Available online at www.coins-indocompt.org

COINS JOURNAL
Computing and Information System Journal

COmputing and INformation Systems Journal

IndoCompt PUBLISHER

Vol. 1, No. 2, Augustus, 2025

EFFECT OF RANDOM UNDER SAMPLING AND RANDOM OVER SAMPLING METHOD ON SVM PERFORMANCE

Agil Dwi Saputra[a*], Deni Arifianto[b], Reni Umilasari[c]

Muhammadiyah University of Jember, Jember 68121, Indonesia

agilputra855@gmail.com

Abstract

Imbalanced data is a common challenge in sentiment analysis, as it can cause the classification model to be biased towards the majority class and ignore important information from the minority class. This study aims to evaluate the effect of resampling methods, namely Random Under Sampling (RUS), and Random Over Sampling (ROS), on the performance of the Support Vector Machine (SVM) algorithm in handling imbalanced sentiment data. Data were collected from social media X (Twitter) with the topic of naturalization of soccer players in Indonesia. The research process includes preprocessing, TF-IDF weighting, and model testing using K-Fold Cross Validation with K = 2, 5, and 10. Evaluation was carried out based on the F1-score matrix, recall, precision, and accuracy. The results show that the ROS method provides the best performance, especially at K = 10 with an F1-score value of 0.80, recall 0.78, precision 0.84, and accuracy 0.85. and RUS shows a lower performance improvement. These results show that selecting an appropriate resampling method can improve the performance of the classification model when faced with imbalanced data.

Keywords: data imbalance, sentiment analysis, RUS, ROS, SVM

1. Introduction

Data imbalance is often a major problem in machine learning. Data imbalance occurs when the distribution of classes in a dataset is uneven, for example, the amount of positive sentiment data is much more than negative sentiment or vice versa. Data imbalance is increasingly significant in the digital era, where public responses are widespread through social media, especially X (Twitter). One topic that is often discussed is the naturalization policy of football players in Indonesia because it causes a lot of debate, which raises various responses, whether positive, negative, or neutral. Inaccurate representation, overfitting, and bias toward the majority class can result from data imbalance, which is another barrier to classification [1]. RUS randomly reduces the amount of data from the majority class so that the distribution is the same as the minority class. This approach is simple but may lead to loss of important information. In contrast, ROS randomly duplicates samples to increase the amount of data of the minority class. While effective in balancing the data, this method runs the risk of overfitting. The application of the Imbalance Data Sampling method greatly affects the performance of classification algorithms, one of which is the Support Vector Machine (SVM). In sentiment analysis, SVM is one of the most widely used classification methods. SVM operates by determining the best hyperlane to separate data into different classes. This algorithm works by weighting through the formation of a line pattern that is used for the weighting and classification process [2]. SVM performance can be affected by data imbalance as the model tends to focus more on the majority class and less on the minority class, even though the minority class plays an important role in accurate detection [3]. This study aims to measure the effect of Imbalance Data Sampling methods such as RUS and ROS on SVM performance in sentiment analysis. By evaluating the performance of the model based on matrices such as accuracy, precision, recall, and F1-score. Applying the right sampling method is essential to improve the performance of SVM, especially in sentiment analysis with imbalanced data. This research provides a better understanding of people's perceptions on a particular topic, as well as providing a scientific understanding of sentiment analysis and machine learning.

2. Methode

The impact of the Imbalance Data Sampling technique on Support Vector Machine performance is the main topic of this study. The steps of the research are as follows:

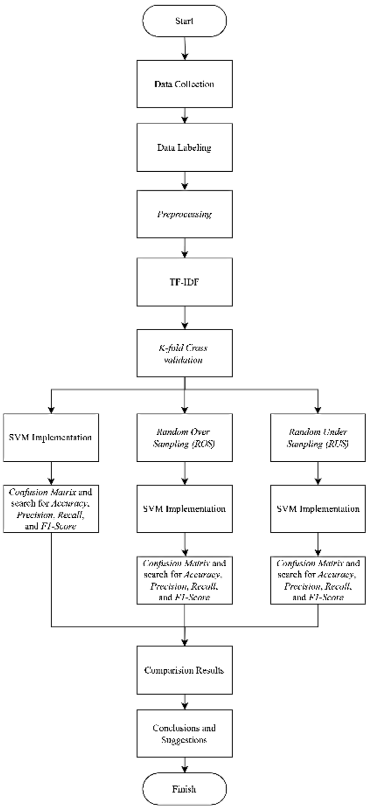
Start
Data Collection
Data Labeling
Preprocessing
TF-IDF
K-fold Cross validation
SVM Implementation
Random Over Sampling (ROS)
Random Under Sampling (RUS)
Confusion Matrix and search for *Accuracy, Precision, Recall,* and *F1-Score*
SVM Implementation
SVM Implementation
Confusion Matrix and search for *Accuracy, Precision, Recall,* and *F1-Score*
Confusion Matrix and search for *Accuracy, Precision, Recall,* and *F1-Score*
Comparision Results
Conclusions and Suggestions
Finish

Figure 1. Research Stages

2.1. Text Preprocessing

The Preprocessing stage consists of several steps, namely Cleaning, Case Folding, Tokenizing, Stopword removal, and stemming. The following is an explanation for each of these steps.

2.1.1 Cleaning

Cleaning is a step of document cleaning that includes deleting components such special characters, symbols, numbers, emoticons, and URL links that are not pertinent to the document's core content [4].

2.1.2 Case Folding

Converting text that has both capital and lowercase characters into solely lowercase letters is known as case folding [5].

2.1.3 Tokenizing

Tokenization divides a document into discrete units known as tokens. Furthermore, several characters that can be regarded as punctuation are eliminated during tokenization [6].

2.1.4 Stopword Removal

The goal of stopword removal is to identify and eliminate the most common words that don't convey important information. Stopwords, which exclude certain verbs, adjectives, or adverbs, also aid in shrinking the text index. Words like "at," "to," "of," "or," "which," and so forth are examples [7].

2.1.5 Stemming

Stemming is the process of reducing words to their most basic form by adding affixes, such as prefixes and suffixes [8].

2.2. TF-IDF

A technique called TF-IDF (Term Frequency-Inverse Document Frequency) weighs each phrase to determine how pertinent it is to the text [9]. word Frequency (TF), the first element in TF-IDF, determines how frequently a word occurs in a given text. TF is calculated by dividing the number of times a word occurs in a document by the total number of words in the document. The second element is Inverse Document Frequency (IDF), which gauges a word's significance within the corpus as a whole. IDF assigns a lesser weight to words that are used often throughout publications because it believes they are less informative. IDF is calculated using the logarithm and ratio of the total number of documents to the number of documents that include a certain term. The formula that describes IDF is as follows:

$$idf_t = log\frac{N}{df_t} \tag{1}$$

The TF-IDF value of a word in the document is calculated by multiplying its TF and IDF values. A greater TF-IDF score indicates a word's significance in the text. Use the following formula to find the TF-IDF value:

$$tfidf_{t,d} = tf_{t,d} \times idf_t \tag{2}$$

2.3. K-Fold Cross Validation

A validation method called K-Fold Cross Validation splits the original dataset into K equal-sized folds at random [10]. This technique enhances the assessment of model performance on unseen data and lessens evaluation bias. In comparison to train-test split, K-Fold Cross Validation necessitates a more sophisticated implementation and a higher computational time. For balanced bias and variance, K values of 5 or 10 are frequently utilized.

