Journal of Informatic Engineering (JOUTICA)
https://jurnalteknik.unisla.ac.id/index.php/informatika

An optimized kernel SVM framework for game review sentiment analysis using particle swarm optimization

Ichwanul Muslim Karo Karo[1*], Sri Dewi[2], Alvi Sahrin Nasution[3]

[1]Computer Science, Faculty of Mathematics and Natural Science, Medan State of University, Medan, 20351, Indonesia
ichwanul@unimed.ac.id

[2]Computer Science, Faculty of Mathematics and Natural Science, Medan State of University, Medan, 20351, Indonesia
sridewi@unimed.ac.id

[3]Mathematics, Faculty of Mathematics and Natural Science, Medan State of University, Medan, 20351, Indonesia
alvisahrin@unimed.ac.id

*Corresponding Author

Abstrak:
Pertumbuhan pesat industri game digital telah meningkatkan interaksi antara pengembang dan pemain melalui platform ulasan online. Ulasan-ulasan ini mengandung informasi penting tentang pengalaman bermain game dan kepuasan, yang berfungsi sebagai pengembangan versi game berikutnya. Analisis sentimen menyediakan pendekatan strategis untuk mengklasifikasikan ulasan secara otomatis, memberikan wawasan berbasis data bagi pengembang. Studi ini berfokus pada peningkatan kinerja analisis sentimen untuk ulasan Player Unknown's Battlegrounds (PUBG) dengan mengintegrasikan Kernel Support Vector Machine (SVM) dengan Particle Swarm Optimization (PSO). Sebuah dataset berisi 1.205 ulasan dari Google Play Store dianalisis menggunakan ekstraksi fitur TF-IDF dan validasi silang 5-fold. Meskipun SVM default Optimization (PSO). Sebuah dataset berisi 1.205 ulasan dari Google Play Store dianalisis menggunakan ekstraksi fitur TF-IDF dan validasi silang 5-fold. Meskipun SVM Kernel default mencapai akurasi 76,78%, akurasinya rendah (56,9%). Implementasi PSO untuk optimasi parameter secara signifikan meningkatkan kinerja, mencapai akurasi 86,42%, presisi 70,98%, dan skor F1 sebesar 83,03%. F1-score. Perbandingan dengan Naïve Bayes, SVM dasar, BERT, dan Lexicon + SVM menunjukkan bahwa model Kernel SVM + PSO memberikan kinerja yang lebih unggul dan stabil. Temuan ini menyoroti efektivitas PSO dalam penyesuaian parameter SVM. Penelitian masa depan sebaiknya menyelidiki kombinasi optimasi metaheuristik dengan model deep learning untuk meningkatkan generalisasi model.

Kata Kunci:
SVM; PSO; PUBG; Kernel; Analisis sentimen.

Abstract:
The digital game industry's rapid growth has increased interaction between developers and players through online review platforms. These reviews contain vital information about gaming experiences and satisfaction, serving as guides for future game versioning. Sentiment analysis provides a strategic approach to automatically classify reviews, offering data-driven insights for developers. This study

---

*Corresponding author: ichwanul@unimed.ac.id

focuses on enhancing sentiment analysis performance for Player Unknown's Battlegrounds (PUBG) reviews by integrating Kernel Support Vector Machine (SVM) with Particle Swarm Optimization (PSO). A dataset of 1,205 reviews from the Google Play Store was analyzed using TF-IDF feature extraction and 5-fold cross-validation. While default Kernel SVM achieved 76.78% accuracy, it suffered from low precision (56.9%). Implementing PSO for parameter optimization significantly improved performance, reaching 86.42% accuracy, 70.98% precision, and an 83.03% F1-score. Comparisons with Naïve Bayes, basic SVM, BERT, and Lexicon + SVM confirm that the Kernel SVM + PSO model provides superior and more stable performance. These findings highlight PSO's effectiveness in SVM parameter tuning. Future research should investigate combining metaheuristic optimization with deep learning models to improve model generalization.

Keywords:
SVM; PSO; PUBG; Kernel; Sentiment analysis.

1. Introduction

The digital gaming industry is growing rapidly and triggering increased interaction between developers and players through various online review platforms. Player Unknown's Battlegrounds (PUBG) is one of the most popular games and is currently ranked in the Top 10 Online Multiplayer Games [1]. Gamer reviews often contain valuable information regarding gameplay experience, satisfaction, and criticism. All of this can be used as insight for game versioning. Sentiment analysis of these reviews is an important approach for automatically identifying positive, negative, and neutral reviews from users. This allows developers to maintain game features that receive positive reviews and improve features that receive negative reviews.

