Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi
Vol.9 No.2 July 2024, P-ISSN : 2502-3470, E-ISSN : 2581-0367

SMOTE-Tomek Re-sampling Based on Random Forest Method to Overcome Unbalanced Data for Multi-class Classification

Dhiaka Shabrina Assyifa[1], Ardytha Luthfiarta[2*]

[1,2]*Informatics Engineering Department, Dian Nuswantoro University, Semarang, Indonesia*

[1]dhiakashabrinaassyifa54@gmail.com

[2]ardytha.luthfiarta@dsn.dinus.ac.id (*)

Received: 2024-06-10; Accepted: 2024-07-18; Published: 2024-07-28

***Abstract*—**Mobile app review data needs to be utilized to understand the app characteristics desired by users. App providers can improve app performance based on user preferences by using sentiment and emotion classification on app review data. However, problems that often arise in text-based analysis are data variation and data imbalance. This can lead to biased and inaccurate classification models. It is necessary to perform pre-processing to comprehend the data requirements and implement feature extraction for word weighting to overcome the variation in the data. In addition, re-sampling techniques are also needed to overcome the imbalance in sample distribution. Re-sampling techniques such as Tomek Links and SMOTE only focus on majority or minority data. This research applies the SMOTE-Tomek merging technique, aiming at not only the minority data but also the majority data. The model performance becomes better because the technique combines oversampling and under-sampling of the majority of data to eliminate the noise of data. The data was modeled using an Ensemble Learning Random Forest for classification. The model performance resulted in a Precision value of 84%, Recall of 84%, F1-Score of 84%, and Accuracy of 84%. Furthermore, the model was optimized using GridSearchCV and obtained an increase in Precision 85%, Recall 85%, F1-Score 85%, and Accuracy 85%.

***Keywords*—** Classification; Combinate Sampling; Sample Distribution Imbalance; SMOTE-Tomek; Random Forest.

I. INTRODUCTION

As part of technological advances, mobile applications have integrated into the lives of the global community. Various groups use mobile applications to meet various needs, ranging from education, health, entertainment, and e-commerce to financial transactions [1]. Based on data from Statista, the use of mobile applications in Indonesia continues to grow. The total number of downloads reached 7.31 billion in 2021, 7.7 billion in 2022, and 7.56 billion in 2023, a reduction from the previous year's figure of 7.31 billion [2]. Despite the decline, there are still many mobile app downloads in Indonesia.

The use of mobile applications has generated many user reviews in text form. The reviews provide deep insights into the user experience of the mobile app used, as well as user feedback regarding the appearance and performance of the app [3]. To determine the conclusions of the arguments presented by the mobile app users, rather than relying on the ratings given by the users, sentiment analysis is needed, one of the topics in Natural Language Processing (NLP). Sentiment analysis aims to study, extract, and identify subjective information from people's expressions, opinions, or emotions toward a topic. This analysis usually categorizes the sentiments into three categories: positive, negative, and neutral [4]. Sentiment analysis and emotional understanding are intertwined in the context of mobile app reviews. Emotions can influence people's opinions, thoughts, or feelings based on biological and psychological states [5]. Mobile app review data can support the classification of the emotions of mobile app users. Companies or stakeholders use the results to understand the feature characteristics and service needs that users want about their products and services.

However, significant challenges need to be overcome in the sentiment analysis process. The two most common challenges are data variation and unbalanced data distribution. Data variation arises from the diversity of languages users use, giving rise to ambiguity in text analysis. In addition, the problem of unbalanced data distribution becomes more complex in the case of multi-class, where each class has a different amount of data. If some classes have more data than others, the model will tend to learn the pattern more easily [6]. As a result, the analysis model becomes biased and inaccurate due to the decrease in classification performance in the minority classes. The model also tends to predict the majority class in the predicted data and ignore the minority class, so the accuracy of the prediction result is biased towards the majority class [7].

To overcome data imbalance, re-sampling techniques such as oversampling using SMOTE (Synthetic Minority Over-sampling Technique) and undersampling using Tomek Links are often applied. The combination of these two techniques, known as SMOTE-Tomek, is often used to handle significant data imbalance between classes [8]. SMOTE-Tomek helps to create a more balanced distribution of data and removes data that creates ambiguity between classes. This combination helps to make the classification model more accurate and reliable by minimizing the bias towards the majority class and increasing the accuracy of the minority class [9].

The research [10] on personality classification on Twitter using the SVM, Decision Tree, Random Forest, Ada Boost, and Gradient Boosti algorithms with the SMOTE-Tomek, Random Undersampling, and SMOTE re-sampling technique, it was found that SMOTE-Tomek performed best among other re-

DOI : https://doi.org/10.25139/inform.v9i2.8410

sampling techniques. In addition, the SVM performed better than the boosting algorithm. The SVM model, with an initial accuracy of 39%, became 55% after SMOTE-Tomek, then optimized with tuning so that the accuracy was 56% [10].

Furthermore, research [7] analyzed the sentiment analysis of application reviews using Random Forest, Neural Network, KNN, and SVM models. Accuracy increases when applying Tomek Links. The SVM model achieved 81% and Random Forest 80%, while KNN and NN accuracy was still below 77%. Tomek Links can clean data that has the potential to become noise from the majority class that has similar characteristics to the minority class, resolve class imbalance, and improve model performance [7]. Additionally, in Research [11] regarding the classification of MBKM program comments using Random Forest, Logistic Regression, MLP, and SVM algorithms, the Tomek Links undersampling technique works better than the Near Miss technique [11]. In the research [12] on text classification using the IndoBERT model, the use of the SMOTE technique produced a better accuracy value of 82% compared to using augmentation techniques which only produced a value of 78%, because SMOTE can improve classification capabilities by adding synthetic data to the minority class based on nearest neighbours [12].

The literature review shows that SMOTE-Tomek, Tomek Links, and SMOTE re-sampling techniques effectively overcome class imbalance and improve model performance in various classification applications. Classification algorithms such as Random Forest (RF) and Support Vector Machine (SVM) are widely used because they can learn complex patterns in textual data, making them suitable for various tasks such as sentiment analysis [7][10][11]. However, the use of this algorithm for multi-class classification in the context of sentiment analysis of mobile application reviews is still rare. Previous studies also did not use feature representations such as TF-IDF [10], and some also limit the number of feature names used, which may affect the classification results [11].

Therefore, this research aims to evaluate the effectiveness of the SMOTE-Tomek technique for multi-class classification using TF-IDF for feature representation. The effectiveness of SMOTE-Tomek will be compared to SMOTE and Tomek Links to determine if it provides additional benefits in enhancing the performance of complex sentiment analysis models. Additionally, *Hyperparameter tuning* using GridSearchCV will be applied to optimize the best-performing classification model to achieve higher accuracy. This research is expected to contribute meaningfully to developing multi-class text classification models by addressing data variations and using re-sampling techniques to overcome dataset imbalances.

II. Research Methodology

This research uses several stages. It starts with the retrieval of the labelled dataset. Then, through data pre-processing, the data is cleaned from noise. Next, feature extraction using TF-IDF. After that, data imbalance is addressed using Tomek Links, SMOTE, and SMOTE-Tomek. Subsequently, the model with the best performance is optimized using GridSearchCv hyperparameter tuning. The following is Figure 1 of the research methodology.

