COMPILER
Vol. 14, No. 2, November 2025, pp. 65-74
ISSN: 2252-2403 (Print), ISSN: 2549-2403 (Online)
Accredited Fourth Grade by Kemenristekdikti, Decree No:10/C/C3/DT.05.00/2025
DOI: 10.28989/compiler.v14i2.3185

Multi-Label Opinion Mining Based on Random Forest with SMOTE and ADASYN

Ricy Ardiansyah[1], Herman Yuliansyah[2,*], Anton Yudhana[3]
[1]Master Program of Informatics, University Ahmad Dahlan, Yogyakarta, Indonesia
[2]Departement of Informatics, University Ahmad Dahlan, Yogyakarta, Indonesia
[3]Departement of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

Article Info

Article history:

Received July 15, 2025
Accepted August 28, 2025
Published November 30, 2025

Keywords:

Aspect Based Sentimen Analysis
Imbalance Data
Random Forest
Candidate Selection
Random Oversampling

ABSTRACT

Multi-label classification is essential to categorize data into multiple labels simultaneously. However, data imbalance poses a challenge, where some labels have much less representation, thus reducing the model performance. This study aims to propose a candidate-based sentiment analysis model on the 2024 Jakarta Presidential and Gubernatorial Election review. The SMOTE and ADASYN oversampling methods are applied to handle class imbalance. Both oversampling methods are compared with the Random Forest machine learning method. The experimental results show that. The experimental results show that in the classification of Presidential candidates, Random Forest achieves an accuracy of 0.947 with SMOTE and 0.948 with ADASYN. For sentiment labels, the accuracy of Random Forest remains high with a result of 0.989 for both SMOTE and ADASYN. In the classification of Jakarta Gubernatorial candidates, Random Forest + SMOTE produces an accuracy of 0.975, while with ADASYN it decreases slightly to 0.973. For sentiment labels, both SMOTE and ADASYN have the highest accuracy of 0.993. The application of SMOTE and ADASYN helps to improve the distribution of the minority class without decreasing the overall accuracy, as well as improving the stability in recognizing various multi-label classes in a balanced manner.

Corresponding Author:

Herman Yuliansyah,
Departement of Informatics,
Universitas Ahmad Dahlan Yogyakarta,
Jl. Ringroad Selatan, Kragilan, Tamanan, Kec. Banguntapan, Bantul, Daerah Istimewa Yogyakarta, Indonesia.
Email: *herman.yuliansyah@tif.uad.ac.id

1. INTRODUCTION

The significant growth of digital content on social media and other online platforms [1] has increased the need for effective text classification techniques [2], especially in the context of public opinion and sentiment analysis. One interesting context is the dynamics of Indonesian politics in 2024, where the Presidential election and the Jakarta Gubernatorial election are the main focus of public discussion. This phenomenon provides a relevant case study in multi-label text classification research, given the high volume of politically polarized public opinion [3]. Multi-label classification emerges as a powerful method in this context, as it is able to categorize a single opinion text into multiple labels at once, both in terms of candidates and sentiment [4]. However, a major challenge faced is data imbalance, where some candidate or sentiment labels have much less data representation than other labels [5]. This condition can cause the machine learning model to be biased towards the majority class, thus reducing the accuracy and generalization ability of the model [6].

In the context of multi-label text classification, each opinion data can contain more than one label [7]. Data imbalance causes machine learning models to have difficulty recognizing minority labels [8]. Therefore, it is necessary to apply oversampling methods such as Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic Sampling (ADASYN) which function to balance the distribution of data between labels [9].

This analysis chose to use Random Forest, an ensemble learning algorithm that is known to be strong in handling high-dimensional data and robust against overfitting [10]. Random Forest works by building many decision trees and combining the results, so that it is able to produce a more stable and accurate classification on multi-label data. However, data imbalance remains a challenge that can cause most trees to tend to predict the majority class, so it is necessary to balance the data using the SMOTE and ADASYN techniques [11]. This research aims to improve the performance of multi-label classification in the context of political opinion by applying Random Forest in combination with SMOTE and ADASYN oversampling techniques. Model evaluation was conducted using Accuracy, Precision, Recall, and F1 Score metrics to measure classification performance for each candidate label and sentiment [12]. Thus, this study is expected to expand the contribution in the development of multi-label opinion mining, specifically the Random Forest approach combined with oversampling strategies, to produce more accurate and fair political opinion classification for all labels, both in terms of candidates and sentiment [13].

The use of SMOTE and ADASYN oversampling over other techniques is due to the main problem in the dataset is class imbalance (uneven candidate labels and sentiment). This technique creates synthetic data on minority labels, helping the model not biased towards the majority. Unlike random oversampling, which only duplicates minority data, SMOTE and ADASYN generate new samples based on interpolation of the original data [14], thereby reducing the risk of overfitting. This study's novelty also includes the integration of SMOTE and ADASYN in the multi-label classification of Indonesian politics (presidential and gubernatorial candidates in 2024). This technique is applied in conjunction with Random Forest in the context of Aspect-Based Sentiment Analysis (ABSA) [15]. Experimental results show that this technique improves the stability and accuracy of minority classification without degrading overall performance.

2. RESEARCH METHOD

Based on Figure 1, the system begins with input of opinion data related to three Presidential candidates and three Governor candidates. This text data is converted into a numeric representation using TF-IDF. To overcome data imbalance, the SMOTE and ADASYN oversampling methods are applied to the candidate multi-label labels. Furthermore, the Random Forest model is trained using balanced data. Model performance evaluation is carried out through Cross Validation, with metrics such as Accuracy, Precision, Recall, and F1 Score to assess the effectiveness of Random Forest-based multi-label opinion classification.