Support Vector Machine (SVM) is one of the machine learning algorithms with reliable performance for sentiment analysis [2]. The algorithm is robust in handling high-dimensional data from text [3]. The SVM performance is greatly influenced by the selection of parameters and kernel functions [4], [5]. Many studies have confirmed that the right kernel can improve the performance of the SVM algorithm [4], [6], [7], [8], [9]. For example, the RBF kernel function has the best performance with an accuracy of 98.8% for classify opinion about product in E-commerce platforms [4], analyzing sentiment on X in the AFC U23 Asian Cup [10], and 84% accurate in analyzing public opinion on online learning policies [6]. The cosine kernel function contributes positively to analyzing public opinion regarding the National Sport Weeks in X [5]. The Gaussian kernel function achieved an accuracy of 95% for analyzing public opinion trends [11]. It applies to other kernel functions.

However, determining the optimal kernel parameters in SVM is not a simple process [4]. There are parameters such as cost (C), gamma (γ), and the kernel function significantly influence the model's performance. Inappropriate parameter selection can lead to overfitting or underfitting, causing classification performance to decline significantly. The search for optimal parameters is typically conducted using conventional methods such as grid search or cross-validation. Nevertheless, these approaches often require high computational time, especially when dealing with large-scale and high-dimensional datasets. To address these limitations, one applicable solution is to utilize metaheuristic optimization techniques, such as Particle Swarm Optimization (PSO).

PSO is a metaheuristic optimization algorithm inspired by the collective behavior of bird flocks and fish schools in searching for food sources [2], [12]. Each particle in PSO represents a candidate solution that moves dynamically within the search space by leveraging both individual and collective best experiences. The integration of PSO with SVM, particularly in the process of adjusting kernel parameters. It has shown great potential for enhancing sentiment analysis accuracy [2].

This study focuses on improving the performance of Kernel SVM through parameter optimization using PSO in the case of PUBG game review sentiment analysis. The findings of this research are expected to contribute to the development of a more effective sentiment analysis system for the gaming industry, while expanding the application of the SVM and PSO combination in the domain of Natural Language Processing (NLP).

2. Research Method

Figure 1 illustrates the research flowchart, which begins with scraping reviews from the Playstore. Subsequently, the reviews undergo a text preprocessing stage to clean and normalize the data until it is ready for use. The feature extraction process is then performed to transform the text data into numerical representations. Cross validation is employed to ensure the reliability of the model. The next stage involves the application of Kernel-based SVM, both directly and through parameter optimization using PSO. The results from both approaches are then evaluated to draw final conclusions.

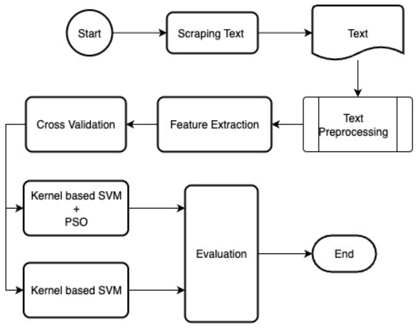


Figure 1 : Flowchart research method

2.1. Scraping Review

The scraping process aims to extract reviews of PUBG Mobile on the Playstore platform. Concurrent with this process, each review is labeled as a positive, negative, or neutral class. Reviews with a rating of 4–5 are categorized as positive, those with a rating of 3 are neutral, while reviews with a rating of 1–2 are categorized as negative. Example of PUBG review shown in Table 1.

Tabel 1: Example of data text

Full Text	Rating	Class
bug mengendarai mamooth tiger turun titik pergi ratusan meter titik	2	Negative
update bug jelek bug visual lobby start match tulisan startmatchnya terbawa sampe match menutupi layar mark step tembakan map bug tombol tembakan dimana terkadang face to face musuh tombil fire block berfungsi bermain pubg bug muncul	1	Negative
tolong perbaikan game kadang gk keliatan tau penghalang	2	Negative
suka event skrng	4	Positive
seru keren the best dah tolong tambahin game battle royale hujannya biar pengalaman bermain bagus	5	Positive

