p-ISSN: 2406-7768
e-ISSN: 2581-2181

WORD EMBEDDING ANALYSIS IN SENTIMENT ANALYSIS USING MACHINE LEARNING: A CASE STUDY OF STEAM RPG GAME REVIEWS

Ardian Adam Alfarisyi[1], Mahendra Dwifebri Purbolaksono[2], Alfian Akbar Gozali[3]

[1,2]Faculty of Informatics Telkom University
[3]Faculty of Applied Science Telkom University Jl. Telekomunikasi – Terusan Buah Batu Bandung
[1]ardianadm@student.telkomuniversity.ac.id
[2]mahendradp@telkomuniversity.ac.id [3]alfian@telkomuniversity.ac.id

Abstract

User reviews on gaming platforms such as Steam have become a crucial source of information for potential players before making purchasing decisions. Due to the varied nature of user opinions, sentiment analysis is essential for processing and interpreting these reviews. This study investigates the application of sentiment analysis to RPG game reviews on Steam, aiming to assist users by summarizing reviews through sentiment results and providing insights into the general perception of a game. To achieve this, the study applies sentiment analysis using Word2Vec and Support Vector Machine (SVM). It focuses on evaluating the impact of lemmatization during preprocessing and analyzing the performance of Word2Vec in sentiment classification. Word2Vec transforms review text into vector representations that capture semantic relationships, enhancing the model's ability to understand context. Meanwhile, SVM is chosen as the classifier for its effectiveness in distinguishing between positive and negative reviews and handling high-dimensional data. The system developed uses Word2Vec with 300-dimensional vectors combined with an SVM Polynomial classifier, resulting in the best performance among the tested models. The final model achieves a macro-average F1-score of 88.6%, indicating a strong capability in accurately classifying sentiments in user reviews. These results highlight the potential of combining word embedding and machine learning techniques for analyzing sentiment in gaming platforms.

Keywords: *sentiment analysis, Word2Vec, SVM, Steam, RPG*

I. INTRODUCTION

The growth of the digital gaming industry in recent years has significantly increased the use of distribution platforms such as Steam. Developed by Valve Corporation, Steam is one of the largest digital game distribution services [1] [2]. It includes features such as review dashboards that help users evaluate games through ratings and feedback [3]. More than just a marketplace, Steam provides an interactive ecosystem that allows users to publish reviews based on their gameplay experiences. These user-generated reviews play a crucial role in influencing the decisions of prospective players. In the case of Role-Playing Games (RPGs), which are known for their complex narratives and immersive gameplay, reviews tend to be more detailed and opinion-rich, often reflecting a highly personalized gaming experience.

However, these reviews are typically written in unstructured text formats, making them difficult to process objectively and efficiently—both for new users seeking quick insights and for developers analyzing player sentiment. It is commonly used to analyze user feedback on social media and measure satisfaction [4]. The variety of expressions, vocabulary, and tone found in such reviews presents a significant challenge for manual analysis, highlighting the need for automated sentiment classification methods that can interpret textual content accurately and contextually. Review analysis supports understanding user needs, identifying issues, and improving applications [5]

A potential approach involves performing sentiment analysis by integrating word embedding techniques like Word2Vec with machine learning classifiers, particularly Support Vector Machines (SVM). Word2Vec transforms words into vector representations that capture semantic meaning based on context [6] [7], while SVM has proven effective in classifying data with high dimensionality, including sentiment polarity [8] [9]. Together, these methods provide a robust framework for building sentiment analysis systems without the need for manual feature engineering.

Despite their popularity in domains like e-commerce and social media, studies focusing on the application of Word2Vec and SVM for analyzing sentiment in game reviews—especially

Received: July13rd, 2025 | Published: September7th, 2025

p-ISSN: 2406-7768
e-ISSN: 2581-2181

within the RPG genre on Steam—remain scarce. Furthermore, there has been limited investigation into how preprocessing techniques such as lemmatization and different SVM kernel types (Linear, RBF, and Polynomial) affect classification accuracy in this domain. This lack of targeted research reveals an important gap that this study seeks to address.