2.4. Support Vector Machine

A data processing technique called Support Vector Machine (SVM) makes use of assumptions that are expressed as linear functions in a high-dimensional feature space. This approach is trained using learning algorithms that are grounded in optimization theory. [11] The stages in the Support Vector Machine (SVM) method include:

1. Determining which phrases appear most frequently in each examined document or tweet.
2. Initial parameter values like ‘=0.001, γ=0.5, λ=0.5, C=1, and α=0.5 are entered.
3. Perform matrix perhitungan by using rumus:

$$D_{ij} = y_i y_j (K(x_i \cdot x_j) + \lambda^2) \tag{3}$$

4. The following formula is used to calculate every nth data=1,2,3, 4...n:

$$E_i = \sum_{j=1}^{n} \alpha_i \, D_{ij} \tag{4}$$

$$\delta\alpha_i = min\,\{max\,[\gamma(1 - E_i), -\alpha_i], C - \alpha_i\} \tag{5}$$

$$\alpha_i = \alpha_i + \delta\alpha_i \tag{6}$$

5. The following formula is used to get the bias value (b):

$$b = -\frac{1}{2}[w \cdot x^+ + w \cdot x^-] \tag{7}$$

6. Evaluation of test documents
7. Calculations to determine the final decision
   The decision is determined by the following rules:

$$h(x) = \begin{cases} +1, & jika\, w \cdot x + b \geq 0 \\ -1, & jika\, w \cdot x + b < 0 \end{cases} \tag{8}$$

If the computation result is larger than or equal to 0, sign(h(x)) = +1 shows that the decision is in the Positive class. On the other hand, if the decision calculation value is less than 0 due to sign(h(x)) = -1, the decision is in the Negative class. Calculations using the following formula are used to make the decision:

$$h(x) = w \cdot x + b \tag{9}$$

2.5. Random Under Sampling

In order to achieve a more equal ratio between the majority and minority classes, a technique known as random under sampling involves removing a portion of the majority class sample. To match the minority class numerically, a subset of the majority class is chosen at random during this procedure. Because it doesn't involve complicated calculations, Random Under Sampling is simple to use and has the benefit of lowering the amount of the dataset, which speeds up the model training time. Although this approach effectively lessens bias towards the majority class, RUS has a significant disadvantage: the possibility of missing crucial information from the majority class, which may eventually impair the model's overall performance [12].

2.6. Random Under Sampling

In order to establish a balanced distribution, the Random Over Sampling technique randomly increases the amount of minority class samples in the training data. Until the quantity of samples in the minority class equals that of the majority class, this process is repeated several times [13]. The benefit of random over sampling is that it can overcome bias towards the majority class while preserving all information from the majority class. The overfitting risk brought on by data duplication and the growing dataset size, which affects training time and necessitates more storage, are the drawbacks of this approach.

3. Result

Data utilizing a variety of balancing strategies, including Random Under Sampling (RUS), Random Over Sampling (ROS)and no balancing strategies, were used to train Support Vector Machine (SVM) models. The K-Fold Cross Validation technique with K = 2, 5, and 10 is used for evaluation. Precision, Recall, F1-score, and Accuracy are among the assessment measures that are employed; they are calculated both on a macro average and per class basis.

3.1. Random Under Sampling

The following are the Precision, Recall, F1-score, and Accuracy values measured on a per-class basis and also on a macro average Random Under Sampling basis in table 1.