2.2. Text Preprocessing

Text preprocessing is a crucial process in NLP that cleans and transforms raw text into a structured and machine-readable format [13]. It removes noise like punctuation, HTML, and stop words, and normalizing text (e.g., lowercasing, stemming) to improve the accuracy and efficiency of models for tasks like sentiment analysis. This study utilizes five text preprocessing methods: text cleaning, tokenization, case folding, normalization, stopword removal, and stemming. Text cleaning is the process of removing unnecessary symbols such as numbers, punctuation, or special characters [14]. This method also removes duplicate data. Tokenizing is the process of breaking review sentences into word-by-word [15], and case folding standardizes to lower case in order to consistent. Normalization is the process of correcting non-standard words into standard words in text or sentences [16]. Stopword removal deletes common words that do not provide a significant impact on content understanding, e.g., punctuation or conjunctions [17]. Stemming is the process of changing words into their base form by removing affixes [18], [19]. This stage aims to reduce the unique words that must be accommodated by the model, thereby improving model performance.

2.3. Feature Extraction

Feature extraction aims to transform text into numerical form [7], [14]. Thus, it can be analyzed using a machine learning approach. The idea is to convert each word into tokens that have gone through the preprocessing stage into vectors that will represent the existing words [13]. Term Frequency – Inverse Document Frequency (TF - IDF) is one of the feature extraction methods with the approach of converting words into vectors [7], [14]. This approach is the product of TF and IDF, as shown in Equation (1).

$$\mathrm{TF}-\mathrm{IDF}(t,d) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t) \tag{1}$$

Term Frequency (TF) is the frequency of a word that appears in a document [14]. In other words, it measures how often a word appears in a particular document, the formula of TF is shown in Equation (2).

$$\mathrm{TF}(t,d) = \frac{\text{Occurrences of word } t \text{ in document } d}{\text{Total number of words in document } d} \tag{2}$$

Inverse Document Frequency (IDF) measures how important a word is in a document [14]. It gives a higher weight to words that frequently appear across documents, thus making them more meaningful. The formula of IDF is shown in Equation (3).

$$\mathrm{IDF}(t) = \frac{\log N}{1 + \text{Number of documents containing } t} \tag{3}$$

2.4. Support Vector Machine + PSO

SVM algorithm classifies data by constructing an optimal hyperplane with a maximum margin, enabling effective separation between classes [20]. To handle nonlinearly separable data, SVM employs kernel functions, such as Radial Basis Function (RBF), Polynomial, and Sigmoid to project input data into higher-dimensional feature spaces [4], [6], [7], [8], [21], [22]. The performance of SVM is highly dependent on appropriate parameter selection, particularly the penalty parameter (*C*), which controls the trade-off between margin maximization and classification error, and kernel parameters such as gamma (γ) in the RBF kernel. Improper parameter settings may result in overfitting or underfitting, thereby reducing classification accuracy.

To overcome this limitation, PSO is integrated with SVM to optimize kernel parameters, as illustrated in Figure 2. In this approach, PSO searches for the optimal combination of (C, γ) by iteratively evaluating candidate solutions using cross-validation accuracy as the fitness function [2]. Each particle updates its position in the search space based on individual best (*pbest*) and global best (*gbest*) solutions, enabling efficient exploration of the parameter space. In this framework, PSO functions as a global optimization mechanism, while SVM serves as the primary sentiment classification model.

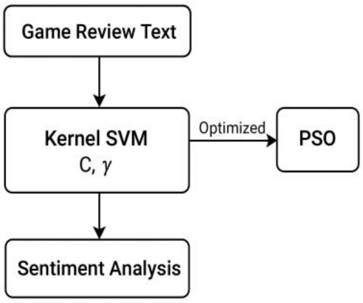


Figure 2 : Integration kernel SVM +PSO

The optimization process begins with the random initialization of particles in the parameter space, followed by iterative training and evaluation of the Kernel SVM model. Particle velocities and positions are updated using inertia, cognitive, and social components to balance exploration and exploitation. The process continues until a termination criterion is met, such as convergence of fitness values or a maximum number of iterations. Compared to conventional search methods such as grid search, PSO demonstrates superior capability in identifying optimal kernel parameter combinations in high-dimensional search spaces, resulting in more stable and accurate sentiment classification performance.