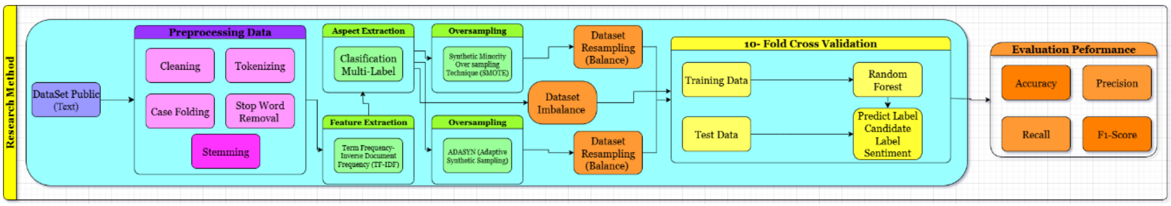

Figure 1. Research methods

2.1. Data Collection

This study consists of two datasets. The first dataset is the President and the second dataset is the Governor. The first dataset uses a data study by Asno Azzawagama https://data.mendeley.com/datasets/7w5zvr8jgp/5 [16]. The second data was collected via Twitter using the Tweet-Harvest method, which utilizes Twitter API authentication tokens to retrieve tweets and user metadata [17]. The presidential candidate dataset consists of three cleaned Excel files. Anies Baswedan with 10,001 tweets (6,455 positive, 3,546 negative). Prabowo Subianto with 10,002 tweets (7,369 positive, 2,633 negative). Ganjar Pranowo with 10,002 tweets (7,831 positive, 2,171 negative). The gubernatorial candidate dataset consists of three Excel files. Pramono Anung with 4,048 tweets (2,945 positive, 1,103 negative). Ridwan Kamil with 4,026 tweets (1,790 positive, 2,236 negative). Dharma Pongrekun with 4,024 tweets (1,919 positive, 2,105 negative). The majority of user tweets are in English because more analysis tools are available [18]. Data were collected using Python module scraping [19]. The scraped datasets were stored in CSV format for easy further analysis. The first and second datasets are shown in Table 1 and Table 2.

Description of the dataset and the suitability of the methods used. First, the presidential dataset: Anies, Prabowo, Ganjar - each = 10,000 tweets. Then, the governors of Jakarta: Pramono, Ridwan Kamil, Dharma- each = 4,000 tweets. The data is labeled with sentiment (Positive/Negative) and candidate. The presidential dataset is sourced from Asno Azzawagama and Twitter scraping results for the governor dataset. The suitability of the methods used are: Multi-label classification is suitable because a single tweet can have more than one candidate label and one sentiment. The dataset shows class imbalance (e.g., positive tweets from Prabowo and

Ganjar are seen in the presidential table). SMOTE and ADASYN are ideal because they are able to balance the label distribution. This improves the generalization of models such as Random Forest in imbalanced conditions.

Table 1. Presidential dataset

Presidential candidate	Number of data	Positive labels	Negative labels
Anies Baswedan	10001	6455	3546
Prabowo Subianto	10002	7369	2633
Ganjar Pranowo	10002	7831	2171

Table 2. Governor dataset

Governor candidate	Number of data	Positive labels	Negative labels
Pramono Anung	4048	2945	1103
Ridwan Kamil	4048	1790	2236
Dharma Pongrekun	4048	1919	2105

2.2. Preprocessing Data

Preprocessing includes data cleaning, case folding, tokenization, stop removal, and stemming [20]. The first process cleans the text and performs various cleaning processes, such as cleaning emojis, hashtags, URLs, and characters. Case Folding is used to change text into a more compact form. Tokenization, which is dividing text into smaller parts, so that tokens are formed which are single words. The stop word removal process removes words that occur frequently but do not contribute to the text. The use of stemming to reduce words to their basic form. Figure 2 explains the flow of pre-processed data for the Presidential and Gubernatorial Candidate datasets, which is part of the research process to obtain clean data, with the aim of facilitating further analysis [21].

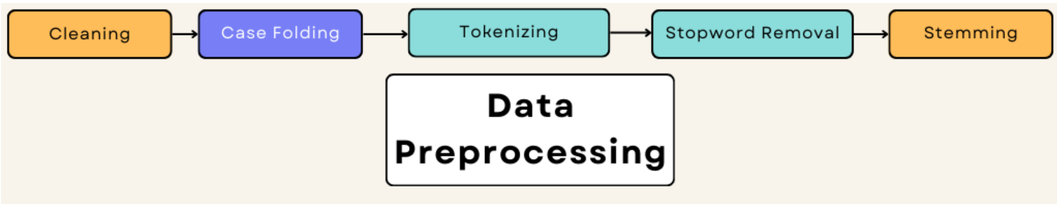

Figure 2. Data preprocessing

2.3. Feature Weighting

The preprocessing data is then converted to the next stage, namely vector form, so that it is easy to classify. Conversion can be done by giving weight to the features by means of weighting. Term Frequency-Inverse Document Frequency, (TF-IDF) is one of the weighting techniques that can be used to calculate the weight of words in text documents by considering the frequency of occurrence of words in the document and the entire corpus [22]. The process in TF-IDF is shown in Figure 3.

```
# TF-IDF Vectorization
tfidf = TfidfVectorizer(max_features=29723, stop_words='english', ngram_range=(1, 3), min_df=3)
X = tfidf.fit_transform(df['Text'])
tfidf = TfidfVectorizer(max_features=12099, stop_words='english', ngram_range=(1, 3), min_df=3)
X = tfidf.fit_transform(df['text'])
```