This research aims to explore the impact of lemmatization in the preprocessing stage, compare the performance of different Word2Vec vector dimensions (100 and 300), and evaluate the effectiveness of various SVM kernels in classifying sentiments of RPG game reviews on Steam. Using a dataset of 10,000 user-labeled reviews, this study is designed to identify the most effective configurations for domain-specific sentiment classification.

Over time, the outcomes of this research could support the advancement of smarter systems for summarizing reviews, building recommendation engines, and tracking user sentiments within the gaming sector. Additionally, the methodology proposed here may serve as a foundation for future research on sentiment analysis in other domains involving complex and unstructured textual data.

II. Literature Review

Research [10] raised the topic of sentiment analysis on public policy using Twitter data and applied a feature expansion approach with Word2Vec to overcome vocabulary mismatches in short text content. The study aimed to improve sentiment classification performance by comparing feature expansion using Word2Vec across three different corpora. Both Support Vector Machine (SVM) and Logistic Regression were employed for classification, where SVM obtained the best accuracy at 78.99%. This research highlights the advantage of using feature expansion combined with Word2Vec in enhancing the model's performance for classifying Indonesian-language tweets related to public policies.

In research [11], a sentiment analysis framework combining Word2Vec and SVM was proposed for analyzing Amazon product reviews. The study utilized the Amazon Fine Food Reviews dataset, focused on preprocessing, and semantic representation through Word2Vec. The results indicated the Word2Vec and SVM combination performed with an accuracy of 83.45%. This study demonstrated that integrating semantic features through Word2Vec enhances the detection of complex sentiment patterns in e-commerce review data. However, the research lacks detailed reporting on the specific number of samples used in each experimental configuration and does not include a comparative analysis across multiple datasets, which may limit the generalizability of the results.

Research [12] analyzed public sentiment toward the "Kampus Merdeka" policy using Twitter data collected between January 2020 and February 2023. The study used a lexicon-based method for initial sentiment labeling and employed the Support Vector Machine (SVM) algorithm for classification. The model was evaluated with the highest accuracy of 91.18%. Moreover, the study revealed that the majority of Twitter users expressed positive rather than negative sentiments toward the "Kampus Merdeka" program, indicating that the public generally responded favorably to it.

III. Methods

A. System Design

This research focuses on analyzing the sentiment of reviews related to an RPG game available on Steam. An outline of the system developed for this purpose is illustrated in Figure 1.

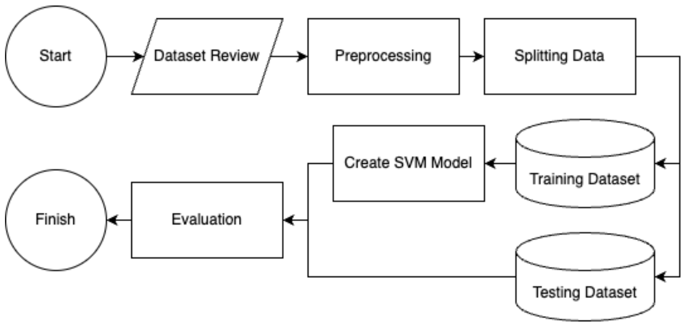

Fig. 1. System Design Flowchart

B. Dataset

This study uses user reviews of role-playing games (RPG) collected from the Steam platform via web scraping. Each review was independently labeled by three human annotators as 1 (positive) or 0 (negative), then validated using a pretrained model to ensure consistency and reliability. Table I shows a sample of the dataset prepared for further analysis.

TABLE I
DATASET

Label	Sentence
1	Best game I ever played my whole life. The entire experience of playing this game is outstanding. The graphics are just clean and incredible, the gameplay feels really fluid and fun, every enemy in the game is perfectly designed for the series. This game is going to impress you from begin to the end.
0	Bad game, not hard, not fun.

