JSCE (Journal of System and Computer Engineering)
Vol.6, No.4, October 2025, pp. 318-328
ISSN (online): 2723-1240
DOI : https://doi.org/10.61628/jsce.v6i4.2158

Comparison Analysis of Naive Bayes and K-Nearest Neighbor Algorithms in Classifying Language Styles in Indonesian Texts

Fika Tsalsabila Tinanda [1,a]; Herry Sujaini [2,b]; Helfi Nasution [3,c,*]
[1,2,3] Program Studi Informatika, Universitas Tanjungpura, Indonesia
[a] fikatsalsabilat15@gmail.coml; [b] hs@untan.ac.id; [c] helfinas@informatika.untan.ac.id
** Corresponding author*

Abstract

The rapid growth of Indonesian-language digital texts often involves figurative language, yet large-scale identification remains challenging due to class imbalance. This study introduces a comparative evaluation of Naïve Bayes and K-Nearest Neighbor (KNN) algorithms for classifying figurative language styles in Indonesian texts, while examining the effect of SMOTE data balancing and hyperparameter tuning. Using 5,155 original samples and 6,240 balanced samples, models were tested under four scenarios (with/without SMOTE and tuning) on an 80:20 split. Results indicate that Naïve Bayes maintained stable performance with an accuracy of 93.19%, whereas KNN reached its best accuracy of 93.43% after SMOTE and tuning. These findings demonstrate that data balancing and parameter optimization significantly enhance classification performance, providing a methodological contribution to computational linguistics and advancing automatic figurative language detection in Indonesian texts.

Keywords— *Language Style Classification, Naive Bayes, K-Nearest Neighbor (KNN), Synthetic Minority Over-sampling Technique (SMOTE), Tuning Hyperparameter*

1. Introduction

In today's digital era, Indonesian-language textual data is proliferating rapidly across social media, online news, blogs, and digital documents, creating new challenges in information management and comprehension. Natural Language Processing (NLP), a branch of artificial intelligence, plays a crucial role in addressing these challenges by enabling computers to automatically process, analyze, and interpret human language (Busiarli et al., 2016). The increasing demand for Indonesian NLP systems stems from the massive growth of textual data, which often contains complex linguistic features such as figurative language that complicate semantic analysis.

Figurative language—including personification, metaphor, hyperbole, euphemism, and irony—serves to enrich meaning and strengthen expression in communication. In Indonesian, where cultural and contextual nuances are central to language use, figurative expressions are pervasive and carry significant implications for sentiment analysis, opinion mining, and adaptive learning systems. However, despite its importance, research on automatic figurative language classification in Indonesian texts remains scarce. Most existing studies focus on general text classification or sentiment analysis, without specifically addressing the identification of figurative styles. This gap highlights the need for more systematic approaches to figurative language processing in Indonesian.

Among the commonly used algorithms, Naïve Bayes and K-Nearest Neighbor (KNN) offer contrasting advantages. Naïve Bayes relies on probabilistic word occurrences and is recognized for efficiency and reliable performance even with limited training data (Dewi et al., 2021). Conversely, KNN classifies text based on similarity to labeled data, making it capable of capturing non-linear patterns and more robust to noise (Hendriyanto & Sari, 2022). While these algorithms have been explored in various text mining tasks, no prior study has directly compared their effectiveness in Indonesian figurative language classification, leaving a methodological gap in computational linguistics research.

Another challenge lies in imbalanced class distribution, where minority figurative styles are underrepresented. This imbalance often degrades model accuracy, limiting its applicability in real-world scenarios. To overcome this, the Synthetic Minority Over-sampling Technique (SMOTE) is employed to improve class representation, enabling the model to better learn from underrepresented categories (Sharfina & Ramadhan, 2023). Moreover, hyperparameter tuning is conducted, as variations in parameter configurations can significantly affect model performance (Arifadilah, 2023).

This study therefore contributes to the field by (1) providing the first comparative evaluation of Naïve Bayes and KNN for Indonesian figurative language classification, (2) demonstrating the role of SMOTE in mitigating data imbalance, and (3) showing the effectiveness of hyperparameter tuning in enhancing accuracy. By filling these gaps, the research not only strengthens Indonesian NLP resources but also offers methodological insights that can be extended to other low-resource languages facing similar challenges.

