Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.1, February 2026, pp. 184~191
e-ISSN: 2548-6861

Comprehensive Comparison of TF-IDF and Word2Vec in Product Sentiment Classification Using Machine Learning Models

Asra Gretya Sinaga [1]*, Robet [2]*, Octara Pribadi [3]*
* Informatics Engineering, STMIK TIME
asraagrtya@gmail.com [1], robertdetime@gmail.com [2], octarapribadi@gmail.com [3]

Article Info

Article history:

Received 2025-10-26
Revised 2025-12-09
Accepted 2025-12-22

Keyword:

Machine Learning,
Sentiment Analysis,
TF-IDF,
Word2Vec,
Product Reviews.

ABSTRACT

Sentiment analysis supports data-driven decisions by turning product reviews into reliable polarity labels. We compare four text representations, TF-IDF, TF-IDF reduced via SVD, Word2Vec (trained from scratch), and a hybrid TF-IDF(SVD-300). Word2Vec, for sentiment classification of Indonesian Shopee product reviews from Kaggle (~2.5k texts). After normalization (with optional emoji handling and Indonesian stemming), ratings are mapped to binary sentiment (≤2 negative, ≥4 positive; 3 discarded). Each representation is evaluated with Logistic Regression, Support Vector Machines (linear/RBF), Naive Bayes, and Random Forest under stratified 5-fold cross-validation. TF-IDF with Logistic Regression (C=1.0) yields the best results (F1-macro = 0.816 ± 0.026; Accuracy = 0.816 ± 0.026), with LinearSVC as a strong runner-up. Word2Vec (scratch) performs lower, consistent with limited data being insufficient to learn stable embeddings, while the hybrid representation offers only modest gains over Word2Vec and does not surpass TF-IDF. These findings indicate that TF-IDF is the most reliable and consistent representation for small, short-text review datasets, and they underscore the impact of feature design on downstream classification performance.

I. INTRODUCTION

The rapid advancement of information technology has influenced consumer shopping behavior, particularly through digital platforms. In this context, the presence of e-commerce has shifted the function of social media beyond mere communication, transforming it into social commerce [1]. The growth of the online shopping market is expected to continue as internet access expands.

The various conveniences offered by e-commerce service providers are a major driver of consumer adoption of online purchasing methods. In many developed countries, online shopping has become part of people's daily lives over the past five years. With an internet connection, users can easily access product availability information from various stores in real time. A similar trend is also observed in Indonesia, where transactions on e-commerce platforms have increased significantly over time [2].

The rapid rise in internet users makes Indonesia a potential market for marketplace platforms. These online applications enable buying and selling transactions from various sellers in one place [3]. The acceleration of digital transformation is inseparable from technological advances, particularly in machine learning, which enables automated analysis of consumer behavior by extracting information from product review data [4].

Sentiment analysis, also known as opinion mining, is a key area of natural language processing that focuses on recognizing and categorizing opinions in text [5]. Sentiment analysis is a technique that extracts subjective information from text data, providing an in-depth understanding of consumers' views or perceptions of a product or service.

Through its application, text data, such as customer reviews, can be analyzed to classify the sentiment contained within, whether positive or negative [6]. The sentiment analysis approach, which continues to be developed on e-commerce platforms, has been proven to make a significant contribution to the implementation of smart cities and to the effectiveness of decision-making systems in the business world [7].

The main challenge in product review sentiment analysis is the complexity of natural language, which contains a wide

variety of emotional expressions, ambiguous contexts, and informal word usage. Furthermore, the large volume of review data makes manual analysis impractical and necessitates an accurate, efficient automated system. Another challenge is how to represent text in a format that is understandable to machine learning algorithms while preserving the semantic meaning of the words.

Various machine learning classification algorithms have been applied to sentiment analysis, including Support Vector Machines (SVMs), Logistic Regression, and Naive Bayes, each with distinct characteristics for handling text data. The selection of an appropriate classification algorithm, combined with effective text representation methods, is crucial in determining the success of a sentiment analysis system.

Previous studies have shown that product reviews play a significant role in shaping consumer purchase intentions. Potential consumers often use reviews from other users as a primary reference when evaluating product quality and building confidence before making a purchase decision [8].

In recent years, developments in sentiment analysis research have shown relatively rapid progress, along with increasing attention to automated text-based opinion processing. Clarisa [9] implemented TF-IDF and Naive Bayes to analyze aspect-based sentiment in female daily reviews.

Research conducted by Wang in 2024 found that combining Word2Vec and SVM algorithms can significantly improve sentiment classification accuracy for product reviews on the Amazon platform [10]. Several recent studies have explored various classification algorithms for sentiment analysis. Research on Support Vector Machine (SVM) has shown progress through testing of different kernel functions, with studies by Mukarrah et al. [11] indicating that polynomial kernels can outperform linear and RBF kernels in specific scenarios.

The study by Jiaxin Lu [12], who evaluated both techniques on the Amazon Fine Food Reviews dataset, found that TF-IDF tended to produce more consistent performance than Word2Vec. In addition to previous research, several other studies are also relevant. Ghatora et al. [13] compared the performance of SVM against LLM models in product review sentiment analysis and found SVM to be more efficient for short data. At the same time, Kayed [14] concluded that Word2Vec is less optimal than other embeddings, such as GloVe, in specific scenarios, especially when data is limited.

Research [15] shows that TF-IDF remains a practical text representation approach for Indonesian, especially when combined with linear classification methods. From a broader perspective, a study found that TF-IDF provides significant improvements in accuracy over N-grams across various text classification datasets [16]. Furthermore, Z. A. Khan and V. Rekha demonstrated that integrating TF-IDF with Word2Vec can enrich semantic information and improve model performance, especially on large-scale data [17].

