BAREKENG: Journal of Mathematics and Its Applications
June 2025 Volume 19 Issue 2 Page 1037–1046
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol19iss2pp1037-1046

SENTIMENT ANALYSIS OF REVIEWS ON 𝕏 APPS ON GOOGLE PLAY STORE USING SUPPORT VECTOR MACHINE AND N-GRAM FEATURE SELECTION

Fahri Aimar Kusumo[1*], Dewi Retno Sari Saputro[2], Purnami Widyaningsih[3]

[1,2,3]*Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Sebelas Maret*
Jln. Ir Sutami No. 36, Surakarta, 57126, Indonesia

*Corresponding author's e-mail: * **fahriaimark@student.uns.ac.id***

ABSTRACT

Article History:

Received: 31st August 2024
Revised: 4th February 2025
Accepted: 5th March 2025
Published: 1st April 2025

Keywords:

AUC;
Feature Selection;
N-gram;
Sentiment Analysis;
SVM.

Sentiment analysis is an application of text mining that is used to find out opinions from a set of textual data about a particular event or topic. The main function of sentiment analysis is to extract information and find the meaning and opinions of a given user. Sentiment analysis requires classification algorithms, such as Support Vector Machine (SVM). SVM is a frequently used algorithm for text data classification because it can handle high-dimensional data. The concept of SVM is to determine the best hyperplane that serves as a separator of two classes in the input space. Text data with a large number of features causes data imbalance and affects the classification process so it is necessary to do feature selection. Feature selection is a technique used to reduce irrelevant attributes in the dataset. N-gram feature selection is a statistics-based approach to classifying text. N-grams are able to classify unknown text with the highest certainty. The characteristics of N-grams in sentiment analysis are that they function well despite textual errors, run efficiently, require simple storage, and fast processing time. This research aims to perform sentiment analysis on 𝕏 application reviews on the Google Play Store with SVM and unigram, bigram, and trigram feature selection. The methodology of this research includes conducting theoretical studies, web scraping, text preprocessing, labeling sentiments with VADER, weighting with TF-IDF, dividing data into training data (80%) and testing data (20%), training and evaluating models, classifying testing data, and interpreting results. Based on the research results, 3151 testing data were classified. SVM classification and unigram feature selection have the highest accuracy value of 90% and AUC of 0.93 (excellent). SVM classification and bigram feature selection have an accuracy value of 78% with an AUC value of 0.81 (good). SVM classification and trigram feature selection had the lowest accuracy value of 68% with an AUC value of 0.66 (poor).

How to cite this article:

F. A. Kusumo, D. R. S. Saputro and P. Widyaningsih., "SENTIMENT ANALYSIS OF REVIEWS ON 𝕏 APPS ON GOOGLE PLAY STORE USING SUPPORT VECTOR MACHINE AND N-GRAM FEATURE SELECTION," *BAREKENG: J. Math. & App.*, vol. 19, iss. 2, pp. 1037-1046, June, 2025.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

The development of the internet has now become the most important part of everyday life in the world, including in Indonesia. The increase in the percentage of internet users from year to year is one of the indicators. Social media is the most frequently accessed internet service by Indonesians, one of which is Twitter [1]. Currently, Twitter's social media logo was changed to 𝕏 by the largest shareholder, Elon Musk. The World of Statistics [2] account released data on the most 𝕏 social media users from various countries in the world. social media 𝕏 has a total of 450 million users and is dominated by the United States, with 95.4 million users.

Every application has reviews and ratings, which are meant to give consumers confidence and benefits. User reviews can take the shape of recommendations, grievances, or critiques. The collection and sorting of these reviews is referred to as text mining. According to [3], the proper method to collect these reviews is the web scraping method. Text mining is a discipline that aims to process text into information obtained from forecasting patterns and trends [4].

Sentiment analysis is a text mining application used to find out opinions from a set of textual data about a particular event or topic [5]. Sentiment analysis is the key to artificial intelligence development of all research related to extracting data information from text [6]. The main function of sentiment analysis is to extract information and find the meaning and opinions of a given user. Sentiment analysis requires classification algorithms, such as support vector machine (SVM) [7]. SVM is one of the algorithms that is often used for text data classification because it can handle high-dimensional data [8]. In general, the concept of SVM is to determine the best hyperplane that serves as a separator of two classes in the input space. SVM determines the hyperplane value by using the support vector and margin value. Margin is the distance between the separator and the closest data from each class. The best hyperplane has the most significant margin. The data closest to the hyperplane is called the support vector [9]. In classifying text data, there are many irrelevant attributes. Feature selection is a technique used to reduce irrelevant attributes in a dataset. In feature selection, N-grams are known as unigram, bigrams, trigrams, and so on [10].

Research using N-grams was conducted by Nugroho in 2018 [11] with the application of N-grams extraction for sentiment analysis on Twitter social media using Naïve Bayes Classifier. The study concluded that the Naïve Bayes algorithm achieved an accuracy rate of 89.67%, and the effect of N-gram feature extraction could increase accuracy by 2.33% so that it became 92.00% in classifying Twitter social media. In 2020, Fitriyani and Arifin [12] conducted research on the application of Word N-gram can be used to analyze sentiment with the SVM method on the Sambara Application. From the results of the study, it was concluded that the highest accuracy value was 89% with an area under curve (AUC) value of 0.944 on a lot of data of 900, but when the application of bigrams and trigrams resulted in a decrease in accuracy. The accuracy value with the highest increase is in the application of trigrams with a lot of data of 1,200. The increase in accuracy value is 0.92% compared to unigram to 88.59% with an AUC value of 0.954.

