SisInfo
JURNAL SISTEM INFORMASI DAN INFORMATIKA

e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 8 No. 1 February 2026

Comparison of Chi-Square and Information Gain Feature Selection Methods for Support Vector Machine-Based Sentiment Analysis
(Case Study: Vidio Application Reviews on Google Play Store)

Vitta Margaret Sinambela, Herlina Napitupulu, Nurul Gusriani
Mathematics, Faculty of Mathematics and Natural Sciences, Padjadjaran University
Email: vitta21001@mail.unpad.ac.id; herlina@unpad.ac.id; nurul.gusriani@unpad.ac.id.

Accepted:	Accepted After Revision:	Published:
4 August 2025	24 February 2026	27 February 2026

Abstract

Vidio is a local streaming platform that dominates the Indonesian market, but still faces challenges in improving user satisfaction as reflected by its 3.5 rating. To enhance the application, user experience insights are needed, which can be identified through sentiment analysis. This study aims to analyze the sentiment of Vidio application user reviews and compare the performance of the Support Vector Machine model using Chi-Square and Information Gain feature selection. The dataset comprises 4,670 reviews collected from July 01 to November 30, 2024. Model evaluation utilizes Balanced Accuracy metrics optimized through hyperparameter tuning to ensure fair assessment on imbalanced data. The experimental results demonstrate that Chi-Square feature selection yields the optimal performance, achieving a peak Balanced Accuracy of 94.78%. Significantly, this result was attained using a computationally efficient Linear Kernel ($\mathrm{C} = 1$). In contrast, the Information Gain method yielded a lower Balanced Accuracy of 94.20% despite utilizing a complex Polynomial Kernel ($\mathrm{C} = 1, \mathrm{r} = -1, \mathrm{p} = 3, \gamma = 0.1$). These findings conclude that Chi-Square provides a superior trade-off between classification accuracy and model complexity, offering a more robust solution for sentiment analysis.

Keywords: Sentiment Analysis, Support Vector Machine, Chi-Square, Information Gain, Vidio.

1 INTRODUCTION

According to [1], the number of video streaming application user in Indonesia continues to increase, reflecting the growing role of streaming platforms in digital life. The local platform that dominates this market in Indonesia is Vidio [2]. The Vidio application has a rating of 3.5 from approximately 670 thousand users, indicating the need for further analysis to identify aspects that require improvement. To process a large volume of reviews, sentiment analysis is conducted to classify user reviews and identify complaint patterns as well as frequently highlighted negative aspects.

According to [3], classification using the Support Vector Machine (SVM) method is more accurate than other classifiers, performs well on nonlinear data, and has a low risk of overfitting. Machine learning methods can be combined with feature selection to reduce the number of features and improve model accuracy. A comparison of Chi-Square, Information Gain, and Principal Component Analysis feature selection methods found that Chi-Square and Information Gain achieved the highest scores [4]. The performance of the SVM model is influenced by hyperparameters. Hyperparameter tuning such as kernel function (k), regularization constant (C), gamma (γ), and polynomial degree (p) can improve SVM performance [5].

Sentiment analysis research on the Vidio application has been conducted previously. Applied the KNN method and obtained 70% accuracy through manual calculation and 50% accuracy using RapidMiner tools [6]. Decision Tree method and achieved the best accuracy of 97.3% with an 80:20 data split ratio [7]. Applied KNN and Naïve Bayes Classifier (NBC), showing that KNN performed better with an accuracy of 93%, recall of 88%, precision of 93%, and F1-Score of 90% compared to NBC with an accuracy of 81%, recall of 91%, precision of 81%, and F1-Score of 86% [8].

In previous sentiment analysis studies on the Vidio application, the review data used was imbalanced. To address this issue, the most appropriate evaluation metric is balanced accuracy [9]. However, balanced accuracy results have not been presented, and feature selection has not been applied, meaning all features were

used without considering their relevance to model performance. Therefore, this study applies the SVM classification method and feature selection by comparing the Chi-Square and Information Gain feature selection methods on Vidio user reviews. In addition, hyperparameter tuning based on balanced accuracy is performed using Grid Search to improve model performance.

2 LITERATURE REVIEW

2.1 Feature Selection

Feature selection is a machine learning technique used to select the minimum number of features required to represent data accurately, making computation more efficient. The use of relevant feature selection can improve accuracy, shorten the machine learning process by reducing dimensionality and removing noise, and produce simpler concepts [8]. In this study, a comparison of feature selection effectiveness is conducted. The feature selection methods compared for sentiment analysis are Chi-Square and Information Gain as individual feature selection techniques.

