Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi

Vol. 13, No. 1, Juli 2025

ISSN 0216 – 0544
e-ISSN 2301– 6914

Comparison of feature extraction in support vector machine (SVM) based sentiment analysis system

Imam Fahrur Rozi[a], Irma Maulidia[b], Mamluatul Hani'ah[c], Rakhmat Arianto[d], Dika Rizky Yunianto[e], Ahmadi Yuli Ananta[f]

[a,b,c,d,e,f]Department of Information Technology, State Polytechnic of Malang, Malang, Indonesia
E-mail: imam.rozi@polinema.ac.id, irmamaulidiaaa@gmail.com, mamluatulhaniah@polinema.ac.id, arianto@polinema.ac.id, dikarizkyyunianto@polinema.ac.id, ahmadi@polinema.ac.id

Abstract

Sentiment analysis plays a crucial role in natural language processing by identifying and categorizing opinions or emotions conveyed in textual data. It is widely applied across diverse fields such as product review analysis, social media monitoring, and market research. To enhance the accuracy and reliability of sentiment classification, various methods and feature extraction techniques have been explored. This study investigates the use of Support Vector Machine (SVM) for sentiment analysis, comparing three feature extraction techniques: Term Frequency-Inverse Document Frequency (TF-IDF), Bag of Words (BoW), and Word2Vec. Our findings indicate that SVM performs effectively with all three feature extraction methods, with TF-IDF yielding the highest accuracy at 0.79. Although the BoW method showed competitive results, it slightly trailed TF-IDF in k-fold validation. Word2Vec, however, exhibited the lowest performance, achieving a maximum accuracy of 0.69. A comparative analysis of accuracy, precision, recall, and F1-score highlight the superiority of TF-IDF in delivering consistent and accurate results. Further statistical analysis using ANOVA revealed no significant differences between the models across any of the evaluation metrics. Additionally, the evaluation was conducted under several scenarios, including tests on balanced and imbalanced datasets, varying dataset sizes, and different CCC parameter values for SVM. These scenarios provided deeper insights into the factors influencing the system's performance, reinforcing that TF-IDF combined with SVM remains the most effective approach in this study.

Key words: ANOVA, Bag of Word, Feature Extraction, Sentiment Analysis, SVM, TF-IDF, Word2Vec.

INTRODUCTION

In the fast-growing digital era, various online information sources are becoming increasingly rapid and significant. Sentiment analysis is a process used to help identify the content of a dataset in the form of text-based opinions or views on an issue or event that are positive or negative [1]. In conducting sentiment analysis, a method that supports classification is needed, such as the Naive Bayes Classifier, which was implemented to analyze the sentiments related to League of Legend. The best result was achieved without stemming, by transforming informal words into formal ones, and by using BOW bigram feature extraction [2]. Another machine learning method that could be used for sentiment analysis is Support Vector Machine (SVM). Based on the results of previous sentiment analysis research conducted by Rima and Pandu, in this study, sentiment analysis was

carried out on public opinion regarding the COVID-19 Booster Vaccination by comparing the Support Vector Machine, Naïve Bayes Classifier, and Decision Tree methods with findings showing the SVM has the biggest average value when compared to the other two methods [3]. Probabilistic Neural Network (PNN) as a machine learning based method obtained a promising result with up to 90% accuracy when classifying the negative and positive e-complaints on campus [4]. In addition to machine learning approaches, deep learning-based methods can also be applied to sentiment analysis. Previous research, for instance, analyzed electric car reviews using the Recurrent Neural Network (RNN) method, achieving an accuracy of 72% [5].

Feature extraction has a central role in the establishment of sentiment analysis methods. Proper feature selection can significantly affect the performance of the method. Therefore, this research focuses on the application of different feature extractions, such as Term Frequency-Inverse Document Frequency (TF-IDF), Bag of Words (BoW) & Word to Vector (Word2Vec). Similar research that has been done before on Indonesian Hoax Detection using SVM Algorithm with Word2Vec Feature Extraction, to achieve optimal results with accuracy above 80% at least more than 20,000 training data is needed. In this case, the more training data used, the better the performance of the SVM model [6].

In addition, the application of TF-IDF feature extraction is also carried out for sentiment analysis of the Corruption Eradication Commission of the Republic of Indonesia, getting the results of testing and evaluation of the accuracy value of precision, recall, F1-Score which produces a good percentage because with the emergence of positive sentiment skew of 77% [7]. Another relevant study explores the detection of hate speech on Facebook by evaluating various feature extraction methods, including BoW, pre-trained word embeddings, and N-gram graphs. These methods were combined with machine learning classifiers such as SVM. The findings indicate that BoW features, when used with SVM classifiers, achieve superior performance in identifying hate speech on Facebook [8]. Additionally, another study discusses the limitations of BoW features due to high false positive rates and suggests that more sophisticated methods can provide better features for classical machine learning algorithms like SVM. These studies provide insights into the effectiveness of combining BoW feature extraction with SVM classifiers for hate speech detection on social media platforms [9].