Table 1. The result value of the SVM model with RUS balancing

Fold	Label	Data Distribution Before RUS	Data Distribution After RUS	precision	Recall	F1-score	Accuracy
K=2							
1	-1	206	134	0.79	0.51	0.62	
	0	134	134	0.43	0.64	0.51	
	1	425	134	0.77	0.79	0.78	
		Macro Avg		**0.66**	**0.65**	**0.64**	**0.69**
2	-1	211	123	0.67	0.59	0.63	
	0	123	123	0.35	0.54	0.42	
	1	431	123	0.81	0.71	0.76	
		Macro Avg		**0.61**	**0.61**	**0.60**	**0.65**
		Average K=2		**0.64**	**0.63**	**0.62**	**0.67**
K=5							
1	-1	340	207	0.88	0.77	0.82	
	0	207	207	0.54	0.8	0.65	
	1	677	207	0.92	0.85	0.88	
		Macro Avg		**0.78**	**0.81**	**0.78**	**0.82**
2	-1	328	209	0.79	0.62	0.69	
	0	209	209	0.41	0.69	0.51	
	1	687	209	0.85	0.78	0.81	
		Macro Avg		**0.68**	**0.70**	**0.67**	**0.72**
3	-1	331	203	0.88	0.6	0.72	
	0	203	203	0.48	0.72	0.57	
	1	690	203	0.82	0.82	0.82	
		Macro Avg		**0.73**	**0.72**	**0.70**	**0.74**
4	-1	342	198	0.80	0.63	0.70	
	0	198	198	0.45	0.69	0.55	
	1	684	198	0.84	0.76	0.80	
		Macro Avg		**0.70**	**0.69**	**0.68**	**0.72**
5	-1	327	211	0.84	0.68	0.75	
	0	211	211	0.29	0.54	0.38	
	1	686	211	0.86	0.74	0.80	
		Macro Avg		**0.66**	**0.65**	**0.64**	**0.69**
		Average K=5		**0.71**	**0.71**	**0.70**	**0.74**
K=10							
1	-1	376	233	0.91	0.76	0.83	
	0	233	233	0.59	0.79	0.68	
	1	768	233	0.87	0.86	0.87	

		Macro Avg		**0.79**	**0.80**	**0.79**	**0.82**
Fold	**Label**	**Data Distribution Before RUS**	**Data Distribution After RUS**	**precision**	**Recall**	**F1-score**	**Accuracy**
2	-1	381	231	0.79	0.75	0.77	
	0	231	231	0.53	0.73	0.61	
	1	765	231	0.88	0.80	0.84	
		Macro Avg		**0.73**	**0.76**	**0.74**	**0.78**
3	-1	377	233	0.75	0.75	0.75	
	0	233	233	0.44	0.62	0.52	
	1	767	233	0.91	0.81	0.86	
		Macro Avg		**0.70**	**0.73**	**0.71**	**0.76**
4	-1	368	233	0.88	0.76	0.81	
	0	233	233	0.47	0.79	0.59	
	1	776	233	0.92	0.81	0.86	
		Macro Avg		**0.76**	**0.79**	**0.76**	**0.79**
5	-1	372	232	0.97	0.67	0.79	
	0	232	232	0.45	0.76	0.57	
	1	773	232	0.85	0.82	0.83	
		Macro Avg		**0.76**	**0.75**	**0.73**	**0.76**
6	-1	376	228	0.80	0.68	0.74	
	0	228	228	0.43	0.62	0.51	
	1	773	228	0.84	0.77	0.81	
		Macro Avg		**0.69**	**0.69**	**0.68**	**0.72**
7	-1	382	225	0.71	0.63	0.67	
	0	225	225	0.48	0.69	0.56	
	1	770	225	0.84	0.74	0.79	
		Macro Avg		**0.68**	**0.69**	**0.67**	**0.71**
8	-1	377	230	0.74	0.57	0.65	
	0	230	230	0.47	0.81	0.59	
	1	770	230	0.88	0.77	0.82	
		Macro Avg		**0.70**	**0.72**	**0.69**	**0.73**
9	-1	367	235	0.93	0.74	0.82	
	0	235	235	0.32	0.50	0.39	
	1	775	235	0.80	0.78	0.79	
		Macro Avg		**0.68**	**0.67**	**0.67**	**0.73**
10	-1	377	233	0.74	0.62	0.68	
	0	233	233	0.32	0.58	0.41	
	1	767	233	0.91	0.76	0.83	
		Macro Avg		**0.65**	**0.66**	**0.64**	**0.70**
		Average K=10		**0.71**	**0.73**	**0.71**	**0.75**

3.2. Random Over Sampling

The following are the Precision, Recall, F1-score, and Accuracy values measured on a per-class basis and also on a macro average Random Under Sampling basis in table 2.