Ahamad et al. [18] applied machine learning and deep learning techniques to sentiment analysis of handwritten and e-text documents, demonstrating progress in understanding key aspects of different types of data. Other studies have also shown that embedding-based models, such as Word2Vec, can achieve competitive performance in sentiment analysis compared to traditional methods such as TF-IDF [19]. Zainottah et al. [20] analyzed 15,000 Tokopedia e-reviews with TF-IDF, SVM, and stacking, achieving 89% accuracy and proving the effectiveness of TF-IDF+SVM for large-scale sentiment analysis.

A review of previous studies shows that there is still room for improvement, particularly in comparing TF-IDF and Word2Vec features across multiple classification algorithms within a single comparative framework. Most existing studies focus on a single classifier, limiting the generalizability of their findings regarding feature extraction effectiveness.

This study provides a controlled comparison of four text representations, TF-IDF, TF-IDF reduced via SVD, Word2Vec (trained from scratch), and a hybrid TF-IDF(SVD-300). Word2Vec, for product-review sentiment classification using four classical classifiers: Support Vector Machines (Linear and RBF kernels), Logistic Regression, Multinomial Naive Bayes, and Random Forest. SVM is examined with Linear and RBF kernels due to their strong performance on sparse high-dimensional and dense features, respectively; Logistic Regression and Multinomial Naive Bayes serve as classical linear baselines; and Random Forest is applied to dense features (SVD/hybrid) for computational and modeling suitability. This multi-classifier framework allows us to assess whether representation effectiveness is algorithm-dependent or consistently superior across classical classifiers and, ultimately, to recommend an optimal representation for product-review sentiment classification under limited data.

All representation classifier pairs are evaluated under stratified 5-fold cross-validation, reporting mean ± standard deviation for Accuracy, Precision, Recall, F1, and macro-F1 on ~2.5k Indonesian product reviews (Shopee/Kaggle). Ratings are binarized as ≤2 = negative and ≥4 = positive (score 3 discarded).

II. METHOD

This research employed a systematic pipeline to compare various TF-IDF and Word2Vec-based feature extraction methods for product sentiment analysis.

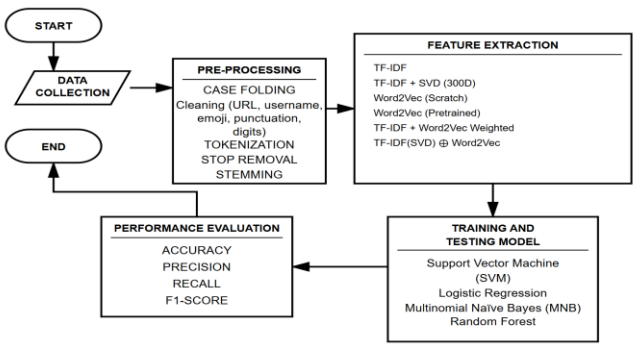

Figure 1. Research Stage

A. Data Collection

These stages include, Data Collection: we use product reviews from the Shopee dataset on Kaggle (~2.5k reviews). Ratings are binarized as ≤2 = negative and ≥4 = positive; rating 3 is discarded to ensure a clear polarity split.

B. Text Preprocessing

This stage aims to clean and standardize the review text for use in modeling. The steps taken are include Case folding: Converting all letters to lowercase. Cleaning URLs, usernames, punctuation, and digits. Tokenization: Splitting sentences into individual words (tokens). Stop Removal: Removing common words that are not meaningful for the analysis. Stemming: Converting words to their base form to unify variations of words with similar meanings using Sastrawi.

C. Feature Extraction

At this stage, the preprocessed text is converted into a numeric representation so the machine learning algorithm can process it. Some of the methods used are TF-IDF: representing text based on term frequency and inverse document frequency, giving higher weight to words that are important within a document but rare across the corpus, Word2Vec (scratch): generating word embeddings by training a Word2Vec model from zero using the dataset, allowing vector representations to adapt specifically to the domain of the reviews, Word2Vec (pretrained): utilizing an existing pretrained Word2Vec model from idwiki_word2vec_300 to obtain high-quality word embeddings learned from a large external corpus, TF-IDF + Word2Vec weighted (hybrid), TF-IDF + Word2Vec weighted: Combining TF-IDF scores with Word2Vec vectors by weighting each word embedding according to its TF-IDF value, producing document-level representations that integrate statistical and semantic information, TF-IDF + SVD (300 dimensions), TF-IDF + SVD 300-dimensi: Reducing high-dimensional TF-IDF vectors into a 300-dimensional semantic space using Singular Value Decomposition (SVD) to capture the most essential latent features, TF-IDF(SVD) ⊕ Word2Vec (concatenation), TF-IDF(SVD) ⊕ Word2Vec: Concatenating low-dimensional TF-IDF (after SVD) with Word2Vec embeddings to form a combined feature vector that integrates both statistical relevance and semantic context.

D. Model Training and Testing

In the Training and testing phase of the model, numerical features from TF-IDF, TF-IDF reduced via SVD, Word2Vec, and the hybrid TF-IDF(SVD-300) ⊕ Word2Vec are used to train Logistic Regression, Support Vector Machines (Linear and RBF), Multinomial Naive Bayes (for sparse TF-IDF), and Random Forest (for dense features: TF-IDF→SVD, Word2Vec, and the hybrid).

