Scientific Journal of Informatics
Vol. 12, No. 1, Feb 2025

p-ISSN 2407-7658 https://journal.unnes.ac.id/journals/sji/index e-ISSN 2460-0040

Improving Random Forest Performance for Sentiment Analysis on Unbalanced Data Using SMOTE and BoW Integration: PLN Mobile Application Case Study

M. R. Fadhlan Rahmatullah[1*], Pulung Nurtantio Andono[2], Affandy[3], M. A. Soeleman[4]
[1, 2, 3, 4]Department of Informatics Engineering, Universitas Dian Nuswantoro, Indonesia

Abstract.
Purpose: This research aims to improve the accuracy of sentiment analysis on PLN Mobile app reviews by overcoming the challenge of data imbalance. This goal is important to provide a better understanding of user opinions and support PT PLN (Persero) in improving mobile application services.
Methods: This research uses the Random Forest algorithm combined with Synthetic Minority Over-sampling Technique (SMOTE) to handle imbalanced data. Data is collected through web scraping reviews from the Google Play Store, followed by preprocessing processes such as data cleaning, stopword removal, tokenization, and stemming. Feature extraction is performed using the Bag of Words (BoW) method, and the data is tested with four sharing schemes.
Result: The results showed that the 90%-10% sharing scheme gave the best performance with an accuracy of 81% and an average precision and recall of 0.79. This finding confirms that the larger the proportion of training data, the better the model performs sentiment classification.
Novelty: This research's novelty lies in combining SMOTE with BoW and Random Forest to overcome data imbalance. This approach is a significant reference for future sentiment analysis research. It provides practical insights that PT PLN (Persero) can use to improve the quality of its application services.

Keywords: Sentiment analysis, PLN mobile, TF-IDF, BoW, SMOTE, Random forest
Received January 2025 / **Revised** March 2025 / **Accepted** March 2025

INTRODUCTION

Perusahaan Listrik Negara, known as PT PLN (Persero), is a state-owned enterprise (BUMN) that provides electricity services in Indonesia. In electricity service and distribution, PLN divides the function of its parent unit into three: generation, transmission, and distribution. [1]. One of the tangible proofs that PLN continues to strive for better service is the satisfaction of customers through the introduction of the PLN mobile app. Customer satisfaction through the launch of the PLN mobile app, which is in customers' hands via Android and iOS smartphones. The PLN mobile app was first launched in 2016 by the PLN Board of Directors on the occasion of the 71st National Electricity Day and relaunched in 2020 with new features. The app is an innovative product of PLN in collaboration with PT Indonesia Comnet Plus, a subsidiary of PLN Mobile. [2].

A machine learning method called sentiment analysis uses ratings and views to assess how people feel about things like people, goods, services, or subjects. Information required for several applications can be obtained using sentiment analysis. Subjective analysis is another name for sentiment analysis. It groups books based on the patterns and tenets of the viewpoints they disclose: neutral, negative, and positive. [3].

Previous research [3] Using Random Forest and TF-IDF methods resulted in 93.14% for the F1-Score. Then in other research [4] managed to get an accuracy of 73% using Naïve Bayes. Furthermore, research [5] It can produce an accuracy of 96% through the decision tree method. The research [6] has 489 reviews with 49% positive sentiment, 365 reviews with 37% neutral sentiment, and 146 reviews with 14% negative sentiment, and its accuracy is 70%.

[*]Corresponding author.
Email addresses: P31202302554@mhs.dinus.ac.id (Rahmatullah), pulung@dsn.dinus.ac.id (Pulung Nurtantio Andono), affandy@dsn.ac.id (Affandy), m.arief.soeleman@dsn.dinus.ac.id (Arief)

DOI: 10.15294/sji.v12i1.19295

In machine learning, data imbalance is an important problem that needs to be addressed. Class imbalance exists when the number of instances of the majority class exceeds the number of instances of the minority class [7]. Unbalanced data can lead to negative results in the classification. This is because an unbalanced amount of data between the majority class and the minority class results in the majority class performing better than the minority class.

One way to solve the problem of unbalanced classes is to use random *sampling*. [7]. The *Synthetic Minority Oversampling Technique* (SMOTE) is a method suitable for large data. The SMOTE technique multiplies the minority data by the same value as the majority data, thus creating a balance among the classes in the dataset. [8]. The weakness of SMOTE is that the data collected does not take into account the majority class, which may lead to overlap. [9].