Pseudocode SVM + PSO

```
Input:
    Dataset $D = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$
    Kernel function $K(\cdot)$
    Number of particles $P$
    Maximum iterations $T$
    Search space for parameters $C$ and $\gamma$
    Inertia weight $w$
    Acceleration coefficients $c_1$, $c_2$
Begin
    Initialize particles $i$ = 1 to $P$
    Randomly initialize position $X_i = (C_i, \gamma_i)$
    Randomly initialize velocity $V_i$
    Set personal best $pbest\_i = X_i$

    Evaluate fitness of each particle
    Train Kernel SVM using parameters $(Ci, \gamma i)$
    Compute accuracy using cross-validation
    Set $fitness\_i$ = accuracy

    Set global best $gbest$ as particle with highest fitness

    For iteration $t$ = 1 to $T$ do
    For each particle $i$ = 1 to P do
    Update velocity:
    $V_i = w * V_i + c_1 * rand() * (pbest\_i - X_i) + c_2 * rand() * (gbest - X_i)$
    Update position:
    $X_i = X_i + V_i$
    Apply boundary constraints on $C_i$ and $\gamma_i$
    Train Kernel SVM using updated $(C_i, \gamma_i)$
    Evaluate $fitness\_i$ using cross-validation

    If $fitness\_i$ > fitness($pbest\_i$) then
```

```
            pbest_i = $X_i$
        End if

        If fitness_i > fitness(gbest) then
            gbest = $X_i$
        End if
        End for
        End for
        Set C_best, γ_best = gbest
        Train final Kernel SVM using optimized parameters
        Return trained model and optimal parameters
End
```

2.5. Model Evaluation

The evaluation process of the sentiment analysis model uses a confusion matrix [23]. Table 2 shows the confusion matrix for a three-class dataset. The confusion matrix is used to calculate accuracy, precision, recall, and F-1 Score of the sentiment analysis results. The accuracy metric is used to measure how accurately the sentiment classification model predicts sentiment labels compared to the actual labels (shown in Equation (4)). Precision is used to measure how many positive predictions are actually positive (shown in Equation (5)). Recall is used to measure the model's ability to find all positive data that actually exists (shown in Equation (6)). The F1-score is the harmonic mean between precision and recall (shown in Equation (7)), this metric is used to balance both values

Table 2 : Confusion matrix

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

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (4)$$

$$Precision\ (P) = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (5)$$

$$Recall\ (R) = \frac{TP}{TP + FN} \times 100\% \quad (6)$$

$$F1\ Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (7)$$

3. Result and Discuss

3.1. Dataset

The number of PUBG game reviews that were successfully scraped is 1205 reviews. The results of scraping and text preprocessing of the reviews are visualized with a word cloud in Figure 3. The size of a word indicates its frequency of occurrence; the larger the word size, the more often the word appears in the review text, and vice versa.

Figure 3 : Word cloud dataset

Based on the results of rating–label conversion for 1,205 reviews (shown in Figure 4), it was found that 54% of PUBG mobile user reviews fall into the neutral category, 35% are in the positive category, while the rest are in the negative category. These results indicate that user perception of the game is relatively good, although there are still a number of criticisms that need to be a concern for the developers. Based on the data in Figure 4, the dataset is class imbalanced. This study has not yet implemented any treatment to address this issue, and it will be addressed in future work.

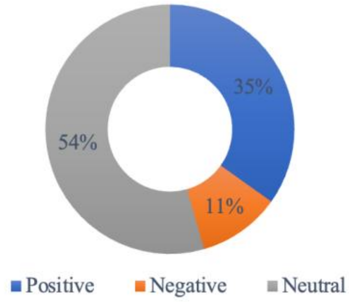


Figure 4 : Sentiment label distribution of dataset

3.2. Cross Validation

This study employs a random K-Fold Cross Validation approach with K =5. Prior to the validation process, the dataset is randomly shuffled and then divided into five folds of approximately equal size for each label. In each iteration, one fold is used as the validation set, while the remaining four folds are used as the training set. In other words, approximately 80% of the dataset is used to train the model, and the remaining 20% is used for testing. This process is repeated for five iterations (as illustrated in Figure 5), so that each fold serves once as the validation set. The final performance of the model is obtained by averaging the evaluation results across all folds, thereby providing a more stable estimate and reducing bias caused by a single data split..