E. Performance Evaluation

The performance of all models was evaluated using stratified 5-fold cross-validation. Vectorizers/SVD/embeddings are fit only on training folds to avoid data leakage. We report mean ± standard deviation for all metrics, with macro-F1 as the primary metric. The performance is measured using four evaluation metrics. Accuracy: Measures how many predictions are correct from all the data. Precision: Assesses the accuracy of the model in predicting positive data. Recall: Measures the model's ability to find all positive data. F1-Score: Harmonic mean of precision and recall, useful when data is imbalanced.

These metrics help determine which combination of feature extraction method (TF-IDF or Word2Vec) and classification model (SVM, LR, MNB) produces the best sentiment analysis performance.

F. System Architecture

This sentiment analysis system compares multiple text representations (TF-IDF, TF-IDF reduced via SVD, Word2Vec trained from scratch, and a hybrid TF-IDF(SVD-300) ⊕ Word2Vec) across classical machine-learning classifiers (Logistic Regression, Support Vector Machines, Linear and RBF, Multinomial/Gaussian Naive Bayes, and Random Forest). The pipeline is implemented in a modular fashion to keep each stage independent and reproducible: preprocessing → feature extraction → model training → cross-validated evaluation.

The system is implemented in Python using NumPy/pandas for data handling, scikit-learn for TF-IDF, SVD, scaling, and classical models, NLTK/Sastrawi for Indonesian text preprocessing, and Gensim for Word2Vec embeddings. All experiments are executed in Jupyter Notebook with stratified 5-fold cross-validation, fitting vectorizers/SVD/embeddings on training folds only to avoid leakage, and reporting mean ± standard deviation for Accuracy, Precision, Recall, F1, and macro-F1.

G. Feature Engineering

This study employs multiple feature engineering techniques, including TF-IDF, Word2Vec (both scratch-trained and pretrained), TF-IDF combined with Word2Vec weighting (hybrid), TF-IDF with dimensionality reduction using SVD, and a concatenation of TF-IDF(SVD) with Word2Vec embeddings.

Feature extraction using TF-IDF was performed using the TfidfVectorizer from the scikit-learn library with the following parameters: max_features=5000 to select the most essential words based on frequency, ngram_range=(1,2) to extract unigrams and bigrams, and sublinear_tf=True for a logarithmic scale for term frequency. The basic TF-IDF formula is:

$$TF - IDF(t, d, D) = TF(t, d) \times IDF(t, D) \quad (1)$$

with;

$$TF(t,d) = f\frac{(t,d)}{|d|} \; dan \; IDF(t,D) = \log\left(\frac{|D|}{|\{d \in D \div t \in d\}|}\right) \quad (2)$$

The Word2Vec-based representations in this study were implemented using two approaches: a scratch-trained model and a pretrained model. For the scratch-trained version, the Gensim library was used with a vector size of 100 dimensions, a window size of 5 to capture contextual relationships among five neighboring words, and the Skip-gram architecture (sg=1), which is more effective for smaller datasets and produces richer semantic embeddings than the CBOW method. The parameters also include min_count=1 to ensure all words are included without frequency filtering and four worker threads for parallel processing.

In addition to the scratch-trained model, a pretrained Word2Vec embedding was utilized to incorporate broader semantic knowledge learned from a large external corpus. This allows the model to benefit from richer vocabulary coverage and more generalizable word representations.

For document-level representation, two strategies were employed. First, the simple averaging method calculates the document vector as the mean of all word vectors contained in the sentence or review:

$$d = \left(\frac{1}{n}\right)\sum_{i=1}^{n} v(w_i) \quad (3)$$

Secondly, a TF-IDF–weighted Word2Vec approach was applied, in which each word embedding is multiplied by its corresponding TF-IDF value. This weighting mechanism gives greater influence to terms that are more informative within the document, resulting in a document vector that reflects both semantic meaning and statistical importance more effectively.

In addition, Word2Vec embeddings were combined with TF-IDF representations, and the resulting representations were reduced to 300 dimensions using SVD. The two vectors were then concatenated to create a more comprehensive hybrid feature representation that captures both the latent statistical structure of TF-IDF and the contextual semantic information encoded by Word2Vec.

H. Model Training and Optimization

Several machine learning classifiers were used to evaluate the various feature engineering methods, including TF-IDF, Word2Vec (scratch and pretrained), hybrid TF-IDF weighted Word2Vec, TF-IDF with SVD reduction, and TF-IDF(SVD) concatenated with Word2Vec. Using multiple models ensures a more reliable comparison and reduces bias toward any single algorithm.

Two SVM variants were applied: a linear SVM for high-dimensional sparse features and an RBF-kernel SVM for capturing nonlinear patterns in dense embeddings. Hyperparameters for SVM were optimized using 5-fold Grid Search over parameters C ∈ [0.1, 1, 10, 100] and gamma ∈ ['scale', 'auto'].

Logistic Regression served as a strong linear baseline, especially suitable for TF-IDF-based representations. Multinomial Naive Bayes was used for TF-IDF features, while Gaussian Naive Bayes was applied to continuous feature spaces such as Word2Vec and hybrid vectors. A Random Forest classifier was also included to assess the performance of an ensemble-based, non-linear approach.

Overall, employing multiple classifiers enables a comprehensive performance comparison across linear, non-linear, probabilistic, and ensemble-based learning methods.

III. Results And Discussion

The comparative analysis of TF-IDF and Word2Vec was implemented using a modular pipeline comprising preprocessing, feature extraction, model training, evaluation, and prediction. This modular structure ensures that each component can be independently improved without disrupting the overall workflow.