Table 2. The result value of the SVM model with ROS balancing

Fold	Label	Data Distribution Before ROS	Data Distribution After ROS	precision	Recall	F1-score	Accuracy
K=2							
1	-1	206	425	0.93	0.73	0.81	
	0	134	425	0.73	0.37	0.49	
	1	425	425	0.79	0.98	0.87	
		Macro Avg		**0.81**	**0.69**	**0.72**	**0.81**
2	-1	211	431	0.88	0.74	0.81	
	0	123	431	0.62	0.38	0.47	
	1	431	431	0.80	0.96	0.87	
		Macro Avg		**0.77**	**0.69**	**0.72**	**0.80**
		Average K=2		**0.79**	**0.69**	**0.72**	**0.81**
K=5							
1	-1	340	677	0.92	0.86	0.89	
	0	207	677	0.84	0.52	0.64	
	1	677	677	0.87	0.99	0.93	
		Macro Avg		**0.88**	**0.79**	**0.82**	**0.88**
2	-1	328	687	0.93	0.83	0.88	
	0	209	687	0.69	0.50	0.58	
	1	687	687	0.86	0.97	0.91	
		Macro Avg		**0.82**	**0.77**	**0.79**	**0.86**
3	-1	331	690	0.92	0.80	0.86	
	0	203	690	0.73	0.50	0.59	
	1	690	690	0.81	0.95	0.88	
		Macro Avg		**0.82**	**0.75**	**0.78**	**0.83**
4	-1	342	686	0.89	0.75	0.81	
	0	198	686	0.74	0.54	0.63	
	1	686	686	0.81	0.95	0.88	
		Macro Avg		**0.82**	**0.75**	**0.77**	**0.82**

Fold	Label	Data Distribution Before ROS	Data Distribution After ROS	precision	Recall	F1-score	Accuracy
5	-1	327	686	0.92	0.78	0.84	
	0	211	686	0.64	0.50	0.56	
	1	686	686	0.86	0.98	0.91	
		Macro Avg		**0.81**	**0.75**	**0.77**	**0.85**
		Average K=5		**0.83**	**0.76**	**0.79**	**0.85**
K=10							
1	-1	376	768	0.97	0.90	0.94	
	0	233	768	0.88	0.62	0.73	
	1	768	768	0.89	0.99	0.94	
		Macro Avg		**0.91**	**0.84**	**0.87**	**0.91**
2	-1	381	765	0.97	0.83	0.90	
	0	231	765	0.75	0.58	0.65	
	1	765	765	0.88	0.99	0.93	
		Macro Avg		**0.87**	**0.80**	**0.83**	**0.88**
3	-1	377	767	0.94	0.80	0.86	
	0	233	767	0.67	0.58	0.62	
	1	767	767	0.86	0.94	0.90	
		Macro Avg		**0.82**	**0.78**	**0.8**	**0.85**
4	-1	368	776	0.93	0.86	0.89	
	0	233	776	0.74	0.58	0.65	
	1	776	776	0.89	0.99	0.93	
		Macro Avg		**0.85**	**0.81**	**0.83**	**0.88**
5	-1	372	773	0.97	0.78	0.86	
	0	232	773	0.67	0.64	0.65	
	1	773	773	0.83	0.93	0.88	
		Macro Avg		**0.82**	**0.78**	**0.80**	**0.84**
6	-1	376	773	0.97	0.88	0.92	
	0	228	773	0.88	0.52	0.65	
	1	773	773	0.83	0.99	0.90	
		Macro Avg		**0.89**	**0.79**	**0.83**	**0.87**
7	-1	382	770	0.84	0.77	0.81	
	0	225	770	0.68	0.41	0.51	
	1	770	770	0.77	0.92	0.84	
		Macro Avg		**0.77**	**0.70**	**0.72**	**0.78**
8	-1	377	770	0.94	0.72	0.82	
	0	230	770	0.71	0.63	0.67	
	1	770	770	0.84	0.95	0.89	
		Macro Avg		**0.83**	**0.77**	**0.79**	**0.84**
9	-1	367	775	0.91	0.80	0.85	
	0	235	775	0.59	0.45	0.51	
	1	775	775	0.85	0.96	0.90	
		Macro Avg		**0.78**	**0.74**	**0.76**	**0.84**
10	-1	377	767	0.91	0.78	0.84	
	0	233	767	0.68	0.62	0.65	
	1	767	767	0.89	0.97	0.92	
		Macro Avg		**0.83**	**0.79**	**0.80**	**0.86**
		Average K=10		**0.84**	**0.78**	**0.80**	**0.85**