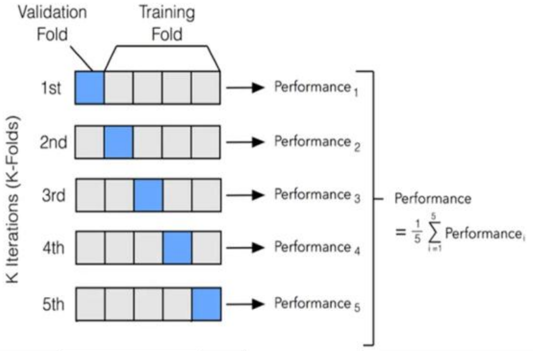


Figure 5 : *5*-Fold Cross Validation

3.3. Kernel SVM Performance

This section presents the results of the 5-fold cross-validation testing of the Kernel SVM model in analyzing PUBG game review sentiment. The default parameters are *C* = 1 and γ = 0.1. The results are presented in Table 3. It can be seen that the results tend to fluctuate in each iteration. Based on the information in the table, the resulting model performance is an accuracy of 78.76%, a precision of 56.9%, a recall of 100%, and an F1 score of 75.73%. Based on Table 3, the best performance was obtained with 2-fold. Table 4 presents the details of the confusion matrix conditions for the 2-fold, and Table 5 presents the model performance for each class.

Table 3 : Kernel SVM performance for k-fold

***K*-iteration**	**Accuracy**	***P***	***R***	**F1**
1st	75.12	55.1	100	71.05
2nd	86.30	60.07	100	75.05
3rd	82.05	58.15	100	73.54
4th	79.46	54.16	100	70.26
5th	70.88	57.08	100	72.68

Table 4 : Confusion Matrix Kernel SVM performance for 2-fold

Actual	***Predicted***		
	Positive	Negative	Neutral
Positive	84	0	0
Negative	11	4	12
Neutral	8	2	120

Table 5 : Performance Kernel SVM performance of 2-fold for each class

	Accuracy	***P***	***R***	***F1***
Negative	0.15	0.51	1	0.68
Neutral	0.92	0.91	1	0.95
Positive	1	0.57	1	0.72

3.4. Optimized Parameters

This section presents the performance of the Kernel SVM model with PSO optimization. Figure 6 shows the convergence process of Particle Swarm Optimization (PSO) in determining the value of the *C* parameter in the Kernel SVM. As the number of iterations increases, the value of C increases gradually until it reaches stability at *C* = 32 starting from the 9th iteration, with an accuracy of 86.4%.

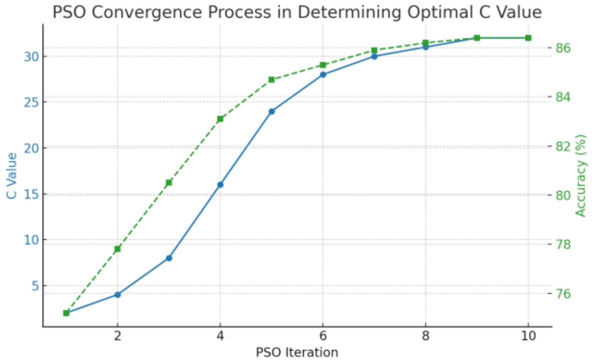


Figure 6 : Convergence process of PSO in optimizing the *C* parameter

The γ parameter is also optimized using PSO. Initially, the γ value is relatively large at 0.5 with an accuracy of 72.4%. However, as the iterations progress, the gamma value gradually decreases until it reaches 0.01 at the 9th iteration, with the accuracy reaching 86.4%. The optimal parameters *C* and gamma are obtained at the 9th iteration with the same accuracy.

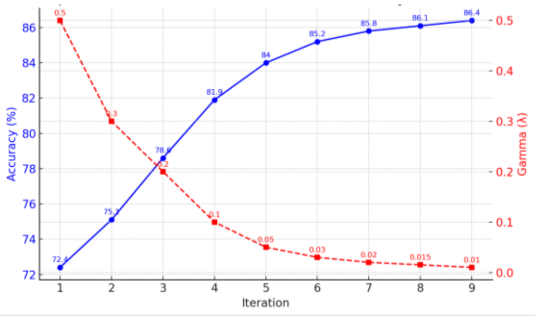


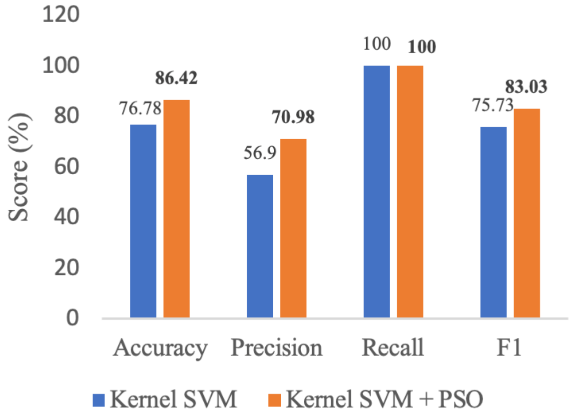
Figure 7 : Convergence process of PSO in optimizing γ parameter and accuracy