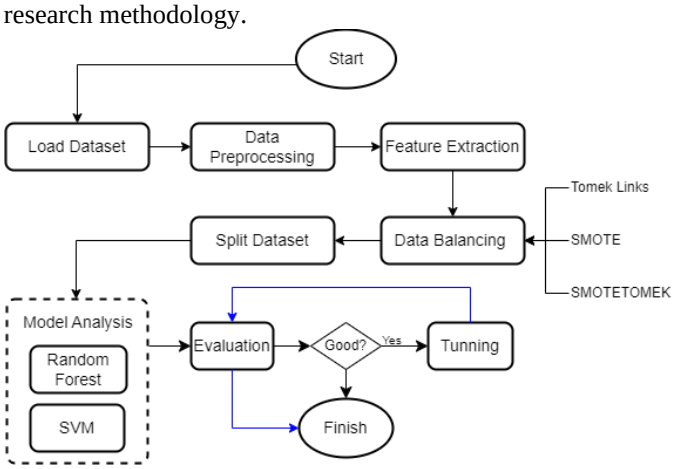

Figure 1. Research Methodology

A. Load Dataset

The dataset is taken from research conducted by Ricco San and Karen, who produced the Multilabel Sentiment and Emotion Dataset from the Indonesian Mobile Application Review [13]. This dataset comes from reviews of 10 mobile applications in Indonesia. User review data on mobile applications is textual. The limited amount of textual data in Indonesian makes this dataset need further development for text analysis-based research.

B. Data Pre-processing

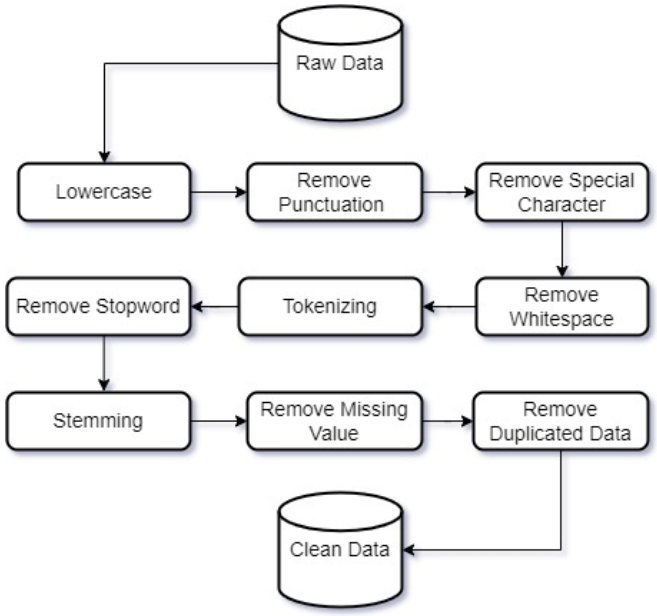

Figure 2. Data Pre-processing Workflow

Data pre-processing is the process of cleaning, transforming, and tidying up data to get better quality data suitable for further analysis. The Multilabel Sentiment and Emotion Dataset from Indonesian Mobile Application Review has gone through several stages in data pre-processing,

DOI : https://doi.org/10.25139/inform.v9i2.8410

including removing base duplicates, removing URLs, removing mention/hashtag/special characters, removing emoji, removing dupes, removing enter or new line format, and remove mobile apps rating data column. However, this dataset needs additional pre-processing because the data is still fairly unstructured, which can affect classification performance. Figure 2 explains the additional pre-processing workflow for this dataset.

1) Lowercase: This process of case folding converts letters in the text to all lowercase. It ensures consistent data and avoids differences in understanding caused by capitalization, which can lead to unnecessary duplication of words [14]. For example, "*Bintang*" becomes "*bintang*" to standardize their representation.

2) Remove Punctuation: Removing punctuation from the text helps focus on important words and improves consistency in text analysis [15]. For example, "*yah..kenapa*" becomes "*yah kenapa*".

3) Remove Special Character: Removing non-alphanumeric characters from the text [16]. These characters do not add meaning to text understanding and can cause noise in the algorithm. For example, "©®®©".

4) Remove Whitespace: This step removes unnecessary whitespace, such as at the beginning or end of a sentence, double spaces, or those resulting cleaning processes [17]. For Example "*tolong perbaiki*" becomes "*tolong perbaiki*".

5) Tokenizing: Tokenizing dividing text into smaller units called tokens. There are two ways methods: sentence tokenizing and word tokenizing. Word tokenizing separates sentences into their constituent words [15]. One of the word tokenizing techniques is RegexpTokenizer, which uses regular expressions to split strings into substrings. For example, "*tolong perbaiki sistem pembayaran juga*" becomes ["*tolong*", "*perbaiki*", "*sistem*", "*pembayaran*", "*juga*"].

6) Remove Stop Words: Stop words have little meaning in the text. Stop words must be filtered and removed so the algorithm can focus more on words that provide meaning to the text. To do the removal process, we can use the Natural Language ToolKit (NLTK) library with Indonesian stopwords.words('Indonesian'), Sastrawi (StopWordRemoverFactory(), and other stop words collected by ourselves [15]. For example ["*tolong*", "*perbaiki*", "*sistem*", "*pembayaran*", "*juga*"] becomes ["*tolong*", "*perbaiki*", "*sistem*", "*pembayaran*"].

7) Stemming: This text normalization technique removes affixes from words to obtain their base form. The Sastrawi library, specifically StemmerFactory(), can be used for this [15]. For example, "*menyenangkan*" becomes "*senang*".

8) Remove Missing Value: Empty or NaN data entries are removed from the dataset [18]. For example, reviews that are empty from the beginning or empty due to cleaning in the pre-processing stage.

9) Remove Duplicated Data: Data entries that are identical and appear more than once in the dataset [18]. For example, if multiple identical reviews exist, only one is retained while the duplicates are removed.

C. Feature Extraction

Term Frequency-Inverse Document Frequency (TF-IDF) is a feature extraction method frequently used to convert text data into a numerical form in NLP using Equations (1) to (3).

$$TF(X,d) = \frac{Count\ of\ term\ X\ in\ document\ d}{total\ number\ of\ term\ in\ document\ d} \quad (1)$$

$$IDF(X) = log\ \frac{Count\ of\ document\ d\ in\ Corpus}{Count\ of\ document\ d\ have\ a\ term\ X} \quad (2)$$

$$Weight(X,d) = TF(X,d) * IDF(X) \quad (3)$$

We can use TfidfVectorizer() from the scikit-learn library to implement TF-IDF. TfidfVectorizer() enables efficient tokenization, weighting, and encoding of new text. Once the text data is converted into numerical representation, these features can be used for various NLP tasks [19].

D. Data Balancing

Data balancing works by balancing the data in each class in the dataset. This can achieved through several methods, namely oversampling, undersampling, and combined re-sampling [8]. Figure 3 flowchart of the Tomek Links, SMOTE, and SMOTE-Tomek re-sampling techniques.

1) Undersampling Tomek Links: This method works by removing samples from the majority class closer to the minority class, called Tomek Links. These pairs consist of the nearest neighbours of different classes. Therefore, although the class distribution does not change, the dataset becomes cleaner, and the boundaries between classes become clearer [20]. Figure 4 is an illustration of the Tomek Links process.

