Available online at www.coins-indocompt.org

COmputing and INformation Systems Journal

E-ISSN: 3109-3248

Vol. 1, No. 2, Augustus, 2025

THE EFFECT OF IMPERFECTIVE DATA SAMPLING METHOD ON SUPPORT VECTOR MACHINE ACCURACY

Ferdy Ilham Maulana[a*], Deni Arifianto[b], Reni Umilasari[c]

Muhammadiyah University of Jember, Jember 68121, Indonesia

ferdyilhamm@gmail.com

Abstract

Sentiment analysis is used to understand the direction of public opinion, but problems arise due to the unbalanced distribution of sentiment data, where one class dominates. This imbalance causes classification models such as Support Vector Machine (SVM) to be biased towards the majority class, which results in decreased accuracy and generalizability of the model. This study aims to assess the effectiveness of two data balancing techniques, namely, SVM-SMOTE, and ADASYN, in improving SVM performance. The research data was taken from social media platform X (Twitter), and testing was conducted using the K-Fold Cross Validation method (K=2, 5, and 10) using evaluation metrics such as accuracy, precision, recall, and F1-score. The results show that without data balancing, the SVM model can only achieve an average accuracy of 76.34% and F1-score of 62.38%, which reflects the weakness in recognizing minority classes. The application of the two balancing methods successfully improved the model performance. ADASYN increased the F1-score to 67.94%, while SVM-SMOTE showed the most optimal results with 82.4% accuracy and 74.02% F1-score. These findings indicate that SVM-SMOTE is the most effective technique in handling data imbalance and improving sentiment classification accuracy equally.

Keywords: data imbalance, sentiment analysis, SVM-SMOTE, ADASYN, SVM

1. Introduction

With the rapid development of the digital era, social media has become a virtual public space that allows people to express their opinions on various issues, including the naturalization of soccer players. These opinions are often divided into positive and negative sentiments, which are analyzed using natural language processing-based sentiment analysis techniques. However, one of the main challenges in applying this analysis is data imbalance, which refers to an uneven distribution of classes within the dataset, causing the model to be biased toward the majority class [1]. To address this issue, data imbalance sampling techniques such as SVM-SMOTE and ADASYN are employed. SVM-SMOTE generates synthetic samples from support vectors [1], while ADASYN adjusts the number of synthetic samples based on the classification difficulty level [2]. The application of these techniques in sentiment analysis of football player naturalization using the Support Vector Machine (SVM) algorithm is expected to improve accuracy and reduce model bias toward the majority class. This study aims to evaluate the effectiveness of these two balancing methods in improving SVM classification performance on imbalanced data, particularly in the context of public opinion regarding the naturalization policy for soccer players in Indonesia. The evaluation is based on accuracy, precision, recall, and F1-score metrics, with the hope of providing recommendations on the most appropriate method for handling data imbalance in social media sentiment analysis.

2. Methode

This study focuses on the effect of the Imbalance Data Sampling method on the performance of Support VectorMachine for sentiment analysis on the naturalization of soccer players. Figure 1 shows the stages of the study.