Another oversampling technique for addressing data imbalance is *ADASYN* (*Adaptive Synthetic Sampling*). *ADASYN* focuses on developing synthetic samples for more elusive minority groups, unlike *SMOTE*. On the other hand, *SMOTE* was selected for this research because of its more steady data distribution and performance in several previous research.

BoW was selected because of its efficacy on app review datasets and its simplicity in quantitatively encoding text. However, unlike *Word2Vec* and *GloVe*, which capture context-based meaning, and *BERT*, which interprets context bidirectionally, *BoW* ignores word order and word relationships. Due to its ease of use, interpretability, and demonstrated efficacy in sentiment analysis, *BoW* is still in use today.

The improvement in this research is by using *SMOTE* to overcome imbalances and then using BOW as a feature extraction technique, and *Random Forest*. By using this, it can increase accuracy that has not previously been achieved and become sentiment literature in further research.

METHODS

This research was conducted using several stages such as data collection, preprocessing, feature extraction, classification, and evaluation. The following is an overview of the method flow.

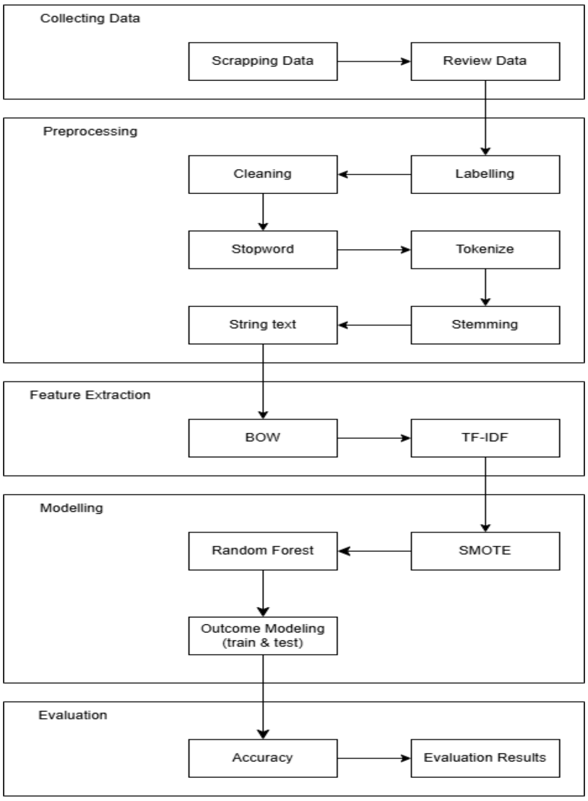


Figure 1. Research steps

Scrapping data

The technique of extracting semi-structured material from the internet, often web pages in a markup language, is known as web scraping. These documents are written in markup languages like HTML or XHTML and analyzed to extract the data [10]. Gathering information on the "PLN Mobile" app that is accessible through the Google Play Store is the primary goal of this phase. This data can include information such as user ratings, opinions, number of downloads, last update, application description, and others. The data obtained is then used to analyze user sentiment for other purposes relevant to the research.

Labeling

Relevant categories are used to label each data in the dataset. In this research, user comments or opinions on the “PLN Mobile” application are used to label the data based on certain categories, such as positive, negative, and neutral.

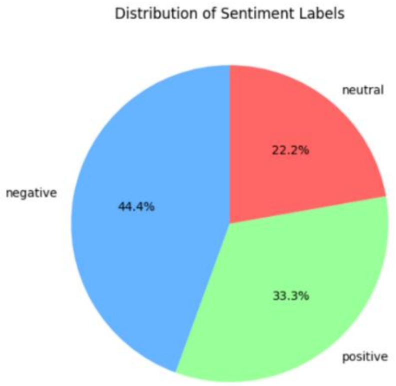


Figure 2. The data distribution before SMOTE

The label distribution in the figure shows that the dataset contains three classes with unbalanced proportions, namely negative (44.4%), positive (33.3%), and neutral (22.2%) classes. This imbalance can be seen from the ratio between the majority (negative) and minority (neutral) classes, which is almost 2:1. This difference indicates that the amount of data is very high. This difference indicates that the amount of data in the neutral class is much less than in the negative class, which may have an impact on how well the machine learning model performs.

Preprocessing

The process of data cleaning and normalization, as the raw data generated by users is usually unstructured and cannot be analyzed to perform sentiment analysis. This technique is often used in Natural Language Processing to prepare the text for classification. [11].