Research using the SVM algorithm has previously been applied to sentiment analysis in detecting anxiety based on social media data using the SVM and Random Forest algorithms. The findings revealed that the Random Forest classifier achieved the highest accuracy of 84.99% with count-vectorization and 82.63% with TF-IDF feature extraction, outperforming other classifiers, including SVM [10]. In another study, still focusing on the application of machine learning for sentiment analysis, researchers compared the performance of SVM and Random Forest for classification tasks. The results showed that SVM achieved slightly higher accuracy (0.80394) than Random Forest (0.78564), making it a more suitable choice when accuracy is the primary consideration [11].

SVM was also used to analyze sentiment towards the Jakarta government lockdown policy. SVM is used by applying TF-IDF feature extraction. Divided into positive sentiment classes of 68.75% and negative 31.25% and produced an accuracy value of 74%, precision of 75%, recall of 92%, and F1-Score of 83%. This means that the SVM algorithm with TF-IDF feature extraction is well used in this research on the Jakarta government lockdown policy [12].

TF-IDF feature extraction by applying the SVM algorithm has also been used to analyze sentiment regarding application reviews of information and documentation management officials of the Kementerian Dalam Negeri Republik Indonesia on the Google Play Store. Researchers achieved 700 data, including 85 positive annotations and 615 negative labels. During the evaluation phase, sentiment analysis yielded an average k-fold of 88%, precision of 94%, recall of 100%, f-measure of 97%, and accuracy of 97%. Based on the evaluation results, also shows that the SVM algorithm by applying TF-IDF feature extraction can be applied well [12].

Building upon the previously outlined background, this study will implement the SVM method to TF-IDF, BoW, and Word2Vec

feature extraction techniques. The purpose of this research is to compare SVM performance through the value of some essential performance metrics when implementing the three feature extractions. The findings from this research are anticipated to serve as a foundation for future advancements in developing more effective classification methods. By building on these results, researchers can refine and enhance techniques to achieve better accuracy and efficiency in classification tasks moving forward. Based on prior studies, this research proposes a hypothesis that TF-IDF is likely to outperform BoW and Word2Vec for this dataset. This is due to its capability to weigh terms based on their importance within a document while considering their rarity in the dataset, making it particularly effective in scenarios with small datasets and straightforward semantic structures [13].

MATERIAL AND METHODS

Reserach Flow

The research flow used in this study can be seen in the Fig. 1.

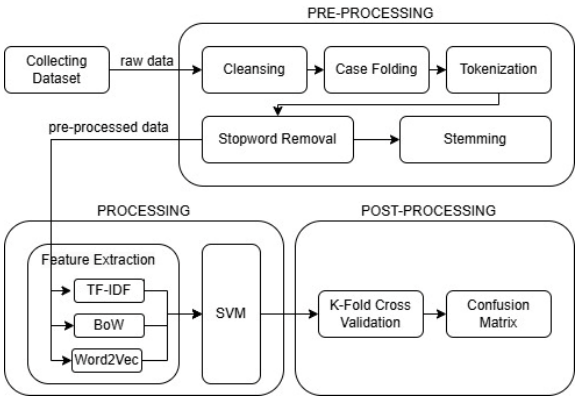

Fig. 1. Research flow

The first step in this research is to collect the data to be used. This dataset contains text or documents to be analyzed that come from Kaggle about beauty product reviews. This data will be the main material that is processed further in the research.

Once the dataset is collected, the next step is preprocessing. At this stage, the raw data is cleaned and prepared for analysis. The preprocessing stages used include Cleansing, Case Folding, Tokenization, Stop Words Removal, and Stemming.

After the data is processed, the next step is feature extraction. Feature extraction is used to convert the processed text into a numerical form that can be understood by machines. Some of the feature extractions used are:

The TF-IDF technique, or Term Frequency-Inverse Document Frequency, serves to calculate the aggregate quantity of documents, measure the complete count of lexical components (words) throughout all documents, and ascertain the weight of each term about distinct documents [14].

Bag of Words (BoW) for processing text data by converting it into a vector of numbers so that it can be processed by a computer. This approach exclusively computes the frequency with which words appear throughout the entirety of the document under examination [15].

Word2Vec to convert words into number vectors based on the meaning of the word in context [16]. After the features are extracted, the data is then classified using the Support Vector Machine (SVM) method. SVM is used to separate the data into different categories based on the features that have been extracted.

The last stage is evaluation, where the performance of the built model is measured using a confusion matrix and k-fold cross validation. This stage is used to help determine how well the model works in classifying the data and whether the research objectives have been achieved.

Data acquition

The data collection stage in sentiment analysis is a crucial first step. In this step, data is collected from Twitter social media. In this research, a dataset of beauty product reviews taken from the Kaggle site by hafidahmusthaanah is used. This dataset contains 1500 textual reviews that have been labeled manually. The labeling process was carried out by three raters, who divided the data into two categories: positive and negative reviews. Each review includes attributes such as username, post date, and review text. However, only the review text column was utilized for feature extraction and classification. The labeled dataset consists of 53% positive reviews and 47% negative reviews. An exploratory analysis of the dataset revealed that the average review length was approximately 20 words, with a standard deviation of 5 words.