Figure 3. TF-IDF

2.4. Synthetic Minority Ovesampling Technique (SMOTE)

Data balancing using a model (SMOTE), functions to add synthetic samples to the minority class to overcome the inequality of data distribution. In this study, SMOTE is implemented through the imblearn.over sampling library in Python [23], with the aim of making the amount of data on both labels, both Presidential Candidates and Governor Candidates, more balanced. This is expected to improve the performance of the classification model by giving fairer weights to all classes. This function of Figure 4 is very important in the ABSA analysis flow because it helps balance the data distribution between candidate aspects, so that the classification model (e.g. Random Forest) can learn fairly from all categories. Thus, the accuracy of the model for the minority class will increase and not only dominate the majority class.

```python
def oversample_data(X_train, y_train, method="SMOTE"):
    if method == "SMOTE":
        sampler = SMOTE(random_state=42, k_neighbors=3)
    elif method == "ADASYN":
        # Define sampling_strategy as a dictionary
        sampling_strategy = {0: int(y_train[:, 0].sum() * 1.5),  # Tambahkan 50% sampel ke kelas 0
                             1: int(y_train[:, 1].sum() * 1.5),  # Tambahkan 50% sampel ke kelas 1
                             2: int(y_train[:, 2].sum() * 1.5)}  # Tambahkan 50% sampel ke kelas 2

        # Initialize ADASYN with the dictionary and increased n_neighbors
        sampler = ADASYN(random_state=42, n_neighbors=3, sampling_strategy=sampling_strategy)
    else:
        raise ValueError("Invalid method. Choose 'SMOTE' or 'ADASYN'.")

    X_resampled, y_resampled = sampler.fit_resample(X_train, y_train)
    return X_resampled, y_resampled
```

Figure 4. Oversampling candidate label

2.5. Adaptive Synthetic Sampling

The concept of ADASYN is to determine the weight distribution of minority samples against the learning difficulty of minority samples. ADASYN generates synthetic minority class samples by focusing on samples that are more difficult to detect in multi-label classification, samples that are difficult to classify get more weighted results of samples with higher weights [24]. Oversampling is done using the Adaptive Synthetic (ADASYN) technique by importing from the imblearn.over_sampling library. The Figure 5 function is an essential part of the machine learning pipeline, especially to handle imbalanced label distributions in training data. Label imbalance often occurs in multi-label classification tasks, such as in Aspect-Based Sentiment Analysis (ABSA), where sentiment aspects are less frequent than others. By using ADASYN, the model trained on oversampled data will be more balanced and have better classification ability for minority classes.

```python
# Oversample Training Data using SMOTE or ADASYN
def oversample_data(X_train, y_train, method="SMOTE"):
    if method == "SMOTE":
        sampler = SMOTE(random_state=42, k_neighbors=3)
    elif method == "ADASYN":
        sampler = ADASYN(random_state=42, n_neighbors=3)
    else:
        raise ValueError("Invalid method. Choose 'SMOTE' or 'ADASYN'.")

    X_resampled, y_resampled = sampler.fit_resample(X_train, y_train)
    return X_resampled, y_resampled
```

Figure 5. Oversampling sentiment label

2.6. Cross Validation

Overall, k-fold cross-validation is a useful method that provides a more accurate picture of how well a model performs, allows for fair model comparison, makes hyperparameter optimization easier, and helps find potential issues [25]. This method is often used in machine learning studies and practices to measure how well a model performs and how well it can generalize. In our study shown in Figure 6, k-fold cross-validation was used to evaluate the model. k-fold cross-validation is the process of dividing the data into k equal-sized datasets. Throughout k iterations, one-fold is selected as the test data, while the remaining k folds serve as the training data.

```python
def cross_validation_evaluation(X, y):
    """Perform cross-validation and display metrics for each fold."""
    model = OneVsRestClassifier(RandomForestClassifier(n_estimators=100, random_state=42))  # Use a fixed hyperparame

    fold = 1

    for train_index, test_index in KFold.split(X, y):
        X_train_fold, X_test_fold = X[train_index], X[test_index]
        y_train_fold, y_test_fold = y[train_index], y[test_index]

    # Train and predict using the model on each fold
        model.fit(X_train_fold, y_train_fold)
#       fit: Melatih model menggunakan data pelatihan (X_train_fold) dan labelnya (y_train_fold) dalam fold saat ini.
        y_pred = model.predict(X_test_fold)
#       Hasil prediksi model untuk set pengujian pada fold saat ini.

        # Calculate evaluation metrics for the current fold
        accuracy = accuracy_score(y_test_fold, y_pred)
        precision = precision_score(y_test_fold, y_pred, average='macro', zero_division=0)
        recall = recall_score(y_test_fold, y_pred, average='macro', zero_division=0)
        f1 = f1_score(y_test_fold, y_pred, average='macro', zero_division=0)
        hamming = hamming_loss(y_test_fold, y_pred)

        # Print fold results
        print(f"\nFold {fold}:")
        print(f"Accuracy: {accuracy:.4f}")
        print(f"Precision: {precision:.4f}")
        print(f"Recall: {recall:.4f}")
        print(f"F1 Score: {f1:.4f}")
        print(f"Hamming Loss: {hamming:.4f}")
        fold += 1
```