Combining SVM with N-gram feature selection is necessary to reduce irrelevant attributes in the dataset [13]. The application of N-gram feature selection has several advantages; namely it works well despite textual errors, runs efficiently, requires simple storage, and has a fast processing time [14]. Thus, in this research, sentiment analysis with SVM and N-gram feature selection on 𝕏 app reviews on the Google Play Store app are conducted.

2. RESEARCH METHODS

The data used in this research is secondary data from 𝕏 app reviews on the Google Play Store [15]. The data is collected through a web scraping process with a range of time from July 2023 to August 2023 with the number of 𝕏 application user reviews totaling 15,757 review data. The web scraping technique is a method used to gather semi-structured information or data from websites, typically in the form of web pages written in markup languages like HTML or XHTML. This process involves extracting specific information or data from these pages for the purpose of analysis [3].

The user reviews in this research are classified into positive and negative sentiments. However, these reviews are still in the form of unstructured text without any clear meaning, so an analysis is needed to classify them into positive and negative sentiments based on the rating assigned to each review. The process of analyzing and exploring this unstructured data to identify patterns and extract meaningful information requires the use of a technology known as text mining [4]. During the text mining process, the first step is to prepare the document text or raw data set, which is referred to as text preprocessing. The objective of text preprocessing is to transform raw, unstructured text data into a structured format, thereby ensuring optimal performance during classification [16]. In general, the stages involved in the text preprocessing can be executed as follows [17].

1. Case Folding is the stage of homogenizing letters. Capital letters will be uniform in lowercase form. Examples of case folding results are shown in Table 1.

Table 1. Result of Case Folding Stage

Review	Case Folding
It used to be good, but now it sucks. Every good feature has either been removed or paywalled. The features left don't work correctly and the app has frequent and severe technical errors. Bugs stay up for weeks and months and most often are just left. The video never worked great but now is even worse. Don't waste your time getting on a sinking ship.	it used to be good but now it sucks every good feature has either been removed or paywalled the features left dont work correctly and the app has frequent and severe technical errors bugs stay up for weeks and months and most often are just left the video never worked great but now is even worse dont waste your time getting on a sinking ship

2. Tokenizing is the stage of cutting documents into small fragments in the form of chapters, subchapters, paragraphs, sentences, and words (tokens). Examples of tokenizing results are shown in Table 2.

Table 2. Result of Tokenizing Stage

Case Folding	Tokenizing
it used to be good but now it sucks every good feature has either been removed or paywalled the features left dont work correctly and the app has frequent and severe technical errors bugs stay up for weeks and months and most often are just left the video never worked great but now is even worse dont waste your time getting on a sinking ship	‘it’, ‘used’, ‘to’, ‘be’, ‘good’, ‘but’, ‘now’, ‘it’, ‘sucks’, ‘every’, ‘good’, ‘feature’, ‘has’, ‘either’, ‘been’, ‘removed’, ‘or’, ‘paywalled’, ‘the’, ‘features’, ‘left’, ‘dont’, ‘work’, ‘correctly’, ‘and’, ‘the’, ‘app’, ‘has’, ‘frequent’, ‘and’, ‘severe’, ‘technical’, ‘errors’, ‘bugs’, ‘stay’, ‘up’, ‘for’, ‘weeks’, ‘and’, ‘months’, ‘and’, ‘most’, ‘often’, ‘are’, ‘just’, ‘left’, ‘the’, ‘video’, ‘never’, ‘worked’, ‘great’, ‘but’, ‘now’, ‘is’, ‘even’, ‘worse’, ‘dont’, ‘waste’, ‘your’, ‘time’, ‘getting’, ‘on’, ‘a’, ‘sinking’, ‘ship’

3. Filtering is the stage of selecting words in the document. Less relevant words are filtered using stopwords. Stopwords are words that appear in large numbers and are irrelevant. Examples of filtering results are shown in Table 3.

Table 3. Result of Filtering Stage

Tokenizing	Filtering
‘it’, ‘used’, ‘to’, ‘be’, ‘good’, ‘but’, ‘now’, ‘it’, ‘sucks’, ‘every’, ‘good’, ‘feature’, ‘has’, ‘either’, ‘been’, ‘removed’, ‘or’, ‘paywalled’, ‘the’, ‘features’, ‘left’, ‘dont’, ‘work’, ‘correctly’, ‘and’, ‘the’, ‘app’, ‘has’, ‘frequent’, ‘and’, ‘severe’, ‘technical’, ‘errors’, ‘bugs’, ‘stay’, ‘up’, ‘for’, ‘weeks’, ‘and’, ‘months’, ‘and’, ‘most’, ‘often’, ‘are’, ‘just’, ‘left’, ‘the’, ‘video’, ‘never’, ‘worked’, ‘great’, ‘but’, ‘now’, ‘is’, ‘even’, ‘worse’, ‘dont’, ‘waste’, ‘your’, ‘time’, ‘getting’, ‘on’, ‘a’, ‘sinking’, ‘ship’	‘used’, ‘good’, ‘sucks’, ‘every’, ‘good’, ‘feature’, ‘either’, ‘removed’, ‘paywalled’, ‘features’, ‘left’, ‘dont’, ‘work’, ‘correctly’, ‘app’, ‘frequent’, ‘severe’, ‘technical’, ‘errors’, ‘bugs’, ‘stay’, ‘weeks’, ‘months’, ‘often’, ‘left’, ‘video’, ‘never’, ‘worked’, ‘great’, ‘even’, ‘worse’, ‘dont’, ‘waste’, ‘time’, ‘getting’, ‘sinking’, ‘ship’