Clean text

The format of the obtained data is incorrect. One technique to make sure the data is accurate, dependable, and useful is to clean it. Generally speaking, datasets need to be cleaned because they include a lot of noise, undesirable data, or outliers. [12]. The data is freed from irrelevant elements or noise, for example by removing unnecessary punctuation, numbers, and special characters. Here is an example of *Clean text* in Table 1:

Table 1. An example of clean text

Before	After
Lanjutkan penangan secara gratis	lanjutkan penangan secara gratis

Stopword

Words that are often repeated and have no meaning [13]. Stopwords in Indonesian such as “yang”, “di”, “untuk”, “dari”, etc. This process uses functions from the Natural Language Toolkit (NLTK) library. [14]. The following is an example of a *Stopword* in Table 2:

Table 2. An example of a stopword

Before	After
aplikasi blank putih saat dibuka sudah install ulang aplikasi tetap sama juga mohon diperbaiki bug tersebut terimakasih	aplikasi blank putih dibuka install ulang aplikasi mohon diperbaiki bug terimakasih

Tokenize

The process by which a text is divided into words, sentences, or other significant parts is called tokens. The tokens are separated by spaces, punctuation marks, and line breaks; characters such as punctuation marks are usually removed during the tokenization process. Tokenization is considered relatively simple compared to other pre-processing techniques. [15]. Then here is an example of *Tokenize* in Table 3:

Table 3. An example of tokenization

Before	After
lelet parah log in aja gak masuk jaringan 5g masuk mah aplikasi nya auto uninstall	[‘lelet’, ‘parah’, ‘log’, ‘in’, ‘aja’, ‘gak’, ‘masuk’, ‘jaringan’, ‘5g’, ‘masuk’, ‘mah’, ‘aplikasi’. ‘nya’. ‘auto’. ‘uninstall’]

Stemming

Matching and examining word forms in their most fundamental expressions. The purpose of stemming is to remove a word's affixes in order to eliminate morphological differences in the word so that the correct term can be provided based on the correct morphological structure [16]. The following is an example of *Stemming* in Table 4:

Table 4. An example of stemming

Before	After
[‘daftar’, ‘akun’, ‘persulit’, ‘email’, ‘maksudnya’, ‘email’, ‘daftar’]	[‘daftar’, ‘akun’, ‘sulit’, ‘maksud’, ‘email’, ‘daftar’]

String text

Recombination of strings or text after stemming to produce a textual representation ready for the next step. Example of *String text* in Table 5:

Table 5. An example of string text

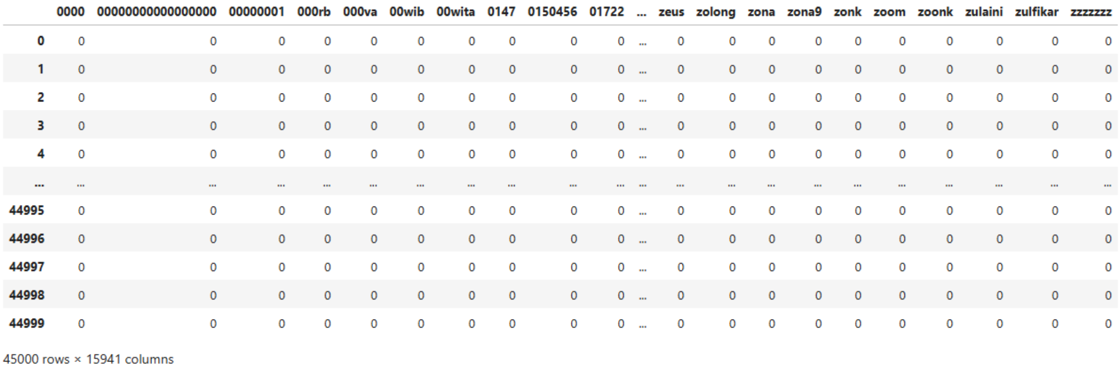
Before	After
[‘aju’, ‘pasang’, ‘listrik’, ‘kendala’, ‘slo’, ‘3’, ‘laku’, ‘bayar’, ‘tf’, ‘email’, ‘info’, ‘pasang’]	pasang listrik kendala laku bayar email info pasang

Bag of words (BOW)