Figure 6. Cross validation

2.7. Aspect Based Sentiment Analysis and Random Forest

Aspect-Based Sentiment Analysis (ABSA) is a branch of sentiment analysis that is more specific and in-depth than the traditional sentiment approach. If conventional sentiment analysis only focuses on determining the overall polarity of an opinion (positive, negative, or neutral), then ABSA aims to identify and analyze sentiments directed at certain aspects of an entity in the text [26]. Overall, Figure 7 code snippets part of an important step in ABSA, namely converting data in text format and label lists into a numeric format that can be processed by machine learning algorithms such as Random Forest. This label binarization process allows the system to perform multi-aspect classification simultaneously, for example identifying public sentiment towards various political candidates or the various issues they support. In the political context, this process supports Opinion Target Extraction (OTE) to detect who is the target of an opinion, Aspect Category Detection (ACD) to determine aspect categories such as candidates or issues, and Sentiment Polarity (SP) to classify public sentiment towards these aspects [27].

```
from sklearn.preprocessing import MultiLabelBinarizer

    # Binarize the multilabel data
    mlb = MultiLabelBinarizer(classes=['Anies Baswedan', 'Prabowo Subianto', 'Ganjar Pranowo'])
    y = mlb.fit_transform(df['candidate'])
    # Binarize the multilabel data
    mlb = MultiLabelBinarizer(classes=['Pramono Anung', 'Ridwan Kamil', 'Dharma Pongrekun'])
    y = mlb.fit_transform(df['candidate'])
    # Clean text column
    df['Text'] = df['Text'].apply(clean_text)

    # Binarize the multilabel data
    mlb = MultiLabelBinarizer()
    y = mlb.fit_transform(df['label'])
```

Figure 7. Klasifikasi multi-label sentiment and candidate

Random Forest is one of the widely used group-based machine learning algorithms for regression and classification. This method produces a number of independently trained decision trees using bootstrap sampling techniques and random feature selection. To improve accuracy and reduce the possibility of overfitting, the final model results can be obtained through majority voting for classification or averaging for regression[28]. One of the main advantages of Random Forest is the ability to handle data with many features and imbalanced data and provide estimates of the importance of features in the prediction process. In addition, Random Forest can also estimate prediction errors automatically. The application of the Random Forest method is shown in Figure 8[29].

```
from sklearn.ensemble import RandomForestClassifier
# Train Model and Evaluate
def train_and_evaluate_model(X_train, y_train, X_test, y_test, mlb, resampling=False):
    model = OneVsRestClassifier(RandomForestClassifier(n_estimators=100, random_state=42))
    model.fit(X_train, y_train)

    y_pred_proba = model.predict_proba(X_test)
    y_pred = (y_pred_proba > 0.5).astype(int)
```

Figure 8. Random forest

2.8. Confusion Matrix Performance

In this study, Accuracy, Precision, and Recall were selected as evaluation indicators, which measure the performance of each classification model shown in Figure 9. Before calculating it, the values of the confusion matrix need to be known, namely TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative). Accuracy shows the proportion of the number of correct observations to the total observations [30]. Precision is the proportion of positive observations that correctly estimate the total number of positive predictions. Recall refers to the proportion of actual positive observations that are correctly identified which is calculated. F1 score or F1 measure is one of the most widely used measures in machine learning (ML) tasks. Calculating the F1 score value provides a way to combine Precision and Recall into a single measure that captures both properties [31].

```
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='macro', zero_division=0)
recall = recall_score(y_test, y_pred, average='macro', zero_division=0)
f1 = f1_score(y_test, y_pred, average='macro', zero_division=0)
hamming = hamming_loss(y_test, y_pred)

print("\n" + ("Balanced Data (SMOTE, ADASYN)" if resampling else "Imbalanced Data"))
print(classification_report(y_test, y_pred, target_names=mlb.classes_, zero_division=0))
print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1 Score: {f1:.4f}")
print(f"Hamming Loss: {hamming:.4f}")

return accuracy, precision, recall, f1, hamming
```

Figure 9. Confusion matrix

3. RESULTS AND ANALYSIS

3.1. Evaluation K-fold Random Forest

A total of 10 data subsets were used to ensure that the SMOTE and ADASYN oversampling models, combined with Random Forest, can produce consistent performance alternately as training data and testing data. Table 3 and Table 4 show the best results of the candidate label process, sentiment from the president and governor datasets.

Table 3. K-fold Labels Presidential Sentiment and Candidates 2024 Random Forest + SMOTE and ADASYN

K-fold	Accuracy (smote)	Precision (smote)	Recall (smote)	F1-score (smote)	Accuracy (adasyn)	Precision (adasyn)	Recall (adasyn)	F1-score (adasyn)
Sentiment Random Forest+SMOTE and ADASYN	0.881	0.871	0.825	0.844	0.881	0.871	0.825	0.844
Candidates Random Forest+SMOTE and ADASYN	0.882	0.927	0.885	0.904	0.882	0.927	0.885	0.904

Table 4. K-fold Labels Jakarta Gubernatorial Sentiment and Candidates 2024 Random Forest + SMOTE and ADASYN

K-fold	Accuracy (smote)	Precision (smote)	Recall (smote)	F1-score (smote)	Accuracy (adasyn)	Precision (adasyn)	Recall (adasyn)	F1-score (adasyn)
Sentiment Random Forest+SMOTE and ADASYN	0.969	0.969	0.968	0.969	0.969	0.969	0.968	0.969
Candidates Random Forest+SMOTE and ADASYN	0.974	0.978	0.976	0.977	0.974	0.978	0.976	0.977