2. Method

This study employs a quantitative approach using machine learning–based text classification methods. Two supervised learning algorithms, Naïve Bayes and K-Nearest Neighbor (KNN), were implemented to classify figurative language styles in Indonesian texts. The overall research process includes data collection, preprocessing with Natural Language Processing (NLP) techniques, classification using Naïve Bayes and KNN, data balancing with SMOTE, hyperparameter tuning, and model evaluation. The evaluation was conducted using accuracy, precision, recall, and F1-score to identify the most effective classification model. Figure 1 illustrates the research flow.

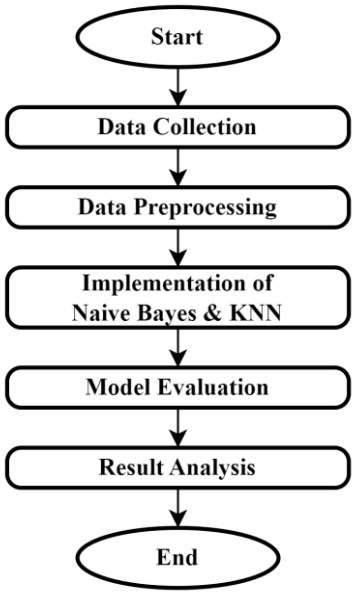

Figure 1. Flowchart of the Research Process

2.1 Data Collection

The dataset was compiled from various sources, including literary works and texts generated with the assistance of the ChatGPT artificial intelligence model. To ensure validity and reliability, all sentences were verified by Indonesian language experts. The final dataset contained 5,155 samples, categorized into five figurative language styles: personification, metaphor, hyperbole, euphemism, and irony (see Table 1).

Tabel 1. Sample Count per Figurative Language Style

No	Language Style	Sample Count
1	Personification	1247
2	Metaphor	1227
3	Hyperbole	1248
4	Euphemism	636
5	Irony	797

The class distribution was imbalanced, particularly for euphemism and irony. To address this, the Synthetic Minority Oversampling Technique (SMOTE) was later applied, increasing the dataset to 6,240 samples with more balanced representation across classes. Both the original and the SMOTE-augmented datasets were split into 80% training data and 20% testing data for model training and evaluation.

2.2 Data Preprocessing

Data preprocessing is a crucial step in Natural Language Processing (NLP) to transform raw text into a clean and structured format suitable for machine learning. In this study, preprocessing was conducted through several stages to ensure that the data was consistent, noise-free, and representative of linguistic patterns relevant to figurative language classification. The stages are explained in detail as follows:

2.2.1 Case Folding

All characters in the text were converted into lowercase letters. For example, words such as *"SAYA"* and *"saya"* were treated as the same token. This step reduced redundancy caused by variations in letter case and ensured uniform representation of words in the dataset. Without case folding, the system might interpret the same word in different cases as separate features, which would unnecessarily increase the feature space.

2.2.2 Cleansing Data

In this step, non-linguistic elements such as punctuation marks, numbers, HTML tags, and special characters were removed from the text. For instance, a sentence like *"Dia berlari!!!"* was converted to *"dia berlari"*. This step aimed to eliminate irrelevant noise that does not contribute to semantic meaning. By focusing only on valid words, data cleansing improved the quality of features extracted for classification and reduced the possibility of models being biased by meaningless symbols.

2.2.3 Stopwords Removal

Stopwords are words that frequently occur in a language but carry minimal semantic weight in text analysis, such as *"dan," "atau," "yang," "adalah,"* in Indonesian or *"and," "is," "or"* in English. These words were removed because they tend to dominate the dataset without providing discriminative information for figurative language classification. For example, the phrase *"dia adalah cahaya"* would still preserve its figurative meaning after removing the word

"adalah." This step helped reduce dimensionality and allowed the algorithms to focus on more informative terms that represent figurative expressions.

2.2.4 TF-IDF Preparation

Before weighting, sentences were segmented into individual tokens (words). For example, the sentence *"Hatinya sekeras batu"* would be tokenized into [*hatinya*, *sekeras*, *batu*]. Tokenization enabled the representation of each word as a basic analysis unit, which is essential for machine learning–based classification.