This method is a fairly simple method for processing text data that is converted into vector numbers so that it can be processed by a computer. This method only calculates the frequency of word occurrence in all processed documents. [17]. If *d* is a document and *v* is a set of words or vocabulary from the whole document, then the mathematical notation of *BoW* can be represented as below

$$BoW(d) = [count(w1, d), count(w2, d \dots count(wn, d)] \tag{1}$$

Below are the features of *BoW*



	0000	0000000000000000	00000001	000rb	000va	00wib	00wita	0147	0150456	01722	...	zeus	zolong	zona	zona9	zonk	zoom	zoonk	zulaini	zulfikar	zzzzzzz
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
44995	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
44996	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
44997	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
44998	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
44999	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

45000 rows × 15941 columns

Figure 3. Data representation by *BoW*

The outcomes of representing text data using the *Bag of Words* (*BoW*) method are displayed in Figure 3. A document is represented by each row in the table, and a unique word found throughout the corpus or collection of documents is represented by each column. The table's data show how frequently each word appears in each document.

For example, if the column value in row 1 contains the number 3 indicating the word “zona”, it indicates that the word “zona” appears three times in the first document. Conversely, if the column value is completely absent, the word does not appear.

The number of rows in the table indicates that there are 45.000 documents in the corpus, and the number of columns indicates that there are 15.941 unique words in the corpus.

Term frequency-inverse document frequency (TF-IDF)

To give weight or value to each word in the text to ensure that a term is relevant to a document or sentence, ensuring that a term is relevant to a document or sentence [18]. *TF* or *Term Frequency* is a matrix that calculates the frequency of occurrence of a word in a document, while *IDF*, or *Inverse Document Frequency* is a matrix that calculates the frequency of occurrence of a word in all documents. [19]. Here is the equation for *Term Frequency*

$$tf_{t,d} = \frac{f_{t,d}}{(Total\ number\ of\ term\ in\ document)} \tag{2}$$

With

$tf_{t,d}$ = number of occurrences of term *t* in document *d*

Then there is the equation of *Inverse Document Frequency*

$$idf_d = \log\left(\frac{N}{n_t}\right) \tag{3}$$

With

N = total number of documents in the collection

n_t= number of documents containing term *t*

In the end, the *TF* and *IDF* will be combined as below

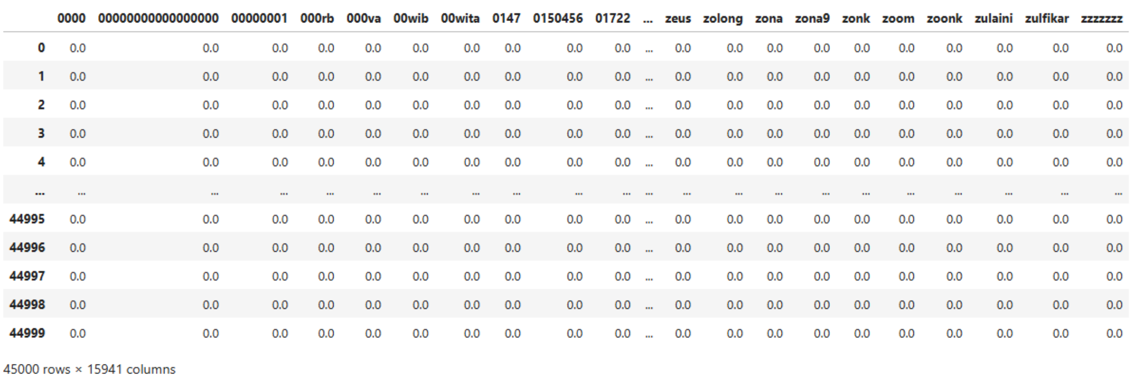
$$TF - IDF(t,d) = TF(t,d) \times IDF\ (t) \tag{4}$$

With

$TF(t,d)$ = frequency of term *t* in document *d*

$IDF(t)$ = *inverse document frequency* of term *t*

Below are the features of *TF-IDF*



	0000	0000000000000000	00000001	000rb	000va	00wib	00wita	0147	0150456	01722	...	zeus	zolong	zona	zona9	zonk	zoom	zoonk	zulaini	zulfikar	zzzzzzz
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
44995	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
44996	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
44997	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
44998	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
44999	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

45000 rows × 15941 columns

Figure 4. Data representation by *TF-IDF*

Figure 4. shows the weight of each specific word (term) in the documents in the dataset, which is the result of *TF-IDF* calculation. Each row shows one document, and each column shows one specific word found in all dataset documents. The value displayed in the table is the result of the *TF-IDF* weight calculation, which shows how important a word is in a particular document compared to the whole dataset documents.