2.1.1 Chi-Square Feature Selection

According [10], Chi-Square feature selection aims to select features using Chi-Square statistical values to measure the dependency between words and their classes. The Chi-Square test function for a word against a category is determined using equation (1) [11],

$$\chi^2(t_u, c_k) = \frac{N\left(A_{c_k}D_{c_k} - C_{c_k}B_{c_k}\right)^2}{\left(A_{c_k}+C_{c_k}\right)\left(B_{c_k}+D_{c_k}\right)\left(A_{c_k}+B_{c_k}\right)\left(C_{c_k}+D_{c_k}\right)}, \quad (1)$$

where t_u is the u-th word, c_k is the k-th class, K is the number of classes, and N is the number of training documents. Meanwhile, A_{c_k} is the number of documents in class c_k containing t_u, B_{c_k} is the number of documents not in class c_k containing t_u, C_{c_k} is the number of documents I class c_k not containing t_u, and D_{c_k} is the number of documents not in class c_k not containing t_u.

The single Chi-Square value of a word is obtained by summing the Chi-Square values of the word across all K classes using equation (2):

$$\chi^2(t_u) = \sum_{k=1}^{K} \chi^2(t_u, c_k). \quad (2)$$

2.1.2 Information Gain Feature Selection

Information Gain feature selection identifies features that contain the most information based on a particular class. The best attribute is determined by calculating entropy. Entropy measures class uncertainty using the probability of occurrence of certain events or attributes. The entropy function before observing attribute A within classification classes is obtained using equation (3) [12]:

$$H(C) = -\sum_{k=1}^{K} P(c_k) \log_2 P(c_k), \quad (3)$$

where the c_k is the k-th class, K is the number of classes, and $P(c_k)$ is the probability of class c_k.

The entropy function after observing the u-th word is obtained using equation (4):

$$H(C|t_u) = \sum_{a \in values\ t_u} \left(P(a)\left(-\sum_{k=1}^{K} P(c_k|a) \log_2 P(c_k|a)\right)\right), \quad (4)$$

where a is the value of t_u, $P(a)$ is the probability of value a for attribute A, and $P(c_k|a)$ is the probability of class c_k given value a.

After obtaining entropy values, Information Gain for the u-th word can be calculated using equation (5):

$$IG(t_u) = H(C) - H(C|t_u). \quad (5)$$

2.2 Word Representation Using TF-IDF

Word representation is performed by transforming textual data into vectors so that it can be processed by machine learning models. [13] stated that Term Frequency–Inverse Document Frequency (TF-IDF) is a feature extraction method in which word frequencies are recalculated by considering how frequently those words appear across all documents. TF-IDF is obtained using the following equations:

$$\text{TF}(t_u, d_i) = \frac{n_{d_i,t_u}}{N_{d_i}}, \quad (6)$$

$$\text{IDF}(t_u) = \log \frac{N}{df(t_u)+1}, \quad (7)$$

$$\text{TF-IDF}(t_u, d_i) = \text{TF}(t_u, d_i) \times \text{IDF}(t_u), \quad (8)$$

where n_{d_i,t_u} is the number of occurrences of the u-th word in the i-th document, N_{d_i} is the total number of words in the i-th document, and $df(t_u)$ is the number of documents containing the u-th word. The resulting TF-IDF vector is then normalized using the Euclidean norm as shown in equation (9):

$$\text{TF-IDF}_{\text{norm}}(t_u, d_i) = \frac{\text{TF-IDF}(t_u, d_i)}{\sqrt{\text{TF-IDF}(t_1, d_i)^2 + \text{TF-IDF}(t_2, d_i)^2 + \cdots + \text{TF-IDF}(t_m, d_i)^2}}, \quad (9)$$

where $u = 1,2,3, \ldots, m$.

The resulting TF-IDF vector is then used as input for the Support Vector Machine classification method.

2.3 Word Representation Using TF-IDF

Support Vector Machine is a statistical method used for classification. SVM is a supervised learning method that analyzes data and recognizes patterns, used for classification and regression analysis [14]. According to [15], SVM aims to find the optimal separating function (hyperplane) that divides one class from another. The optimal hyperplane is determined by maximizing the distance (margin) between the hyperplane and the nearest data points (support vectors) from each class. The components of SVM include the optimal hyperplane, positive class hyperplane, negative class hyperplane, and margin.