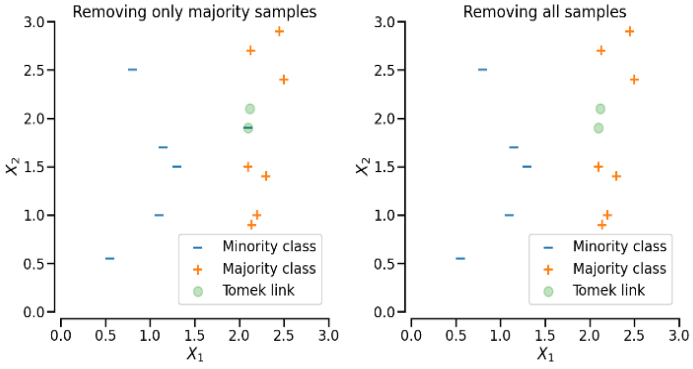

Figure 3. Illustration of the definition of Tomek links [8]

DOI : https://doi.org/10.25139/inform.v9i2.8410

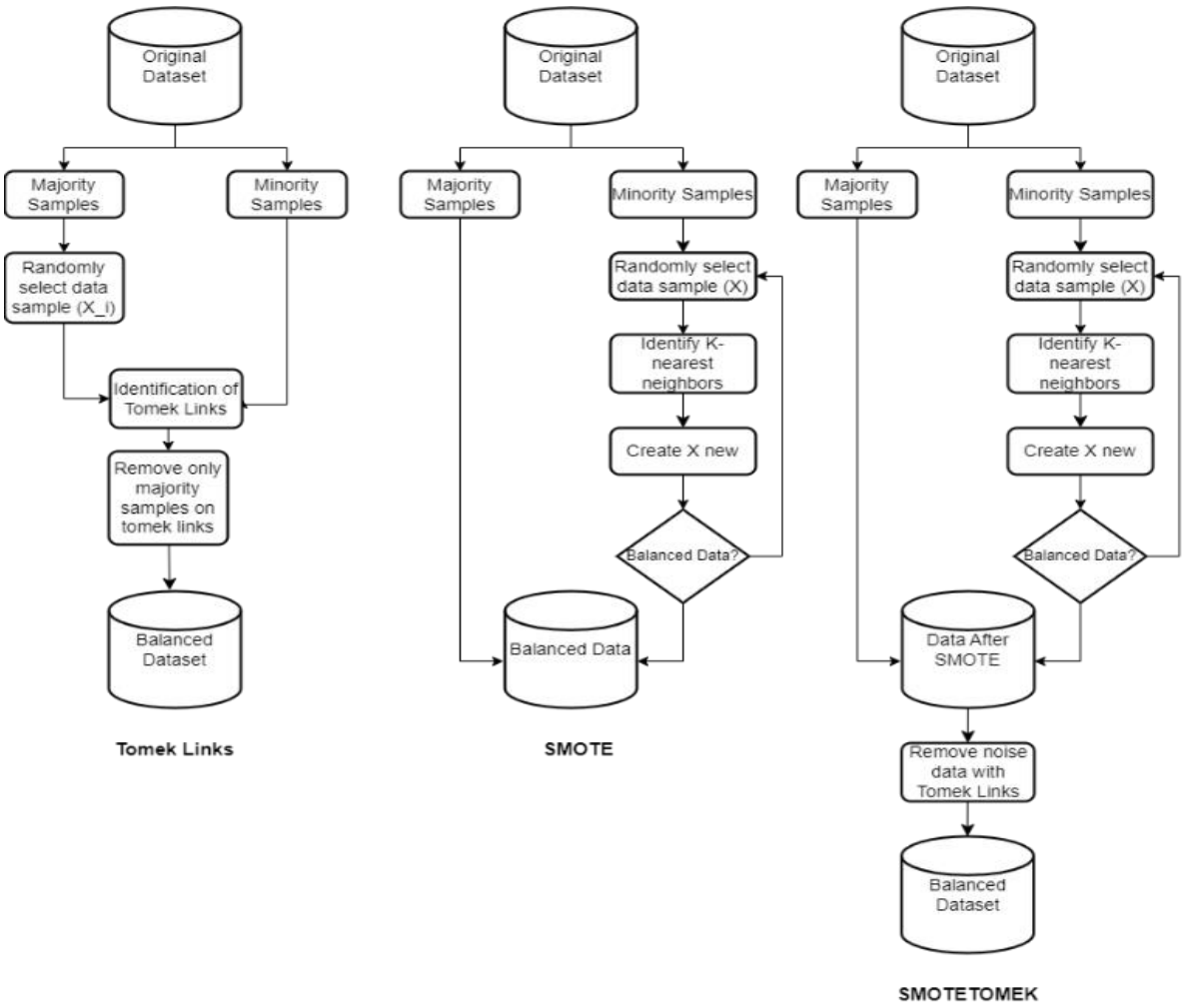

Figure 4. Flowchart Re-sampling Technique

2) Oversampling SMOTE: The synthetic minority oversampling technique (SMOTE) generates new samples from minority classes. The process is done by randomly taking k nearest neighbours from each sample in the minority class. It then creates new synthetic samples between samples with k randomly selected nearest neighbours [21]. Using SMOTE can balance the class distribution in the dataset, thus avoiding the problem of overfitting. Figure 5 is an illustration of the SMOTE process [22].

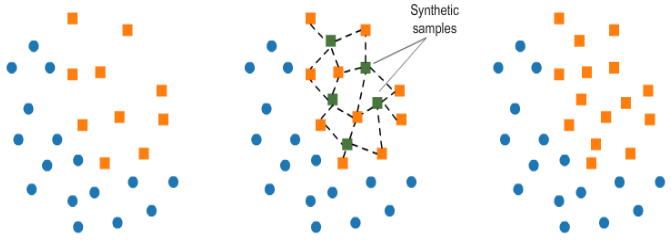

Figure 5. Illustration of The Definition of a SMOTE

3) Combined Re-sampling (SMOTE-Tomek): A combination of SMOTE oversampling and Tomek Links under-sampling techniques [21]. The process is done by combining the capabilities of SMOTE, which generates synthesized data for minority classes, and Tomek Links, which removes data from majority classes identified as Tomek links [23]. Figure 6 is an illustration of the SMOTE-Tomek process, where Figure 6(a) uses the original dataset, Figure 6(b) uses the dataset after SMOTE, Figure 6(c) uses Tomek link identification, and Figure 6(d) uses examples of removed boundaries and noise [24]**.**

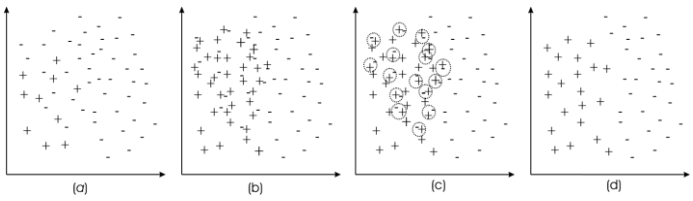

Figure 6. Balancing The Dataset

E. Split Dataset

After handling the data imbalance, the next step is to divide the data into training data and test data, with a ratio of 80:20.

F. Model Analysis

Classification models can be used to identify and understand the emotions contained in each review. One of them is by

DOI : https://doi.org/10.25139/inform.v9i2.8410

utilizing the Random Forest and Support Vector Machine algorithms.