2.2.5 TF-IDF Weighting

After tokenization and stopwords removal, features were transformed into numerical vectors using Term Frequency–Inverse Document Frequency (TF-IDF) weighting (Saleh, 2015). TF-IDF calculates the importance of a word in a document relative to its frequency across the entire dataset. Words that appear frequently in one class but rarely across others (e.g., "batu" in metaphoric contexts) received higher weights. Conversely, very common words across all classes (e.g., "dia") received lower weights. This ensured that the classification algorithms focused on terms that were distinctive for identifying figurative language styles.

Through these preprocessing steps, the dataset was converted into a structured representation where each sentence was transformed into a weighted vector of relevant tokens. This process significantly enhanced the ability of the machine learning models to recognize patterns associated with different figurative language styles.

2.3 Classification

The classification process in this study was carried out through several steps. First, if the experiment involved hyperparameter tuning, the tuning process was conducted beforehand to determine the best parameters, such as the alpha value in Naive Bayes and the number of neighbors (k) in KNN, using the Grid Search method. Once the optimal parameters were obtained or if tuning was not applied, the next step was to process the dataset according to the selected approach. If SMOTE was used, the data was balanced first before being split into training and testing sets. If SMOTE was not applied, the dataset was directly split without modifying the distribution of samples across classes.

After the data was prepared, the classification algorithms, namely Naive Bayes and K Nearest Neighbor (KNN), were implemented. Naive Bayes calculates the probability of words in a document based on their frequency of occurrence (Annur, 2018), while KNN determines the class based on the majority vote of the k nearest neighbors (Putri et al., 2023). In addition, Term Frequency Inverse Document Frequency (TF-IDF) was used to assign weights to words based on their relevance to specific language styles in the context of classification tasks (Saleh, 2015).

To address class imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was applied to generate synthetic samples for the minority class, resulting in a more balanced data distribution (Fatiya, 2021). This technique solves the problem by generating synthetic data for the minority class so that it becomes balanced with the majority class (Farizki, 2023). The classification results were then evaluated using accuracy, precision, recall, and F1 score metrics. Each combination of approach (SMOTE or non-SMOTE, Naive Bayes or KNN, tuning or non-tuning) was analyzed to determine the best-performing model. This study involved eight models based on different method combinations.