The optimal hyperplane is defined in equation (10) [16]:

$$(\mathrm{w} \cdot \mathrm{x}) + b = 0, \quad (10)$$

where x is the input data vector, w is the weight vector, and b is a scalar bias term.

The nearest point from the positive class (positive hyperplane) is defined as:

$$(\mathrm{w} \cdot \mathrm{x}_{i+}) + b = 1, \quad (11)$$

where x_{i+} is the positive support vector

The nearest point from the negative class (negative hyperplane) is defined as:

$$(\mathrm{w} \cdot \mathrm{x}_{i-}) + b = -1, \quad (12)$$

where x_{i-} is the negative support vector.

According to [17], for all data i, the negative hyperplane can be formulated as:

$$(\mathrm{w} \cdot \mathrm{x}_i) + b \leq -1, \quad (13)$$

and the positive hyperplane can be formulated as:

$$(\mathrm{w} \cdot \mathrm{x}_i) + b \geq 1. \quad (14)$$

From equation (13) and (14), equation (15) is obtained:

$$y_i[(\mathrm{w} \cdot \mathrm{x}_i) + b] \geq 1 \quad (15)$$

where y_i is the label of the i-th document, x_i is the i-th input vector from x, and $\mathrm{x}_i \in R^n$, $y_i \in \{-1, +1\}$ for $i = 1,2, \ldots, N$.

The distance between the two hyperplanes (margin) is obtained by calculating the projection length of the difference between the support vectors of each class onto the vector w as shown in equation (16):

$$\rho = \frac{\mathrm{w}}{||\mathrm{w}||} \cdot (\mathrm{x}_{i+} - \mathrm{x}_{i-}). \quad (16)$$

Substituting equation (11) and (12) into equation (16), we obtain:

$$\rho = \frac{\mathrm{w}}{||\mathrm{w}||} \cdot (\mathrm{x}_{i+} - \mathrm{x}_{i-}) = \frac{(\mathrm{w} \cdot \mathrm{x}_{i+}) - (\mathrm{w} \cdot \mathrm{x}_{i-})}{||\mathrm{w}||} = \frac{1 - b - (-1 - b)|}{||\mathrm{w}||} = \frac{2}{||\mathrm{w}||}. \quad (17)$$

Thus, the margin ρ is:

$$\rho = \frac{2}{||\mathrm{w}||}. \quad (18)$$

The largest margin can be obtained by maximizing $\frac{1}{||\mathrm{w}||}$, which is equivalent to minimizing $||\mathrm{w}||$. The problem of finding the hyperplane with the largest margin can be formulated as a quadratic programming problem in equation (19):

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}||\mathrm{w}||^2 \\ &\text{subject to} && y_i[(\mathrm{w} \cdot \mathrm{x}_i) + b] \geq 1, \forall i. \end{aligned} \quad (19)$$

In practice, two classification classes cannot always be perfectly separated. As a result, an optimal hyperplane cannot always be found. To overcome this, SVM is designed using the soft-margin technique. According to [18], the constraint function is modified by adding a slack variable (ξ_i) as follows:

$$y_i[(\mathrm{w} \cdot \mathrm{x}_i) + b] \geq 1 - \xi_i, \forall i. \quad (20)$$

To minimize the slack ξ_i variables, a regularization parameter C is introduced to control tolerance for misclassification errors. Thus, equation (19) becomes:

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\left|\left|\mathrm{w}\right|\right|^2 + C\sum_{i=1}^{N}\xi_i \\ &\text{subject to} && y_i[(\mathrm{w}\cdot \mathrm{x}_i)+b] \geq 1-\xi_i, \forall i \\ & && \xi_i > 0, \forall i, \end{aligned} \tag{21}$$

where $C > 0$.

A larger value of C gives a larger penalty for classification errors.

The optimization problem in equation (21) can be solved using the Lagrange Multiplier method:

$$L(\mathrm{w}, b, \xi, \alpha, \beta) = \frac{1}{2}\left|\left|\mathrm{w}\right|\right|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i(y_i[(\mathrm{w}\cdot \mathrm{x}_i)+b]-1+\xi_i) - \sum_{i=1}^{N}\beta_i\xi_i, \tag{22}$$

where α_i and β_i are Lagrange multipliers. The optimal values of the multipliers are obtained by partially differentiating L with respect to w, b, and ξ and setting them equal to zero, resulting in:

$$\mathrm{w} = \sum_{i=1}^{N}\alpha_i y_i \mathrm{x}_i, \tag{23}$$

$$\sum_{i=1}^{N}\alpha_i y_i = 0, \tag{24}$$

$$C = \sum_{i=1}^{N}\alpha_i + \sum_{i=1}^{N}\beta_i. \tag{25}$$

Substituting equation (23), (24), dan (25) into equation (22) yields:

$$L(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j(\mathrm{x}_i \cdot \mathrm{x}_j). \tag{26}$$