The system was implemented in Python using libraries such as pandas and NumPy for data processing, scikit-learn for model training, Gensim for Word2Vec, and matplotlib/seaborn for visualization.

The preprocessing stage applies several normalization steps: lowercase conversion, removal of URLs/usernames/punctuation/digits, tokenization, Indonesian stopword removal, and Sastrawi stemming. The resulting text is then prepared for feature extraction, recombined into strings for TF-IDF, or kept as token lists for Word2Vec (with subsequent normalization/scaling for dense features).

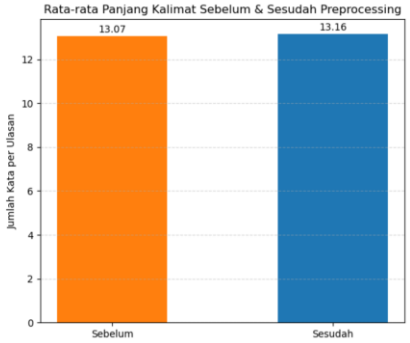

Figure 2. Average Words per Review Before and After Preprocessing

The preprocessing procedure successfully cleaned and normalized the review text. Although the average word count reduction was minimal, the semantic content of the reviews was preserved, ensuring that both TF-IDF and Word2Vec produced reliable and representative features for classification.

Model evaluation was conducted using four machine-learning algorithms: Logistic Regression, Support Vector Machines (Linear and RBF kernels), Naive Bayes (Multinomial and Gaussian), and Random Forest, applied to multiple feature representations: TF-IDF, Word2Vec

(scratch), TF-IDF reduced via SVD, TF-IDF-weighted Word2Vec, and concatenated TF-IDF(SVD-300) ⊕ Word2Vec vectors. All models were evaluated using stratified 5-fold cross-validation, and we report macro-averaged F1 (mean ± std) as the primary metric, providing a balanced assessment across classes.

The combined ranking across all representation–model pairs reveals clear performance tiers. TF-IDF consistently produced the strongest results, especially when paired with Logistic Regression and Linear SVM. Word2Vec (scratch) yielded lower performance, confirming that embeddings trained on a small, domain-specific corpus carry limited semantic richness. Hybrid representations, both TF-IDF-weighted Word2Vec and TF-IDF(SVD-300) ⊕ Word2Vec, achieved moderate scores but did not surpass pure TF-IDF.

The TF-IDF analysis is shown in Fig. 3.

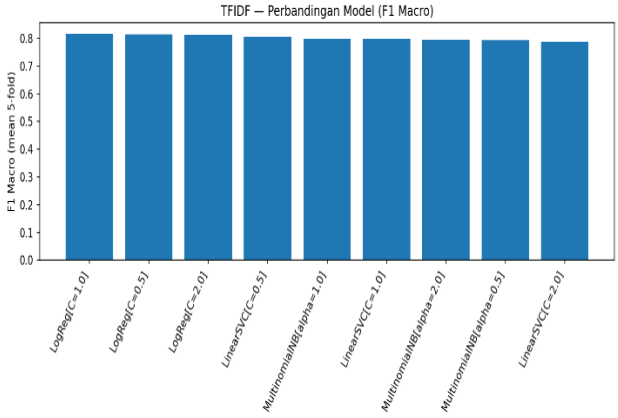

Figure 3. Bar TFIDF f1 macro

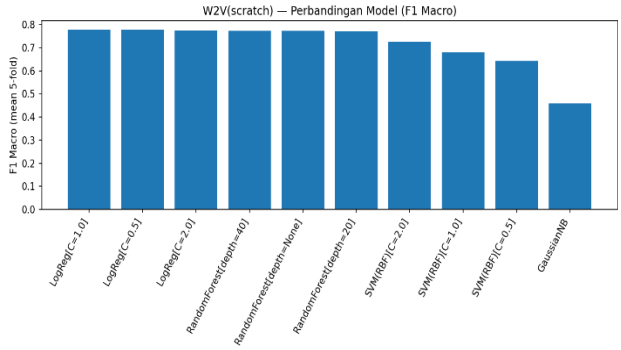

Figure 4. Bar W2V(scratch) f1 macro

Evaluation of the Word2Vec representations trained from scratch was conducted using several classification algorithms, including Logistic Regression, Random Forest, RBF-based SVM, and Gaussian Naive Bayes. The performance of each model was measured using 5-fold cross-validation, with F1 Macro as the primary metric. The results showed that Logistic Regression and SVM provided the most consistent performance, while Random Forest remained competitive, and Gaussian Naive Bayes ranked lowest.

Evaluation on the scratch-trained Word2Vec representation shows that Logistic Regression and RBF-SVM achieve the highest and most consistent F1-macro scores. Random Forest follows closely with competitive performance, while Gaussian Naive Bayes ranks the lowest, reflecting its limitations when modeling dense vector embeddings.

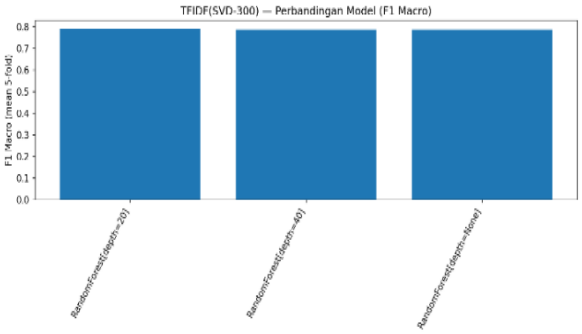

Figure 5. Bar TFIDF(SVD-300) f1 macro

Across the TF-IDF(SVD-300) representation, all Random Forest configurations deliver nearly identical F1-macro performance, indicating that tree depth has minimal impact on the model's effectiveness under this reduced feature space.