Based on testing Table 3 and Table 4 using 10-fold cross-validation, the best results were obtained in the classification of Presidential sentiment, where both oversampling methods, SMOTE and ADASYN, produced high accuracy of 0.989. Precision and recall for positive and negative sentiments showed very balanced performance, with precision and recall values ranging from 0.97–1.00, and a consistent F1-score of 0.98–0.99. This shows that Random Forest is able to recognize sentiment accurately, without significant differences between the two oversampling methods. On the Presidential candidate label, the results were also very good. Both SMOTE and ADASYN recorded high accuracy, namely 0.947 (SMOTE) and 0.948 (ADASYN), with stable precision at 0.97–0.98, recall at 0.94–0.96, and F1-score reaching 0.96–0.97. This shows that Random Forest successfully recognizes each candidate with a high level of accuracy. In the DKI Jakarta Governor data, the model performance increased more significantly. For the gubernatorial candidate label, both SMOTE and ADASYN recorded an accuracy of 0.975 (SMOTE) and 0.973 (ADASYN), with precision in the range of 0.94–0.99, recall reaching 1.00, and F1-score consistently at 0.96–1.00, indicating very good prediction stability. Meanwhile, for the gubernatorial sentiment label, the accuracy is very high, namely 0.993, both with SMOTE and ADASYN. Precision and recall for positive and negative sentiments reached 0.99–1.00, and the F1-score was stable at 0.99. Overall, both oversampling techniques successfully corrected the unbalanced data distribution and significantly improved the performance of Random Forest, especially for the gubernatorial candidate and sentiment labels. There is no significant difference between SMOTE and ADASYN in terms of accuracy or stability of evaluation metrics, indicating that Random Forest is optimal in handling multi-label data, both with SMOTE and ADASYN.

3.2. Method Performance

The combination results of the Random Forest and SMOTE+ADASYN methods in this study are able to handle data imbalance from both labels effectively, as can be seen in the accuracy results for the multi-label classification context. The test results are shown in Table 5 and Table 6 for both labels with the presidential candidate dataset and Table 7 and Table 8 for the gubernatorial candidate dataset.

Table 5. 2024 Presidential Candidate Labels Random Forest SMOTE and ADASYN

Labels	Random Forest	Random Forest+SMOTE	Random Forest+ADASYN
Accuracy	0.946	0.947	0.948
Precision (Anies Baswedan)	0.97	0.97	0.98
Precision (Prabowo Subianto)	0.97	0.97	0.96
Precision (Ganjar Pranowo)	0.97	0.97	0.97
Recall (Anies Baswedan)	0.96	0.96	0.96
Recall (Prabowo Subianto)	0.94	0.94	0.95
Recall (Ganjar Pranowo)	0.95	0.95	0.95
F1-Score (Anies Baswedan)	0.97	0.97	0.97
F1-Score (Prabowo Subianto)	0.95	0.95	0.95
F1-Score (Ganjar Pranowo)	0.96	0.96	0.96

Table 6. 2024 Presidential Sentiment Labels Random Forest SMOTE and ADASYN

Labels	Accuracy	Precision (Positive)	Precision (Negative)	Recall (Positive)	Recall (Negative)	F1-Score (Positive)	F1-Score (Negative)
Random Forest	0.990	0.99	0.98	0.99	0.98	0.99	0.98
Random Forest+SMOTE	0.989	1.00	0.97	0.99	0.99	0.99	0.98
Random Forest+ADASYN	0.989	1.00	0.97	0.99	0.99	0.99	0.98

Based on the test results in Table 5 to Table 8, the evaluation of Random Forest performance with SMOTE and ADASYN oversampling shows very high and stable results, both on candidate labels and Presidential and Governor sentiments. These results indicate that Random Forest provides very good accuracy and consistency even without oversampling. In the classification of Presidential candidate labels, Random Forest without oversampling has recorded a high accuracy of 0.946. The combination with ADASYN provides an increase in accuracy to 0.948, better than SMOTE 0.947. In the precision metric for Anies, Prabowo, and Ganjar, the results remain stable with high numbers (0.97–0.98). For recall, the increase is not significant but remains stable (0.94–0.96), with a consistent F1-score for all candidates. In Presidential sentiment, Random Forest without oversampling has recorded an accuracy of 0.990, superior. Oversampling using SMOTE and ADASYN gave stable accuracy results (0.989), with precision reaching 1.00 on positive sentiment and stable recall at 0.99–0.97. F1-score was also consistent at 0.98–0.99. For the gubernatorial candidates, Random Forest without oversampling gave an accuracy of 0.975, and remained stable with SMOTE (0.975) and decreased slightly with ADASYN (0.973). The highest precision was obtained with ADASYN for Pramono (0.99), while Ridwan Kamil was better with SMOTE (0.96), and Dharma Pongrekun remained stable (1.00). Recall and F1-score for each candidate remained consistent at very high numbers (0.96–1.00).