Equation (26) is used to find the best hyperplane by transforming it into a maximization problem to determine α_i. The problem is solved using the Sequential Minimal Optimization (SMO) heuristic method:

$$\begin{aligned} &\text{maximize} && \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j(\mathrm{x}_i \cdot \mathrm{x}_j) \\ &\text{subject to} && \sum_{i=1}^{N}\alpha_i y_i = 0, \forall i \\ & && 0 \leq \alpha_i \leq C, \forall i. \end{aligned} \tag{27}$$

After obtaining α_i values, equation (23) is used to compute w. To obtain b, equations (11) and (12) can be rewritten as:

$$b_s = y_s - \mathrm{w}^T\mathrm{x}_s, \tag{28}$$

where b_s is the bias term for the support vector and y_s is the support vector label.

After obtaining w and b, the input vector x can be classified using:

$$f(\mathrm{x}) = \mathrm{w}\cdot \mathrm{x} + b. \tag{29}$$

Substituting equation (23) into equation (29):

$$f(\mathrm{x}) = \sum_{i=1}^{N}\alpha_i y_i \mathrm{x}_i \cdot \mathrm{x} + b. \tag{30}$$

The classification class is determined as:

$$sign(f(\mathrm{x}) = \begin{cases} +1, & \sum_{i=1}^{N}\alpha_i y_i \mathrm{x}_i \cdot \mathrm{x} + b \geq 0 \\ -1, & \sum_{i=1}^{N}\alpha_i y_i \mathrm{x}_i \cdot \mathrm{x} + b < 0. \end{cases} \tag{31}$$

In real-world applications, datasets are not always linearly separable. To solve nonlinear problems, SVM is modified using kernel functions. This approach is known as the kernel trick. Data is transformed into a space where it can be separated linearly. The kernel trick maps low-dimensional nonlinear data into a higher-dimensional space [19].

In nonlinear SVM, input data x is first mapped using $\Phi(\mathrm{x})$ into a higher-dimensional vector space so that the classes can be separated linearly by a hyperplane [20]:

$$\begin{aligned} &\Phi: \mathbb{R}^d \to \mathbb{R}^q, d < q \\ &\mathrm{x} \mapsto \Phi(\mathrm{x}). \end{aligned} \tag{32}$$

With this mapping, the equations for w and b become:

$$\mathrm{w} = \sum_{i=1}^{N}\alpha_i y_i \Phi(\mathrm{x}_i), \tag{33}$$

$$b_s = y_s - \mathrm{w}^T\Phi(\mathrm{x}_s). \tag{34}$$

The SVM learning process depends on the dot product $\Phi(\mathrm{x}_i) \cdot \Phi(\mathrm{x}_j)$. Based on Mercer's theorem, this can be replaced with a kernel function $K(\mathrm{x}_i, \mathrm{x}_j)$:

$$K(\mathrm{x}_i, \mathrm{x}_j) = \Phi(\mathrm{x}_i) \cdot \Phi(\mathrm{x}_j). \tag{35}$$

Thus, equation (27) becomes:

$$\text{maximize} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathrm{x}_i, \mathrm{x}_j). \quad (36)$$

Classification of input data becomes:

$$f(\Phi(\mathrm{x})) = \sum_{i=1}^{N_s} \alpha_i y_i K(\mathrm{x}_i, \mathrm{x}_j) + b. \quad (37)$$

The classification result is:

$$sign\left(f(\Phi(\mathrm{x}))\right) = \begin{cases} +1, & \sum_{i=1}^{N_s} \alpha_i y_i K(\mathrm{x}_i, \mathrm{x}_j) + b \geq 0 \\ -1, & \sum_{i=1}^{N_s} \alpha_i y_i K(\mathrm{x}_i, \mathrm{x}_j) + b < 0. \end{cases} \quad (38)$$

where N_s is the number of support vectors.
Commonly used kernel functions are shown in Table 1.