4. Stemming is the stage of the process of finding the basic word in English by removing affixes. Examples of stemming results are shown in Table 4.

Table 4. Result of Stemming Stage

Filtering	Stemming
'used', 'good', 'sucks', 'every', 'good', 'feature', 'either', 'removed', 'paywalled', 'features', 'left', 'dont', 'work', 'correctly', 'app', 'frequent', 'severe', 'technical', 'errors', 'bugs', 'stay', 'weeks', 'months', 'often', 'left', 'video', 'never', 'worked', 'great', 'even', 'worse', 'dont', 'waste', 'time', 'getting', 'sinking', 'ship'	use good suck everi good featur either remov paywal featur left dont work correctli app frequent sever technic error bug stay week month often left video never work great even wors dont wast time get sink ship

Following the completion of the text preprocessing stage, it is essential to model the data to facilitate the processing and computation of the information, which remains in a textual format. This is achieved by transforming the data into vectors, where each word is assigned a specific value and weight. The present research employs the TF-IDF weighting method, which serves to establish the relationship of words (terms) within each document in the corpus [8]. The formula for TF-IDF is written as

$$W_{dt} = (tf)_{dt} \times \ln\left(\frac{N}{(df)_t}\right) \tag{1}$$

Sentiment analysis is a text mining application used to determine the opinion of a set of textual data about a particular event or topic. Sentiment analysis involves examining reviews or opinion trends regarding a particular issue or object, irrespective of whether the opinions expressed are negative or positive [5]. The VADER lexicon is employed to automatically categorize sentiment classes in English review data by assessing the intensity of emotional strength based on the existing lexicon data dictionary [18][19].

The data labeled with sentiment categories will subsequently be divided into training and testing sets, following an 80:20 split. In this research, the amount of training data is 80% or 12,602 data, and test data is 20% or 3151 data. Then, SVM is used to separate between positive sentiment and negative sentiment. SVM is a supervised learning method by determining the best hyperplane that serves as a separator of two classes in the input space [9]. SVM determines the hyperplane value by using the support vector and margin value. The margin is the distance between the separation and the closest data from each class. The best hyperplane has the largest margin. The data closest to the hyperplane is called the support vector. It is assumed that the two classes can be perfectly separated by a hyperplane in the D-dimensional feature space. Feature space is a set of data that has been selected and transformed to define a hyperplane. The hyperplane on two-dimensional data is written as [20]:

$$\boldsymbol{w} \cdot \boldsymbol{x_i} + \boldsymbol{b} = \boldsymbol{0} \tag{2}$$

Data x_i belonging to the negative class satisfies the inequality

$$\boldsymbol{w} \cdot \boldsymbol{x_i} + \boldsymbol{b} \leq -\boldsymbol{1} \tag{3}$$

The x_i data that belongs to the positive class satisfies the inequality

$$\boldsymbol{w} \cdot \boldsymbol{x_i} + \boldsymbol{b} \geq \boldsymbol{1} \tag{4}$$

where $\boldsymbol{w}$ is the weight vector, x_i is the i-th data, and b is the bias.

The determination of the best hyperplane in SVM is visualized in Figure 1, which shows the separation of the data depicted in blue (circle) and orange (plus symbol).

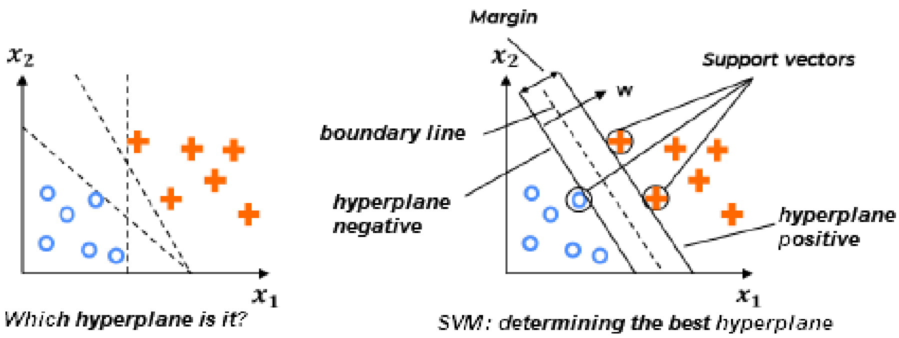


Figure 1. Hyperplane Separating Positive and Negative Classes

The largest margin is calculated by maximizing the distance between the hyperplane and the closest data, which is $\frac{1}{\|\boldsymbol{w}\|}$ where $\|\boldsymbol{w}\|$ is the norm of the weight vector $\boldsymbol{w}$. The SVM formulation for the linear classification case is written as [20], [21]:

$$\min_{\boldsymbol{w},\boldsymbol{b}} \frac{1}{2}\|\boldsymbol{w}\|^2 \tag{5}$$

subject to

$$\boldsymbol{y_i}(\boldsymbol{w}\cdot\boldsymbol{x_i}+\boldsymbol{b}) \geq \mathbf{1}, \boldsymbol{i}=\mathbf{1},\ldots,\boldsymbol{m}.$$