Synthetic minority over-sampling technique (SMOTE)

A method that uses samples from the minority class to produce synthetic samples and combines them with neighboring samples to create new synthetic samples [20]. *SMOTE* can help balance data, which has labels for negative, positive, and neutral sentiments that do not exist in proportional numbers in the dataset. [21].

The parameter used in this research is random_state, using this parameter ensures that every time this parameter is run the results will remain the same.

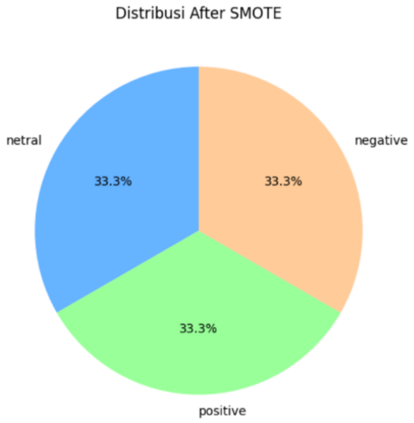


Figure 5. The data distribution after SMOTE

Random forest

An algorithm for supervised learning used for classification and regression issues. The simplest way to describe Random Forest is as a group of trees, each of which is unique. Every tree has its unique characteristics. constructing several decision trees, then integrating them to get a stable, absolute value that is mostly utilized for class formation and execution [22].

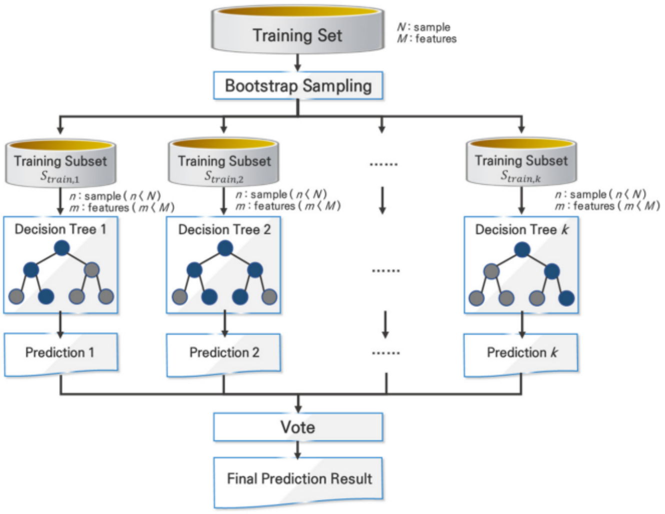


Figure 6. *Random forest* model structure [23]

In *Random Forest*, each tree selection greatly affects the classification prediction results because the more *Decision Tree* with the highest number of votes, the easier it is for the machine to determine the target class [24]. The parameter used in this study is *random_state*, to ensure that the results produced by the model are consistent every time it is run.

Train & test

The model was tested using four data split schemes (90%-10%, 80%-20%, 70%-30%, and 60%-40%) to evaluate the effect of the amount of training data on model performance. This method uses train-test split without cross-validation.

Evaluation

The confusion matrix is useful for analyzing classifiers to identify tuples belonging to different classes. When measured with the confusion matrix, four terms describe the result of the classification process [25]. These four terms are: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative

(FN). True Negative (TN) is a negative statement that is recognized as a negative statement, while False Positive (FPP) is a negative statement that is recognized as a positive statement. A True Positive (TP) is a positive statement that is recognized as a positive statement. A False Negative (FN) is the opposite of a False Positive, where a positive statement is recognized as a negative statement [26]. The Confusion Matrix equation is shown below.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

$$precision = \frac{TP}{TP + FP} \tag{6}$$

$$\mathrm{recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{7}$$

$$\mathrm{f1} - \mathrm{score} = 2\ \times \frac{\mathrm{precision}\ \times \mathrm{recall}}{\mathrm{precision}\ + \mathrm{recall}} \tag{8}$$

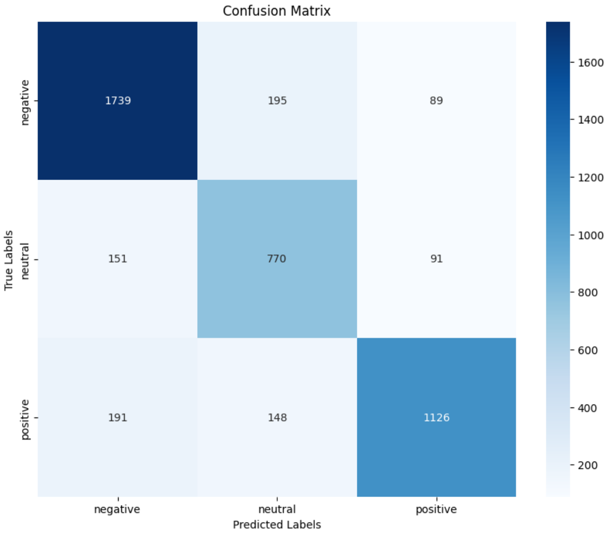


Figure 7. Confusion matrix (90% train – 10% test)

$$accuracy = \frac{1739 + 770 + 1126}{4500} = \frac{3635}{4500} = 0.8078\ (80.78\ \%) \tag{9}$$