Frequently occurring terms in positive reviews included 'terbaik,' 'suka', 'kualitas' etc., while 'jelek', 'kecewa', 'sampah' etc. were common in negative reviews. These observations provide insight into the dataset but are independent of the sentiment classification process, which was conducted using a machine learning approach (SVM) with feature extraction methods such as TF-IDF, BoW, and Word2Vec. The dataset was preprocessed by removing stopwords, punctuation, and special characters to ensure consistency and prepare the text for machine learning feature extraction.

Preprocessing

After the data acquisition stage is complete, the next step is data preprocessing to convert unstructured text data into structured data [17]. The preprocessing stages carried out include:

1. The first step in preprocessing is Cleansing. This step is used to clean the document from irrelevant words, to reduce noise in the data. (Rahman Isnain et al., 2021).
2. The next stage is Case Folding and Tokenization. In the Case Folding stage, all letters in the document are converted into lowercase letters. Only the letters 'a' to 'z' are retained, while characters other than letters are removed and considered as delimiters. After that, Tokenization is performed to break the document into tokens and identify keywords separated by spaces and remove punctuation characters [18].
3. The next step is Stop Words Removal which aims to remove words that have no significant meaning in the analysis [18].
4. The last stage of preprocessing is Stemming, which is used to remove prefixes or suffixes from words, so that the words are returned to their basic form [19].

Feature Extraction

The data that has been generated from preprocessing will then be represented in the form of features in numeric format. There are three feature extraction approaches used in the research TF-IDF, Bag of Words, and Word2Vec. The use of these three feature extraction approaches is intended to evaluate their influence on the performance of the classification process. The feature extraction used in this research includes:

TF-IDF

The first feature extraction used is TF-IDF which is used to convert text data into numerical data by giving weight to each word or feature. This aids in determining the significance of a word within a document. TF-IDF assesses a word's relevance specifically within the context of that document [20].

TF, or Term Frequency, refers to how often a word appears in a particular document, which indicates how relevant the word is in that document.

$$TF(t,d) = \frac{\text{the number of times the word } t \text{ appears in the document } d}{\text{total number of words in a document } d} \quad (1)$$

DF represents the frequency of documents that include a specific term, thereby reflecting the prevalence of that term within the corpus. IDF, on the other hand, serves as the reciprocal of the DF metric. The outcome of the term weighting process utilizing TF-IDF is derived from the multiplication of TF by IDF.

$$IDF(t,D) = \frac{\text{total number of collection documents } d}{\text{number of collection documents containing the word } t} \quad (2)$$

After calculating TF and IDF, then multiply the two to get the TF-IDF value for a word in a document.

$$TF - IDF(t,d,D) = TF(t,d)\ x\ IDF(t,D) \quad (3)$$

The weight of a word increases if it appears frequently in a document and decreases if it appears in many documents.

Bag of Words

The next feature extraction used is Bag of Words (BoW) which is a simple way to represent the words in each sentence by converting them into vectors. Basically, BoW works by counting how often each word appears in a sentence, regardless of order or grammar. BoW is one of the most basic techniques for converting text into numerical data and is widely used in Natural Language Processing (NLP) and Information Retrieval (IR). In BoW, a text, be it a sentence or a document, is transformed into a “bag” containing the set of words it contains, without regard to their order but retaining the variety of the words [21].

The formula used was as follows:

$$V = \{w_1, w_2, w \dots, w_{lvl}\} \quad (1)$$

$$[w = [0, \dots, 0, 1, 0, \dots, 0] \quad (2)$$

The equation employed denotes V as a lexicon (vocabulary) of terms utilized, while w represents a one-hot encoding in which a value of 1 signifies the presence of a term from the lexicon within the pertinent document, whereas a value of 0 denotes the absence of such a term. The establishment of this one-hot encoding is achieved through the computation of the frequency with which the term appears within the sentence, rather than relying exclusively on the binary indicators of 1 and 0 [21].

Word2Vec

The latest feature extraction used is Word2Vec which is used to transform document operations into vector calculations in word vector space. Semantic relations in documents can be characterized based on the similarity of words in the vector space. The initial stage of the Word2Vec process is to build a word dictionary from the training text data and then learn a vector representation of the word set. The resulting vectors serve as features for various applications in NLP and machine learning. Word2Vec is capable of learning word representations in a high-dimensional vector space and identifying the semantic relationships between words by calculating the cosine similarity. This allows it to detect how closely related words are in meaning within a document [22].