In the infeasible case, there is data that is not classified correctly, then the SVM formulation is written as

$$\underset{\boldsymbol{w},\boldsymbol{b},\boldsymbol{t}}{\mathbf{Min}} \frac{1}{2}\|\boldsymbol{w}\|^2 + \boldsymbol{C}\sum_{\boldsymbol{i}=1}^{\boldsymbol{m}} \boldsymbol{t_i} \tag{6}$$

subject to

$$\boldsymbol{y_i}(\boldsymbol{w}\cdot\boldsymbol{x_i}+\boldsymbol{b}) + \boldsymbol{t_i} \geq \mathbf{1}, \boldsymbol{t_i} \geq \mathbf{0}, \boldsymbol{i}=\mathbf{1},\ldots,\boldsymbol{m}.$$

Equation (6) is used to minimize the error in classification. t_i is a slack variable used to overcome the infeasible case. The constant C is used to minimize t_i. In solving these cases, a Lagrange multiplier is needed to be more efficient and make computation easier, which is written as

$$\boldsymbol{L}(\boldsymbol{w},\boldsymbol{b},\boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{\boldsymbol{i}=1}^{\boldsymbol{N}} \boldsymbol{\alpha_i}\left(\boldsymbol{y_i}\big((\boldsymbol{w}\cdot\boldsymbol{x_i}+\boldsymbol{b})-\mathbf{1}\big)\right) \tag{7}$$

where $\boldsymbol{w}$ is the weight vector, x_i is the i-th data, b is the bias, y_i is the i-th data class, and α_i is the Lagrange multiplier, which is zero or positive. The optimal value of **Equation (7)** can be calculated by minimizing L with $\boldsymbol{w}$ and b, and maximizing L with α. By modifying **Equation (7)**, the maximization problem is represented in α_i by applying the constraint $\alpha_i \geq 0, \sum_{i=1}^{N} \alpha_i y_i$ which is written as:

$$\sum_{\boldsymbol{i}=1}^{\boldsymbol{N}} \boldsymbol{\alpha_i} - \frac{1}{2}\sum_{\boldsymbol{i},\boldsymbol{j}=1}^{\boldsymbol{N}} \boldsymbol{\alpha_i}\cdot\boldsymbol{\alpha_j}\cdot\boldsymbol{y_i}\cdot\boldsymbol{y_j}\cdot\boldsymbol{x_i}\cdot\boldsymbol{x_j} \tag{8}$$

The result of **Equation (8)** produces many α_i with positive values. The data that corresponds to α_i that is not zero is the support vector. The formula provided illustrates how to determine the hyperplane that acts as the boundary between two classes. Furthermore, in this research, N-gram feature selection is carried out with three approaches, namely unigram, bigrams, and trigrams. An N-gram is characterized as a sequence of elements of length n, which may consist of either characters or words. N-grams serve as a fundamental technique for the categorization of text. The variable N represents the number of keywords utilized to segment the input text. N-grams possess the capability to classify unfamiliar text with a high degree of accuracy [22].

Table 5. Confusion matrix

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

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{9}$$

Table 5 above is used to calculate the accuracy written in **Equation (9)** and the area under the curve (AUC) written in **Equation (10)**. Accuracy is the degree of closeness between the classification result and the actual [23]. AUC serves as an evaluative metric that relies on sensitivity or specificity for its measurement basis [24]. According to [25], five classification categories based on AUC values are shown in **Table 6**.

Table 6. Classification Categories Based on AUC values

AUC values	**Classification categories**
0.9 – 1.0	Excellent
0.8 – 0.9	Good
0.7 – 0.8	Fair
0.6 – 0.7	Poor
0.5 – 0.6	Failure

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \tag{10}$$

with

$$TP_{rate} = \frac{TP}{TP+FN},$$

$$FP_{rate} = \frac{FP}{FP+TN},$$

TP_{rate} is the ratio of positive correct predictions compared to all positive actual data and FP_{rate} is the ratio of positive false predictions compared to all negative actual data.

This research uses SVM and N-gram feature selection with three keywords: unigram, bigram, and trigram. Then, in this research, accuracy and AUC are used to compare the classification performance of the three keywords. The stages of analysis utilized in this research are illustrated in Figure 2.