The confusion matrix above shows that the model is quite good at classifying negative and positive sentiments, but still has difficulty in distinguishing neutral sentiments. A total of 195 negative samples were misclassified as neutral, while 151 neutral samples were classified as negative, suggesting that the model struggles to distinguish texts that have ambiguous or less explicit meanings. In addition, 191 positive samples were misclassified as negative, which could be due to the presence of negative words in a positive context.

RESULTS AND DISCUSSIONS

Random Forest was chosen for its ability to handle complex and variable data using an ensemble learning approach, resulting in stable and accurate predictions. Its performance is optimal at 90%-10% data split with 81% accuracy, average precision of 0.80, and average recall of 0.80, more consistent than other algorithms such as *Naïve Bayes* or *Decision Tree*. *SMOTE* is used to overcome data imbalance by increasing the number of samples of minority classes such as neutral and positive. This method helps improve recall for minority classes, although it has the risk of overfitting which can be minimized with proper parameter settings.

BoW was chosen for its simplicity in converting text into numerical representation. Although it does not consider context, it is quite effective for sentiment analysis of short reviews. Methods such as TF-IDF or embedding can be considered for further research.

The test results show that 90%-10% data split gives the best performance with 81% accuracy. Below are the test results for each data-sharing scenario:

Table 6. Confusion matrix (90% train - 10% test)

True / Predicted	Negative	Neutral	Positive
Negative	1739	195	89
Neutral	151	770	91
Positive	191	148	1129

At a 90%-10% data split, the model performed best for the positive and neutral classes.

Table 7. Confusion matrix (80% train - 20% test)

True / Predicted	Negative	Neutral	Positive
Negative	3394	396	201
Neutral	344	1433	226
Positive	378	324	2304

The 80%-20% data split showed a decrease in accuracy but was still good enough for all classes.

Table 8. Confusion matrix (70% train - 30% test)

True / Predicted	Negative	Neutral	Positive
Negative	5008	719	320
Neutral	618	2017	340
Positive	572	521	3385

At 70%-30% data split, there is a drop in performance, especially for the neutral class.

Table 9. Confusion matrix (60% train - 40% test)

True / Predicted	Negative	Neutral	Positive
Negative	6596	999	426
Neutral	996	2482	482
Positive	741	728	4550

At 60%-40% data split, the accuracy decreased to 76%, but the performance was relatively even for all classes.

The test results (Table 6-9) demonstrate that, particularly for the Positive and Neutral classes, the model predictions are more accurate when the percentage of training data is larger (90%-10%) and 80%-20%). On the other hand, the model predictions are more accurate (e.g., 70%-30%) and 60%-40%) when the percentage of training data is lower. The data split technique has an impact on the model's performance, as evidenced by the results, which range from 90% to 10%. However, because the outcomes rely on the data split, the train-test split may have drawbacks. As a result, cross-validation techniques can be applied to future studies to guarantee more broadly applicable findings.

Table 10. F1-Score comparison for each division scheme

Train & Test	Negative	Neutral	Positive
90% & 10%	0.85	0.73	0.81
80% & 20%	0.84	0.69	0.80
70% & 30%	0.82	0.65	0.79
60% & 40%	0.81	0.61	0.79

From Table 10, the model gives the best results at 90% data split for training and 10% for testing, especially in the Negative and Positive classes. However, the 60%:40% split shows a more even performance for all classes, although it slightly lowers the F1-Score in the Neutral class.

Table 11. Comparison of accuracy, precision, and recall

Train & Test	Precision Avg	Recall Avg	Accuracy
90% & 10%	0.80	0.80	0.81
80% & 20%	0.78	0.78	0.79
70% & 30%	0.75	0.75	0.77
60% & 40%	0.74	0.73	0.76

As the percentage of train data increases, the average values of precision, recall, and accuracy also tend to increase. At 90% train data and 10% test data, the highest precision and recall values are 0.79, and the accuracy is 0.81. Conversely, when the percentage of train data is lower (60% train and 40% test), the precision and recall values decrease to 0.74 with an accuracy of 0.76.