Table 1. Kernel Functions [16]

Kernel Name	$K(\mathbf{x}_i, \mathbf{x}_j)$
Linear	$\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$
Polynomial	$\left(\gamma\left(\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)\right) + r\right)^p$
Gaussian Radial Basis Function (RBF)	$\exp\left(-\gamma \left\|\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)\right\|^2\right)$
Sigmoid	$\tanh\left(\gamma\left(\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)\right) + r\right)$

2.4 Model Performance Evaluation

Model performance evaluation in this study uses the balanced accuracy metric. Balanced accuracy is used to evaluate how well the model predicts binary classes [21]. Balanced accuracy is calculated as:

$$Balanced\ Accuracy = \frac{1}{2} \cdot \left(\frac{TP}{TP + FN} + \frac{TF}{TN + FP}\right), \quad (39)$$

where TP is the number of positive samples correctly predicted as positive, TN is the number of negative samples correctly predicted as negative, FP is the number of negative samples incorrectly predicted as positive, and FN is the number of positive samples incorrectly predicted as negative. In addition to balanced accuracy, accuracy, precision, recall, and F1-score are also calculated to complement model evaluation.

2.5 Model Performance Evaluation

Hyperparameter tuning is conducted to improve SVM model performance. This study uses Grid Search to identify the optimal hyperparameter combination. The hyperparameters used include kernel type (linear, polynomial, Gaussian RBF, sigmoid), regularization constant ($C \in \{\frac{1}{100}, \frac{1}{10}, 1, 10, 100\}$), gamma ($\gamma \in \{\frac{1}{100}, \frac{1}{10}, 1, 10, 100\}$), polynomial degree ($p \in \{2, 3, 4, 5\}$), and offset parameter ($r \in \{-1, 0, 1\}$).

3 RESEARCH METHODS

3.1 Data

The data collected in this study consists of Vidio application reviews obtained from Google Play Store between July 1, 2024 and November 30, 2024. Data collection was conducted through web scraping using Python with the google-play-scraper library, resulting in a total of 4,670 reviews.

The Vidio reviews were manually labeled as positive or negative. To reduce subjective judgment, labeling was conducted by three individuals who reviewed each entry one by one. Reviews were only labeled as positive or negative because neutral sentiment is difficult to determine. Labeling was based on the intent and meaning of the review, and ratings were also considered when there was uncertainty in determining the sentiment label.

3.2 Text Preprocessing

Text preprocessing applied to the Vidio review dataset includes case folding to convert all uppercase letters to lowercase, punctuation removal to eliminate all non-alphabetic characters, normalization to convert informal words into standard forms, tokenization to split text into smaller units, stemming to extract root words, and stopwords removal to eliminate less meaningful words.

3.3 Data Splitting

In this step, the dataset is partitioned into two subsets: training data and testing data with an 80% and 20% ratio, respectively.

3.4 Feature Selection

3.4.1 Chi-Square Feature Selection

Chi-Square feature selection applied to training data to reduce less relevant features across all documents. This process selects words that best represent the dataset by testing the independence of each term with its class. The best features are determined by calculating Chi-Square values for each word using equations (1) and (2).

3.4.2 Information Gain Feature Selection

Information Gain feature selection is applied to the training data to reduce less relevant features across all documents. The best features are determined by calculating the Information Gain value for each word using equations (3) through (5).

3.5 Word Representation Using TF-IDF

One of the most widely used word representation methods is TF-IDF. TF-IDF is used to measure the relative importance of a word in a document. Word representation is applied to the training data after the feature selection stage. Each word must be transformed into a numeric vector so that it can be processed by the classification model. TF-IDF vector calculation is performed using equations (6) through (9).

3.6 Classification Using Support Vector Machine

The $\mathbf{TF\text{-}IDF}_{\text{norm}}(t_u, d_1) = \begin{bmatrix} \text{TF-IDF}_{\text{norm}}(t_1, d_1) \\ \text{TF-IDF}_{\text{norm}}(t_2, d_1) \\ \vdots \\ \text{TF-IDF}_{\text{norm}}(t_m, d_1) \end{bmatrix}$ vector obtained from the training data used as the input vector for the SVM classification model. The SVM method aims to find the best hyperplane that separates two data classes with maximum margin. The hyperplane determined by calculating $\mathbf{w}$ using (33) and b using (34).

3.7 Determining the Class of Testing Data

The class of the testing data is determined by substituting the testing input vectors into the sign function in equation (37), and the class is determined using equation (38). Testing data must also undergo text preprocessing and feature selection before being transformed into TF-IDF vectors.

3.8 Model Performance Evaluation

Model performance evaluation is carried out by calculating performance metrics. For imbalanced data, model evaluation uses balanced accuracy as defined in equation (39). Afterward, hyperparameter tuning is conducted to improve the performance of the existing SVM model.