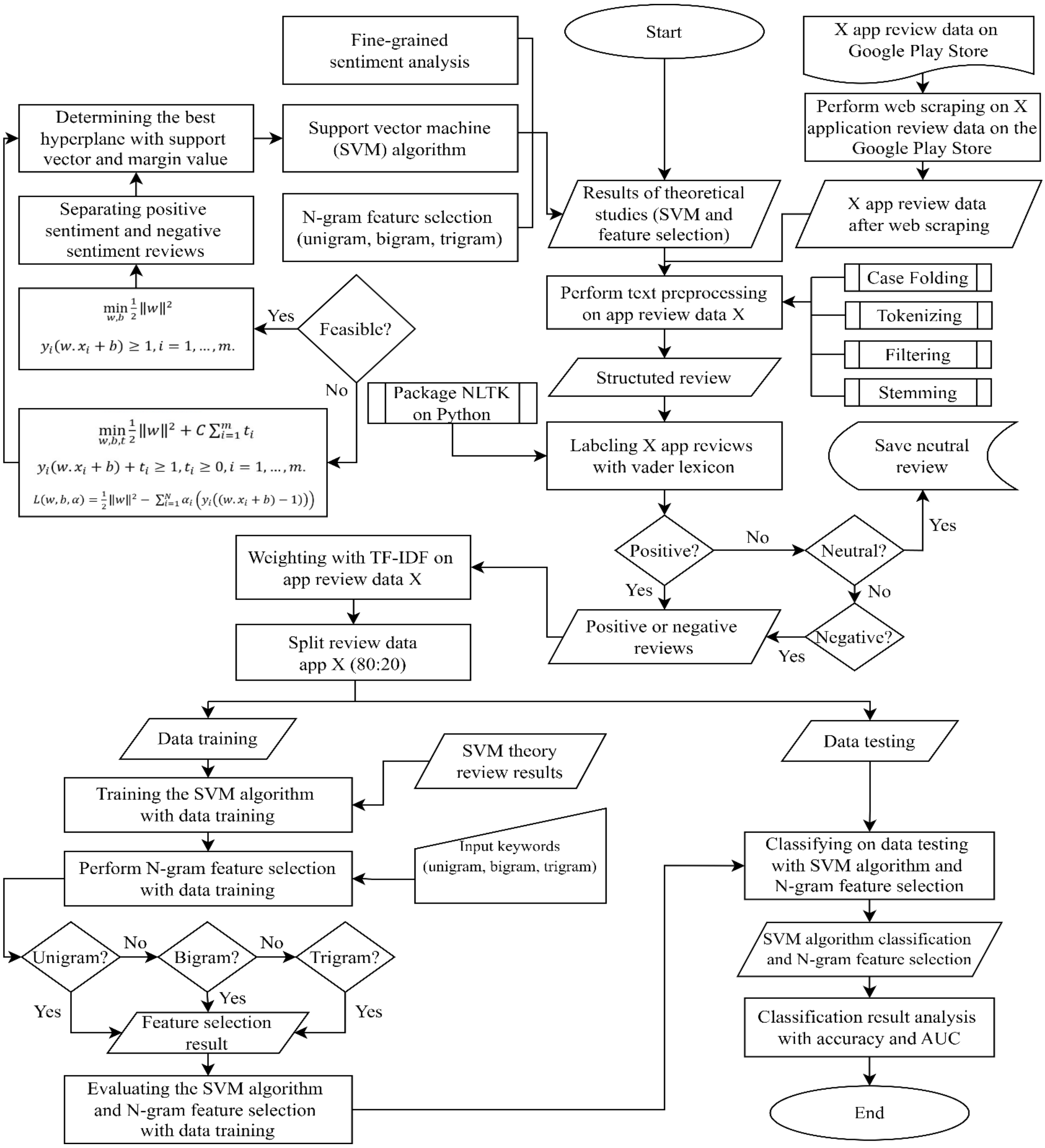


Figure 2. Flowchart of Research Steps

3. RESULTS AND DISCUSSION

The results and discussion are divided into seven sections, namely text preprocessing, sentiment analysis, data labeling, TF-IDF weighting, and classification, with each description stated as follows.

3.1 Text Preprocessing

Text preprocessing is essential for standardizing word spellings and letter forms, minimizing vocabulary size, and removing extraneous characters that lack significance, thereby enhancing the structure and informativeness of the data. This phase involves various data-cleaning techniques, including case folding, tokenizing, filtering, and stemming. The outcomes of text preprocessing are presented in **Table 7** below.

Table 7. Simulation of Text Preprocessing

Review	Text Preprocessing Results
It used to be good, but now it sucks. Every good feature has either been removed or paywalled. The features left don't work correctly and the app has frequent and severe technical errors. Bugs stay up for weeks and months and most often are just left. The video never worked great but now is even worse. Don't waste your time getting on a sinking ship.	use good suck everi good featur either remov paywal featur left dont work correctli app frequent sever technic error bug stay week month often left video never work great even wors dont wast time get sink ship

3.2 Sentiment Analysis

This research employs the VADER lexicon library to automate the sentiment categorization of English review data. The sentiment analysis will classify the reviews into two categories: positive and negative sentiments. Following the categorization process, the comparative analysis of the number of reviews classified as positive versus those classified as negative is presented in **Figure 3** below.