The data gathered consists of reviews or comments on the application, with a total of 10,000 data points collected as shown in Figure 2 below.

C. Preprocessing

The preprocessing stage is conducted to address issues encountered during data processing. Data preprocessing has a vital impact on the performance of traditional machine learning [13]. The steps involved in this stage include Cleansing, Case Folding, Stopword Removal, Lemmatization, and Tokenization. Figure 3 below illustrates the flowchart of the preprocessing steps applied.

1) Cleansing: The initial phase, known as Cleansing, serves as a preprocessing step designed to remove unnecessary

Received: July13rd, 2025 | Published: September7th, 2025

p-ISSN: 2406-7768
e-ISSN: 2581-2181

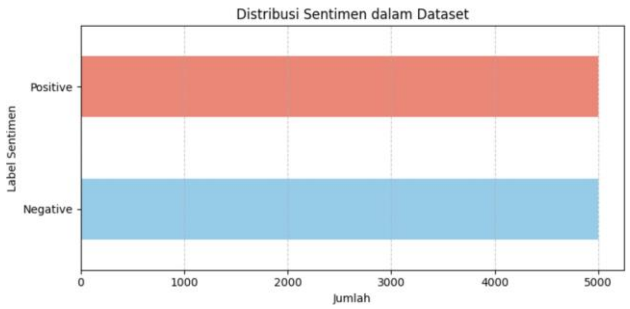

Fig. 2. Total Dataset

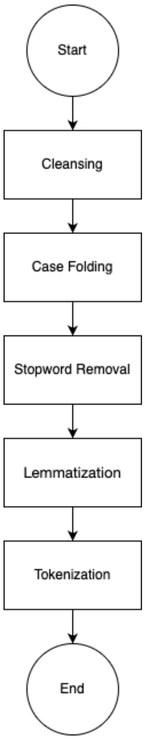

Fig. 3. Preprocessing Flowchart

TABLE II
CLEANSING

Sentence Before Cleansing	Sentence After Cleansing
From a fan of DSR, this game is fun but not in a playable state due to a garbage port to PC. Unstable-Crashing more frequently as I progress. Unfinished- Some spells have broken stats where a ””0”” was left off of a value. Poorly Optimized- Performance. Server Issues- ””connection error”” when trying to play with others as of 19 March. DO NOT PLAY WITH TWO MONITORS.	From a fan of DSR this game is fun but not in a playable state due to a garbage port to PC Unstable Crashing more frequently as I progress Unfinished Some spells have broken stats where a was left off of a value Poorly Optimized Performance Server Issues connection error when trying to play with others as of March DO NOT PLAY WITH TWO MONITORS

terms and punctuation. Table II illustrates an example of this procedure.

2) Case Folding: The second stage, referred to as Case Folding, involves converting every uppercase character in a word into its lowercase form. An illustration of this process is provided in Table III.

TABLE III
CASE FOLDING

Sentence Before Case Folding	Sentence After Case Folding
From a fan of DSR this game is fun but not in a playable state due to a garbage port to PC Unstable Crashing more frequently as I progress Unfinished Some spells have broken stats where a was left off of a value Poorly Optimized Performance Server Issues connection error when trying to play with others as of March DO NOT PLAY WITH TWO MONITORS	from a fan of dsr this game is fun but not in a playable state due to a garbage port to pc unstable crashing more frequently as i progress unfinished some spells have broken stats where a was left off of a value poorly optimized performance server issues connection error when trying to play with others as of march do not play with two monitors

3) Stopword Removal: The third step, Stopword Removal, involves eliminating less meaningful words that have minimal impact on the classification process. An illustration of this process is presented in Table IV.