The results of parameter optimization by PSO show a significant difference compared to the previous experiment (shown in Table 6). In the Kernel SVM model, the parameters were set to *C* = *1 and* γ = 0.1.

Table 6 : Parameters comparison

Parameter	Kernel SVM (Default)	Kernel SVM + PSO
C	1	32
γ	0.1	0.01
Number of Particle	30	30
Iteration Maximum	50	50
Inertia Weigh	0.7	0.7
C1	2	2
C2	2	2

3.5. Kernel SVM + PSO Performance



3.

Figure 8 : Comparison of kernel SVM and PSO-Optimized kernel SVM performance

The experimental results show that the integration of PSO on Kernel SVM is able to improve the model performance (Figure 8). The model accuracy increases from 76.78% to 86.42% and the precision from 56.9% to 70.98%. Although the recall value remains at the maximum level (100%), the application of PSO results in a significant improvement in the F1-score, from 75.73% to 83.03%. This improvement indicates that kernel parameter optimization with PSO is able to produce a more balanced and effective classification model in managing the trade-off between precision and recall. This trend shows that PSO is able to find more optimal kernel parameters through exploration of the search space, thereby resulting in improved performance of the SVM model in sentiment analysis.

3.6. Comparison with Previous Studies

There are several previous studies that have also developed sentiment analysis of PUBG game reviews [24], [25], [26]. Table 7 provides a brief summary of previous studies and the results of this study. The Naïve Bayes and basic SVM algorithms produce relatively low accuracy, namely 69.83% and 70.95%, respectively, with balanced precision, recall, and F1 values but still limited. BERT, although able to achieve an accuracy of 83.71%, shows significant weaknesses in precision (44.44%), recall (23.53%), and F1 (30.77%), making its performance unstable in sentiment classification. The Lexicon + SVM approach provides better results with an accuracy of 82.49% as well as precision, recall, and F1 around 80%, thus being more consistent.

Table 7 : Performance comparison

Model	Accuracy	*P*	*R*	*F1*
Naïve Bayes [26]	69.83	70.3	69.83	68.82
SVM [26]	70.95	70.81	70.95	70.80
BERT [25]	83.71	44.44	23.53	30.77
Lexicon+ SVM [24]	82.49	**81.39**	80.47	80.42
Kernel SVM	76.78	56.9	**100**	75.73
Kernel SVM + PSO	**86.42**	70.98	**100**	**83.03**

Meanwhile, the results of this study show that Kernel SVM without optimization achieves an accuracy of 76.78% with perfect recall (100%), but low precision (56.9%), indicating that the model tends to produce excessive positive predictions (over-prediction). After being optimized using PSO, the performance increases significantly: accuracy reaches 86.42%, precision increases to 70.98%, recall remains stable at 100%, and the F1-score increases to 83.03%. This proves that the integration of PSO into Kernel SVM is able to improve the balance between accuracy, precision, and F1, while maintaining the maximum recall level.

4. Conclusions

This study shows that the use of Kernel SVM integrated with PSO optimization is able to provide a significant improvement in performance in the sentiment analysis of PUBG game reviews. From the data perspective, the study uses 1,205 player reviews obtained through scraping from Google Playstore, which are then processed through preprocessing and TF-IDF–based feature extraction to ensure the quality of text representation. From the methodological perspective, the *K*-Fold Cross Validation approach (K = 5) is used to produce reliable model evaluation, while PSO acts as an optimal parameter search mechanism (C and γ) for the Kernel SVM. The experimental results show that Kernel SVM without optimization only achieves an accuracy of 76.78% with relatively low precision (56.9%), although the recall remains perfect (100%). After being optimized with PSO, the performance increases significantly with an accuracy of 86.42%, precision of 70.98%, recall remaining at 100%, and an F1-score of 83.03%. When compared with previous studies (Naïve Bayes, basic SVM, BERT, and Lexicon + SVM), this approach is proven to be more consistent, balanced, and superior in maintaining the trade-off between accuracy, precision, recall, and F1-score. The implications of these findings confirm that PSO can be an effective optimization strategy to strengthen SVM models in the domain of text-based sentiment analysis. For future research, it is recommended to integrate deep learning–based models (such as BERT or LSTM) that are also optimized with metaheuristics, as well as to explore multi-domain and multilingual datasets in order to test the generalization of the model on a broader scale.