3.3. Support Vector Machine

The following are the Precision, Recall, F1-score, and Accuracy values measured on a per-class basis and also on a macro average Random Under Sampling basis in table 3.

Table 3. Evaluation Results of SVM Model Without Balancing

Fold	Label	Training Data Distribution	precision	Recall	F1-score	Accuracy
K=2						
1	-1	206	0.85	0.43	0.57	
	0	134	0.80	0.03	0.06	
	1	425	0.65	0.99	0.79	
	Macro Avg		**0.77**	**0.48**	**0.47**	**0.68**
2	-1	211	0.90	0.58	0.71	
	0	123	0.63	0.13	0.21	
	1	431	0.69	0.98	0.81	
	Macro Avg		**0.74**	**0.56**	**0.58**	**0.73**
	Average K=2		**0.75**	**0.52**	**0.53**	**0.70**
K=5						
1	-1	340	0.97	0.81	0.88	
	0	207	0.85	0.22	0.35	
	1	677	0.78	1.00	0.88	
	Macro Avg		**0.87**	**0.68**	**0.70**	**0.82**

Fold	Label	Distribusi Data Latih	precision	Recall	F1-score	Accuracy
2	-1	328	0.97	0.71	0.82	
	0	209	0.57	0.17	0.26	
	1	687	0.74	0.99	0.84	
	Macro Avg		**0.76**	**0.62**	**0.64**	**0.78**
3	-1	331	0.97	0.73	0.83	
	0	203	0.81	0.24	0.37	
	1	690	0.73	0.99	0.84	
	Macro Avg		**0.84**	**0.65**	**0.68**	**0.78**
4	-1	342	0.88	0.71	0.79	
	0	198	0.75	0.15	0.25	
	1	684	0.71	0.97	0.82	
	Macro Avg		**0.78**	**0.61**	**0.62**	**0.75**
5	-1	327	0.93	0.69	0.79	
	0	211	0.53	0.22	0.31	
	1	686	0.76	0.99	0.86	
	Macro Avg		**0.74**	**0.63**	**0.65**	**0.78**
	Average K=5		**0.80**	**0.64**	**0.66**	**0.78**
K=10						
1	-1	376	0.95	0.88	0.91	
	0	233	0.88	0.29	0.44	
	1	768	0.82	1.00	0.90	
	Macro Avg		**0.88**	**0.72**	**0.75**	**0.86**
2	-1	381	0.93	0.78	0.85	
	0	231	0.78	0.27	0.40	
	1	765	0.80	1.00	0.89	
	Macro Avg		**0.84**	**0.68**	**0.71**	**0.82**
3	-1	377	0.97	0.80	0.88	
	0	233	0.62	0.21	0.31	
	1	767	0.78	0.98	0.87	
	Macro Avg		**0.79**	**0.66**	**0.68**	**0.81**
4	-1	368	0.97	0.80	0.88	
	0	233	0.57	0.17	0.26	
	1	776	0.75	1.00	0.86	
	Macro Avg		**0.77**	**0.65**	**0.66**	**0.80**
5	-1	372	1.00	0.76	0.86	
	0	232	0.70	0.28	0.40	
	1	773	0.74	0.98	0.84	
	Macro Avg		**0.81**	**0.67**	**0.70**	**0.80**
6	-1	376	1.00	0.80	0.89	
	0	228	0.89	0.28	0.42	
	1	773	0.74	0.99	0.85	
	Macro Avg		**0.88**	**0.69**	**0.72**	**0.80**
7	-1	382	0.90	0.74	0.81	
	0	225	0.75	0.19	0.30	
	1	770	0.72	0.98	0.83	
	Macro Avg		**0.79**	**0.64**	**0.65**	**0.76**
8	-1	377	0.90	0.68	0.77	
	0	230	0.86	0.22	0.35	
	1	770	0.73	0.99	0.84	
	Macro Avg		**0.83**	**0.63**	**0.66**	**0.77**
9	-1	367	0.95	0.74	0.83	
	0	235	0.50	0.23	0.31	
	1	775	0.78	1.00	0.88	
	Macro Avg		**0.74**	**0.66**	**0.67**	**0.80**
10	-1	377	0.91	0.72	0.81	
	0	233	0.56	0.21	0.30	
	1	767	0.79	0.99	0.88	
	Macro Avg		**0.75**	**0.64**	**0.66**	**0.80**
	Average K=10		**0.81**	**0.66**	**0.69**	**0.80**