TABLE IV
STOPWORD REMOVAL

Sentence Before Stopword Removal	Sentence After Stopword Removal
from a fan of dsr this game is fun but not in a playable state due to a garbage port to pc unstable crashing more frequently as i progress unfinished some spells have broken stats where a was left off of a value poorly optimized performance server issues connection error when trying to play with others as of march do not play with two monitors	fan dsr game fun playable state due garbage port pc unfinished spells broken stats left value poorly optimized performance server issues connection error trying play others march play two monitors

4) Lemmatization: The next step is Lemmatization, which refers to the process of mapping various word forms to their base form or lemma [14], taking into account both context and part of speech. This ensures that the resulting word is grammatically correct and semantically appropriate within the sentence. Table V illustrates an example of its implementation.

TABLE V
LEMMATIZATION

Sentence Before Lemmatization	Sentence After Lemmatization
fan dsr game fun playable state due garbage port pc unstable crashing frequently progress unfinished spells broken stats left value poorly optimized performance server issues connection error trying play others march play two monitors	fan dsr game fun playable state due garbage port pc unstable crash frequently progress unfinished spell break stats leave value poorly optimize performance server issue connection error try play others march play two monitor

5) Tokenization: The last stage, Tokenization, is the process of dividing a sentence into separate words. An example of this step is shown in Table VI.

Received: July13rd, 2025 | Published: September7th, 2025

p-ISSN: 2406-7768
e-ISSN: 2581-2181

TABLE VI
TOKENIZATION

Sentence Before Tokenization	Sentence After Tokenization
fan dsr game fun playable state due garbage port pc unstable crashing frequently progress unfinished spells broken stats left value poorly optimized performance server issues connection error trying play others march play two monitors	'fan', 'dsr', 'game', 'fun', 'playable', 'state', 'due', 'garbage', 'port', 'pc', 'unstable', 'crash', 'frequently', 'progress', 'unfinished', 'spell', 'break', 'stats', 'leave', 'value', 'poorly', 'optimize', 'performance', 'server', 'issue', 'connection', 'error', 'try', 'play', 'others', 'march', 'play', 'two', 'monitor'

D. Data Splitting

Following preprocessing, the dataset is divided into two parts: one designated for training the model and the other for assessing its performance. In this study, 9,000 out of 10,000 game reviews (90%) are used for training and as the corpus for Word2Vec, while the remaining 1,000 reviews (10%) are reserved for testing the classification models.

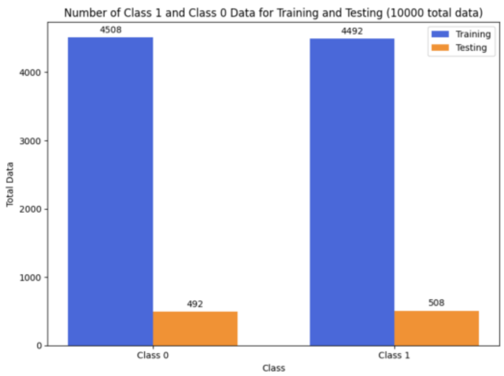

Fig. 4. Training and Testing Dataset

Figure 4 presents the distribution of Class 0 and Class 1 samples across the training and testing datasets. From a total of 10,000 instances, 4,508 instances from Class 0 and 4,492 instances from Class 1 were allocated to the training set, while the remaining 492 and 508 instances from Class 0 and Class 1, respectively, were assigned to the testing set. The relatively balanced class distribution in both subsets ensures that the model can be trained and evaluated without class bias, thereby contributing to more robust and generalizable performance.

E. Word2Vec

Word2vec is well known and widely used in learning word embedding from raw text introduced by Mikolov et al. [15] to transform words in a text into vector representations. The process works by using a text corpus as input and producing vector representations for each word within it.

$$P(w_{t+j}|w_t) = \frac{e^{x_t \cdot x_{t+j}}}{\sum_{j'=-c, j' \neq 0} e^{x_t \cdot x_{t+j'}}} \quad (1)$$

This formula represents the probability of a context word w_{t+j} appearing given a target word w_t in the Skip-Gram model of Word2Vec. It uses the softmax function over the dot product between the vector representations of the target and context words. A higher dot product indicates a stronger semantic similarity, leading to a higher probability. The denominator normalizes this value by summing over all words within the context window (excluding the target itself), ensuring the output is a valid probability distribution. The primary objective is to develop word embeddings that reflect the semantic connections among words in the text.