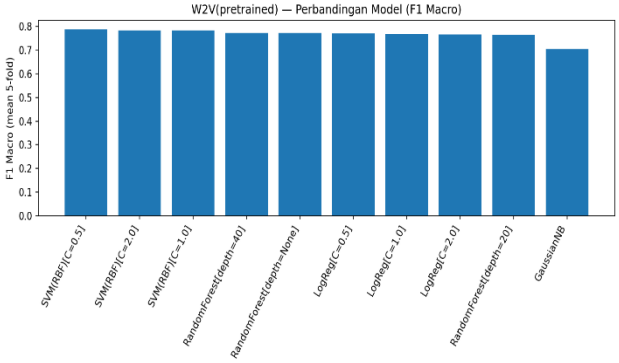

Figure 6. Bar W2V(pretrained) f1 macro

The results on W2V(pretrained) show stable and close model performance, with SVM, Logistic Regression, and Random Forest being the best, while GaussianNB is the lowest

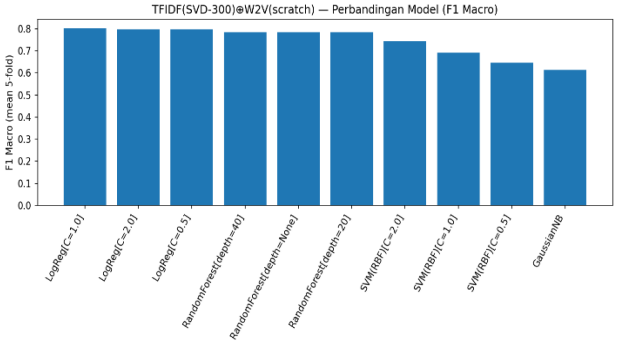

Figure 7. Bar TFIDF(SVD-300)⊕W2V(scratch) f1 macro

On the hybrid TF-IDF(SVD-300) and Word2Vec representation, Logistic Regression consistently attains the highest F1-macro scores, with Random Forest and RBF-SVM following closely. SVM models with smaller C values show a moderate drop in performance, while Gaussian Naive Bayes remains the weakest performer.

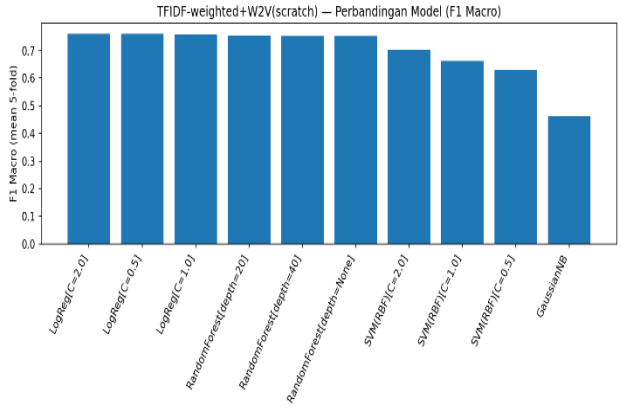

Figure 8. Bar TFIDF-weightedplusW2V(scratch) f1 macro

Using the TF-IDF-weighted Word2Vec representation, Logistic Regression continues to yield the strongest F1-macro scores, followed closely by Random Forest. RBF-SVM shows slightly lower but stable performance, while Gaussian Naive Bayes remains the weakest among all models.

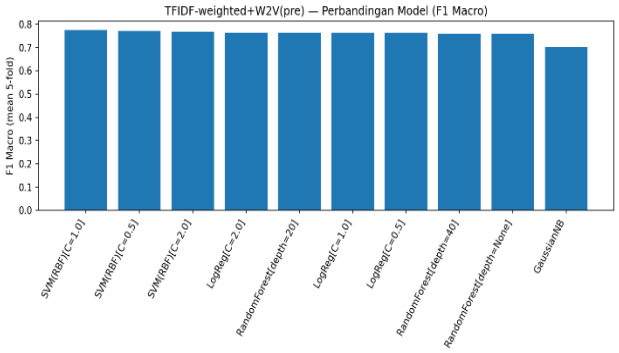

Figure 9. Bar TFIDF-weightedplusW2V(pre) f1 macro

Model comparison on TF-IDF-weighted + Word2Vec (pretrained). SVM (RBF) and Logistic Regression achieve the highest and most stable scores (F1-macro ≈ 0.769–0.774), Random Forest is consistently mid-range (≈ 0.760–0.764), while GaussianNB trails the group (≈ 0.703).

Overall across representations. TF-IDF consistently delivers the strongest performance, Logistic Regression (C=1.0) reaches F1-macro = 0.816 ± 0.026 (best overall), with Linear SVM a close second (≈ 0.806 ± 0.020). The TF-IDF(SVD-300) ⊕ Word2Vec hybrid is competitive but remains below pure TF-IDF (≈ 0.799 ± 0.017). Random Forest provides stable mid-tier results on dense features, while Multinomial Naive Bayes shows high precision but lower recall, yielding moderate F1-macro scores.