CONCLUSION

Based on sentiment analysis conducted on PLN Mobile application user reviews, the Bag of Words (BoW) feature extraction method and Random Forest algorithm proved effective in improving sentiment classification accuracy. Of the four experiments conducted, the composition of 90% training data and 10% test data produced the best performance, with precision and recall values of 0.79 each, and an accuracy of 0.81. These results show that the higher the proportion of training data, the more accurate the model predictions. This research can be a reference for future studies in sentiment analysis of similar applications and has the potential to assist PT PLN (Persero) in improving the quality of its application services based on user feedback.

REFERENCES

[1] S. Syafrizal, M. Afdal, and R. Novita, “Analisis Sentimen Ulasan Aplikasi PLN Mobile Menggunakan Algoritma Naïve Bayes Classifier dan K-Nearest Neighbor,” *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 1, pp. 10–19, 2023, doi: 10.57152/malcom.v4i1.983.

[2] F. Claudia, N. Siahaan, and B. K. Fawzeea, “SENTIMENT ANALYSIS ON PLN MOBILE APPLICATION USERS ’ OPINIONS TO IMPROVE THE QUALITY OF PLN MOBILE SERVICES,” pp. 1191–1198.

[3] M. D. Rizkiyanto, M. D. Purbolaksono, and W. Astuti, “Sentiment Analysis Classification on PLN Mobile Application Reviews using Random Forest Method and TF-IDF Feature Extraction,” *INTEK J. Penelit.*, vol. 11, no. 1, p. 37, 2024, doi: 10.31963/intek.v11i1.4774.

[4] Y. Astuti, Yova Ruldeviyani, Faris Salbari, and Aldiansah Prayogi, “Sentiment Analysis of Electricity Company Service Quality Using Naïve Bayes,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 2, pp. 389–396, 2023, doi: 10.29207/resti.v7i2.4627.

[5] Ihsan Zulfahmi, “Analisis Sentimen Aplikasi PLN Mobile Menggunakan Metode Decission Tree,” *J. Penelit. Rumpun Ilmu Tek.*, vol. 3, no. 1, pp. 11–21, 2023, doi: 10.55606/juprit.v3i1.3096.

[6] Y. Asri, W. N. Suliyanti, D. Kuswardani, and M. Fajri, “Pelabelan Otomatis Lexicon Vader dan Klasifikasi Naive Bayes dalam menganalisis sentimen data ulasan PLN Mobile,” *Petir*, vol. 15, no. 2, pp. 264–275, 2022, doi: 10.33322/petir.v15i2.1733.

[7] R. Ridwan, E. H. Hermaliani, and M. Ernawati, “Penerapan: Penerapan Metode SMOTE Untuk Mengatasi Imbalanced Data Pada Klasifikasi Ujaran Kebencian,” *Comput. Sci.*, vol. 4, no. 1, pp. 80–88, 2024.

[8] A. K-nn, Y. P. Banjarnahor, A. Gultom, A. Siagian, and P. D. P. Silitonga, “Penanganan Data Ketidakseimbangan dalam Pendekatan SMOTE Guna Meningkatkan akurasi,” vol. 1, no. 2, pp. 473–478, 2024.

[9] W. I. Sabilla and C. Bella Vista, “Implementasi SMOTE dan Under Sampling pada Imbalanced Dataset untuk Prediksi Kebangkrutan Perusahaan,” *J. Komput. Terap.*, vol. 7, no. 2, pp. 329–339, 2021, doi: 10.35143/jkt.v7i2.5027.

[10] Nurdin, M. Hutomi, M. Qamal, and B. Bustami, “Sistem Pengecekan Toko Online Asli atau Dropship pada Shopee Menggunakan Algoritma Breadth First Search,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 6, pp. 1117–1123, 2020, doi: 10.29207/resti.v4i6.2514.

[11] D. Rifaldi, Abdul Fadlil, and Herman, “Teknik Preprocessing Pada Text Mining Menggunakan Data Tweet ‘Mental Health,’” *Decod. J. Pendidik. Teknol. Inf.*, vol. 3, no. 2, pp. 161–171, 2023, doi: 10.51454/decode.v3i2.131.

[12] J. Hemanth, R. B. Joy, and I.-Z. Chen, *Lecture Notes on Data Engineering and Communications Technologies 57 Intelligent Data Communication Technologies and Internet of Things Proceedings of ICICI 2020*. 2021.