F. Support Vector Machine

Support Vector Machine (SVM) is a supervised learning technique that can be applied to both classification and regression problems. Compared to many other classification methods, SVM offers a more structured and efficient approach, especially in modeling classification problems. SVM is a discriminant technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter [16]. This margin-based optimization makes SVM particularly effective in handling high-dimensional data. Figure 5 below illustrates a visual representation of the SVM model used in this study.

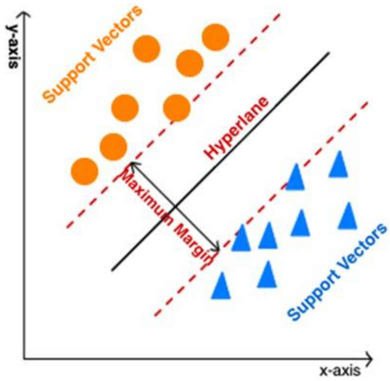

Fig. 5. VM Model Representation [17]

SVM offers three types of kernel functions that can be utilized, namely Polynomial, RBF (Radial Basis Function), and Linear kernels.

This study employs the Support Vector Machine (SVM) to establish the most effective decision boundary for classifying reviews as either positive or negative. Unlike many other classification approaches, SVM is recognized for its precision and efficiency in forming decision boundaries. Choosing the appropriate kernel is essential for model performance, as its effectiveness may differ depending on the dataset's characteristics.

G. Confusion Matrix

Model performance is analyzed using a confusion matrix, a common statistical technique for classification assessment. Confusion matrix is a statistical table used to summarize the

Received: July13rd, 2025 | Published: September7th, 2025

p-ISSN: 2406-7768
e-ISSN: 2581-2181

predictions of the applied model [18]. The confusion matrix is composed of four key components that represent prediction results: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These metrics help in understanding how well the model distinguishes between the sentiment classes, and are essential for calculating precision, recall, and other performance indicators. Table VII shows the visualization of the confusion matrix.

TABLE VII
CONFUSION MATRIX

	Positive	Negative
Positive	TP	FP
Negative	FN	TN

1) Recall:

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

In performance evaluation, recall highlights the sensitivity of a model toward positive data. It is quantified by comparing the number of true positives detected to the total positives that actually exist in the dataset.

2) Precision:

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

As an evaluation indicator, precision reflects the correctness of positive classifications by comparing correctly predicted positives with all instances labeled positive by the model.

3) F1-Score:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

To achieve a comprehensive measure, the F1-score integrates both recall and precision via their harmonic mean. This metric is often applied to deliver a balanced evaluation of classification models, particularly in contexts where one class significantly outweighs the other.

IV. RESULTS AND DISCUSSIONS

This study investigates three experimental scenarios to determine the most effective way of handling preprocessing and classification. In the initial scenario analyzes the impact of lemmatization by comparing results obtained with and without its application. The second scenario assesses differences in performance between Word2Vec models trained with 100 and 300 dimensions. In the final scenario, the study evaluates SVM classifiers by testing various kernels, namely RBF, Polynomial, and Linear, with the aim of identifying the kernel that yields the best classification performance.

A. Scenario 1 : Lemmatization and Without Lemmatization

Scenario 1 involved experiments comparing text preprocessing with lemmatization and without lemmatization, using 300-dimensional and Polynomial Kernel Word2Vec for feature extraction and SVM for classification. The results are presented in Table VIII.