4 RESULTS AND DISCUSSION

4.1 Data

The dataset used in this study consists of Vidio application reviews collected from Google Play Store between July 1, 2024 and November 30, 2024, totaling 4,670 reviews. The collected data was labeled based on sentiment into positive or negative categories. Of the 4,670 reviews, 3,431 reviews (73.5%) were labeled as negative sentiment, while 1,239 reviews (26.5%) were labeled as positive sentiment. This indicates that the dataset is imbalanced. Sample data is shown in Table 2.

Table 2. Sample User Review Data of the Vidio Application

Reviews	Sentiment
Kebanyakan iklan👎👎	Negative
Mantap	Positive
Download ini untuk nonton byon 4 😑	Positive
Beli Paket Byon Combat Vol.4 Ga bisa di tonton , kocak	Negative

4.2 Text Preprocessing

Text preprocessing was applied to the labeled dataset. The preprocessing steps included case folding to convert uppercase letters into lowercase, punctuation removal to remove non-alphabetic characters, normalization to convert informal words into standard forms, tokenization to split text into smaller units, stemming to extract root words, and stopword removal to remove less meaningful words. Reviews that contained only stopwords or non-alphabetic characters became empty and were removed, reducing the number of reviews to 4,584. Samples of preprocessing results are shown in Table 3.

Table 3. Sample Results of Text Preprocessing for Vidio Application Reviews

Reviews	Text Preprocessing	Sentiment
Kebanyakan iklan👎👎	'iklan'	Negative
Mantap	'mantap'	Positive
Download ini untuk nonton byon 4 😑	'download', 'tonton', 'byon'	Positive
Beli Paket Byon Combat Vol.4 Ga bisa di tonton , kocak	'beli', 'paket', 'byon', 'combat', 'volume', 'tidak', 'tonton', 'kocak'	Negative

4.3 Data Splitting

The data was divided into 80% training data (3,667 reviews) and 20% testing data (917 reviews).

4.4 Feature Selection

4.4.1 Chi-Square Feature Selection

Chi-Square feature selection was performed on the training data using equations (1) and (2). Sample results of Chi-Square values for each word are presented in Table 4.

Table 4. Sample Single Chi-Square Values for Each Word in Vidio Application Reviews

t_u	$\chi^2(t_u)$
tidak	1.282,747
bagus	678,6296
mantap	578,9626
⋮	⋮
broadcast	0,001989
bicara	0,001989

From Table 4, the words with the highest Chi-Square values are "tidak", "bagus", and "mantap", indicating that these words are the most effective for determining classification classes. Meanwhile, the words with the lowest Chi-Square values are "cerita", "broadcast," and "bicara", indicating that these words are less representative for determining classification classes.

4.4.2 Information Gain Feature Selection

Information Gain feature selection was performed on the training data using equations (3), (4), and (5). Sample results of Information Gain values for each word are shown in Table 5.

Table 5. Sample Information Gain Values for Each Word in Vidio Application Reviews

t_u	$IG(t_u)$
tidak	0,1064
langgan	0,041977
bagus	0,040818
⋮	⋮
broadcast	$1{,}36\times 10^{-7}$
bicara	$1{,}36\times 10^{-7}$

From Table 5, the words with the highest Information Gain values are "tidak", "langgan", and "bagus", indicating that these words are the most effective for determining classification classes. The words with the lowest Information Gain values are "broadcast" and "bicara", indicating that these words are less representative for classification.

4.5 Classification Using Support Vector Machine

The testing dataset consists of 917 reviews. The training dataset that has undergone text preprocessing, feature selection, and TF-IDF word representation produces input vectors **x** for the SVM classification method. The default model performance results for each feature selection method are shown in Table 6.

Table 6. Default Performance Results of Vidio Application Reviews for Each Feature Selection Method

Feature Selection	*Accuracy*	*Precision*	*Recall*	F1-*Score*	*Balanced accuracy*
Chi-Square	94,77%	92,67%	93,63%	93,13%	93,63%
Information Gain	93,02%	92,64%	88,44%	90,28%	88,44%

Hyperparameter tuning was then performed on $C, r, p, \gamma,$ and kernel parameters defined by the researcher. The best model performance results for each feature selection method are presented in Table 7.