1) Random Forest: Random Forest is a decision tree-based ensemble learning method and a powerful approach to classification. This model creates a set of decision trees from a randomly selected subset. Each tree provides a prediction, and the predictions from all trees are combined through a voting process. The final prediction is determined based on the majority of votes [25]. Each tree is built using training data samples with a bootstrap sampling technique. The training data is then classified based on the trees built. Each tree classifies the test data, which is classified into the category with the most votes, known as the majority vote. This method's main advantage lies in overcoming overfitting problems and providing more stable and accurate predictions [23]. Figure 7 illustrates the stages of the Random Forest algorithm [23].

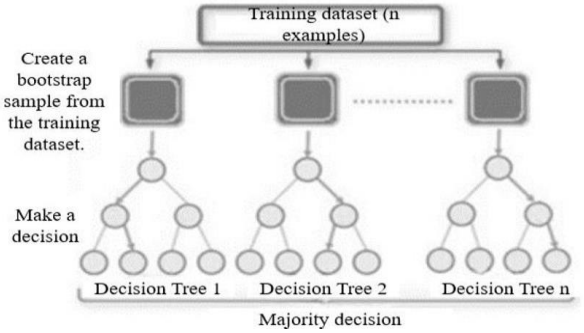

Figure 7. Stages of Random Forest

2) Support Vector Machine (SVM): SVM can be used for regression and classification because it finds the optimal hyperplane that separates classes in the feature space by maximizing the margin, the distance between the hyperplane and the support vector. The advantage of this method is its ability to handle complex data with good performance [23]. Figure 8 illustrates a visual representation of the SVM algorithm.

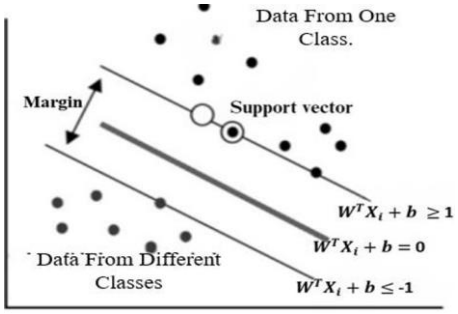

Figure 8. Visualization of SVM [23]

G. Evaluation

The Confusion Matrix evaluates model performance, which provides an overview of how well the model classifies the data. The variables measured in the Confusion Matrix consist of true positive (TP), which is the amount of data that is correct and correctly detected by the model; true negative (TN), which is the amount of incorrect data detected incorrectly by the model, false negative (FN) which is the amount of data detected correctly but detected incorrectly by the model, and false positive (FP) which is the amount of data detected correctly but detected incorrectly by the model. From the Confusion Matrix, various evaluation metrics such as accuracy, precision, recall, and F1-Score can be calculated using Equation (4) to (7), respectively [26].

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

$$F1 - Score = \frac{2*(Recall*Precision)}{Recall+Precision} \quad (7)$$

H. Tunning

Hyperparameter tuning is used to improve model performance. It allows the model to fit the training data better and produce better results when used to predict new data. GridSearchCV is a hyperparameter tuning method that tries all parameter combinations in the search space to find the parameter combination with the smallest error [27].

III. Result and Discussion

In this research, a series of stages have been carried out to obtain optimal results in text data classification. This section describes the results of each stage of the research that has been carried out.

A. Load Dataset

The dataset' Multilabel Sentiment and Emotion Dataset from Indonesian Mobile Application Review' by Ricco San and Karen has gone through data pre-processing to remove noise with a total collection of 21,697 reviews and data annotating process to identify the type of sentiment and emotion in the review text. The data is annotated into three sentiments: positive, negative, and neutral. There are six emotions: anger, fear, sadness, happiness, love, and neutrality. Table I is the sample dataset.

Table I
Dataset Samples

Content	Label	
	Sentiment	Emotion
Lemot loadingnya..banyak bug	Negative	Anger
Kok game nya kaga bisa masuk sih gimana caranya game nya mahal lagi ga bisa masuk lagi pilih folder nya yang mana devlover yang mana oy	Negative	Fear
Untuk pelayanan oke,, cma sayang harga pulsa/pket data mahal. Dan sekarang tranfer malah kena bea admin	Negative	Sad
Mantab stabil & iklan nya ga bikin ganggu	Positive	Happy
Bus simulator sangat baguss wajib donload	Positive	Love
Yang Penting Aplikasi Berjalan Lancar dan Bagus	Neutral	Neutral

DOI : https://doi.org/10.25139/inform.v9i2.8410

B. Data Preprocessing

At this stage, the data is cleaned to create more structured data to avoid irrelevant data that contains errors and can reduce the accuracy and efficiency of the model. Table II shows the flow of data pre-processing.

TABLE II
DATA PRE-PROCESSING

Raw Text	
Terima kasih banyak untuk Kemendikbud karena sudah menghadirkan apk yang sangat bermanfaat ini. (⊃•́‿•̀)⊃	
Step	**Text Preprocessing**
Lowercase	*terima kasih banyak untuk kemendikbud karena sudah menghadirkan apk yang sangat bermanfaat ini. (⊃•́‿•̀)⊃*
Remove punctuation	*terima kasih banyak untuk kemendikbud karena sudah menghadirkan apk yang sangat bermanfaat ini ⊃•́‿•̀⊃*
Remove special character	*terima kasih banyak untuk kemendikbud karena sudah menghadirkan apk yang sangat bermanfaat ini*
Remove whitespace	*terima kasih banyak untuk kemendikbud karena sudah menghadirkan apk yang sangat bermanfaat ini*
Tokenizing	*['terima', 'kasih', 'banyak', 'untuk', 'kemendikbud', 'karena', 'sudah', 'menghadirkan', 'apk', 'yang', 'sangat', 'bermanfaat', 'ini']*
Remove stopword	*['terima', 'kasih', 'kemendikbud', 'menghadirkan', 'apk', 'bermanfaat']*
Stemming	*['terima', 'kasih', 'kemendikbud', 'hadir', 'apk', 'manfaat']*
Clean Text	
terima kasih kemendikbud hadir apk manfaat	

TABLE III
CLASS DISTRIBUTION BEFORE AND AFTER PRE-PROCESSING

Sentiment	Emotion	Total			
		Before	Missing Value	Duplicated Data	After
Negative	Anger	2697	7	66	2624
	Fear	1271	6	17	1248
	Sad	3753	11	173	3569
Positive	Happy	6330	19	685	5626
	Love	193	0	24	169
Neutral	Neutral	7453	144	821	6488
Total		**21697**	**187**	**1786**	**19724**

After the steaming process, the data list is returned to string form. Then, clean the dataset from missing values and duplicate data due to empty data or the cleaning process results. Table III compares class distribution results before and after pre-processing, with total data from 21697 to 19724.

C. Label Encoding

Label encoding is done to simplify the classification process. We are changing the class name on the Sentiment label, namely, 0 for Negative, 1 for Neutral, and 2 for Positive. For Emotion labels, namely, 0 for anger, 1 for fear, 2 for sad, 3 for happy, 4 for love, and 5 for neutral.

D. Feature Extraction

The cleaned data needs feature extraction to convert the data into numeric. Table IV shows an example of TF-IDF calculation results for feature extraction. The higher the value, the more words are important or influential in the sentence in the document.