Word2Vec provides two steps to get similar words, including:

- CBOW

  The Continuous Bag of Words (CBOW) model, widely recognized in the field of natural language processing, employs adjacent lexical items to infer the intended target word. The architecture of CBOW facilitates the prediction of a present word by analyzing its contextual surroundings [22]. Formula 6 is the formula of CBOW.

$$v_w = \frac{1}{C}\sum_{c \in C} v_c \quad (6)$$

- Skip-gram

  Uses words to predict neighboring words in a sentence. Skip- gram architecture predicts neighboring words based on word flow [22]. Formula 7 is the formula of Skip-gram.

$$P(w_0|w_1) = \frac{v_{w0} \cdot v_{w1}}{\sum_{w=1}^{W} e^{v_{w0} \cdot v_{w1}}} \quad (7)$$

In this research, Word2Vec was implemented using the Continuous Bag of Words (CBOW) method. CBOW was chosen due to its efficiency in processing smaller datasets and its ability to predict a target word based on its surrounding context, which aligns well with the characteristics of the dataset used in this study [23]. The embedding dimensions were set to 100, with a context window size of 5, a minimum word frequency of 1 (min_count), and 4 worker threads to optimize training performance.

Classification

After the feature extraction stage is carried out, the next stage is classification. The classification used in this research is SVM. SVM classification works based on the principle of Structural Risk Minimization (SRM). SVM is used to find the best hyperplane that separates two classes in the input space. The hyperplane is created in such a way as to separate the closest data points from each class. The main advantage of this approach is that it can work on very high dimensional areas [24].

SVM is one of the well-known classification methods because it can find the right classification boundary by using the so-called support vector, hyperplane and kernel. Many SVM applications have been carried out by several researchers. The study [25], the authors applied linear SVM kernel to classify skincare product reviews. SVM uses a function or hyperplane to separate two classes of patterns. In Linear Discriminant Analysis, the worst case can occur, when all classes have SVM will try to find the optimal hyperplane pattern where two classes can be maximally separated. SVM is a binary classifier, but SVM can also be used for many class problems or multiclass problems [26].

The SVM model runs by projecting data into a rich feature landscape, thereby granting the ability to classify data points, even in scenarios where linear separation is unfeasible. In situations where a separation among various categories emerges, the information may be reconfigured such that this distinction can be termed a hyperplane. Subsequently, the

attributes of the modified data may be utilized to infer the appropriate classification for new records. Some data points are separated into 2 categories, namely black spheres and white spheres.

The two categories are then separated by a curve, after transformation, the boundary between the two categories can be determined by the hyperplane. The two points that become the benchmark of the hyperplane are called the support vector. It can be seen that there are two groups of data called classification, then the task of SVM is to divide these two groups as best as possible or determine the best hyperplane, the division where the boundary line can separate the two groups with the farthest distance between the outermost point in each group and the boundary line itself. The non-linear problem can be solved by modifying the kernel trick into SVM which will be the class separator or hyperplane into two classes in the vector space.

Evaluation

After the model is successfully created, the last step that must be taken is evaluation. This step is used to determine the performance of the model that has been created. The evaluation results obtained by applying different feature extractions will be compared to determine the effect of their performance on classification results. There are two evaluation methods used:

1. The first evaluation method used is tje Confusion Matrix to calculate the performance or accuracy of the classification process. The accuracy of the results is measured by the recall, precision, accuracy, and f1-score values. Where recall (True Positive Rate) is the ratio of true positive identifications compared to all true positive data. Precision (Positive Predictive Value) is the ratio of true positive identification to all positive identification results. Accuracy is the ratio of true positive identifications for all data [27].
2. The second evaluation method is K-Fold Cross Validation which is used to get the best accuracy value when the data is divided into test data and training data. The sample data used will be divided randomly and then will be grouped as much as the K value used. After being divided into several partitions, the data will be processed K times with each K trials, the testing data used is the Kth partition data and the rest of the other partitions are used as training data. And so on until the processing ends according to the number of K in the k-fold used in the study [28]. This research employed a 10-fold cross-validation approach to evaluate the model's performance. n each fold, the dataset was divided into 90% training and 10% testing subsets. Evaluation metrics such as accuracy, precision, recall, and F1-score were computed for each fold and averaged to provide a comprehensive assessment.

In this study, several evaluation conditions were conducted to analyze the effects of data balance, dataset size, and the C parameter of SVM on classification performance. These evaluations were designed to provide a comprehensive understanding of the factors that may influence the effectiveness of the classifier.

RESULT AND DISCUSSION

Effect of using balanced and unbalanced data

In the first test, two types of training datasets were evaluated, a balanced dataset and an unbalanced dataset, each consisting of 1100 data. This study evaluated the classifier's performance on both balanced and imbalanced datasets to assess how class distribution affects classification outcomes. The imbalanced dataset reflects the natural distribution of the data, where positive reviews make up 53% and negative reviews account for 47%.

In contrast, the balanced dataset was constructed by oversampling the minority class (negative reviews) to ensure equal representation of both classes. The test results can be seen in Table 1.

Table 1. Effects of using balanced and unbalanced data

Testing Scenario	Evaluation Metrics	TF-IDF	BoW	W2V
Balanced Data	Accuracy	0.79	0.75	0.58
	Precision	0.79	0.75	0.58
	Recall	0.79	0.75	0.58
	F1 Score	0.79	0.75	0.58
Unbalanced Data	Accuracy	0.45	0.44	0.50
	Precision	0.44	0.43	0.20
	Recall	0.45	0.44	0.50
	F1 Score	0.43	0.42	0.34

Based on the test results above, it shows that the SVM model is able to classify data quite accurately, both in recognizing positive and negative classes. The even distribution of classes on a balanced dataset allows the model to learn well from each class, so the model can predict new data more accurately. In contrast, when using an unbalanced dataset, the model's performance drops dramatically. This decrease is due to the unequal distribution of classes in the dataset.