Table 7. 2024 Gubernatorial Candidate Labels Random Forest SMOTE and ADASYN

Labels	Random Forest	Random Forest+SMOTE	Random Forest+ADASYN
Accuracy	0.975	0.975	0.973
Precision (Pramono Anung)	0.97	0.98	0.99
Precision (Ridwan Kamil)	0.96	0.96	0.94
Precision (Dharma Pongrekun)	1.00	1.00	1.00
Recall (Pramono Anung)	0.96	0.96	0.93
Recall (Ridwan Kamil)	0.97	0.97	0.99
Recall (Dharma Pongrekun)	1.00	1.00	1.00
F1-Score (Pramono Anung)	0.97	0.97	0.96
F1-Score (Ridwan Kamil)	0.97	0.97	0.96
F1-Score (Dharma Pongrekun)	1.00	1.00	1.00

On gubernatorial sentiment, Random Forest accuracy was at 0.993 both without oversampling and after applying SMOTE and ADASYN. Precision, recall, and F1-score for both positive and negative sentiments reached maximum values (0.99–1.00), indicating very optimal performance without significant changes after oversampling. From the results of this test, it can be concluded that: ADASYN oversampling is slightly superior in increasing precision on presidential candidates (0.98) and gubernatorial candidates (0.99 on Pramono). Random Forest showed more stable and high results overall, especially in handling imbalanced multi-label data.

Table 8. 2024 Gubernatorial Sentiment Labels Random Forest SMOTE and ADASYN

Labels	Accuracy	Precision (Positive)	Precision (Negative)	Recall (Positive)	Recall (Negative)	F1-Score (Positive)	F1-Score (Negative)
Random Forest	0.993	0.99	1.00	1.00	0.99	0.99	0.99
Random Forest+SMOTE	0.993	0.99	1.00	1.00	0.99	0.99	0.99
Random Forest+ADASYN	0.993	0.99	1.00	1.00	0.99	0.99	0.99

3.3. Oversampling Random Forest

The results after the oversampling technique was applied, on both labels gave significant changes especially in the differences before and after accuracy. This can be seen in the graph of Figure 10 of both labels in the presidential and gubernatorial candidate datasets.

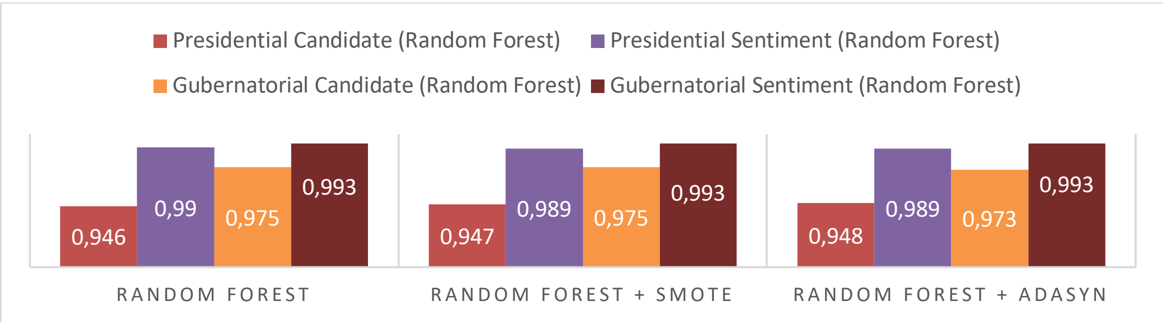

Figure 10. Random forest oversampling

The visualization results in Figure 2 show the accuracy after oversampling and the precision, recall, f1-score matrices for each label become stable compared to before oversampling, both in the Presidential Candidate and Governor Candidate datasets. The most significant increase occurred in the sentiment label, where previously there was an imbalance in the number of samples between categories, and after oversampling it became more even. With this more balanced data distribution, Random Forest is able to work more optimally in classifying candidate and sentiment labels, both in the Presidential Candidate and Governor Candidate data. This makes the model better able to recognize patterns that were previously underrepresented in training. The use of SMOTE and ADASYN has proven effective in increasing the amount of data in the minority class, thereby helping to improve the overall performance of the classification model. Random Forest, which previously had high performance, becomes even more optimal in handling data imbalance, maintaining stable accuracy in both the majority and minority classes.

The implications of the SMOTE and ADASYN approaches on model results significantly impact classification performance, particularly in the context of imbalanced labeled data. The SMOTE (Synthetic Minority Oversampling Technique) technique is suitable for situations where the minority is quite dense in the vector space. It improves recall and F1-score without compromising the accuracy of the majority. SMOTE's weaknesses arise when the effective distribution of minority labels is highly dispersed, as the interpolation tends to be less representative of the actual data patterns. Meanwhile, ADASYN (Adaptive Synthetic Sampling) has advantages in working with difficult-to-classify samples and focuses more on difficult-to-classify examples, giving greater weight to ambiguous portions of the dataset. It can improve precision, especially for minority candidates (e.g., Pramono Anung). It is suitable for multi-label applications because it detects "difficult regions" between labels more sensitively. The overall implication of applying SMOTE and ADASYN oversampling is that the model becomes fairer in classification, rather than solely relying on the majority label. This shows that Random Forest + ADASYN is slightly superior in some precision and stability metrics. However, high performance is maintained without oversampling, indicating the dataset is sufficiently rich.

4. CONCLUSION

Limitations of the results due to the specific dataset are limited. Excellent results on presidential and gubernatorial data may not be applicable to other domains. The quality of the Tweet data collected using scraping methods may contain noise or platform bias. Mild overfitting is possible because some labels were already very dominant before oversampling. Only two domains were tested (national politics & Jakarta), not including other provinces or non-political issues.