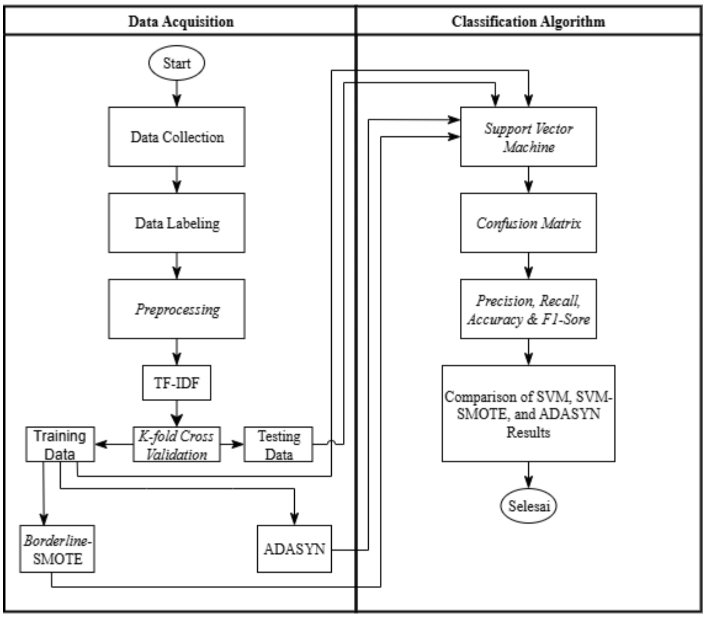

Figure 1. Research Stages

2.1. Data Collection

In this stage, the data used was obtained from comments posted on Twitter using the hashtag Naturalisasi. The data collection process was carried out using scraping techniques covering the period from January 1, 2021, to December 31, 2024.

2.2. Text Preprocessing

The initial stage of text processing, known as text preprocessing, involves removing noise from the comment data so that sentiment analysis can use it. The preprocessing stages consist of several steps, namely cleaning, case folding, tokenizing, stopword removal, and stemming.

2.2.1 Cleaning

Cleaning is the dataset cleaning stage, which includes removing ASCII characters, numbers, RTs, mentions, links, hashtags, URLs, punctuation marks, and spaces. The goal is to clean the dataset of meaningless characters or those that can influence sentiment [3].

2.2.2 Case Folding

Case Folding is a text normalization technique that transforms all alphabetical characters in the dataset to lowercase, from ‘A’ to ‘Z’ to ‘a’ to ‘z’. For example, the words ‘Program’ and ‘program’ will be considered different. In this step, all punctuation marks and numbers are removed so that the system can read the data more efficiently [4].

2.2.3 Tokenizing

Tokenization is the process of dividing a dataset into symbols or blocks to facilitate further processing [3]. For example, the sentence ‘I went to the market’ will be broken down into [‘I’, ‘went’, ‘to’, ‘market’].

2.2.4 Stopword Removal

Words that are frequently used in sentences but do not have significant meaning are removed [3]. Examples of words that are removed include ‘in’, ‘to’, ‘from’, ‘on’, ‘is’, ‘this’, ‘that’, and so on.

2.2.5 Stemming

Stemming is the process of removing prefixes or suffixes and converting words to their base form [3], to ensure that words related to the base word with the same meaning are spelled consistently. For example, the words ““selection”, “to choose”, and “selected” will become “choose”.

2.3. TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) is a technique used to assign weights to each word or phrase in a text to determine its relevance to a document [5]. Through this approach, words with high weight can be identified, which typically reflect the core or main topic of a text [6]. The first component of TF-IDF is Term Frequency (TF), which measures the frequency of a word's occurrence in a specific document. The TF value is obtained by dividing the number of occurrences of the word by the total number of words in the same document.

The second component is Inverse Document Frequency (IDF), which indicates how significant the word is across the entire document collection. Words that appear in many documents are considered less important and receive a low IDF weight. The IDF value is calculated using the logarithm of the ratio of the total number of documents to the number of documents containing the word. IDF is expressed by the following formula:

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

The TF and IDF values are multiplied to determine the TF-IDF value of a word in a document. The prominence of a word in a text is indicated by a higher TF-IDF value. The TF-IDF value is determined using the following formula:

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

2.4. K-Fold Cross Validation

K-Fold Cross Validation determines the accuracy of a method by partitioning data into training and testing sets, then applying the model and measuring its performance [7]. According to a study [8], cross-validation as a rotation estimate is a very important model validation technique in machine learning and applied statistics.