This imbalance makes the model more likely to predict the majority class and ignore the minority class. This difference in results proves the importance of an even distribution of classes in the dataset to achieve good classification performance. On a balanced dataset, the SVM model can learn from each class proportionally, leading to better and more consistent performance. In contrast, on unbalanced datasets, the model tends to be biased towards the majority class, which reduces its ability to recognize minority classes well. This suggests that to improve performance on unbalanced datasets, techniques such as oversampling minority classes, undersampling majority classes, or using appropriate class weighting methods are necessary.

Effect of the Data Size

The second test was conducted using balanced training data but with varying amounts of 200, 500, 700, and 1100 samples. The test results can be seen in the following Table 2.

It can be seen in the figure above that there is an increase in the performance of the SVM model with the three feature extractions as the number of samples in the dataset increases. The accuracy results obtained show a consistent pattern. The greater the number of samples in the dataset, the higher the accuracy obtained. This can be attributed to the concept of machine learning which generally shows that the more data the model has to learn, the better the performance can be achieved.

When the number of samples in the dataset increases, the model has more examples to learn from, so it can better recognize patterns and data variability. The increase in accuracy as the number of samples increases is also due to the model's ability to find better representations of the features. With larger datasets, the model can discover more complex patterns or more informative features that may not be seen when using smaller datasets. Therefore, as the dataset grows larger, the model has the potential to improve its ability to better distinguish between different classes. However, the Word2Vec feature extraction remained consistent at 0.58 for all dataset sizes. This is due to the way Word2Vec builds the feature representation. Word2Vec generates numerical representations based on the context of the words in the text, and its performance is highly dependent on the quality of the plate data.

Table 2. Effect of amount of data used

Testing Scenario	Evaluation Metrics	TF-IDF	BoW	W2V
200	Accuracy	0.53	0.52	0.58
	Precision	0.53	0.52	0.58
	Recall	0.53	0.52	0.58
	F1 Score	0.53	0.52	0.58
500	Accuracy	0.67	0.67	0.58
	Precision	0.68	0.67	0.58
	Recall	0.67	0.67	0.58
	F1 Score	0.67	0.66	0.58
700	Accuracy	0.73	0.70	0.57
	Precision	0.74	0.71	0.57
	Recall	0.74	0.71	0.57
	F1 Score	0.73	0.70	0.57
1100	Accuracy	0.79	0.75	0.58
	Precision	0.79	0.75	0.58
	Recall	0.79	0.75	0.58
	F1 Score	0.79	0.75	0.58

Effect of C Parameters

The last test was carried out adding a parameter, namely the C value. The C values used are 0.1, 1, 10, 100 and 1000. The parameter C in SVM controls the trade-off between maximizing the margin and minimizing misclassification errors. A smaller C improves generalization by prioritizing a wider margin, while a larger C focuses on reducing training errors, which can risk overfitting. Testing C values is crucial to optimize classification performance. Table 3 depicts the results of testing SVM with the three feature extractions.

In the last test, the addition of the C parameter to the SVM classification with the three feature extractions was carried out. The results of testing the C parameter in several scenarios, there was an increase in accuracy. The C parameter controls the degree of

regularization in the SVM model, where higher values indicate the model is more likely to adjust to the training data. The increase in accuracy indicates that the SVM model becomes more flexible and is able to adapt well to the training data. Therefore, choosing the right C parameter is key in improving the performance of the SVM model in this case.

Table 3. Effect of C parameters

Testing Scenario	Evaluation Metrics	TF-IDF	BoW	W2V
0.1	Accuracy	0.67	0.79	0.58
	Precision	0.67	0.79	0.58
	Recall	0.67	0.79	0.58
	F1 Score	0.67	0.79	0.58
1	Accuracy	0.79	0.75	0.58
	Precision	0.79	0.75	0.58
	Recall	0.79	0.75	0.58
	F1 Score	0.79	0.75	0.58
10	Accuracy	0.78	0.75	0.60
	Precision	0.78	0.75	0.60
	Recall	0.78	0.75	0.60
	F1 Score	0.78	0.75	0.60
100	Accuracy	0.78	0.75	0.63
	Precision	0.78	0.75	0.63
	Recall	0.78	0.75	0.63
	F1 Score	0.78	0.75	0.63
1000	Accuracy	0.78	0.75	0.69
	Precision	0.78	0.75	0.69
	Recall	0.78	0.75	0.69
	F1 Score	0.78	0.75	0.69

Overall Comparisons

From the three test scenarios that have been carried out, the overall comparison produced is as presented in Table 4. The testing employed a 10-fold cross-validation method. In this approach, the data is divided into 10 subsets. Each subset is used as the testing set in turn, while the remaining nine subsets serve as the training data. This process is repeated 10 times to ensure that every part of the data is used for both training and testing. At the end of each iteration, the accuracy, precision, recall, and F1-score metrics are computed and averaged to obtain a comprehensive evaluation. Additionally, the standard deviation is calculated to measure the variability in the model's performance across the different folds, ensuring a more reliable and stable performance assessment. TF-IDF works by weighting the importance of a word in a document relative to the document collection. It assigns a higher value to words that appear frequently in a particular document but rarely appear in other documents, thus better at finding key words that are important for classification. Therefore, TF-IDF often gives good results on text classification tasks.