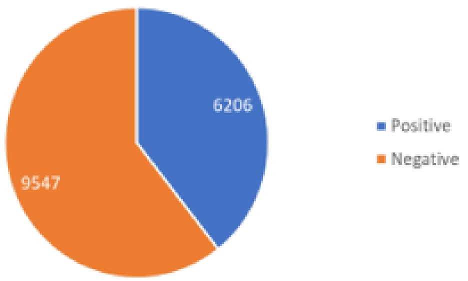


Figure 3. Number of Review Data in Sentiment Class

3.3 Data Labeling

Following the analysis of the reviewed data that has undergone text preprocessing, it is possible to conduct a simulation for calculating sentiment scores. The formula employed for this sentiment score calculation is as follows [25]:

$$Score = (number\ of\ positive\ words) - (number\ of\ negative\ words)$$

Should the outcome of the calculation simulation exceed zero, the classification of the review will indicate a positive sentiment. The results of the sentiment score calculation simulation are presented in **Table 8**.

Table 8. Sentiment Score Calculation Simulation

Review	Score	Sentiment Label
use good suck everi good featur either remov paywal featur left dont work correctli app frequent sever technic error bug stay week month often left video never work great even wors dont wast time get sink ship	1 – 4 = -3	Negative

3.4 TF-IDF Weighting

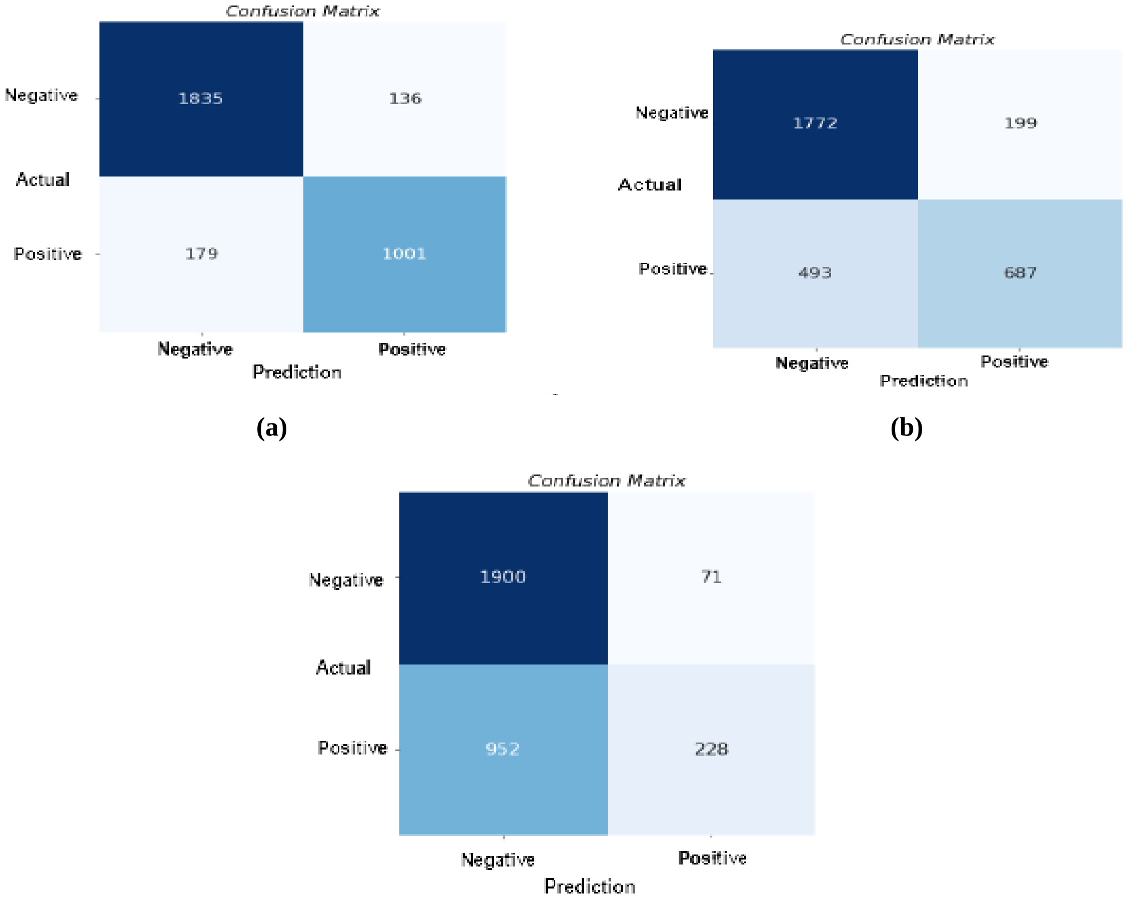
Weighting the results of text preprocessing with TF-IDF is used to determine the relationship of words (terms) in each document in the corpus. The TF value is calculated based on the number of occurrences of a word in the data divided by the number of words in the data. The more often the word appears in the data, the greater the TF value. The TF value does not always reflect how important the word is in the data because words that appear frequently in the data are not always important. Therefore, IDF is used to reduce the weight of words that appear frequently in the document. The results of the TF-IDF calculation simulation are presented in **Table 9**.