TABLE IV
TF-IDF SAMPLES

Term	TF			IDF	TF-IDF		
	D 18640	D 18641	D 18642		D 18640	D 18641	D 18642
aja	0.3682 94	0.0000 00	0.0000 00	3.78 9615	1.3956 93	0.0000 00	0.000 000
apik	0.0000 00	0.0000 00	0.6662 88	7.89 3910	0.0000 00	0.0000 00	5.259 615
bagus	0.0000 00	0.3600 98	0.0000 00	2.72 1439	0.0000 00	0.9799 85	0.000 000
banyak	0.0000 00	0.7405 02	0.0000 00	5.59 6337	0.0000 00	4.1440 98	0.000 000
berita	0.3750 16	0.0000 00	0.0000 00	3.85 8785	1.4471 07	0.0000 00	0.000 000
bug	0.0000 00	0.0000 00	0.3587 03	4.24 9766	0.0000 00	0.0000 00	1.524 403
hoax	0.6440 42	0.0000 00	0.0000 00	6.62 6962	4.2680 41	0.0000 00	0.000 000
iklan	0.0000 00	0.5674 38	0.0000 00	4.28 8412	0.0000 00	2.4334 10	0.000 000
loading	0.0000 00	0.0000 00	0.4247 03	5.03 1709	0.0000 00	0.0000 00	2.136 981
map	0.0000 00	0.0000 00	0.3732 35	4.42 1943	0.0000 00	0.0000 00	1.650 426
media	0.5558 17	0.0000 00	0.0000 00	5.71 9158	3.1788 04	0.0000 00	0.000 000
udah	0.0000 00	0.0000 00	0.3282 01	3.88 8397	0.0000 00	0.0000 00	1.276 176

The TF-IDF weighting result is stored in a sparse matrix. Table V is a sparse matrix of TF-IDF weighting results in the first row (0, 15339), with the first number (0) indicating this value comes from the first document in the corpus.

TABLE V
MATRIX TF-IDF

Matrix	TF-IDF
(0, 15339)	0.12312022870173846
(0, 10713)	0.12429499075794163
(0, 1581)	0.08029525689981273
:	:
(19723, 856)	0.17884837513222326
(19723, 3252)	0.36425591215067443
(19723, 12043)	0.34775786618491417

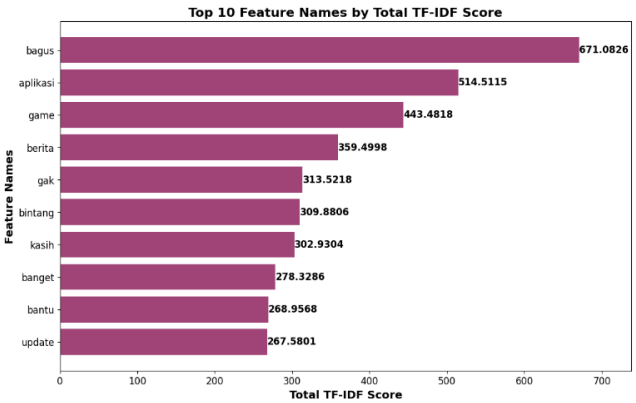

Figure 9. Top 10 Feature Names

The second number (15339) shows the index of the word in the list of all words (feature name). Then, the third number

DOI : https://doi.org/10.25139/inform.v9i2.8410

(0.12312022870173846) shows the TF-IDF weight. Feature extraction using TF-IDF resulted in 20141 feature names. Figure 9 shows the top 10 feature names.

E. Data Balancing

Data balancing is required for this dataset. In Figure 10, it can be seen that the data is not balanced in each class for both Sentiment and Emotion labels. Data balancing in this research is focused on emotion as the target label because Sentiment and Emotion labels have a close relationship. The Negative class is connected to the Angry, Scared, or Sad class. In contrast, the Positive class is only connected to the Happy or Love class. In comparison, the neutral class is connected only to the neutral class. Emotion knowledge can be used to determine sentiment, but not vice versa.

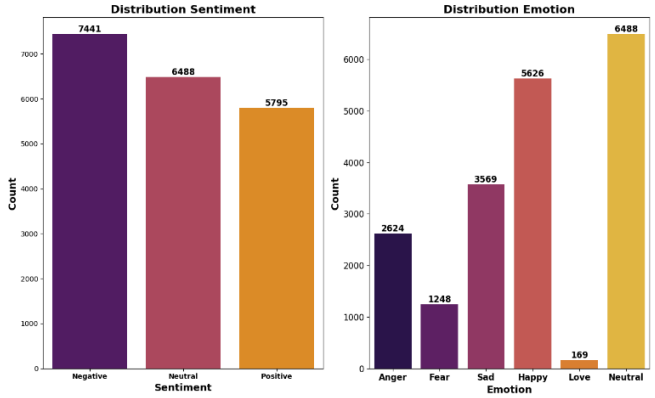

Figure 10. Distribution Emotion Before Re-sampling

Data balancing in this study uses three techniques: oversampling, undersampling, and combined re-sampling.

1) Undersampling (Tomek Links): Tomek Links are removed, leaving more representative data. The amount of data in each class that was removed was that the anger class lost 234 samples, fear lost 194 samples, sad lost 394 samples, happy lost 461 samples, love did not lose data samples, and neutral lost 691 samples. Figure 11 compares the class distribution before and after applying Tomek Links undersampling.

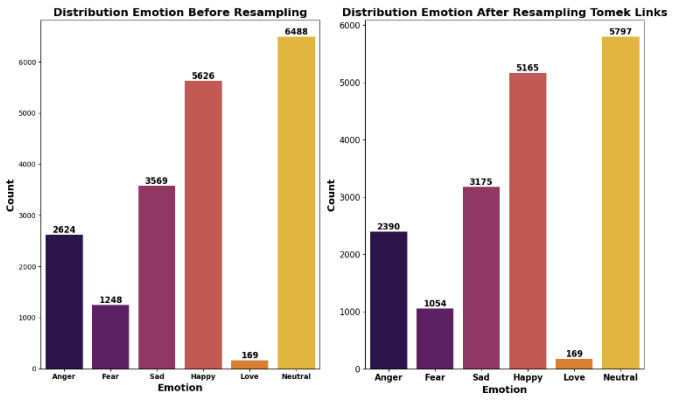

Figure 11. Comparison of Tomek Links Re-sampling Results

2) Oversampling (SMOTE): New data samples in each class. The results of the SMOTE process are that the Anger class produces 3864 samples, fear produces 5240 samples, sad produces 2919 samples, happy produces 862 samples, love produces 6319 samples, and neutral is the majority class, so there are no additional samples. Figure 12 compares the class distribution before and after applying SMOTE oversampling.