[13] A. W. Pradana and M. Hayaty, "The Effect of Stemming and Removal of Stopwords on the Accuracy of Sentiment Analysis on Indonesian-language Texts," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 3, pp. 375–380, 2019, doi: 10.22219/kinetik.v4i4.912.

[14] S. Riyadi, N. Gita Mahardika, C. Damarjati, and S. Ramli, "Modified Convolutional Neural Network for Sentiment Classification: A Case Study on The Indonesian Electoral Commission," *Sci. J. Informatics*, vol. 11, no. 2, p. 493, 2024, doi: 10.15294/sji.v11i2.4929.

[15] N. A. S. Abdullah and N. I. A. Rusli, "Multilingual sentiment analysis: A systematic literature review," *Pertanika J. Sci. Technol.*, vol. 29, no. 1, pp. 445–470, 2021, doi: 10.47836/pjst.29.1.25.

[16] E. Sutoyo and A. Almaarif, "Twitter sentiment analysis of the relocation of Indonesia's capital city," *Bull. Electr. Eng. Informatics*, vol. 9, no. 4, pp. 1620–1630, 2020, doi: 10.11591/eei.v9i4.2352.

[17] K. Tri Putra, M. Amin Hariyadi, and C. Crysdian, "Perbandingan Feature Extraction Tf-Idf Dan Bow Untuk Analisis Sentimen Berbasis Svm," *J. Cahaya MAndalika*, p. 1449, 2023.

[18] J. A. Putra, A. Dharmawan, and J. Gondohanindijo, "Sentimen analisis aplikasi digitalent mobile menggunakan naïve bayes dan svm dengan ekstraksi fitur tf-idf sentimen analysis digitalent mobile application using naïve bayes and svm with tf-idf fitur extraction," vol. 7, 2024.

[19] M. Rahardi, A. Aminuddin, F. F. Abdulloh, and R. A. Nugroho, "Sentiment Analysis of Covid-19 Vaccination using Support Vector Machine in Indonesia," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 6, pp. 534–539, 2022, doi: 10.14569/IJACSA.2022.0130665.

[20] M. Rahayu, A. Luthfiarta, L. Cahyaningrum, and A. Nurfaiza Azzahra, "Pengaruh Oversampling dan Cross Validation Pada Model Machine Learning Untuk Sentimen Analisis Kebijakan Luaran Kelulusan Mahasiswa," *J. Media Inform. Budidarma*, vol. 8, no. 1, pp. 163–172, 2024, doi: 10.30865/mib.v8i1.7012.

[21] L. Cahyaningrum, A. Luthfiarta, and M. Rahayu, "Sentiment Analysis on the Impact of MBKM on Student Organizations Using Supervised Learning with Smote to Handle Data Imbalance," *Inf. J. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 9, no. 1, pp. 58–66, 2024.

[22] P. Karthika, R. Murugeswari, and R. Manoranjithem, "Sentiment Analysis of Social Media Network Using Random Forest Algorithm," *IEEE Int. Conf. Intell. Tech. Control. Optim. Signal Process. INCOS 2019*, pp. 1–5, 2019, doi: 10.1109/INCOS45849.2019.8951367.

[23] M. Park, D. Jung, S. Lee, and S. Park, "Heatwave damage prediction using random forest model in Korea," *Appl. Sci.*, vol. 10, no. 22, pp. 1–12, 2020, doi: 10.3390/app10228237.

[24] N. Istiqamah and M. Rijal, "Klasifikasi Ulasan Konsumen Menggunakan Random Forest dan SMOTE," *J. Syst. Comput. Eng.*, vol. 5, no. 1, pp. 66–77, 2024, doi: 10.61628/jsce.v5i1.1061.

[25] I. B. Prakoso, D. Richasdy, and M. D. Purbolaksono, "Sentiment Analysis of Telkom University as the Best BPU in Indonesia Using the Random Forest Method," *J. Media Inform. Budidarma*, vol. 6, no. 4, p. 2050, 2022, doi: 10.30865/mib.v6i4.4567.

[26] C. D. Angelina and Painem, "Penerapan Algoritma Naive Bayes Classifier Pada Analisis Sentimen Masyarakat Indonesia Terhadap Childfree Pada Twitter," *Semin. Nas. Mhs. Fak. Teknol. Inf.*, vol. 2, no. 2, pp. 398–407, 2023.