2.5. Support Vector Machine

Support Vector Machine is a method in machine learning for classification and regression. SVM is a supervised learning model-based method, involving training data to create a model that can predict the category or class of new data [9]. The steps in the Support Vector Machine (SVM) method include:

1. Identifying the most frequently occurring terms in each document or tweet analyzed.
2. Initial parameter values like ʿ ϵ=0.001, γ=0.5, λ=0.5, C=1, and α=0.5 are entered.
3. Perform matrix calculations using the formula:

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

Notes:
D_{ij} = the value of the element in the data matrix at position i, j
y_i = the label or class of data point i
y_j = the label or class of data point j
λ = the theoretical upper bound
$K(x_i \cdot x_j)$ = the kernel function used in the data mapping process

4. Each data point $n = 1,2,3,4, \ldots n$ is calculated using the following equation:

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

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

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

Notes:

E_i = error rate for data-i
γ = weight update rate
$max(i)D_{ij}$ = highest value of the diagonal elements of the Hessian matrix

5. The bias value (b) is calculated using the following equation:

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

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

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

$sign(h(x)) = +1$indicates that the decision belongs to the Positive class if the calculation result is greater than or equal to 0. Conversely, the decision belongs to the Negative class if the calculation value is less than 0 because $sign(h(x)) = -1$. The decision is determined through calculation using the following formula:

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

2.6. SVM-SMOTE

SVM-SMOTE is an oversampling technique that integrates Support Vector Machine (SVM) in selecting minority class samples that have the greatest impact on the decision boundary. This method uses the SVM principle to determine minority class support vectors, which are the points closest to the separating hyperplane between the majority and minority classes [2].

2.7. ADASYN

Adaptive Synthetic Sampling (ADASYN) is an adaptive oversampling method that generates synthetic data based on the local density of minority classes. This algorithm focuses on the most difficult areas to classify, where minority classes are weakly represented [2]. ADASYN improves model performance when dealing with imbalanced data sets, adjusting the total number of synthetic samples generated according to the local density of minority data [1].

3. Result

The Support Vector Machine (SVM) model was trained using data with various balancing techniques, namely SVM-SMOTE, ADASYN, and without balancing techniques. The evaluation was carried out using the K-Fold Cross Validation method with K = 2, 5, and 10. The evaluation metrics used included Precision, Recall, F1-score, and Accuracy, which were measured per class and also as a macro average.

3.1. SVM without Balancing

Table 1. SVM model results without balancing

K-Fold	*Accuracy*	*Precision*	*Recall*	*F1-Score*
2	0.7039	0.7544	0.5242	0.5250
5	0.7836	0.7964	0.6383	0.6594
10	0.8026	0.8078	0.6642	0.6871

Based on Table 1, it can be concluded that the larger the K value in K-Fold Cross Validation, the better the performance of the resulting model. This is demonstrated by the increase in accuracy, precision, recall, and F1-score values, with the best performance achieved at K=10.

3.2. SVM-SMOTE

Table 2. SVM model results with SVM-SMOTE balancing

Fold	*Accuracy*	*Precision*	*Recall*	*F1 Score*
K=2				
1	0,7621	0,7716	0,6151	0,6457
2	0,7908	0,8046	0,6599	0,6750
Average	0.7765	0.7881	0.6374	0.6603
K=5				
1	0,8170	0,8184	0,7254	0,7498
2	0,7876	0,8418	0,7611	0,7880
3	0,7908	0,8511	0,7324	0,7623
4	0,7614	0,8252	0,7382	0,7652

Fold	*Accuracy*	*Precision*	*Recall*	*F1 Score*
5	0,8235	0,8572	0,7684	0,7979
Average	0,8431	0,8387	0,7451	0,7726
K=10				
1	0,9020	0,9131	0,8133	0,8413
2	0,8431	0,8257	0,7340	0,7535
3	0,8366	0,8247	0,7221	0,7535
4	0,9085	0,9070	0,8365	0,8611
5	0,8562	0,8687	0,7812	0,8128
6	0,8235	0,8673	0,7214	0,7426
7	0,7908	0,8159	0,7065	0,7345
8	0,8301	0,8239	0,7530	0,7800
9	0,8693	0,8516	0,7700	0,7956
10	0,8627	0,8625	0,7675	0,8005
Average	0,8523	0,8560	0,7606	0,7875