Table 9. TF-IDF Calculation Simulation

Word	TF	IDF	TF-IDF	Word	TF	IDF	TF-IDF
use	0.027	2.080	0.056	error	0.027	4.902	0.132
good	0.054	2.390	0.129	bug	0.027	4.147	0.112
suck	0.027	3.655	0.098	stay	0.027	4.789	0.129
everi	0.027	3.373	0.091	week	0.027	5.445	0.147
featur	0.054	3.219	0.174	month	0.027	4.308	0.116
either	0.027	5.079	0.137	often	0.027	5.675	0.153
remov	0.027	3.887	0.105	video	0.027	3.200	0.086
paywal	0.027	4.222	0.114	never	0.027	3.575	0.096
left	0.054	4.964	0.268	great	0.027	3.098	0.083
dont	0.054	2.457	0.132	even	0.027	2.791	0.075
work	0.054	3.096	0.167	wors	0.027	3.150	0.085
correctli	0.027	7.099	0.191	wast	0.027	5.090	0.137
app	0.027	1.007	0.027	time	0.027	3.055	0.082
frequent	0.027	6.109	0.165	get	0.027	2.541	0.068
sever	0.027	5.752	0.155	sink	0.027	6.199	0.167
technic	0.027	6.445	0.174	ship	0.027	6.027	0.162

3.5 Classification

This research uses SVM and N-gram with three keywords, namely unigram, bigram, and trigram. Then, the dataset will be divided into two parts with a ratio of 80% as training data, or 12602 data, and 20% as testing data, or 3151 data. The dataset that has undergone text preprocessing will proceed to the learning phase utilizing SVM and unigram, bigram, and trigram. The classification results of SVM and unigram algorithms are evaluated by a confusion matrix using 3151 testing data for positive and negative sentiment classification shown in **Figure 4** (a). The classification results of SVM and bigram algorithms evaluated by the confusion matrix are shown in **Figure 4** (b). It appears that 2459 data were correctly classified out of 687 positive reviews that were correctly classified as positive and 1772 negative reviews that were correctly classified as negative. Furthermore, classification prediction is carried out using the learning algorithm.



(a) **(b)** **(c)**

Figure 4. The Classification Results of (a) Unigram, (b) Bigram, and (c) Trigram

The classification results of SVM and trigram algorithms evaluated by the confusion matrix are shown in **Figure 4** (c); it can be seen that 2128 data were correctly classified from 228 positive reviews that were correctly classified to positive and 1900 negative reviews that were correctly classified to negative.

From the results of the classification method used, the accuracy and AUC values are obtained. The classification results of the SVM algorithm with unigram, bigram, and trigram are shown in **Table 10**, each with its accuracy and AUC values.

Table 10. Comparison of SVM Results with Unigram, Bigram, and Trigram in Application 𝕏

Classification	Accuracy	AUC Values
SVM + unigram	90%	0.93
SVM + bigram	78%	0.81
SVM + trigram	68%	0.66

Based on **Table 10**, the highest accuracy and AUC value in this research is 90%, with an AUC value of 0.93 (excellent) when classification is carried out using SVM and unigram. SVM classification and bigram feature selection have an accuracy value of 78% with an AUC value of 0.81 (good). SVM classification and trigram feature selection had the lowest accuracy value of 68% with an AUC value of 0.66 (poor).

4. CONCLUSIONS

Based on the research results, a classification of 3151 testing data is obtained. SVM classification and unigram feature selection have the highest accuracy value of 90% and AUC of 0.93 (very good). SVM classification and bigram feature selection had an accuracy value of 78% with an AUC value of 0.81 (good). SVM classification and trigram feature selection had the lowest accuracy value of 68% with an AUC value of 0.66 (poor). SVM algorithms can only address binary classification. Therefore, readers who are interested in this topic can continue this research topic with the Multiclass Support Vector Machine (MSVM) algorithm. The MSVM algorithm is a technique developed from the SVM algorithm to solve multi-class problems, such as one-vs-one (OVO) and one-vs-rest (OVR) methods.

ACKNOWLEDGMENT

We express our sincere appreciation to the Soft Computing Mathematics Research Group for their moral support and invaluable input, which significantly improved the quality of this paper. In addition, we would also like to thank the Institute of Research and Community Service of Universitas Sebelas Maret (UNS) for the financial support through this research group grant.