This study shows that ADASYN provides higher accuracy in improving precision on Presidential and gubernatorial candidates, compared to SMOTE. Random Forest shows more stable and high results overall. Handling imbalanced data in multi-label text classification is essential to improve classifier performance, especially for underrepresented labels. Random Forest, although powerful, can achieve competitive results when combined with techniques such as SMOTE and ADASYN. However, overall, SMOTE and ADASYN provide very similar results, with Random Forest maintaining high performance even without oversampling.

ACKNOWLEDGEMENTS

This research was supported by Universitas Ahmad Dahlan under Penelitian Tesis Magister (Master's Thesis Research) with grant number: PTM-025/SP3/LPPM-UAD/XII/2024 (09-12-2024).

REFERENCES

[1] P. Rita, N. António, dan A. Patrícia, "Social Media Discourse and Voting Decisions Influence : Sentiment Analysis in Tweets During an Electoral Period," *Soc. Netw. Anal. Min.*, vol. 13, no. 1, hal. 1–16, 2023, doi: 10.1007/s13278-023-01048-1.

[2] A. Patel, P. Oza, dan S. Agrawal, "Sentiment Analysis of Customer Feedback and Reviews for Airline Services Using Language Representation Model," *Procedia Comput. Sci.*, vol. 218, hal. 2459–2467, 2023, doi: 10.1016/j.procs.2023.01.221.

[3] A. Nahid, D. Pramesti, A. Fathurahman, dan H. Fakhrurroja, "Exploring Sentiment Analysis for the Indonesian Presidential Election Through Online Reviews Using Multi-Label Classification with a Deep Learning Algorithm," *Multidiscip. Digit. Publ. Inst. Inf.*, vol. 15, hal. 1–33, 2024, doi: https://doi.org/10.3390/info15110705.

[4] G. Manias, A. Mavrogiorgou, A. Kiourtis, C. Symvoulidis, dan D. Kyriazis, "Multilingual Text Categorization and Sentiment Analysis : A Comparative Analysis of The Utilization of Multilingual Approaches for Classifying Twitter Data," *Neural Comput. Appl.*, vol. 35, no. 29, hal. 21415–21431, 2023, doi: 10.1007/s00521-023-08629-3.

[5] A. Ameur, S. BenHamdi, dan S. Ben Yahia, "Multi-Label Learning for Aspect Category Detection of Arabic Hotel Reviews Using AraBERT," *Sci. Technol. Publ.*, vol. 2, hal. 241–250, 2023, doi: 10.5220/0011694800003393.

[6] Erlin, Y. Desnelita, N. Nasution, L. Suryati, dan F. Zoromi, "Dampak SMOTE terhadap Kinerja Random Forest Classifier berdasarkan Data Tidak Seimbang," *J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 21, no. 3, hal. 677–690, 2022, doi: 10.30812/matrik.v21i3.1726.

[7] A. A. Firdaus, A. Yudhana, dan I. Riadi, "Public opinion analysis of presidential candidate using naïve bayes method," *Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 2, hal. 563–570, 2023, [Daring]. Tersedia pada: https://doi.org/10.22219/kinetik.v8i2.1686

[8] I. A. Rahma dan L. H. Suadaa, "Penerapan Text Augmentation Untuk Mengatasi Data Yang Tidak Seimbang Pada Klasifikasi Teks Berbahasa Indonesia Studi Kasus: Deteksi Judul Clickbait Dan Komentar Hate Speech Pada Berita Online," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 6, hal. 1329–1340, 2023, doi: 10.25126/jtiik.2023107325.

[9] A. Azzawagama, A. Yudhana, dan I. Riadi, "Indonesian presidential election sentiment : Dataset of response public before 2024," *Data Br.*, vol. 52, hal. 109993, 2024, doi: 10.1016/j.dib.2023.109993.

[10] A. Basuki, "Sentiment Analysis of Customers ' Review on Delivery Service Provider on Twitter Using Naive Bayes Classification," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 9, no. 2, hal. 420–428, 2023, doi: 10.26555/jiteki.v9i2.26327.

[11] B. Erkantarci dan G. Bakal, "An Empirical Study of Sentiment Analysis Utilizing Machine Learning and Deep Learning Algorithms," *J. Comput. Soc. Sci.*, vol. 7, no. 1, hal. 241–257, 2024, doi: 10.1007/s42001-023-00236-5.

[12] H. Yuliansyah, S. A. Mulasari, F. A. Ghozali, dan B. Sudarsono, "Sentiment Analysis of the Waste Problem based on YouTube comments using VADER and Deep Translator," *J. MEDIA Inform. BUDIDARMA*, vol. 8, no. 1–11, hal. 663–673, 2024, doi: 10.30865/mib.v8i1.6918.

[13] R. Ardiansyah, H. Yuliansyah, dan A. Yudhana, "Multi-Label Classification for Opinion Mining in Presidential Election Using Tf-Idf With NB And SVM," *J. Tek. Elektro, Teknol. Inf. dan Komput.*, vol. xx, hal. 1–13, 2025.

[14] M. P. Pulungan, A. Purnomo, dan A. Kurniasih, "Penerapan SMOTE Untuk Mengatasi Imbalance Class Dalam Klasifikasi Kepribadian MBTI Menggunakan Naive Bayes Classifier," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, no. 5, hal. 1033–1042, 2024, doi: 10.25126/jtiik.2024117989.

[15] H. Y. Hisyam Agus Setiawan, "Analisis Sentimen Berbasis Aspek Terhadap Ulasan Pengguna pada Game Honkai: Star Rail Menggunakan Naïve Bayes Classifier," *J. Sist. Inf.*, vol. 13, no. 5, hal. 1956–1971, 2024.