Based on the evaluation results in Table 2, the performance of the SVM model with the SVM-SMOTE balancing technique shows an increasing trend as the K value increases in K-Fold Cross Validation. At K=2, the average accuracy is 0.7765 and the F1-score is 0.6603. These values increase at K=5 with an average accuracy of 0.8431 and an F1-score of 0.7726. The best performance is achieved at K=10 with an average accuracy of 0.8523 and an F1-score of 0.7875. These results indicate that selecting a larger K value can provide a more stable and representative model evaluation, as well as demonstrate the effectiveness of SVM-SMOTE in enhancing the model's ability to recognize data from minority classes.

3.3. ADASYN

Table 3. SVM model results with ADASYN balancing

Fold	*Accuracy*	*Precision*	*Recall*	*F1 Score*
K=2				
1	0,7255	0,7456	0,5714	0,6022
2	0,7477	0,7634	0,6106	0,6347
Average	0,7366	0,7545	0,5910	0,6185
K=5				
1	0,8235	0,8252	0,6991	0,7307
2	0,7908	0,7796	0,6636	0,6938
3	0,7876	0,8305	0,6664	0,6987
4	0,7516	0,7644	0,6428	0,6740
5	0,8170	0,8467	0,6898	0,7266
Average	0,7941	0,8093	0,6723	0,7048
K=10				
1	0,8431	0,8721	0,7228	0,7631
2	0,8105	0,7904	0,6898	0,7147
3	0,8170	0,8013	0,6657	0,6909
4	0,8301	0,8559	0,7195	0,7529
5	0,7974	0,8232	0,6823	0,7184
6	0,7908	0,8280	0,6849	0,7164
7	0,7386	0,7717	0,6324	0,6583
8	0,7712	0,7664	0,6620	0,6936
9	0,8235	0,8554	0,6978	0,7286
10	0,8105	0,8296	0,6740	0,7116
Average	0,8033	0,8194	0,6831	0,7149

Based on the evaluation results in Table 3, the SVM model using the ADASYN balancing technique shows improved performance as the K value increases in K-Fold Cross Validation. At K=2, the average accuracy and F1-score are 0.7366 and 0.6185, respectively. When the K value is increased to 5, the model performance also improves with an accuracy of 0.7941 and an F1-score of 0.7048. The best results were obtained at K=10, where the accuracy reached 0.8033 and the F1-score was 0.7149. These findings indicate that ADASYN can improve the model's ability to classify imbalanced data, especially in recognizing minority classes, and that larger K values result in more stable and representative evaluations.

Table 4. Evaluation Results for Each Method

Method	*K-fold*	*Accuracy*	*Precision*	*Recall*	*F1 Score*
Support Vector Machine	2	0.7039	0.7544	0.5242	0.5250
	5	0.7836	0.7964	0.6383	0.6594
	10	0.8026	0.8078	0.6642	0.6871
Average		0.7634	0.7862	0.6089	0.6238
SVM + SVM-SMOTE	2	0.7765	0.7881	0.6375	0.6603
	5	0.8431	0.8387	0.7451	0.7726
	10	0.8523	0.8560	0.7606	0.7875
Average		0.8240	0.8276	0.7144	0.7402
SVM + ADASYN	2	0.7366	0.7545	0.5910	0.6185
	5	0.7941	0.8093	0.6723	0.7048
	10	0.8033	0.8194	0.6831	0.7149
Average		0.7780	0.7944	0.6488	0.6794

Based on the evaluation results in Table 4, it shows that the application of data balancing techniques can improve the performance of the Support Vector Machine (SVM) model. Without balancing, SVM recorded an average accuracy of 0.7634, precision of 0.7862, recall of 0.6089, and F1-score of 0.6238. The low recall value indicates that the model is not yet optimal in recognizing all classes, especially those with fewer data points.