REFERENCES

[1] S. Kemp, "DIGITAL 2023: INDONESIA," 2023. https://datareportal.com/reports/digital-2023-indonesia.
[2] World of Statistics, "TWITTER USERS," 2023. https://twitter.com/stats_feed/status/1661437849978417152?s=21.
[3] D. D. A. Yani, H. S. Pratiwi, and H. Muhardi, "IMPLEMENTASI WEB SCRAPING UNTUK PENGAMBILAN DATA PADA SITUS MARKETPLACE," *J. Sist. dan Teknol. Inf.*, vol. 7, no. 4, p. 257, 2019, doi: 10.26418/justin.v7i4.30930.
[4] J. Han, M. Kamber, and J. Pei, *DATA MINING: CONCEPTS AND TECHNIQUES*. 2011.
[5] B. Liu, "SENTIMENT ANALYSIS AND MINING OF OPINIONS," *Morgan Claypool Publ.*, 2012, doi: 10.1007/978-3-319-60435-0_20.
[6] E. Cambria, D. Das, and S. Bandyopadhay, A PRACTICAL GUIDE TO SENTIMENT ANALYSIS. Switzerland: Springer International Publishing, 2017.
[7] F. Aftab *et al.*, "A COMPREHENSIVE SURVEY ON SENTIMENT ANALYSIS TECHNIQUES," *Int. J. Technol.*, vol. 14, no. 6, pp. 1288–1298, 2023, doi: 10.14716/ijtech.v14i6.6632.
[8] N. Arifin, U. Enri, and N. Sulistiyowati, "PENERAPAN ALGORITMA SUPPORT VECTOR MACHINE (SVM) DENGAN TF-IDF N-GRAM UNTUK TEXT CLASSIFICATION," *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 6, no. 2, p. 129, 2021, doi: 10.30998/string.v6i2.10133.
[9] A. S. Nugroho, A. B. Witarto, and D. Handoko, "APPLICATION OF SUPPORT VECTOR MACHINE IN BIOINFORMATICS," 2003.
[10] G.-T. Tsvetanka and M. Duraku, "RESEARCH ON N-GRAMS FEATURE SELECTION METHODS FOR TEXT CLASSIFICATION," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1031, no. 1, 2021, doi: 10.1088/1757-899X/1031/1/012048.
[11] A. Nugroho, "ANALISIS SENTIMEN PADA MEDIA SOSIAL TWITTER MENGGUNAKAN NAIVE BAYES CLASSIFIER DENGAN EKSTRASI FITUR N-GRAM," *J-SAKTI (Jurnal Sains Komput. dan Inform.*, vol. 2, no. 2, p. 200, 2018, doi: 10.30645/j-sakti.v2i2.83.
[12] F. Fitriyani and T. Arifin, "PENERAPAN WORD N-GRAM UNTUK SENTIMENT ANALYSIS REVIEW MENGGUNAKAN METODE SUPPORT VECTOR MACHINE (STUDI KASUS: APLIKASI SAMBARA)," *Sistemasi*, vol. 9, no. 3, p. 610, 2020, doi: 10.32520/stmsi.v9i3.954.
[13] P. Koncz and J. Paralic, "AN APPROACH TO FEATURE SELECTION FOR SENTIMENT ANALYSIS," *INES 2011 - 15th Int. Conf. Intell. Eng. Syst. Proc.*, pp. 357–362, 2011, doi: 10.1109/INES.2011.5954773.
[14] K. O. Ogada, "N-GRAMS FOR TEXT CLASSIFICATION USING SUPERVISED MACHINE LEARNING ALGORITHMS," 2016.
[15] X Corp., "[Online]," *[Online]*. https://play.google.com/store/apps/details?id=com.twitter.android&pcampaignid=web_share.
[16] E. Haddi, X. Liu, and Y. Shi, "THE ROLE OF TEXT PRE-PROCESSING IN SENTIMENT ANALYSIS," *Procedia Comput. Sci.*, vol. 17, pp. 26–32, 2013, doi: 10.1016/j.procs.2013.05.005.
[17] A. K. Uysal and S. Gunal, "THE IMPACT OF PREPROCESSING ON TEXT CLASSIFICATION," *Inf. Process. Manag.*, vol. 50, no. 1, pp. 104–112, 2014, doi: 10.1016/j.ipm.2013.08.006.
[18] C. J. Hutto and E. Gilbert, "VADER: A PARSIMONIOUS RULE-BASED MODEL FOR SENTIMENT ANALYSIS OF SOCIAL MEDIA TEXt," *Eighth Int. AAAI Conf. Weblogs Soc. Media*, vol. 8, no. 1, pp. 216–225, 2014, [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14550.
[19] S. Elbagir and J. Yang, "LANGUAGE TOOLKIT AND VADER SENTIMENT," *Proc. Int. MultiConference Eng. Comput. Sci.*, vol. 0958, pp. 12–16, 2019.
[20] A. Kowalczyk, "SUPPORT VECTOR MACHINES SUCCINCTLY," p. 114, 2017, [Online]. Available: www.syncfusion.com.
[21] B. Santosa, "TUTORIAL SUPPORT VECTOR MACHINES," 2010, pp. 1–19.
[22] S. H. Yadav and P. M. Manwatkar, "AN APPROACH FOR OFFENSIVE TEXT DETECTION AND PREVENTION IN SOCIAL NETWORKS," *ICIIECS 2015 - 2015 IEEE Int. Conf. Innov. Information, Embed. Commun. Syst.*, pp. 3–6, 2015, doi: 10.1109/ICIIECS.2015.7193018.
[23] F. Gorunescu, DATA MINING: CONCEPTS, MODELS AND TECHNIQUES. Berlin, 2011.
[24] H. He and Y. Ma, *I*MBALANCED LEARNING: FOUNDATIONS, ALGORITHMS, AND APPLICATIONS. 2013.
[25] M. Bekkar, H. K. Djemaa, and T. A. Alitouche, "EVALUATION MEASURES FOR MODELS ASSESSMENT OVER IMBALANCED DATA SETS," *J. Inf. Eng. Appl.*, vol. 3, no. 10, pp. 27–38, 2013, [Online]. Available: http://www.iiste.org/Journals/index.php/JIEA/article/view/7633.