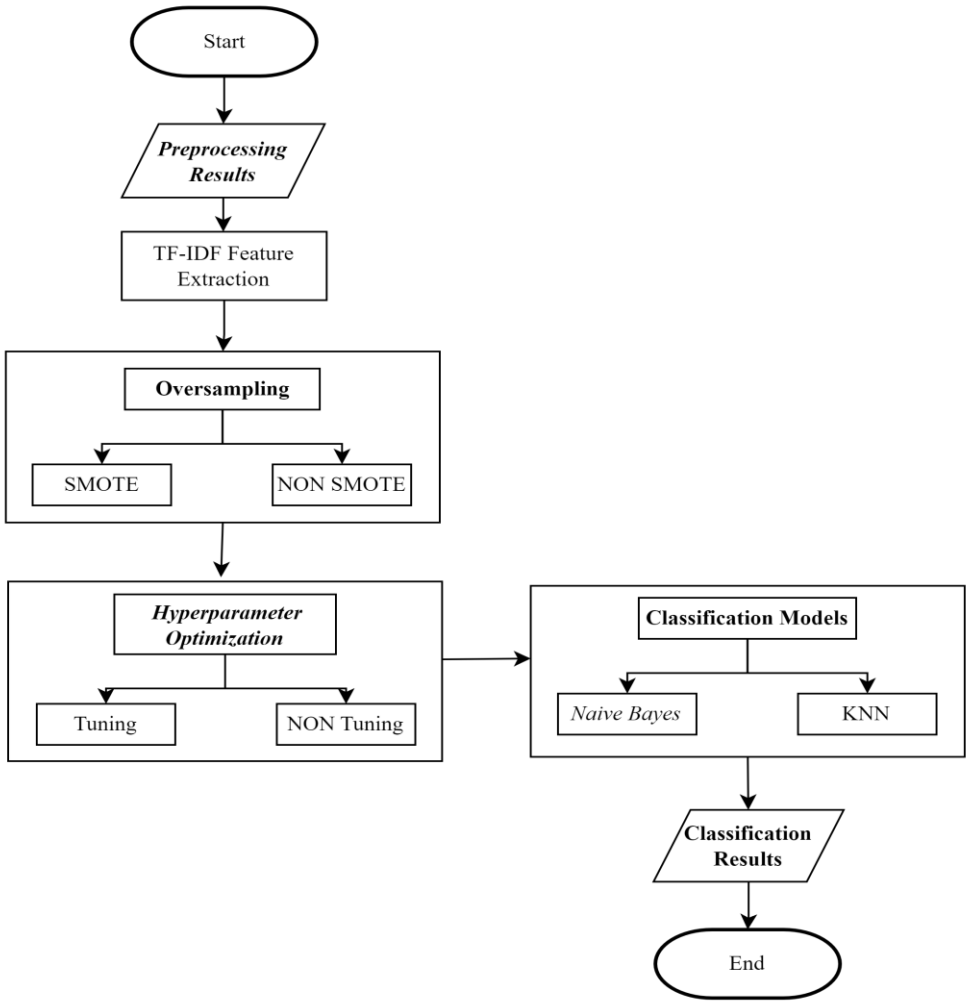

Figure 2. Classification Process

2.4 Data Balancing with SMOTE

Since the dataset was imbalanced, SMOTE was applied to generate synthetic data points for underrepresented classes. This approach produces new samples by interpolating between existing minority instances, thereby balancing the dataset (Fatiya, 2021; Farizki, 2023). After applying SMOTE, the dataset expanded from 5,155 to 6,240 samples, which reduced bias toward majority classes and improved model learning.

2.5 Model Evaluation

All models were evaluated using accuracy, precision, recall, and F1-score. These metrics enabled a comprehensive assessment of classification performance beyond overall accuracy, particularly for imbalanced data. The comparative evaluation of Naïve Bayes and KNN across all experimental conditions allowed identification of the best-performing algorithm and demonstrated the impact of SMOTE and hyperparameter tuning on model effectiveness.

3. Results And Discussion

This study compared eight experimental models combining Naïve Bayes and KNN with different conditions of SMOTE and hyperparameter tuning. Model performance was evaluated

using accuracy, precision, recall, F1-score, and confusion matrices to provide both quantitative measurement and qualitative insights into classification stability.

3.1 Data Collection Result

A total of 5,155 sentences were collected and categorized into five figurative language styles: hyperbole, personification, metaphor, irony, and euphemism. Each class contained a different number of samples, which led to data imbalance. The dataset was then saved in CSV format to enable further computational processing (Figure 3).

```
Jumlah Data Masing-masing Gaya Bahasa:
Label
Hiperbola        1248
Personifikasi    1247
Metafora         1227
Ironi             797
Eufemisme         636
Name: count, dtype: int64
Total Data: 5155
```

Figure 3. Data Collection Result

3.2 Data Processing Result

The dataset underwent preprocessing to ensure quality and consistency. The steps included case folding, cleansing, stopwords removal, tokenization, and TF-IDF transformation (Saleh, 2015). These processes reduced noise and emphasized distinctive linguistic features necessary for figurative language classification. The preprocessing output is illustrated in Figure 4.

```
Hasil Preprocessing:
     Gaya Bahasa                                    Contoh Kalimat  \
0      Eufemisme  Paranormal berkedok agama itu akhirnya dihukum...
1      Hiperbola  Kelelahan ini seperti beban berat yang mengika...
2          Ironi  Saya tahu anda seorang gadis yang paling canti...
3       Metafora  Maling itu mengambil langkah seribu ketika dik...
4  Personifikasi  Angin sore mengalir dengan lembut, membawa aro...

                                  Case Folding  \
0  paranormal berkedok agama itu akhirnya dihukum...
1  kelelahan ini seperti beban berat yang mengika...
2  saya tahu anda seorang gadis yang paling canti...
3  maling itu mengambil langkah seribu ketika dik...
4  angin sore mengalir dengan lembut, membawa aro...

                                 Cleansing Data  \
0  paranormal berkedok agama itu akhirnya dihukum...
1  kelelahan ini seperti beban berat yang mengika...
2  saya tahu anda seorang gadis yang paling canti...
3  maling itu mengambil langkah seribu ketika dik...
4  angin sore mengalir dengan lembut membawa arom...

                              Stopwords Removal
0  paranormal berkedok agama dihukum masuk jeruji...
1            kelelahan beban berat mengikat tubuhku
2                    gadis cantik dunia terhormat
3     maling mengambil langkah seribu dikejar warga
4  angin sore mengalir lembut membawa aroma musim...
```