TABLE VIII
PERFORMANCE COMPARISON OF LEMMATIZATION AND NON-LEMMATIZATION

Preprocessing	Precision	Recall	F1-Score
Lemmatization	**88.7%**	**88.5%**	**88.6%**
Non-Lemmatization	80.9%	77.7%	79.3%

Lemmatization addresses morphological variations by converting words into their base or dictionary form, referred to as a lemma. This step is particularly beneficial in natural language processing tasks, as it enhances consistency in textual data representation. As for the language modeling techniques, lemmatization produced better precision [19]

B. Scenario 2 : Word2Vec Dimensions

Following the completion of the first scenario, the second scenario was conducted to perform additional experiments. In this stage, the Polynomial kernel is still used, and lemmatization is used in this scenario. The results obtained from this set of experiments are presented in Table IX below.

TABLE IX
PERFORMANCE COMPARISON FOR DIFFERENT WORD2VEC DIMENSIONS

Word2Vec Dimensions	Precision	Recall	F1-Score
100	73.2%	72%	72.6%
300	**88.7%**	**88.5%**	**88.6%**

Table IX demonstrates that the larger 300-dimensional Word2Vec model performs both more effectively and more consistently than the smaller 100-dimensional version. A larger vector space allows for richer representation of word relationships within the language. The increase in dimensionality contributes to stronger correlations between word vectors. This is in line with the approach used in paper [20], which conducted a comparable experiment achieved the highest accuracy of 80,8% when utilizing Word2Vec with 300 dimensions.

C. Scenario 3 : SVM Kernel Linear, RBF, and Polynomial

In the final scenario, the focus is on comparing the kernel presented in Table IX with other kernel types, specifically Polynomial, RBF, and Linear. This stage continues to use lemmatization and the 300-dimensional Word2Vec model, as both configurations demonstrated strong performance in the preceding experiments. The outcomes of this comparison are detailed in Table X below.

According to the results presented in Table X, the Polynomial kernel surpassed both the RBF and Linear kernels in classification performance. This superiority may result from its

Received: July13rd, 2025 | Published: September7th, 2025

p-ISSN: 2406-7768
e-ISSN: 2581-2181

TABLE X
PERFORMANCE COMPARISON FOR DIFFERENT SVM KERNELS

SVM Kernel	Precision	Recall	F1-Score
Polynomial	**88.7%**	**88.5%**	**88.6%**
RBF	88.6%	87.7%	88.2%
Linear	88.1%	86.4%	87.2%

capability to expand the feature space into higher dimensions, enabling clearer linear boundaries between classes. Contrary to its typical dominance, the RBF kernel was marginally less effective in this experiment, while the Linear kernel showed moderate but still weaker results. These findings are in line with the work of Muflikhah et al. [21], which concluded that Polynomial kernels provide better performance than RBF for product review analysis.

V. CONCLUSION

The results suggest that incorporating lemmatization provides substantial improvements, generating cleaner and more context-aware word representations. This highlights that lemmatization functions not only as a reduction process but also as a way of preserving semantic meaning. Likewise, 300-dimensional Word2Vec embeddings outperform their 100-dimensional counterparts by offering a broader representation space and enhancing classification performance. Among the kernel options, the Polynomial kernel stands out as the most effective, especially when combined with lemmatization and 300-dimensional Word2Vec, achieving peak performance of 88.6%.

Based on the analysis of all scenarios, the Polynomial kernel demonstrated the ability to adapt to different preprocessing variations and efficiently leverage the detailed word vector representations in high-dimensional space, making it the most suitable kernel.

Future research is recommended to examine additional preprocessing aspects—such as stemming, word normalization, and data cleaning—that may impact classification outcomes. These studies may provide a better understanding of text classification model performance and contribute to the development of more effective models.

REFERENCES

[1] Suparna Wijaya, Naufal Setyo, and Waidatin Azizah. Potential analysis and supervision of vat on the utilization of digital contents (case study: Steam platform). *Dinasti International Journal of Digital Business Management*, 1:342–352, 04 2020.

[2] Andrazˇ Luisa, Jan Hartman, David Nabergoj, Samo Pahor, Marko Rus, Bozhidar Stevanoski, Jure Demsˇar, and Erik Sˇtrumbelj. Predicting the popularity of games on steam, 10 2021.