TABLE 1.
PERFORMANCE COMPARISON ACROSS ALL REPRESENTATIONS AND MODELS

NO	Metric Evaluation	Algorithms
1	TF-IDF	LogReg[C=1.0]
	Accuracy	0.816 ± 0.026
	Precision	0.842 ± 0.026
	Recall	0.778 ± 0.031
	F1-Score	0.809 ± 0.028
	F1-Macro	0.816 ± 0.026
2	TF-IDF	LogReg[C=0.5])
	Accuracy	0.815 ± 0.030
	Precision	0.847 ± 0.034
	Recall	0.768 ± 0.030
	F1-Score	0.805 ± 0.031
	F1-Macro	0.814 ± 0.030
3	TF-IDF	LogReg[C=2.0)
	Accuracy	0.814 ± 0.024
	Precision	0.832 ± 0.024
	Recall	0.785 ± 0.029
	F1-Score	0.808 ± 0.026
	F1-Macro	0.813 ± 0.024
4	TF-IDF	LinearSVC[C=0.5]
	Accuracy	0.806 ± 0.020
	Precision	0.818 ± 0.022
	Recall	0.787 ± 0.025
	F1-Score	0.802 ± 0.020
	F1-Macro	0.806 ± 0.020
5	TF-IDF	MultinomialNB[alpha=1.0]
	Accuracy	0.800 ± 0.020
	Precision	0.889 ± 0.029
	Recall	0.687 ± 0.026
	F1-Score	0.775 ± 0.023
	F1-Macro	0.798 ± 0.020
6	TF-IDF	MultinomialNB[alpha=2.0]
	Accuracy	0.799 ± 0.022
	Precision	0.905 ± 0.034
	Recall	0.670 ± 0.027
	F1-Score	0.770 ± 0.026
	F1-Macro	0.796 ± 0.023
7	TFIDF(SVD-300)⊕W2V(scratch)	LogReg[C=0.5]
	Accuracy	0.799 ± 0.017
	Precision	0.829 ± 0.022
	Recall	0.754 ± 0.026
	F1-Score	0.789 ± 0.019
	F1-Macro	0.799 ± 0.017
8	TFIDF	LinearSVC[C=1.0]
	Accuracy	0.797 ± 0.018
	Precision	0.805 ± 0.021
	Recall	0.785 ± 0.022
	F1-Score	0.795 ± 0.019
	F1-Macro	0.797 ± 0.018
9	TFIDF(SVD-300)⊕W2V(scratch)	LogReg[C=1.0]
	Accuracy	0.797 ± 0.015
	Precision	0.824 ± 0.025
	Recall	0.757 ± 0.028
	F1-Score	0.788 ± 0.016
	F1-Macro	0.797 ± 0.015
10	TFIDF	MultinomialNB[alpha=0.5]
	Accuracy	0.797 ± 0.021
	Precision	0.875 ± 0.031

	Recall	0.694 ± 0.029
	F1-Score	0.774 ± 0.024
	F1-Macro	0.795 ± 0.021
11	TFIDF(SVD-300)⊕W2V(scratch)	RandomForest[depth=None]
	Accuracy	0.795 ± 0.032
	Precision	0.818 ± 0.037
	Recall	0.759 ± 0.030
	F1-Score	0.787 ± 0.032
	F1-Macro	0.795 ± 0.032
12	TFIDF(SVD-300)⊕W2V(scratch)	RandomForest[depth=40]
	Accuracy	0.795 ± 0.032
	Precision	0.818 ± 0.037
	Recall	0.759 ± 0.030
	F1-Score	0.787 ± 0.032
	F1-Macro	0.795 ± 0.032
13	TFIDF(SVD-300)⊕W2V(scratch)	LogReg[C=2.0]
	Accuracy	0.794 ± 0.018
	Precision	0.817 ± 0.027
	Recall	0.759 ± 0.027
	F1-Score	0.786 ± 0.018
	F1-Macro	0.794 ± 0.018
14	TFIDF(SVD-300)⊕W2V(pretrained)	RandomForest[depth=20]
	Accuracy	0.793 ± 0.022
	Precision	0.828 ± 0.021
	Recall	0.740 ± 0.029
	F1-Score	0.781 ± 0.024
	F1-Macro	0.792 ± 0.022
15	TFIDF(SVD-300)⊕W2V(scratch)	RandomForest[depth=20]
	Accuracy	0.791 ± 0.033
	Precision	0.813 ± 0.036
	Recall	0.756 ± 0.035
	F1-Score	0.783 ± 0.034
	F1-Macro	0.791 ± 0.033

The experiments indicate that feature representation is the principal driver of performance. In this setting, TF-IDF consistently dominates. The combination TF-IDF + Logistic Regression (C = 1.0) ranks first with F1-macro = 0.816 ± 0.026 (identical accuracy), with C = 0.5 and C = 2.0 close behind, suggesting a broad optimum; the decision boundary is not overly sensitive to mild regularization changes. Linear SVM is a strong runner-up (F1-macro ≈ 0.806 ± 0.020), reinforcing that linear margins align well with the high-dimensional, sparse nature of n-gram features. Meanwhile, Multinomial Naive Bayes on TF-IDF yields high precision but lower recall (F1-macro ≈ 0.795–0.800), a familiar pattern given independence assumptions and term-count features.

When TF-IDF is reduced with SVD (300 dimensions) to yield dense features, non-linear or distance-based models such as SVM-RBF, Random Forest, or GaussianNB become more computationally suitable. However, even though dense projections are friendlier to these models, the low-rank mapping can discard sharp lexical cues (e.g., negations, intensifiers, brand/variant tokens) that make TF-IDF + linear classifiers excel. This also explains the behavior of hybrid representations: both TF-IDF-weighted Word2Vec and TF-IDF(SVD-300) ⊕ Word2Vec deliver moderate performance and do not surpass pure TF-IDF. The best hybrid with Logistic Regression reaches F1-macro = 0.799 ± 0.017, while Random Forest on hybrid features sits around ≈ 0.791–0.795. Notably, the pretrained hybrid variant is comparable in one configuration (e.g., RF depth = 20, F1-macro = 0.792 ± 0.022), indicating that immense external corpora help, but still not enough to overtake TF-IDF on this task and dataset.