On the other hand, BoW only calculates word frequency without considering the importance of the word in the context of a larger document collection. Although simple and effective, this approach can be less accurate as it does not consider the relative significance of words across different documents.

Word2Vec works by mapping words into high-dimensional vectors based on their surrounding context. While this technique is capable of capturing semantic relationships between words, its performance can be suboptimal as the model does not always capture variations that are relevant for specific tasks such as classification. Moreover, if the training dataset is not large enough or not representative, the resulting Word2Vec model is not robust enough to capture the text required for classification tasks.

Overall, TF-IDF has a better performance compared to Bag of Words and Word2Vec. This can be due to its ability to pay attention to the most relevant features for classification, while Bag of Words and Word2Vec do not capture the same information well for that specific purpose.

While TF-IDF showed the highest performance in this study, previous research highlights that BoW and Word2Vec often outperform TF-IDF in other scenarios. This discrepancy can be understood by considering factors such as dataset size and semantic complexity. several studies have shown that BoW and Word2Vec often achieve higher accuracy in various applications. For instance, research has found that combining TF-IDF and Word2Vec improves text classification accuracy compared to using TF-IDF alone [29]. Additionally, another study reported that document representation using Word2Vec yielded better classification performance than traditional methods for Mandarin text [30]. These differences can be attributed to the size and semantic complexity of the dataset. BoW and Word2Vec tend to perform better with larger datasets [31] and those with complex semantic structures, as they can effectively leverage statistical and contextual relationships. Conversely, TF-IDF is more

effective for smaller and simpler datasets, such as the one used in this study. For example, research has shown that the TF-IDF approach enables classifiers to achieve higher accuracy when working with smaller datasets [32]. Therefore, the specific characteristics of the dataset and research objectives in this study created favorable conditions for TF-IDF to deliver superior results, consistent with the research hypothesis.

Table 4. Value of evaluation metrics and standard deviation

No.	Fitur Extraction	Accuracy		Precision		Recall		F1-Score	
		Mean	Std. Deviation	Mean	Std. Deviation	Mean	Std. Deviation	Mean	Std. Deviation
1.	TF-IDF	0.790	0.025	0.787	0	0.787	0	0.786	0
2.	*Bag of Words*	0.777	0.026	0.779	0.024	0.777	0.026	0.777	0.026
3.	*Word to Vector*	0.669	0.027	0.673	0.025	0.669	0.028	0.669	0.028

Furthermore, from Table 4, a further process can be carried out to determine how significant the difference in performance is brought about using these three types of feature extraction methods on the performance of SVM-based sentiment classification. The ANOVA approach, as one of the statistical-based approaches, can be used to statistically assess how significant the differences are from the application of TF-IDF, Bag of Words, and Word to Vector on the classification results using SVM. Table 5 shows the F-Statistic and P-Value from the further analysis on the evaluation metrics and standard deviation from Table 4. F-Statistics measures the ratio of variance between the models to the variance within the models. A high F-statistic would indicate significant differences between groups. While P-Value indicates whether the observed differences are statistically significant. A P-Value below a significance level (typically 0.05) would indicate that there are statistically significant differences between the models.

Table 5. ANOVA result

No.	Evaluation Metrics	F-Statistic	P-Value
1.	Accuracy	0.0081	0.9920
2.	Precision	0.0059	0.9942
3.	Recall	0.0061	0.9939
4.	F1-Score	0.0061	0.9939

The p-values reveal that none of the models show a significant difference in terms of accuracy, precision, recall, or F1-score, suggesting that they perform similarly in these areas. Furthermore, the low F-statistics confirm that the variance between the models is almost negligible. Since the ANOVA results point to no meaningful statistical differences, the choice of model may not heavily influence the outcomes. Consequently, it might be more practical to base the decision on factors like computational efficiency, user-friendliness, or the clarity of the model's interpretations.

CONCLUSION

From the research conducted, it was found that the SVM method was successfully applied to three feature extraction techniques, namely TF-IDF, BoW, and Word2Vec for text classification. Each of these methods proved effective in training the SVM model. The implementation process involved several main steps, including data preprocessing, feature extraction, and training the SVM model with the extracted features. The results showed that TF-IDF and BoW performed well, with SVM showing higher accuracy, while Word2Vec gave slightly lower results in terms of accuracy. Overall, a comparison of the accuracy, precision, recall, and f1-score values showed that TF-IDF with SVM gave the best performance, followed by BoW with slightly lower results, and Word2Vec with the lowest performance, although still quite good in some cases. The evaluation was conducted across several scenarios, including tests on balanced and imbalanced datasets, varying dataset sizes, and different CCC parameter values for SVM. These scenarios provided insights into how class distribution and parameter tuning influenced the model's performance. The findings suggest that TF-IDF with SVM is particularly robust in delivering consistent results, especially in datasets with

characteristics similar to those used in this study. However, further experimentation with larger and more diverse datasets is recommended to validate the generalizability of these conclusions.