It has been demonstrated that adding an imbalance data sampling strategy may enhance the SVM model's performance, particularly in the areas of recall and F1-score. Performance is rather poor in the SVM model without balancing, particularly in the 2-Fold scheme with precision 0.75, recall 0.52, F1-score 0.53, and accuracy 0.70. This model only attains precision 0.81, recall 0.66, F1-score 0.69, and accuracy 0.80, even in the 10-Fold scheme. With the best results in the 10-Fold scheme of precision 0.71, recall 0.73, F1-score 0.71, and accuracy 0.75, the Random Under Sampling (RUS) technique outperforms the baseline. Its performance still falls short of other approaches, though, maybe as a result of crucial data being lost during the undersampling procedure. The best performance is provided by Random Over Sampling (ROS), especially in the 10-Fold with precision 0.84, recall 0.78, F1-score 0.80, and accuracy 0.85. It is demonstrated that ROS may increase the model's overall performance and mitigate data skews without compromising the accuracy of the information. The evaluation results of each balancing technique and without balancing are presented in table 4.

Table 4. Evaluation Results for Each Method

Method	K-Fold	Precision	Recall	F1-score	Accuracy
SVM	2	0.75	0.52	0.53	0.70
	5	0.80	0.64	0.66	0.78
	10	0.81	0.66	0.69	0.80
SVM + RUS	2	0.64	0.63	0.62	0.67
	5	0.71	0.71	0.70	0.74
	10	0.71	0.73	0.71	0.75
SVM + ROS	2	0.79	0.69	0.72	0.81
	5	0.83	0.76	0.79	0.85
	10	0.84	0.78	0.80	0.85
SVM + SMOTE	2	0.76	0.59	0.62	0.74
	5	0.82	0.68	0.71	0.80
	10	0.83	0.69	0.72	0.81

4. Conclusion

This study demonstrates that the Support Vector Machine (SVM) model's performance in sentiment analysis of imbalanced data may be considerably enhanced by the application of data balancing or resampling techniques. Random Over Sampling (ROS) produced the best outcomes out of the two techniques that were examined. The accuracy, precision, recall, and F1-score of the SVM + ROS model were 0.85, 0.78, and 0.84 at K=10. This demonstrates that a useful tactic for enhancing the classification model's accuracy and balance is to duplicate data from the minority class without eliminating information from the majority class. Random Under Sampling (RUS), on the other hand, tends to be less ideal even while it performs better than the model without balancing. SVM + RUS only achieved accuracy 0.75, precision 0.71, recall 0.73, and F-score 0.71 at K=10. The possible loss of knowledge from the ruling class is the cause of this. The stability and dependability of model assessment are enhanced by the application of K-Fold Cross Validation techniques with higher K values.