Table 7. Best Performance Results of Vidio Application Reviews for Each Feature Selection Method

Feature Selection	Best parameter	*Accuracy*	*Precision*	*Recall*	F1-*Score*	*Balanced accuracy*
Chi-Square	Linear $C = 1$	95,2%	92,88%	94,78%	93,78%	94,78%
Information Gain	*Polynomial* $C = 10, r = 1, p = 3, \gamma = 0{,}1$	94,33%	91,49%	94,2%	92,72%	94,2%

Based on the balanced accuracy metric in Table 7, the best model performance was achieved using Chi-Square feature selection with a balanced accuracy of 94.78%, using a linear kernel and hyperparameter $C =$ 1. In contrast, the Information Gain method yielded a slightly lower Balanced Accuracy of 94.20%. Notably, this method required a much more complex configuration to reach its peak performance, utilizing

a Polynomial Kernel with parameters $C = 10$, $r = 1$, $p = 3$, and $\gamma = 0{,}1$. The comparison highlights that Chi-Square not only outperforms Information Gain in accuracy but does so with a significantly simpler model architecture.

4.6 Sentiment Analysis Results of Vidio Application Reviews

The classification results of the testing data using the best tuned model are shown in Table 8.

Table 8. Classification Results of Vidio Application Review Testing Data After Hyperparameter Tuning

Reviews	Actual Sentiment	Prediction Sentiment
Top markotop dah	Positive	Positive
Download krn pngen nnton zona merah.. tp trnyata lemotnya bukan maen...	Negative	Negative
Aplikasinya bagus Cukup menghibur.	Positive	Positive

Based on sentiment analysis results of Vidio user reviews, it was found that out of 917 reviews, 670 reviews (73.06%) were classified as negative sentiment, while 247 reviews (26.94%) were classified as positive sentiment.

The classification performance comparison between SVM with Chi-Square feature selection and SVM with Information Gain feature selection shows a difference in balanced accuracy. SVM with Chi-Square feature selection achieved a slightly higher balanced accuracy of 94.78% compared to SVM with Information Gain feature selection which achieved 94.2%. Although the difference in peak Balanced Accuracy is 0.58% (94.78% vs 94.20%), the practical significance of the proposed method lies in its computational efficiency.

5 CONCLUSION

Based on the results and discussion in this study, the following conclusions can be drawn:

1. Sentiment analysis results of Vidio application user reviews using SVM classification with Chi-Square feature selection produced 73.06% negative sentiment and 26.94% positive sentiment. Meanwhile, using Information Gain feature selection produced 72.19% negative sentiment and 27.81% positive sentiment.
2. The Chi-Square method demonstrated superior stability compared to Information Gain. In the default SVM mode (without hyperparameter tuning), Chi-Square achieved a baseline Balanced Accuracy of 93.63%, significantly outperforming Information Gain, which only reached 88.44%. This 5.19% gap indicates that features selected by Chi-Square are fundamentally more separable and less dependent on complex parameter optimization.
3. Through hyperparameter tuning, the SVM model based on Chi-Square proved to be the optimal approach, achieving the highest Balanced Accuracy of 94.78%. Crucially, this peak performance was attained using a computationally efficient Linear Kernel ($C = 1$). In contrast, the Information Gain method yielded a lower Balanced Accuracy of 94.20% despite requiring a complex Polynomial Kernel ($C = 10$, $r = 1$, $p = 3$, $\gamma = 0{,}1$) to converge. This confirms that the proposed method offers a superior trade-off between accuracy and computational cost.

For future research, deep learning classification methods such as Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Bidirectional Gated Recurrent Unit (Bi-GRU), and others may be applied.

REFERENCES

[1] H. Nurhayati-Wolff, “Number of users of video streaming (SVoD) in Indonesia from 2017 to 2027,” Statista.

[2] Google Play Store, “Vidio: Sports, Movies, Series.”

[3] A. Palanivinayagam, C. Z. El-Bayeh, and R. Damaševičius, “Twenty Years of Machine-Learning-Based Text Classification: A Systematic Review,” *Algorithms*, vol. 16, no. 5, p. 236, Apr. 2023, doi: 10.3390/a16050236.

[4] M. Iqbal, M. M. Abid, M. N. Khalid, and A. Manzoor, “Review of feature selection methods for text classification,” *International Journal of Advanced Computer Research*, vol. 10, no. 49, pp. 138–152, Jul. 2020, doi: 10.19101/IJACR.2020.1048037.

[5] R. Guido, M. C. Groccia, and D. Conforti, “A hyper-parameter tuning approach for cost-sensitive support vector machine classifiers,” *Soft comput.*, vol. 27, no. 18, pp. 12863–12881, Sep. 2023, doi: 10.1007/s00500-022-06768-8.