References

[1] R. Narwal and H. Aggarwal, "Predicting Online Game-Addicted Behaviour with Sentiment Analysis Using Twitter Data," 2022, pp. 505–517. doi: 10.1007/978-981-16-2354-7_45.

[2] S. Fahmi, L. Purnamawati, G. F. Shidik, M. Muljono, and A. Z. Fanani, "Sentiment Analysis of Student Review in Learning Management System Based on Sastrawi Stemmer and SVM-PSO," in *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, IEEE, Sep. 2020, pp. 643–648. doi: 10.1109/iSemantic50169.2020.9234291.

[3] R. Kashef, "A boosted SVM classifier trained by incremental learning and decremental unlearning approach," *Expert Syst. Appl.*, vol. 167, p. 114154, Apr. 2021, doi: 10.1016/j.eswa.2020.114154.

[4] A. P. Gopi, R. N. S. Jyothi, V. L. Narayana, and K. S. Sandeep, "Classification of tweets data based on polarity using improved RBF kernel of SVM," *International Journal of Information Technology*, vol. 15, no. 2, pp. 965–980, Feb. 2023, doi: 10.1007/s41870-019-00409-4.

[5] A. Mansyur *et al.*, " Impact of Cosine Similarity Function on SVM Algorithm for Public Opinion Mining About National Sports Week 2024 on X," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 11, no. 3, pp. 263–275, May 2025.

[6] Husni, Arif Muntasa, and Mochamad Dani Hartanto, "Classification of Public Opinion on Online Learning Policies using Various Support Vector Machine's Kernel," *Technium: Romanian Journal of Applied Sciences and Technology*, vol. 17, pp. 427–434, Nov. 2023, doi: 10.47577/technium.v17i.10119.

[7] P. H. Prastyo, I. Ardiyanto, and R. Hidayat, "Indonesian Sentiment Analysis: An Experimental Study of Four Kernel Functions on SVM Algorithm with TF-IDF," in *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, IEEE, Oct. 2020, pp. 1–6. doi: 10.1109/ICDABI51230.2020.9325685.

[8] N. Mohd Hatta, Z. Ali Shah, and S. Kasim, "Evaluate the Performance of SVM Kernel Functions for Multiclass Cancer Classification," *International Journal of Data Science*, vol. 1, no. 1, pp. 37–41, May 2020, doi: 10.18517/ijods.1.1.37-41.2020.

[9] Z. Gao, R. Xia, and P. Zhang, "Prediction of Anti-proliferation Effect of [1,2,3]Triazolo[4,5-<i>d</i>]pyrimidine Derivatives by Random Forest and Mix-Kernel Function SVM with PSO," *Chem. Pharm. Bull. (Tokyo).*, vol. 70, no. 10, pp. c22-00376, Oct. 2022, doi: 10.1248/cpb.c22-00376.

[10] H. Prasetya, Z. Situmorang, and R. Rosnelly, "SVM Optimization with Kernel Function for Sentiment Analysis on Social Media twitter (X) in AFC U23 Asian Cup Case Study," *Proceeding of International Conference on Science and Technology UISU*, pp. 227–233, Dec. 2024, doi: 10.30743/wjxmmr59.

[11] P. S. Ramesh, S. Surendran, S. Kerthy, R. B, U. G, and S. K. Jain, "Analysis of Text Emotional Tendency and Prediction of Internet Public Opinion Trend Based on Support Vector Machine Algorithm," in *2023 Second International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, IEEE, Aug. 2023, pp. 399–404. doi: 10.1109/SmartTechCon57526.2023.10391708.

[12] A. H. Ali and M. Z. Abdullah, "An efficient model for data classification based on SVM grid parameter optimization and PSO feature weight selection," *International Journal of Integrated Engineering*, vol. 12, no. 1, pp. 1–12, 2020, doi: 10.30880/ijie.2020.12.01.001.