Figure 4. Data Processing Result

3.3 Data Splitting

The dataset was partitioned into training and testing subsets using the train_test_split function, with an 80:20 ratio. This split ensured that the majority of data was allocated for model learning while reserving a smaller portion for unbiased evaluation. To address class imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was applied exclusively to the training set, thereby preventing data leakage from synthetic samples into the test set. This approach guaranteed that the evaluation phase would reflect the model's true performance on unseen, real data. In the original (imbalanced) dataset, the training set consisted of 4,124 samples, while the test set contained 1,031 samples. After applying SMOTE, the training set

was expanded to 4,992 samples, and the test set to 1,248 samples. Importantly, SMOTE balanced the dataset by generating synthetic samples for the minority class, ensuring that each class contained exactly 1,248 samples. This balance mitigates bias during model training and improves the classifier's ability to generalize across both majority and minority classes.

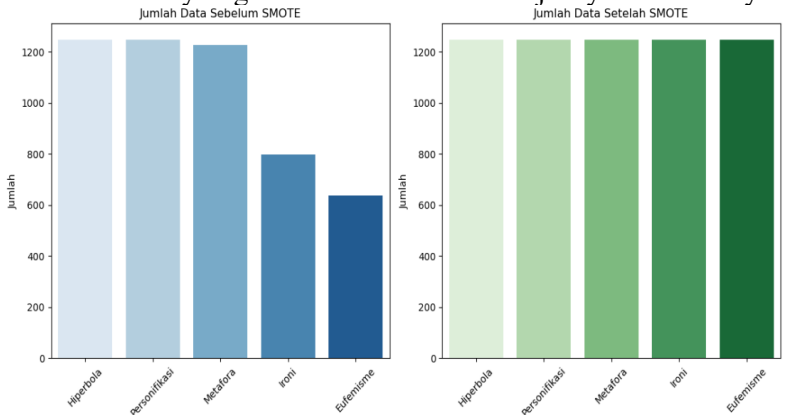

Figure 5. SMOTE Result

The distribution of samples across different scenarios (with and without SMOTE) is summarized in Table 2, providing a clear comparison of how the dataset changed after preprocessing.

Tabel 2. Data Distribution for Each Model Testing

Scenario	Algorithm	Training Data	Testing Data	Total Data
1. No SMOTE & No Tuning	*Naive Bayes*	4.124	1.031	5.155
	K-nearest neighbor			
2. SMOTE & No Tuning	*Naive Bayes*	4.992	1.248	6.240
	K-nearest neighbor			
3. No SMOTE & Tuning	*Naive Bayes*	4.124	1.031	5.155
	K-nearest neighbor			
4. SMOTE & Tuning	*Naive Bayes*	4.992	1.248	6.240
	K-nearest neighbor			

3.4 Model Evaluation Results

Model evaluation was conducted by comparing Naive Bayes and KNN under four conditions: without SMOTE and tuning, with SMOTE without tuning, without SMOTE with tuning, and with both SMOTE and tuning. The testing was performed on 5,155 samples without SMOTE and 6,240 samples with SMOTE, classified into five language styles: euphemism, hyperbole, irony, metaphor, and personification. The evaluation results can be seen in Table 4. A bar chart is used to compare the accuracy of each model under different conditions and their correct predictions.