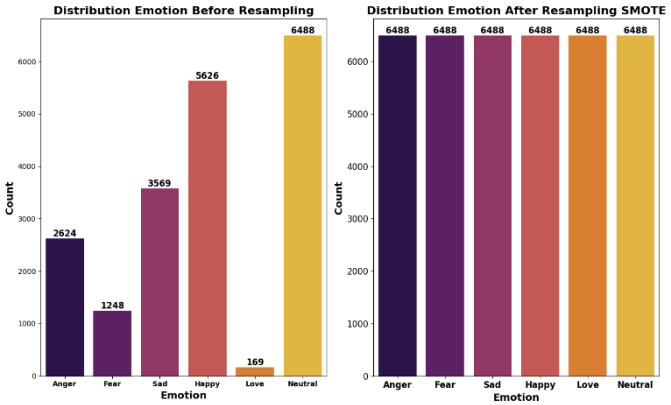

Figure 12. Comparison of SMOTE Re-sampling Results

*3) Combined Resamping (*SMOTE-Tomek): SMOTE-Tomek works by performing SMOTE to oversample the minority data and then cleaning the majority data that is indicated to have Tomek Links. The amount of SMOTE result data for each class becomes 6488. Then the Tomek links are removed, the anger class loses 25 samples, the fear class loses 3 samples, the sad class loses 74 samples, the happy class loses 256 samples, and the love class has no Tomek links. Hence, it uses the SMOTE result data, and the Neutral class loses 300 samples. Figure 13 compares the class distribution before and after applying combined re-sampling with SMOTE.

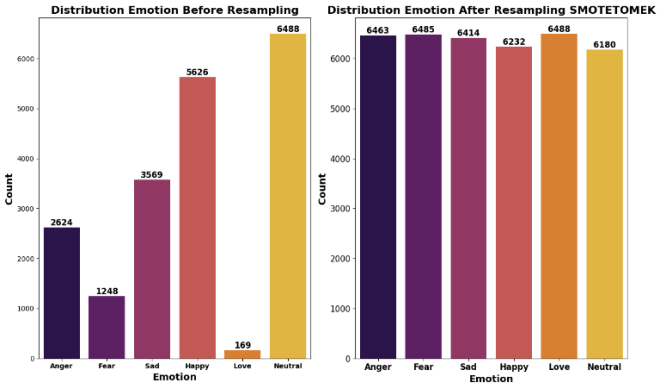

Figure 13. Comparison of SMOTE-Tomek re-sampling results

The results of using T-SNE to visualize the distribution of data before and after applying various re-sampling techniques are shown in Figure 14. In the original data, there is a very significant overlap between classes. After implementing Tomek Links, although there was still overlap between classes, the neutral class (5) experienced a slight decrease in data. Implementing SMOTE produces a more balanced data distribution. The data after the SMOTE-Tomek process is similar to SMOTE, but the effective data overlap is reduced if you look closely.

DOI : https://doi.org/10.25139/inform.v9i2.8410

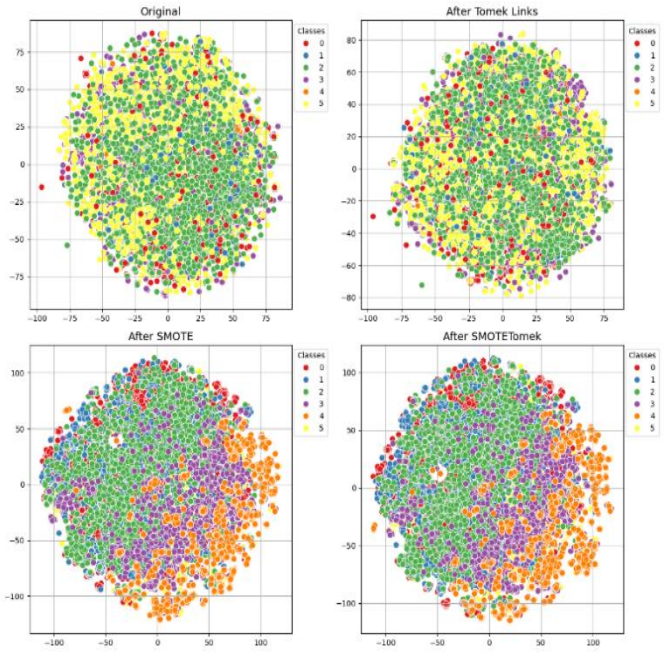

Figure 14. T-SNE Visualisation Comparing Data Distributions Using Tomek Links, SMOTE, and SMOTE-Tomek Re-sampling Techniques

F. Split Dataset

Datasets with various sampling techniques are divided with a ratio of 80% as training data and 20% as test data. Table VI compares data distribution for training and test sets on each sampling data.

TABLE VI
DATASET SPLIT

Sampling Type	Sampling Quantity	Data Split	
		Data Train (80%)	Data Test (20%)
Original	19724	15779	3945
Tomek Links	17750	14200	3550
SMOTE	38928	31142	7786
SMOTE-Tomek	38262	30609	7653

G. Model Analysis

Research data from several sampling techniques are trained using Random Forest and SVM algorithms to learn patterns in the training data. Then, the models were tested using the test data to predict the class labels. Table VII shows the performance comparison of Random Forest and SVM models with various sampling techniques.

TABLE VII
COMPARISON OF TESTING RESULTS

Sampling Type	Model	Result			
		Precision	Recall	F1-Score	Accuracy
Original	RF	0.57	0.58	0.56	0.58
	SVM	0.60	0.59	0.57	0.59
Tomek Links	RF	0.59	0.59	0.57	0.59
	SVM	0.61	0.60	0.59	0.61
SMOTE	RF	0.83	0.83	0.83	0.83
	SVM	0.76	0.77	0.77	0.77
SMOTE-Tomek	**RF**	**0.84**	**0.84**	**0.84**	**0.84**
	SVM	0.77	0.77	0.77	0.77

Based on the evaluation of Random Forest and SVM models on original data and re-sampled data using Tomek Links, SMOTE, and SMOTE-Tomek, It was found that the Random Forest model performed best on data re-sampled using SMOTE-Tomek. This model achieved 84% precision, 84% recall, 84% F1 score, and 84% accuracy. These results show that SMOTE-Tomek is effective in overcoming data imbalances and provides additional benefits by improving the performance of complex sentiment analysis models compared to other re-sampling techniques.

H. GridSearchCV

The best parameter combinations with the smallest error in GridSearchCV Hyperparameter tuning for SMOTE-Tomek data using the Random Forest model include.

a) 'bootstrap': False (Not using bootstrap)
b) 'max_depth': None (no limit on tree depth)
c) 'min_samples_leaf': 1 (the minimum number of samples at each leaf node is 1)
d) 'min_samples_split': 2 (the minimum number of samples required to split a node is 2)
e) 'n_estimators': 200 (the number of estimators in the ensemble is 200)

With these parameters, the Random Forest model can provide the best performance with a small error rate when applied to data processed with the SMOTE-Tomek technique.

I. Evaluation

Figure 15 shows the Confusion matrix for the Random Forest model with the SMOTE-Tomek technique. Out of six classes, the model correctly classified a number of data in each class. The values 1118, 1233, 1003, 943, 1294, and 828 represent the number of correctly classified data for each class according to the experiments' results.

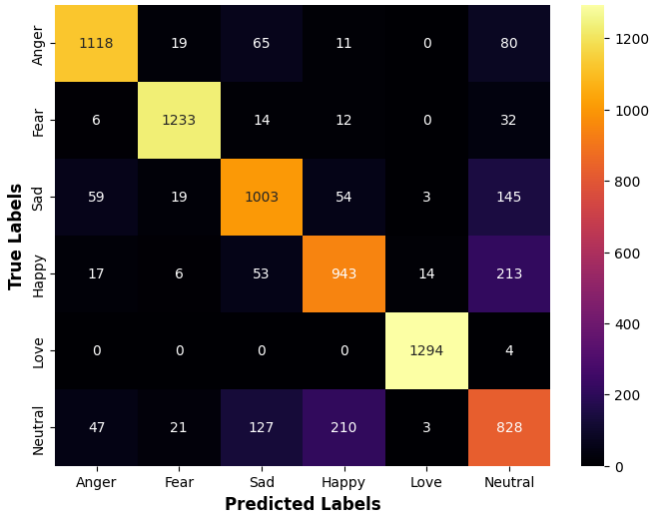

Figure 15. Confusion Matrix: RF Model on SMOTE-Tomek Data

Next, Figure 16 shows the Random Forest classification model evaluation report using SMOTE-Tomek re-sampling.