[6] M. Fudhail Ferio Supeli and S. Setiaji, “Klasifikasi Sentimen Positif Dan Negatif Pada Aplikasi Vidio Dengan Algoritma K-Nearest Neighbor,” *Indonesian Journal Computer Science*, vol. 2, no. 1, pp. 7–15, Apr. 2023, doi: 10.31294/ijcs.v2i1.1874.

[7] I. L. Kharisma, D. A. Septiani, A. Fergina, and K. Kamdan, “Penerapan Algoritma Decision Tree

untuk Ulasan Aplikasi Vidio di Google Play," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 9, no. 2, pp. 218–226, Sep. 2023, doi: 10.25077/TEKNOSI.v9i2.2023.218-226.

[8] A. Maulana, "Analisis Sentimen Menggunakan Algoritma K-Nearest Neighbor dan Naive Bayes pada Aplikasi Vidio," Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, 2024.

[9] scikit-learn developers, "SVC - scikit-learn 1.8.0 documentation," https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html.

[10] I. S. Thaseen, Ch. A. Kumar, and A. Ahmad, "Integrated Intrusion Detection Model Using Chi-Square Feature Selection and Ensemble of Classifiers," *Arab. J. Sci. Eng.*, vol. 44, no. 4, pp. 3357–3368, Apr. 2019, doi: 10.1007/s13369-018-3507-5.

[11] C. F. Suharno, M. A. Fauzi, and R. S. Perdana, "Klasifikasi Teks Bahasa Indonesia Pada Dokumen Pengaduan Sambat Online Menggunakan Metode K-Nearest Neighbors Dan Chi-square," *Systemic: Information System and Informatics Journal*, vol. 3, no. 1, pp. 25–32, Aug. 2017, doi: 10.29080/systemic.v3i1.191.

[12] E. Odhiambo Omuya, G. Onyango Okeyo, and M. Waema Kimwele, "Feature Selection for Classification using Principal Component Analysis and Information Gain," *Expert Syst. Appl.*, vol. 174, p. 114765, Jul. 2021, doi: 10.1016/j.eswa.2021.114765.

[13] R. N. Waykole and A. D. Thakare, "A Review of Feature Extraction Methods for Text Classification," *International Journal of Advance Engineering and Research Development*, vol. 5, no. 4, Apr. 2018.

[14] D. Oktavia, Y. R. Ramadahan, and M. Minarto, "Analisis Sentimen Terhadap Penerapan Sistem E-Tilang Pada Media Sosial Twitter Menggunakan Algoritma Support Vector Machine (SVM)," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 1, pp. 407–417, Aug. 2023.

[15] M. A. Reddy, T. T. Kumar, and G. S. S. R. Krishna, "Malaria Cell-Image Classification using InceptionV3 and SVM," *International Journal of Engineering Research & Technology (IJERT)*, vol. 10, no. 8, pp. 6–10, Aug. 2021.

[16] R. A. Ariyanto and N. Chamidah, "Sentiment Analysis for Zoning System Admission Policy Using Support Vector Machine and Naive Bayes Methods," *J. Phys. Conf. Ser.*, vol. 1776, no. 1, p. 012058, Feb. 2021, doi: 10.1088/1742-6596/1776/1/012058.

[17] N. M. S. Hadna, P. I. Santosa, and W. W. Winarno, "Studi Literatur Tentang Perbandingan Metode Untuk Proses Analisis Sentimen Di Twitter," in *Seminar Nasional Teknologi Informasi dan Komunikasi (SENTIKA 2016)*, Yogyakarta, Mar. 2016.

[18] Naiyang. Deng, Yingjie. Tian, and Chunhua. Zhang, *Support vector machines : optimization based theory, algorithms, and extensions*. CRC Press, Taylor & Francis Group, 2013.

[19] Md. S. Reza, U. Hafsha, R. Amin, R. Yasmin, and S. Ruhi, "Improving SVM performance for type II diabetes prediction with an improved non-linear kernel: Insights from the PIMA dataset," *Computer Methods and Programs in Biomedicine Update*, vol. 4, p. 100118, 2023, doi: 10.1016/j.cmpbup.2023.100118.

[20] A. S. Nugroho, A. B. Witarto, and D. Handoko, "Support Vector Machine: Teori dan Aplikasinya dalam Bioinformatika," https://asnugroho.net/papers/ikcsvm.pdf.

[21] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, Jan. 2021, doi: 10.1016/j.aci.2018.08.003.