[16] A. A. Firdaus, A. Yudhana, dan R. Imam, "Analisis Sentimen Pada Proyeksi Pemilihan Presiden 2024 Menggunakan Metode Support Vector Machine," *J. Pendidik. Teknol. Inf.*, vol. 3, no. 2, hal. 236–245, 2024, [Daring]. Tersedia pada: http://dx.doi.org/10.51454/decode.v3i2.172

[17] M. M. Adam, N. M. Sabri, U. M. Fatihah, N. Hrishvanthika, dan N. Isa, "Sentiment Analysis on Acceptance of

COVID-19 Vaccine for Children based on Support Vector Machine," *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 2, no. 2, hal. 252–270, 2026, [Daring]. Tersedia pada: https://doi.org/10.37934/araset.58.2.252270

[18] Y. Qi dan Z. Shabrina, "Sentiment Analysis Using Twitter Data : A Comparative Application of Lexicon - and Machine - Learning - Based Approach," *Soc. Netw. Anal. Min.*, vol. 13, no. 1, hal. 1–14, 2023, doi: 10.1007/s13278-023-01030-x.

[19] M. Verma *et al.*, "People ' s Perceptions on COVID - 19 Vaccination : an Analysis of Twitter Discourse from Four Countries," *Sci. Rep.*, hal. 1–11, 2023, doi: 10.1038/s41598-023-41478-7.

[20] N. C. Anggista Oktavia Praneswara, "Analisis Sentimen Ulasan Aplikasi TikTok Shop Seller Center di Google Playstore Menggunakan Algoritma Naive Bayes," *Indones. J. Comput. Sci.*, vol. 12, no. 1, hal. 3925–3940, 2023.

[21] Yuyun, N. Hidayah, dan S. Supriadi, "Algoritma Multinomial Naïve Bayes Untuk Klasifikasi Sentimen Pemerintah Terhadap Penanganan Covid-19 Menggunakan Data Twitter," *Rekayasa Sist. dan Teknol. Inf.*, vol. 5, no. 10, hal. 820–826, 2021.

[22] R. Saputra dan M. G. Pradana, "Implementasi Algoritma Cosine Similarity dan TF- IDF dalam Menentukan Rumpun Jabatan," *J. Tek. Inform.*, vol. 12, no. 1, hal. 1–11, 2024, doi: 10.32832/kreatif.v12i1.15470.

[23] V. Pranatawijaya, N. Noor, A. Resha, E. Christian, dan G. Septian, "Unveiling User Sentiment : Aspect-Based Analysis and Topic Modeling of Ride-Hailing and Google Play App Reviews," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 10, no. 3, hal. 328–339, 2024, doi: http://dx.doi.org/10.20473/jisebi.10.3.328-339.

[24] M. Zakariah, S. A. Alqahtani, dan M. S. Al-Rakhami, "Machine Learning-Based Adaptive Synthetic Sampling Technique for Intrusion Detection," *Multidiscip. Digit. Publ. Inst. Appl. Sci.*, vol. 13, hal. 2–31, 2023, [Daring]. Tersedia pada: https://doi.org/10.3390/app13116504

[25] V. W. Lumumba dan D. Kiprotich, "Comparative Analysis of Cross-Validation Techniques : LOOCV , K-folds Cross-Validation , and Repeated K-folds Cross-Validation in Machine Learning Models," *Am. J. Theor. Appl. Stat.*, vol. 13, no. 5, hal. 127–137, 2024, doi: 10.11648/j.ajtas.20241305.13.

[26] Y. Cathy, D. Paul, W. Jörg, dan T. Katerina, *A systematic review of aspect - based sentiment analysis : domains , methods , and trends*, vol. 57, no. 11. Springer Netherlands, 2024. doi: 10.1007/s10462-024-10906-z.

[27] E. Xu, J. Zhu, L. Zhang, Y. Wang, dan W. Lin, "Research on Aspect-Level Sentiment Analysis Based on Adversarial Training and Dependency Parsing," *Multidiscip. Digit. Publ. Inst.*, hal. 1–16, 2024, [Daring]. Tersedia pada: https://doi.org/10.3390/electronics13101993

[28] T. M. Lange, M. Gültas, A. O. Schmitt, dan F. Heinrich, "optRF : Optimising random forest stability by determining the optimal number of trees," *BioMed Cent. Bioinforma. J.*, hal. 1–21, 2025, doi: 10.1186/s12859-025-06097-1.

[29] J. Ding, J. Du, H. Wang, dan S. Xiao, "OPEN A novel two-stage feature selection method based on random forest and improved genetic algorithm for enhancing classification in machine learning," *Sci. Rep.*, hal. 1–16, 2025, [Daring]. Tersedia pada: https://doi.org/10.1038/s41598-025-01761-1

[30] S. Sathyanarayanan dan B. R. Tantri, "Confusion Matrix-Based Performance Evaluation Metrics," *African J. Biomed. Res.*, vol. 27, no. 4, hal. 1–9, 2024, [Daring]. Tersedia pada: doi: https://doi.org/10.53555/AJBR.v27i4S.4345

[31] D. A. Pamungkas, I. L. Kharisma, D. S. Simatupang, dan Kamdan, "Implementation of Deep Neural Network in the Design of Ethereum Blockchain Scam Token Detection Applications," *J. Sist. Inf.*, vol. 12, no. 3, hal. 927–937, 2023.