5. References

[1] R. Syaputra, T. A. Y. Siswa, and W. J. Pranoto, “Model Optimasi SVM Dengan PSO-GA dan SMOTE Dalam Menangani High Dimensional dan Imbalance Data Banjir,” *Teknika*, vol. 13, no. 2, pp. 273–282, 2024, doi: 10.34148/teknika.v13i2.876.

[2] H. Faisal, A. Febriandirza, and F. N. Hasan, “Analisis Sentimen Terkait Ulasan Pada Aplikasi PLN Mobile Menggunakan Metode Support Vector Machine,” *KESATRIA J. Penerapan Sist. Inf. (Komputer Manajemen)*, vol. 5, no. 1, pp. 303–312, 2024.

[3] U. Islam and S. Agung, “Hal. 942,” vol. 2, no. 3, pp. 942–951, 2025.

[4] M. W. A. Putra, Susanti, Erlin, and Herwin, “Analisis Sentimen Dompet Elektronik Pada Twitter Menggunakan Metode Naïve Bayes Classifier,” *IT J. Res. Dev.*, vol. 5, no. 1, pp. 72–86, 2020, doi: 10.25299/itjrd.2020.vol5(1).5159.

[5] B. W. Sari and F. F. Haranto, “Implementasi Support Vector Machine Untuk Analisis Sentimen Pengguna Twitter Terhadap Pelayanan Telkom Dan Biznet,” *J. Pilar Nusa Mandiri*, vol. 15, no. 2, pp. 171–176, 2019, doi: 10.33480/pilar.v15i2.699.

[6] D. Alita and A. R. Isnain, “Pendeteksian Sarkasme pada Proses Analisis Sentimen Menggunakan Random Forest Classifier,” *J. Komputasi*, vol. 8, no. 2, pp. 50–58, 2020, doi: 10.23960/komputasi.v8i2.2615.

[7] M. Graciela and dan Hafiz Irsyad, “Klasifikasi Opini Masyarakat Terhadap Naturalisasi Pemain Sepak Bola Menggunakan KNN dan SMOTE,” *Aicoms*, vol. 3, no. 1, pp. 21–27, 2024, [Online]. Available: https://jurnal.politap.ac.id/index.php/aicoms

[8] A. Sitanggang, Y. Umaidah, Y. Umaidah, R. I. Adam, and R. I. Adam, “Analisis Sentimen Masyarakat Terhadap Program Makan Siang Gratis Pada Media Sosial X Menggunakan Algoritma Naïve Bayes,” *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 3, 2024, doi: 10.23960/jitet.v12i3.4902.

[9] A. Deolika, K. Kusrini, and E. T. Luthfi, “Analisis Pembobotan Kata Pada Klasifikasi Text Mining,” *J. Teknol. Inf.*, vol. 3, no. 2, p. 179, 2019, doi: 10.36294/jurti.v3i2.1077.

[10] V. K. S. Que, A. Iriani, and H. D. Purnomo, “Analisis Sentimen Transportasi Online Menggunakan Support Vector Machine Berbasis Particle Swarm Optimization,” *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 9, no. 2, pp. 162–170, 2020, doi: 10.22146/jnteti.v9i2.102.

[11] R. Tineges, A. Triayudi, and I. D. Sholihati, “Analisis Sentimen Terhadap Layanan Indihome Berdasarkan Twitter Dengan Metode Klasifikasi Support Vector Machine (SVM),” *J. Media Inform. Budidarma*, vol. 4, no. 3, p. 650, 2020, doi: 10.30865/mib.v4i3.2181.

[12] I. K. Dharmendra, I. M. Agus, W. Putra, and Y. P. Atmojo, “Evaluasi Efektivitas SMOTE dan Random Under Sampling pada Klasifikasi Emosi Tweet,” vol. 9, no. 2, pp. 192–193, 2024.

[13] S. Diantika, “Penerapan Teknik Random Oversampling Untuk Mengatasi Imbalance Class Dalam Klasifikasi Website

Phishing Menggunakan Algoritma Lightgbm,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 1, pp. 19–25, 2023, doi: 10.36040/jati.v7i1.6006.