the 75:25 ratio (Figure 9) confirms the findings from the accuracy analysis. GridSearchCV and BayesSearchCV consistently produce the lowest log loss values, indicating that the models they produce are more accurate and more confident in their predictions. Conversely, RandomisedSearchCV has the highest log loss, indicating a higher level of model uncertainty.

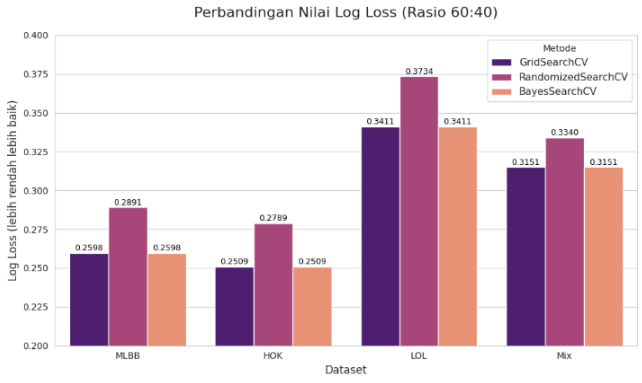

Figure 10. Comparison Of Log Loss Values 60:40 Ratio

At a ratio of 60:40 (Figure 10), the log loss values for GridSearchCV and BayesSearchCV are again nearly identical, reinforcing the convergence findings under limited data conditions. In general, the log loss increases slightly for all methods due to the smaller training data set. The most visually challenging scenario is confirmed on the LOL dataset optimised with RandomisedSearchCV, which shows the highest log loss value.

IV. Conclusion

XGBoost effective for classifying MOBA game review sentiment, consistently achieving accuracy between 86.58% and 92.87%. Among the optimisation methods tested, BayesSearchCV proved to be the most superior strategy overall, as it was able to match the high performance of GridSearchCV with much better time efficiency. Model performance was also found to be highly influenced by data characteristics, with the HOK dataset consistently yielding the highest accuracy while the LOL dataset proved to be the most challenging. Additionally, this study confirmed that larger training data volumes, such as a 75:25 split ratio, generally result in more accurate and robust models.

V. Acknowledgment

I would like to express my sincere and heartfelt gratitude to Widyagama University Malang for the opportunity, facilities, and supportive academic environment provided to me to pursue my education and complete this research. I would also like to thank all the lecturers of the Informatics Study Programme for the valuable knowledge and insights they have shared with me throughout my studies. In particular, I would like to extend my deepest gratitude to Mr. Syahroni Wahyu Iriananda and Mr. Istiadi. Thank you for your invaluable time, effort, and patience in providing guidance, direction, motivation, and constructive criticism, which enabled this research to be successfully completed.

References

[1] D. Sinaga and C. Jatmoko, *Analisis Sentimen Untuk Mengetahui Kesan Player Game Mobile Legends Menggunakan Naïve Bayes Classifier*. 2020. [Online]. Available: www.netlytic.org

[2] V. Fazrian, T. Suprapti, and R. Narasati, "Penerapan Algoritma Naive Bayes Terhadap Analisis Sentimen Aplikasi Game Multiplayer Online Battle Arena (Studi Kasus: Mobile Legend)," 2024.

[3] R. W. Abie and S. Rosmilawati, "Perilaku Toxic Dalam Komunikasi Virtual Di Game Online Mobile Legends: Bang Bang Pada Mahasiswa Fakultas Ilmu Muhammadiyah Palangkaraya University," *Restorica: Jurnal Ilmiah Ilmu Administrasi Negara dan Ilmu Komunikasi*, vol. 9, pp. 44–48, 2023, doi: 10.33084/restorica.v9i1.

[4] D. Ikasari and Widiastuti, "Sentiment Analysis Review Novel 'Goodreads' Berbahasa Indonesia Menggunakan Naïve Bayes Classifier," Jan. 2021.

[5] A. Okta, K. Adi, F. Prayoganing Gusti, and F. Wijaya, "Analisis Sentimen Ulasan Pengguna Aplikasi Mobile Legends Pada Google Playstore Menggunakan Naïve Bayes," 2025.

[6] A. F. Panjalu, S. Alam, and M. I. Sulistyo, "MOBA GAME REVIEW SENTIMENT ANALYSIS USING SUPPORT VECTOR MACHINE ALGORITHM," *JIKO (Jurnal Informatika dan Komputer)*, vol. 6, no. 2, Aug. 2023, doi: 10.33387/jiko.v6i2.6388.

[7] S. W. Iriananda, R. W. Budiawan, A. Y. Rahman, and I. Istiadi, "Optimasi Klasifikasi Sentimen Komentar Pengguna Game Bergerak Menggunakan Svm, Grid Search Dan Kombinasi N-Gram," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 4, pp. 743–752, Aug. 2024, doi: 10.25126/jtiik.1148244.

[8] J. M. A. S. Dachi and P. Sitompul, "Analisis Perbandingan Algoritma XGBoost dan Algoritma Random Forest Ensemble Learning pada Klasifikasi Keputusan Kredit," *Jurnal Riset Rumpun Matematika Dan Ilmu Pengetahuan Alam (Jurrimipa)*, vol. 2, no. 2, pp. 87–103, Oct. 2023.

[9] R. G. Gunawan, Erik Suanda Handika, and Edi Ismanto, "Pendekatan Machine Learning Dengan Menggunakan Algoritma Xgboost (Extreme Gradient Boosting) Untuk Peningkatan Kinerja Klasifikasi Serangan Syn," *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 3, no. 3, pp. 453–463, Dec. 2022, doi: 10.37859/coscitech.v3i3.4356.

[10] M. Zlobin and V. Bazylevych, "Bayesian Optimization For Tuning Hyperparametrs Of Machine Learning Models: A Performance Analysis In Xgboost," *Computer systems and information technologies*, no. 1, pp. 141–146, Mar. 2025, doi: 10.31891/csit-2025-1-16.

[11] P. Fajar and Y. I. Aviani, "Hubungan Self-Efficacy dengan Penyesuaian Diri: Sebuah Studi Literatur," vol. 6, no. 1, pp. 2186–2194, 2022.

[12] A. Akbar *et al.*, "Pelatihan Dan Pengembangan Sdm Dalam Perspektif Ilmu Manajemen: Sebuah Studi Literatur," 2023.

[13] U. Mufidah and M. Siahaan, "Perancangan Aplikasi Perbanndingan Harga Produk (Historical Data) Menggunakan Teknik Scraping Web," 2021.

[14] M. Rizqi, A. Rustiawan, and P. T. Prasetyaningrum, "Analisis Sentimen Terhadap Klinik Natasha Skincare di Yogyakarta Dengan Metode Google Review," *Journal of Information Technology Ampera*, vol. 5, no. 1, pp. 2774–2121, 2024, doi: 10.51519/journalita.v5i1.556.