REFERENCES

[1] P. Nandwani and R. Verma, "A review on sentiment analysis and emotion detection from text," Soc Netw Anal Min, vol. 11, no. 1, p. 81, Dec. 2021, https://doi.org/10.1007/s13278-021-00776-6

[2] K. Khadijah, N. Sabilly, and F. A. Nugroho, "Sentiment Analysis of League of Legends: Wild Rift Reviews on Google Play Using Naive Bayes Classifier," Jurnal Ilmiah Kursor, vol. 12, no. 1, pp. 23-30, Jul. 2023, https://doi.org/10.21107/kursor.v12i01.328

[3] R. T. Aldisa and P. Maulana, "Analisis Sentimen Opini Masyarakat Terhadap Vaksinasi Booster COVID-19 Dengan Perbandingan Metode Naive Bayes, Decision Tree dan SVM," Building of Informatics, Technology and Science (BITS), vol. 4, no. 1, pp. 106-109, Jun. 2022, https://doi.org/10.47065/bits.v4i1.1581

[4] M. Zoqi Sarwani, "Campus Sentiment Analyss E-Complaint Using Probabilistic Neural Network Algorithm," Jurnal Ilmiah Kursor, vol. 8, no. 3, p. 135, Mar. 2017, https://doi.org/10.28961/kursor.v8i3.88

[5] F. Handayani and M. Mustikasari, "Sentiment Analysis of Electric Cars Using Recurrent Neural Network Method in Indonesian Tweets," Jurnal Ilmiah Kursor, vol. 10, no. 4, Dec. 2020, https://doi.org/10.21107/kursor.v10i4.233

[6] M. S. Asramanggala, S. S. Prasetyowati, and Y. Sibaroni, "Optimal Number Data Trains in Hoax News Detection of Indonesian using SVM and Word2Vec," Building of Informatics, Technology and Science (BITS), vol. 5, no. 1, Jun. 2023, https://doi.org/10.47065/bits.v5i1.3516

[7] D. Darwis, E. S. Pratiwi, and A. F. O. Pasaribu, "Penerapan Algoritma SVM untuk Analisis Sentimen pada DataTwitter Komisi Pemberantasan Korupsi Republik Indonesia," Edutic - Scientific Journal of Informatics Education, vol. 7, no. 1, Nov. 2020, https://doi.org/10.21107/edutic.v7i1.8779

[8] F. Del et al., "Hate me, hate me not: Hate speech detection on Facebook," in Italian Conference on Cybersecurity, 2017. [Online]. Available: http://www.alexa.com/topsites

[9] G. Kovács, P. Alonso, and R. Saini, "Challenges of Hate Speech Detection in Social Media," SN Comput Sci, vol. 2, no. 2, p. 95, Apr. 2021, https://doi.org/10.1007/s42979-021-00457-3

[10] S. Saifullah, Y. Fauziyah, and A. S. Aribowo, "Comparison of machine learning for sentiment analysis in detecting anxiety based on social media data," Jurnal Informatika, vol. 15, no. 1, p. 45, Feb. 2021, https://doi.org/10.26555/jifo.v15i1.a20111

[11] T. Ahmed Khan, R. Sadiq, Z. Shahid, M. M. Alam, and M. Mohd Su'ud, "Sentiment Analysis using Support Vector Machine and Random Forest," Journal of Informatics and Web Engineering, vol. 3, no. 1, pp. 67-75, Feb. 2024, https://doi.org/10.33093/jiwe.2024.3.1.5

[12] A. Rahman Isnain, A. Indra Sakti, D. Alita, and N. Satya Marga, "Sentimen Analisis Publik terhadap Kebijakan Lockdown Pemerintah Jakarta Menggunakan Algoritma SVM," JDMSI, vol. 2, no. 1, pp. 31-37, 2021, https://doi.org/10.33365/jdmsi.v2i1.1021

[13] N. A. Semary, W. Ahmed, K. Amin, P. Pławiak, and M. Hammad, "Enhancing machine learning-based sentiment analysis through feature extraction techniques," PLoS One, vol. 19, no. 2, p. e0294968, Feb. 2024, https://doi.org/10.1371/journal.pone.0294968

[14] A. Yodi Prayoga, A. Id Hadiana, and F. Rakhmat Umbara, "Deteksi Hoax pada Berita Online Bahasa Inggris Menggunakan Bernoulli Naïve Bayes dengan Ekstraksi Fitur Tf-Idf," Jurnal Syntax Admiration, vol. 2, no. 10, pp. 1808-1823, Oct. 2021, https://doi.org/10.46799/jsa.v2i10.327