DOI : https://doi.org/10.25139/inform.v9i2.8410

	precision	recall	f1-score	support
0	0.90	0.86	0.88	1293
1	0.95	0.95	0.95	1297
2	0.79	0.78	0.79	1283
3	0.77	0.76	0.76	1246
4	0.98	1.00	0.99	1298
5	0.64	0.67	0.65	1236
accuracy			0.84	7653
macro avg	0.84	0.84	0.84	7653
weighted avg	0.84	0.84	0.84	7653

Figure 16. Classification Report: RF Model on SMOTE-Tomek Data

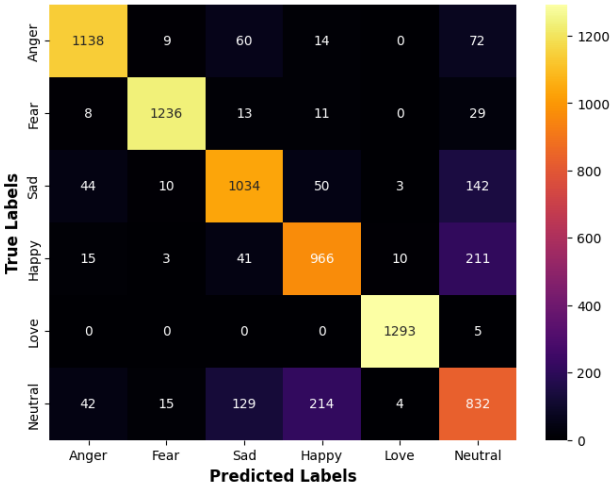

Figure 17. Confusion Matrix: RF Model with GridSearchCV on SMOTE-Tomek Data

Figure 17 shows the Confusion matrix for the Random Forest model with the SMOTE-Tomek technique in GridSearchCV hyperparameter tuning. Out of six classes, the model correctly classified a number of data in each class. The values 1138, 1236, 1034, 966, 1293, and 832 represent the number of correctly classified data for each class according to the experiments' results. Figure 18 shows the Random Forest classification model evaluation report using SMOTE-Tomek re-sampling with GridSearchCV hyperparameter tuning.

	precision	recall	f1-score	support
0	0.91	0.88	0.90	1293
1	0.97	0.95	0.96	1297
2	0.81	0.81	0.81	1283
3	0.77	0.78	0.77	1246
4	0.99	1.00	0.99	1298
5	0.64	0.67	0.66	1236
accuracy			0.85	7653
macro avg	0.85	0.85	0.85	7653
weighted avg	0.85	0.85	0.85	7653

Figure 18. Classification Report: RF Model with GridSearchCV on SMOTE-Tomek Data

The confusion matrix evaluation of the Random Forest model with the SMOTE-Tomek re-sampling technique before and after using Hyperparameter tuning GridSearchCV. Precision, Recall, F1-Score, and Accuracy values are compared in Table VIII.

TABLE VIII
COMPARISON BEFORE AND AFTER TUNING

Model	Result			
	Precision	Recall	F1-Score	Accuracy
Random Forest	0.84	0.84	0.84	0.84
RF + GridSearchCV	0.85	0.85	0.85	0.85

It can be visualized in Figure 19 the increase in accuracy after applying GridSearchCV to the Random Forest model with SMOTE-Tomek re-sampling.

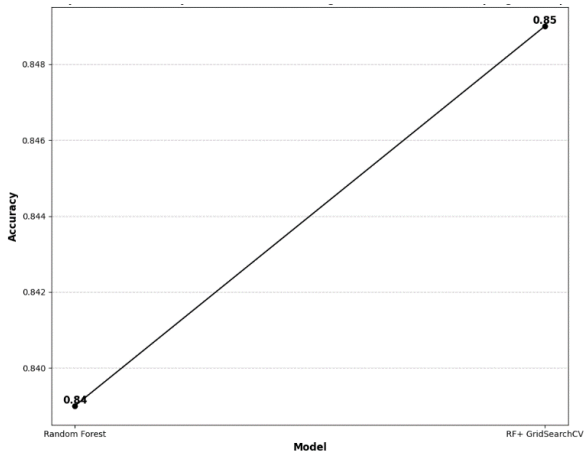

Figure 19. Comparison of Accuracy before and after tunning in SMOTE-Tomek re-sampling technique

IV. CONCLUSION

Classification based on user emotions is very important. To solve the problem of multi-class imbalance in application review data, we can use the SMOTE-Tomek re-sampling technique with the Random Forest method. The results of research using this method show that the implementation of a set of pre-processing techniques, especially the SMOTE-Tomek technique, to overcome data imbalance by oversampling the minority class and then eliminating samples that are at risk of noise in the majority class can improve the performance of the Random Forest model from 58% become 84%. This model was optimized using GridSearchCV hyperparameter tuning, increasing the accuracy to 85%. This improvement shows that combining SMOTE-Tomek with GridSearchCV can improve model performance. However, for future research, it is recommended that other re-sampling techniques using Pipeline be explored.

REFERENCES

[1] A. Rafid Rizqullah, A. Wedhasmara, R. Izwan Heroza, A. Putra, and P. Putra, “Analisis Masalah Pada Data Review Aplikasi Terhadap Layanan E-Commerce Menggunakan Metode Text Classification,” *Jurnal Tekno Kompak*, vol. 16, no. 1, pp. 186–198, 2022, doi: 10.33365/jtk.v16i1.1448.
[2] Ceci Laura, "Number of mobile app downloads worldwide from 2021 to 2023 by country," Data.ai. Accessed: May 22, 2024. [Online]. Available: https://www.statista.com/statistics/1287159/app-downloads-by-country
[3] P. Br Sihotang, F. Dameka Br Sitanggang, N. Azriansyah, and E. Indra, “Penerapan Natural Language Processing Untuk Analisis Sentimen Terhadap Aplikasi Streaming,” *Jurnal Ilmiah Betrik*, vol. 14, no. 02, pp. 273–282, 2023, doi: 10.36050/betrik.v14i02%20AGUSTUS.96.
[4] F. A. Larasati, D. E. Ratnawati, and B. T. Hanggara, “Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest,” *Jurnal*

DOI : https://doi.org/10.25139/inform.v9i2.8410

Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 6, no. 9, pp. 4305–4313, 2022.

[5] D. R. Wulandari, C. Setianingsih, and F. M. Dirgantara, "Deteksi Emosi Berbasis Teks Untuk Menganalisis Kuliah Daring Selama Masa Pandemi Menggunakan Algoritme Naive Bayes Text Based Emotion Detection For Analysis Online Lecture During Pandemic Using Naive Bayes Algorithm," in *eProceedings of Engineering*, 2022, pp. 1908–1915.