After applying the SVM-SMOTE technique, all evaluation metrics showed a significant improvement. The average accuracy increased to 0.8240, precision to 0.8276, recall to 0.7144, and F1-score to 0.7402. This proves that SVM-SMOTE helps the model to be more effective in recognizing minority classes, resulting in more balanced classification results.

The ADASYN technique also has a positive impact on model performance. With this method, the average accuracy is 0.7780, precision 0.7944, recall 0.6488, and F1-score 0.6794. Although still below the performance of SVM-SMOTE, ADASYN shows a significant improvement compared to the model without balancing.

Overall, the combination of SVM with SVM-SMOTE yields the best results across all evaluation metrics, followed by SVM with ADASYN, and finally SVM without balancing techniques. These findings reinforce that the use of data balancing methods, particularly SVM-SMOTE, plays a crucial role in improving classification accuracy on imbalanced datasets. This technique enhances model generalization and provides more stable evaluations, especially in validation schemes with higher K values.

4. Conclusion

This study was conducted to assess the effectiveness of the Support Vector Machine (SVM) algorithm in classifying sentiment in unbalanced data, both before and after applying the balancing method. Based on the test results, the SVM model without balancing techniques showed an average accuracy of 76.34%, but the F1-score was quite low at 62.38%. This indicates a tendency for the model to be more accurate in predicting the majority class, while performance on the minority class remains suboptimal, resulting in an uneven classification distribution.

To address this issue, two balancing techniques were applied, namely SVM-SMOTE and ADASYN, which significantly improved the performance of the SVM model. The ADASYN technique achieved an average accuracy of 77.80% with an F1-score of 67.94%. On the other hand, SVM-SMOTE showed the best results, with an average accuracy of 82.40% and an F1-score of 74.02%. Thus, it can be concluded that the SVM-SMOTE method is the most effective balancing approach in improving sentiment classification performance on imbalanced data using the SVM algorithm.

5. References

[1] Q. Meidianingsih, D. E. Wardani, E. Salsabila, and A. N. Mutia, “Perbandingan Performa Metode Berbasis Support Vector Machine untuk Penanganan Klasifikasi Multi Kelas Tidak Seimbang,” vol. 23, no. 1, pp. 8–18, 2023.

[2] M. Tiara *et al.*, “PEMANFAATAN ALGORITMA ADASYN DAN SUPPORT VECTOR MACHINE DALAM MENINGKATKAN AKURASI PREDIKSI KANKER PARU-PARU,” vol. 8, no. 5, pp. 8773–8778, 2024.

[3] A. Herdhianto, *Sentiment Analysis Menggunakan Naïve Bayes Classifier (NBC) pada Tweet Tentang Zakat*. 2020.

[4] E. Sonalitha, D. Setyawati, and S. Haryanto, “University transformation towards a learning experience facing the world of work and industry,” *J. Penelit.*, vol. 18, no. 2, pp. 40–54, 2021, doi: 10.26905/jp.v18i2.7066.

[5] 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.

[6] M. Y. Khan, A. Qayoom, M. S. Nizami, M. S. Siddiqui, S. Wasi, and S. M. K. U. R. Raazi, “Automated Prediction of Good Dictionary EXamples (GDEX): A Comprehensive Experiment with Distant Supervision, Machine Learning, and Word Embedding-Based Deep Learning Techniques,” *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/2553199.

[7] D. Cahyanti, A. Rahmayani, and S. A. Husniar, “Analisis performa metode Knn pada Dataset pasien pengidap Kanker Payudara,” *Indones. J. Data Sci.*, vol. 1, no. 2, pp. 39–43, 2020, doi: 10.33096/ijodas.v1i2.13.

[8] F. Tempola, M. Muhammad, and A. Khairan, “Perbandingan Klasifikasi Antara KNN dan Naive Bayes pada Penentuan Status Gunung Berapi dengan K-Fold Cross Validation,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 5, pp. 577–584, 2018, doi: 10.25126/jtiik.201855983.

[9] 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.