[13] I. M. Karo Karo, M. F. Md Fudzee, S. Kasim, and A. A. Ramli, "Karonese Sentiment Analysis: A New Dataset and Preliminary Result," *JOIV: International Journal on Informatics Visualization*, vol. 6, no. 2–2, p. 523, Aug. 2022, doi: 10.30630/joiv.6.2-2.1119.

[14] I. M. K. Karo, A. Perdana, and S. Dewi, "Automatic Text Review Summarization of Digital Library System Application using TextRank Algorithm and TF-IDF," in *2024 4th International Conference of Science and Information Technology in Smart Administration (ICSINTESA)*, IEEE, Jul. 2024, pp. 570–575. doi: 10.1109/ICSINTESA62455.2024.10747952.

[15] A. Toleu, G. Tolegen, and A. Makazhanov, "Character-based Deep Learning Models for Token and Sentence Segmentation," 2017, *Kazan, Tatarstan*.

[16] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Appl. Soft Comput.*, vol. 97, p. 105524, Dec. 2020, doi: 10.1016/j.asoc.2019.105524.

[17] Arif Bijaksana Putra Negara, "The Influence Of Applying Stopword Removal And Smote On Indonesian Sentiment Classification," *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, vol. 14, no. 03, pp. 172–185, Oct. 2025, doi: 10.24843/LKJITI.2023.v14.i03.p05.

[18] I. M. Karo Karo, M. F. Md Fudzee, S. Kasim, and A. A. Ramli, "Sentiment Analysis in Karonese Tweet using Machine Learning," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 10, no. 1, Mar. 2022, doi: 10.52549/ijeei.v10i1.3565.

[19] A. P. Wibawa, F. A. Dwiyanto, I. A. E. Zaeni, R. K. Nurrohman, and A. Afandi, "Stemming javanese affix words using nazief and adriani modifications," *Jurnal Informatika*, vol. 14, no. 1, p. 36, Jan. 2020, doi: 10.26555/jifo.v14i1.a17106.

[20] P. Kiran Kumar, N. Jahna Tejaswi, M. L. Vasanthi, L. L. Srihitha, and B. Phanindra Kumar, "Sentimental Analysis on Multi-domain Sentiment Dataset Using SVM and Naive Bayes Algorithm," 2022, pp. 201–213. doi: 10.1007/978-3-030-95502-1_16.

[21] C. B. Pande *et al.*, "Comparative Assessment of Improved SVM Method under Different Kernel Functions for Predicting Multi-scale Drought Index," *Water Resources Management*, vol. 37, no. 3, pp. 1367–1399, Feb. 2023, doi: 10.1007/s11269-023-03440-0.

[22] Y. Li, J. Lou, X. Tan, Y. Xu, J. Zhang, and Z. Jing, "Adaptive Kernel Learning Kalman Filtering With Application to Model-Free Maneuvering Target Tracking," *IEEE Access*, vol. 10, pp. 78088–78101, 2022, doi: 10.1109/ACCESS.2022.3193101.

[23] D. Valero-Carreras, J. Alcaraz, and M. Landete, "Comparing two SVM models through different metrics based on the confusion matrix," *Comput. Oper. Res.*, vol. 152, p. 106131, Apr. 2023, doi: 10.1016/j.cor.2022.106131.

[24] M. A. P. Putri, N. Y. Setiawan, and B. T. Hanggara, "Analisis Sentimen Ulasan Pengguna Aplikasi PUBG Mobile Menggunakan Karakteristik Quality in Use: Effectiveness and Satisfaction," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 9, no. 10, pp. 2548–964, Oct. 2025, [Online]. Available: http://j-ptiik.ub.ac.id

[25] Handrizal, F. Y. Manik, and H. A. Misbah, "Sentiment Analysis Based On Pubgm Player Aspects From App Store Reviews Using Bidirectional Encoder Representation From Transformer (BERT)," *J. Theor. Appl. Inf. Technol.*, vol. 102, 2024.

[26] P. R. Sari, D. R. Indah, E. Rasywir, M. A. Firdaus, and G. Athalina, "Comparison of Naive Bayes and SVM Algorithms for Sentiment Analysis of PUBG Mobile on Google Play Store," *SISTEMASI*, vol. 13, no. 6, p. 2767, Nov. 2024, doi: 10.32520/stmsi.v13i6.4814.