[3] Jalaluddin Fadhlurrahman, Neng Herawati, Hayyu Aulya, Ira Puspasari, and Nugraha Utama. Sentiment analysis of game reviews on steam using bert, bilstm, and crf. pages 1–6, 10 2023.

[4] Aldira Fazri, Mahendra Dwifebri Purbolaksono, and Widi Astuti. Aspect-based sentiment analysis of beauty product reviews on female daily website using support vector machine and word2vec. pages 230– 235, 07 2024.

[5] Romario Onsu, Daniel Sengkey, and Feisy Kambey. Implementasi bi-lstm dengan ekstraksi fitur word2vec untuk pengembangan analisis sentimen aplikasi identitas kependudukan digital. *Jurnal Teknologi Terpadu*, 10:46–55, 07 2024.

[6] Maite Jaca-Madariaga, Enara Zarrabeitia, Rosa Rio-Belver, and Marie- Francine Moens. Sentiment analysis model using word2vec, bi-lstm and attention mechanism. 03 2023.

[7] Ruoyu Liu. Exploring the impact of word2vec embeddings across neural network architectures for sentiment analysis. *Applied and Computational Engineering*, 94:106–111, 11 2024.

[8] Nanjin Li. Logistics demand forecasting and simulation based on support vector machine. *Advances in Economics, Management and Political Sciences*, 139:143–148, 12 2024.

[9] Zhen Liu. *Support Vector Machines*, pages 129–140. 01 2025.

[10] Alvi Royyan and Erwin Setiawan. Feature expansion word2vec for sentiment analysis of public policy in twitter. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 6:78–84, 02 2022.

[11] Haili Wang. Word2vec and svm fusion for advanced sentiment analysis on amazon reviews. *Highlights in Science, Engineering and Technology*, 85:743–749, 03 2024.

[12] Nafiatul Risa, Didik Prasetya, Wahyu Nur Hidayat, Putrinda Maula, I Made Wirawan, and Satria Setiawan. Sentiment analysis of 'kampus merdeka' on twitter using support vector machine (svm) algorithm. pages 163–168, 11 2024.

[13] Jiale Deng. The data preprocessing technique in machine learning. *Science and Technology of Engineering, Chemistry and Environmental Protection*, 1, 12 2024.

[14] Botir Elov, Shahlo Khamroeva, and Zilola Khusainova. The pipeline processing of nlp. *E3S Web of Conferences*, 413, 08 2023.

[15] Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*, 2013, 01 2013.

[16] Mariette Awad and Rahul Khanna. *Support Vector Machines for Classification*, pages 39–66. 04 2015.

Received: July13rd, 2025 | Published: September7th, 2025

p-ISSN: 2406-7768
e-ISSN: 2581-2181

[17] Mahfuzul Nissan, James Wagner, and Sharmin Aktar. Database memory forensics: A machine learning approach to reverse-engineer query ac- tivity. *Forensic Science International: Digital Investigation*, 44:301503, 03 2023.

[18] Ahmad Alshammari. Implementation of model evaluation using confu- sion matrix in python. *International Journal of Computer Applications*, 186:42–48, 12 2024.

[19] Vimala Balakrishnan and Lloyd-Yemoh Ethel. Stemming and lemma- tization: A comparison of retrieval performances. *Lecture Notes on Software Engineering*, 2:262–267, 01 2014.

[20] Putra Muhammad, Retno Kusumaningrum, and Adi Wibowo. Senti- ment analysis using word2vec and long short-term memory (lstm) for indonesian hotel reviews. *Procedia Computer Science*, 179:728–735, 01 2021.

[21] Lailil Muflikhah and Dimas Haryanto. High performance of polynomial kernel at svm algorithm for sentiment analysis. *Journal of Information Technology and Computer Science*, 3:194, 11 2018.

Received: July13rd, 2025 | Published: September7th, 2025