The relatively lower performance of Word2Vec (scratch) confirms that embeddings are sensitive to corpus size. With ~2.5k reviews, training from scratch produces vectors with limited stability and coverage, and simple document averaging tends to blur polarity contexts (e.g., bigram negations such as “tidak bagus”). By contrast, TF-IDF preserves explicit lexical signals, especially unigrams and bigrams, that linear models exploit effectively.

In terms of reliability, the standard deviations for the top TF-IDF models are minor (e.g., LR ± 0.026; LinearSVC ± 0.020), indicating stable generalization across folds. The closeness of Accuracy and F1-macro for the best runs suggests balanced class proportions after discarding rating 3, and no dramatic collapse on either class.

Practically, these findings suggest that for small, short Indonesian review corpora, TF-IDF (uni/bi-gram) with linear classifiers, in particular Logistic Regression (C ≈ 1.0) or Linear SVM, is the most reliable and efficient choice. Dense representations (SVD/hybrids) are helpful if one must deploy SVM-RBF/RF for computational or architectural reasons, but in this study, they do not surpass TF-IDF + linear. Strengthening embedding-based approaches will likely require much larger Indonesian pretraining corpora (or modern sentence encoders), along with linguistic handling such as slang normalization and negation heuristics to address the remaining error sources.

TABLE 2.
COMPARISON WITH RELATED RESEARCH

Study	**Dataset Size**	**Method**	**Accuracy**	**Domain**
Ahamad et al. [18]	20,000 + samples	ML + DL + Lexicon (ESIHE_AML)	CNN: 90%, Bi-LSTM: 89%	Multidomain (Twitter, Handwritten, Reviews)
Ghatora[13]	10,000 reviews	LLM + SVM	85.3%	E-commerce Review
Kayed[14]	5,000 sentences	Word2Vec + GloVe + Deep NN	88.1%	Mixed Sentiment Data
Ours	2,500 reviews	TF-IDF, TF-IDF(SVD-300), Hybrid TF-IDF ⊕ Word2Vec + ML Models	0.816 (TF-IDF), 0.803 (Hybrid)	Product Reviews

The studies span different dataset sizes, domains, and model families, so results are not directly comparable. For cross-paper readability, we report Accuracy as given by each source, while our primary metric is macro-F1 (mean ± std) under stratified 5-fold CV. On our ~2.5k Indonesian product reviews, TF-IDF + Logistic Regression (C=1.0) achieves

macro-F1 = 0.816 ± 0.026 (Accuracy 0.816 ± 0.026), and the best hybrid TF-IDF(SVD-300) ⊕ Word2Vec reaches macro-F1 = 0.799 ± 0.017 (Accuracy ≈ 0.803). These findings emphasize that feature representation substantially impacts classification performance: TF-IDF remains the most reliable choice for short, small review texts, while hybrid/embedding approaches help moderately but do not surpass TF-IDF in this setting.

IV. CONCLUSION

This study built a modular sentiment-analysis pipeline and compared four text representations, TF-IDF, TF-IDF reduced via SVD, Word2Vec (scratch), and a hybrid TF-IDF(SVD-300) ⊕ Word2Vec, across Logistic Regression, SVM (linear/RBF), Multinomial/Gaussian Naive Bayes, and Random Forest using stratified 5-fold cross-validation on ~2.5k Indonesian product reviews (ratings mapped to binary: ≤2 negative, ≥4 positive; score 3 discarded).

Empirically, TF-IDF with Logistic Regression (C=1.0) delivered the best and most consistent performance (F1-macro = 0.816 ± 0.026; Accuracy = 0.816 ± 0.026), while LinearSVC was a strong runner-up (F1-macro = 0.806 ± 0.020). Multinomial Naive Bayes on TF-IDF yielded high precision but lower recall, consistent with term-count assumptions.

The hybrid TF-IDF(SVD-300) ⊕Word2Vec attained moderate scores (best F1-macro ≈ 0.799–0.803) and did not surpass the TF-IDF baseline; Word2Vec (scratch) performed lower overall, indicating that effective embeddings require a larger and more diverse corpus than the dataset used here. TF-IDF(SVD-300) features were relatively stable across Random Forest settings, but still below full TF-IDF with LR. Overall, the results underscore that feature representation dominates downstream performance: TF-IDF remains the most robust and practical choice for small, short-text review datasets, while embedding-based representations benefit from more data. These findings provide an apparent comparative reference for pairing representations with classical classifiers in similar Indonesian e-commerce scenarios.

REFERENCES

[1] W. Zhao, F. Hu, J. Wang, T. Shu, and Y. Xu, "A systematic literature review on social commerce: Assessing the past and guiding the future," *Electron. Commer. Res. Appl.*, vol. 57, 2023, doi: 10.1016/j.elerap.2022.101219.

[2] L. A. Huwaida *et al.*, "Generation Z and Indonesian Social Commerce: Unraveling key drivers of their shopping decisions," *J. Open Innov. Technol. Mark. Complex.*, vol. 10, no. 2, 2024, doi: 10.1016/j.joitmc.2024.100256.