[6] R. Dwi Fitriani, H. Yasin, D. Statistika, and F. Sains dan Matematika, "Penanganan Klasifikasi Kelas Data Tidak Seimbang Dengan Random Oversampling Pada Naive Bayes (Studi Kasus: Status Peserta Kb Iud Di Kabupaten Kendal)," *Jurnal Gaussian*, vol. 10, no. 1, pp. 11–20, 2021, doi: 10.14710/j.gauss.10.1.11-20.

[7] I. Ayu Mirah Cahya Dewi, I. Komang Dharmendra, N. Wayan Setiasih, F. Informatika dan Komputer, and I. Teknologi dan Bisnis STIKOM Bali, "Analisis Sentimen Review Aplikasi Satu Sehat Mobile Menggunakan Model Sampling Tomek Links," *Jurnal Teknologi Informasi dan Komputer*, vol. 9, no. 5, pp. 497–504, 2023, doi: 10.36002/jutik.v9i5.2644.

[8] G. Lemaitre, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning," *Journal of Machine Learning Research}*, vol. 18, no. 17, pp. 1–5, 2017, Accessed: Apr. 25, 2024.

[9] Z. Wang, C. Wu, K. Zheng, X. Niu, and X. Wang, "SMOTETomek-Based Re-sampling for Personality Recognition," *IEEE Access*, vol. 7, pp. 129678–129689, 2019, doi: 10.1109/ACCESS.2019.2940061.

[10] E. Utami, I. Oyong, S. Raharjo, A. Dwi Hartanto, and S. Adi, "Supervised learning and re-sampling techniques on DISC personality classification using Twitter information in Bahasa Indonesia," *Applied Computing and Informatics*, 2021, doi: 10.1108/ACI-03-2021-0054.

[11] A. Nurhopipah and C. Magnolia, "Perbandingan Metode Resampling Pada Imbalanced Dataset Untuk Klasifikasi Komentar Program Mbkm," *JUPIKOM (Jurnal Publikasi Ilmu Komputer Dan Multimedia)*, vol. 1, no. 2, 2022, doi: 10.55606/jupikom.v2i1.862.

[12] L. D. Cahya, A. Luthfiarta, J. I. T. Krisna, S. Winarno, and A. Nugraha, "Improving Multi-label Classification Performance on Imbalanced Datasets Through SMOTE Technique and Data Augmentation Using IndoBERT Model," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 9, no. 3, pp. 290–298, Jan. 2024, doi: 10.25077/teknosi.v9i3.2023.290-298.

[13] Riccosan and K. E. Saputra, "Multilabel multiclass sentiment and emotion dataset from indonesian mobile application review," *Data Brief*, vol. 50, Oct. 2023, doi: 10.1016/j.dib.2023.109576.

[14] O. I. Gifari, M. Adha, I. Rifky Hendrawan, F. Freddy, and S. Durrand, "Analisis Sentimen Review Film Menggunakan TF-IDF dan Support Vector Machine," *JIFOTECH (Journal Of Information Technology)*, vol. 2, no. 1, 2022, doi: 10.46229/jifotech.v2i1.330.

[15] V. W. D. Thomas and F. Rumaisa, "Analisis Sentimen Ulasan Hotel Bahasa Indonesia Menggunakan Support Vector Machine dan TF-IDF," *Jurnal Media Informatika Budidarma*, vol. 6, no. 3, p. 1767, Jul. 2022, doi: 10.30865/mib.v6i3.4218.

[16] A. F. Rahman, "Klasifikasi Tweet di Twitter dengan Menggunakan Metode K-Nearest Neighbor," *Jurnal Sistim Informasi dan Teknologi*, pp. 64–69, Mar. 2022, doi: 10.37034/jsisfotek.v4i2.125.

[17] A. Rizki Bramantyo and A. R. Pratama, "Analisis Sentimen Kebijakan Protokol Kesehatan Pada Masa Pandemi Di Media Sosial Facebook dengan Crowdtangle," *Jurnal Sains Komputer & Informatika (J-SAKTI*, vol. 6, no. 2, pp. 947–960, 2022, doi: 10.30645/j-sakti.v6i2.505.

[18] S. Faira Huwaida, R. Kusumawati, B. Isnaini, P. Korespondensi, and R. Artikel, "Analisis sentimen komentar youtube terhadap pemindahan ibu kota negara menggunakan metode Naïve Bayes," *Jambura Journal of Informatics*, vol. 6, no. 1, pp. 26–39, 2024, doi: 10.37905/jji.v6i1.24718.

[19] J. E. Br Sinulingga and H. C. K. Sitorus, "Analisis Sentimen Opini Masyarakat terhadap Film Horor Indonesia Menggunakan Metode SVM dan TF-IDF," *Jurnal Manajemen Informatika (JAMIKA)*, vol. 14, no. 1, pp. 42–53, Feb. 2024, doi: 10.34010/jamika.v14i1.11946.

[20] E. F. Swana, W. Doorsamy, and P. Bokoro, "Tomek Link and SMOTE Approaches for Machine Fault Classification with an Imbalanced Dataset," *Sensors*, vol. 22, no. 9, May 2022, doi: 10.3390/s22093246.

[21] Y. A. Sir and A. H. H. Soepranoto, "Pendekatan Resampling Data Untuk Menangani Masalah Ketidakseimbangan Kelas," *Jurnal Komputer dan Informatika*, vol. 10, no. 1, pp. 31–38, Mar. 2022, doi: 10.35508/jicon.v10i1.6554.

[22] R. A. Danquah, "Handling Imbalanced Data: A Case Study for Binary Class Problems," Oct. 2020, doi: 10.6084/m9.figshare.13082573.v2.

[23] A. J. Dahur, A. Wahyul Syafei, and T. Prahasto, "Analysis of Visitor Review Data Using Lexicon Based, Support Vector Machine, Random Forest in Determining the Priority Scale of Building Labuan Bajo Tourism Objects," in *E3S Web of Conferences*, EDP Sciences, Nov. 2023. doi: 10.1051/e3sconf/202344802043.

[24] G. E. A. P. A. Batista, A. L. C. Bazzan, and M. C. Monard, "Balancing Training Data for Automated Annotation of Keywords: a Case Study," in *II Brazilian Workshop on Bioinformatics*, 2003.

[25] K. Marzuki, L. Ganda Rady Putra, H. Hairani, L. Zazuli Azhar Mardedi, and J. Ximenes Guterres, "Performance Improvement of The Random Forest Method Based on Smote-Tomek Link on Lombok Tourism Analysis Sentiment," *Jurnal Bumigora Information Technology (BITe)*, vol. 5, no. 2, pp. 151–158, 2023, doi: 10.30812/bite/v5i1.3166.

[26] K. Rahayu, V. Fitria, D. Septhya, R. Rahmaddeni, and L. Efrizoni, "Klasifikasi Teks untuk Mendeteksi Depresi dan Kecemasan pada Pengguna Twitter Berbasis Machine Learning," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 3, no. 2, pp. 108–114, Sep. 2023, doi: 10.57152/malcom.v3i2.780.

[27] A. Baita, I. A. Prasetyo, and N. Cahyono, "Hyperparameter Tuning On Random Forest For Diagnose Covid-19," *JIKO (Jurnal Informatika dan Komputer)*, vol. 6, no. 2, Aug. 2023, doi: 10.33387/jiko.v6i2.6389.

DOI : https://doi.org/10.25139/inform.v9i2.8410