[15] A. A. Firdaus, A. Id Hadiana, and A. K. Ningsih, "Klasifikasi Sentimen pada Aplikasi Shopee Menggunakan Fitur Bag of Word dan Algoritma Random Forest," R2J, vol. 6, no. 5, 2024, https://doi.org/10.38035/rrj.v6i5

[16] A. Nurdin, B. Anggo, S. Aji, A. Bustamin, and Z. Abidin, "Perbandingan Kinerja Word Embedding Word2Vec, Glove, dan FastText pada Klasifikasi Teks," Jurnal TEKNOKOMPAK, vol. 14, no. 2, p. 74, 2020, https://doi.org/10.33365/jtk.v14i2.732

[17] L. Efrizoni, S. Defit, M. Tajuddin, and A. Anggrawan, "Komparasi Ekstraksi Fitur dalam Klasifikasi Teks Multilabel Menggunakan Algoritma Machine Learning," MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer, vol. 21, no. 3, pp. 653-666, Jul. 2022, https://doi.org/10.30812/matrik.v21i3.1851

[18] A. N. Syafia, M. F. Hidayattullah, and W. Suteddy, "Studi Komparasi Algoritma SVM Dan Random Forest Pada Analisis Sentimen Komentar Youtube BTS," vol. 8, no. 3, 2023, https://doi.org/10.30591/jpit.v8i3.5064

[19] D. Darwis, E. S. Pratiwi, A. Ferico, and O. Pasaribu, "Penerapan Algoritma SVM untuk Analisis Sentimen pada DataTwitter Komisi Pemberantasan Korupsi Republik Indonesia," 2020. https://doi.org/10.21107/edutic.v7i1.8779

[20] T. M. F. A. N. Jeremy Andre Septian, "Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor," 2019. [Online]. Available: https://t.co/9WloaWpfD5

[21] D. Sugiarto, E. Utami, and A. Yaqin, "Perbandingan Kinerja Model TF-IDF dan BOW untuk Klasifikasi Opini Publik Tentang Kebijakan BLT Minyak Goreng".

[22] M. R. Faisal, "Ekstraksi Fitu Menggunakan Model Word2vec Untuk Analisis Sentimen Pada Komentar Facebook." [Online]. Available: https://www.researchgate.net/publication/343057288

[23] S. Sivakumar, L. S. Videla, T. Rajesh Kumar, J. Nagaraj, S. Itnal, and D. Haritha, "Review on Word2Vec Word Embedding Neural Net," in 2020 International Conference on Smart Electronics and Communication (ICOSEC), IEEE, Sep. 2020, pp. 282-290. https://doi.org/10.1109/ICOSEC49089.2020.9215319

[24] S. Khairunnisa, A. Adiwijaya, and S. Al Faraby, "Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19)," Jurnal Media Iinformatika BUDIDARMA, vol. 5, no. 2, p. 406, Apr. 2021, https://doi.org/10.30865/mib.v5i2.2835

[25] S. Hardiristanto et al., "Determining the Abnormality of Bull Sperm Tail Morphology Using Support Vector," 2013.

[26] A. Asri and W. Setiawan, "Multiple Discriminant Analysis with Fukunaga Koontz Transfor and Support Vector Machine for Image-Based Face Detection and Recognition," 2013.

[27] G. Rininda, I. H. Santi, and S. Kirom, "Penerapan SVM dalam Analisis Sentimen pada EdLink Menggunakan Pengujian Confusion Matrix," 2023. https://doi.org/10.36040/jati.v7i5.7420

[28] N. Nurainun, E. Haerani, F. Syafria, and L. Oktavia, "Penerapan Algoritma Naïve Bayes Classifier Dalam Klasifikasi Status Gizi Balita dengan Pengujian K-Fold Cross Validation," Journal of Computer System and Informatics (JoSYC), vol. 4, no. 3, pp. 578-586, May 2023, https://doi.org/10.47065/josyc.v4i3.3414

[29] J. Lilleberg, Y. Zhu, and Y. Zhang, "Support vector machines and Word2vec for text classification with semantic features," in 2015 IEEE 14th International Conference on Cognitive Informatics &

Cognitive Computing (ICCI*CC), IEEE, Jul. 2015, pp. 136-140, https://doi.org/10.1109/ICCI-CC.2015.7259377

[30] L. Zhu, G. Wang, and X. Zou, "A Study of Chinese Document Representation and Classification with Word2vec," in 2016 9th International Symposium on Computational Intelligence and Design (ISCID), IEEE, Dec. 2016, pp. 298-302, https://doi.org/10.1109/ISCID.2016.1075

[31]D. E. Cahyani and I. Patasik, "Performance comparison of TF-IDF and Word2Vec models for emotion text classification," Bulletin of Electrical Engineering and Informatics, vol. 10, no. 5, pp. 2780-2788, Oct. 2021, Accessed: Dec. 04, 2024, https://doi.org/10.11591/eei.v10i5.3157

[32]R. Dzisevic and D. Sesok, "Text Classification using Different Feature Extraction Approaches," in 2019 Open Conference of Electrical, Electronic and Information Sciences (eStream), IEEE, Apr. 2019, pp. 1-4, https://doi.org/10.1109/eStream.2019.8732167