[3] D. A. Agustina, S. Subanti, and E. Zukhronah, "Implementasi Text Mining Pada Analisis Sentimen Pengguna Twitter Terhadap Marketplace di Indonesia Menggunakan Algoritma Support Vector Machine," *Indones. J. Appl. Stat.*, vol. 3, no. 2, p. 109, 2021, doi: 10.13057/ijas.v3i2.44337.

[4] A. Daza, N. D. González Rueda, M. S. Aguilar Sánchez, W. F. Robles Espíritu, and M. E. Chauca Quiñones, "Sentiment Analysis on E-Commerce Product Reviews Using Machine Learning and Deep Learning Algorithms: A Bibliometric Analysisand Systematic Literature Review, Challenges and Future Works," *Int. J. Inf. Manag. Data Insights*, vol. 4, no. 2, 2024, doi: 10.1016/j.jjimei.2024.100267.

[5] C. D. Sasongko, R. Isnanto, and A. P. Widodo, "Review of Systematic Literature about Sentiment Analysis Techniques," *J. Sist. Inf. Bisnis*, vol. 15, no. 2, pp. 227–236, 2025, doi: 10.14710/vol15iss2pp227-236.

[6] B. Bansal and S. Srivastava, "Sentiment classification of online consumer reviews using word vector representations," *Procedia Comput. Sci.*, vol. 132, pp. 1147–1153, 2018, doi: 10.1016/j.procs.2018.05.029.

[7] J. A. Aguilar-Moreno, P. R. Palos-Sanchez, and R. Pozo-Barajas, "Sentiment analysis to support business decision-making. A bibliometric study," *AIMS Math.*, vol. 9, no. 2, pp. 4337–4375, 2024, doi: 10.3934/math.2024215.

[8] Dr. Kochuthresia Jose, "the Rise of Social Commerce: Analyzing Consumer Buying Behavior on Social Media," *Intersecting Realms New Dimens. Multidiscip. Res. Vol.*, 2020, doi: 10.25215/9358091800.16.

[9] C. H. Yutika, A. Adiwijaya, and S. Al Faraby, "Analisis Sentimen Berbasis Aspek pada Review Female Daily Menggunakan TF-IDF dan Naïve Bayes," *J. Media Inform. Budidarma*, vol. 5, no. 2, p. 422, 2021, doi: 10.30865/mib.v5i2.2845.

[10] H. Wang, "Word2Vec and SVM Fusion for Advanced Sentiment Analysis on Amazon Reviews," *Highlights Sci. Eng. Technol.*, vol. 85, pp. 743–749, 2024, doi: 10.54097/sw4pft19.

[11] R. Mukarramah, D. Atmajaya, and L. B. Ilmawan, "Performance comparison of support vector machine (SVM) with linear kernel and polynomial kernel for multiclass sentiment analysis on twitter," *Ilk. J. Ilm.*, vol. 13, no. 2, pp. 168–174, 2021, doi: 10.33096/ilkom.v13i2.851.168-174.

[12] J. Lu, "Text vectorization in sentiment analysis: A comparative study of TF-IDF and Word2Vec from Amazon Fine Food Reviews," *ITM Web Conf.*, vol. 70, p. 03001, 2025, doi: 10.1051/itmconf/20257003001.

[13] P. S. Ghatora, S. E. Hosseini, S. Pervez, M. J. Iqbal, and N. Shaukat, "Sentiment Analysis of Product Reviews Using Machine Learning and Pre-Trained LLM," *Big Data Cogn. Comput.*, vol. 8, no. 12, 2024, doi: 10.3390/bdcc8120199.

[14] A. Mabrouk, R. P. D. Redondo, and M. Kayed, "Deep Learning-Based Sentiment Classification: A Comparative Survey," *IEEE Access*, vol. 8, pp. 85616–85638, 2020, doi: 10.1109/ACCESS.2020.2992013.

[15] C. Apriansyah Hutagalung and V. Budi Lestari, "Data Mining Approach: K-Means Clustering and Naïve Bayes Classifier for Graduate Quality Analysis," *J-KOMA J. Ilmu Komput. dan Apl.*, vol. 8, no. 1, pp. 33–42, 2025, doi: 10.21009/j-koma.v8i1.05.

[16] M. Das, S. Kamalanathan, and P. Alphonse, "A Comparative Study on TF-IDF feature weighting method and its analysis using unstructured dataset," *CEUR Workshop Proc.*, vol. 2870, pp. 98–107, 2021.

[17] Z. A. Khan and V. Rekha, "Fake News Detection Using TF-IDF Weighted with Word2Vec: An Ensemble Approach," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 3, pp. 1065–1076, 2023.

[18] R. Ahamad and K. N. Mishra, "Exploring sentiment analysis in handwritten and E-text documents using advanced machine learning techniques: a novel approach," *J. Big Data*, vol. 12, no. 1, 2025, doi: 10.1186/s40537-025-01064-2.

[19] R. Mulyawan, H. Naparin, and W. M. Fatihia, "Comparison of Text Vectorization Methods for IMDB Movie Review Sentiment Analysis Using SVM," *J. Appl. Informatics Comput.*, vol. 9, no. 5, pp. 2270–2277, 2025.

[20] M. Z. Zainottah, R. S. Rengga, Y. S. Yustian, and I. R. Isa, "Critical Sentiment Analysis of Tokopedia Electronic Products Using SVM-Logistic & TF-IDF Ensemble Methods," *J. Artif. Intell. Eng. Appl.*, vol. 4, no. 3, pp. 2476–2482, 2025, doi: 10.59934/jaiea.v4i3.1194.