Tabel 3. Model Evaluation Results

Scenario	Naïve Bayes				KNN			
	P	R	F	A (%)	P	R	F	A (%)
No SMOTE & No Tuning	0.88	0.87	0.87	87.29	0.77	0.79	0.80	77.11
SMOTE & No Tuning	0.92	0.92	0.92	91.67	0.84	0.84	0.84	83.97
No SMOTE & Tuning	0.89	0.91	0.90	90.11	0.79	0.79	0.79	79.24
SMOTE & Tuning	0.93	0.93	0.93	93.19	0.93	0.93	0.93	93.43

Explanation:

P : Precision; R : Recall; F : F1-Score; A : Accuracy

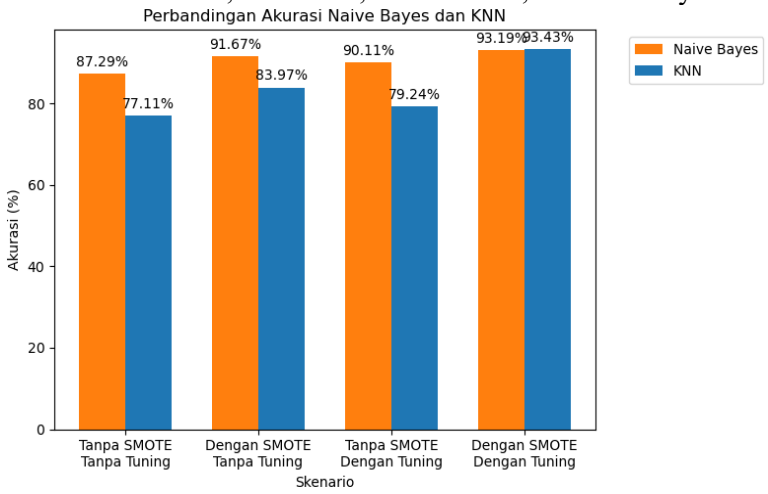

Figure 6. Accuracy Comparison Results of Naive Bayes and KNN in Various Scenarios

Figure.6 shows the accuracy comparison between the Naïve Bayes and K-Nearest Neighbor (KNN) algorithms across four different scenarios. In the initial condition with no SMOTE and no tuning, Naïve Bayes achieved an accuracy of 87.29%, significantly outperforming KNN, which only reached 77.11%. This indicates that Naïve Bayes performs more consistently when handling imbalanced data, whereas KNN tends to be less optimal in such cases.

After applying SMOTE, both algorithms saw an increase in accuracy. Naïve Bayes improved to 91.67%, while KNN rose to 83.97%. Meanwhile, in the condition without SMOTE but with tuning, Naïve Bayes reached 90.11% accuracy, and KNN increased to 79.24%. Tuning had a positive effect on both models' performance, but Naïve Bayes still outperformed KNN in these scenarios, demonstrating its robustness across different conditions.

In the best-performing condition, where both SMOTE and tuning were applied, KNN finally surpassed Naïve Bayes with an accuracy of 93.43%, compared to 93.19%. This is the only scenario where KNN showed higher performance, indicating that this algorithm heavily depends on balanced data distribution and optimal parameter selection.

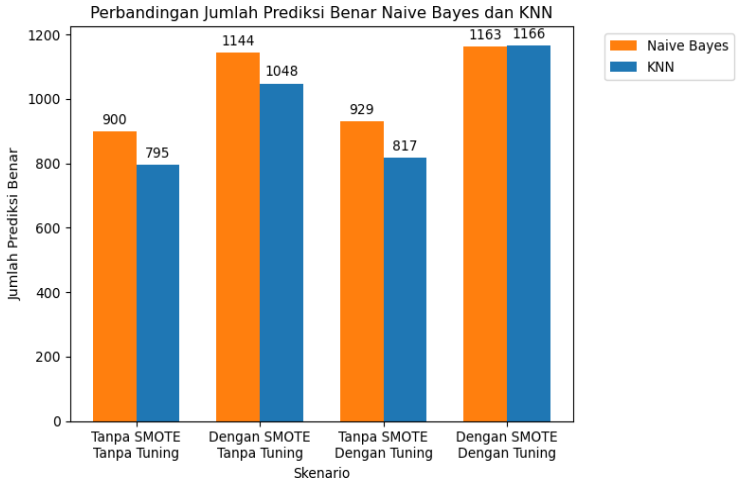

Figure 7. Correct Predictions from the Confusion Matrix Across Different Conditions

Figure 7 illustrates the number of correct predictions produced by each model in various testing scenarios. In the initial condition without SMOTE and without tuning, out of a total of 1,031 test samples, Naïve Bayes achieved 900 correct predictions, while KNN reached only 795. This suggests that Naïve Bayes is more reliable under basic conditions, while KNN shows lower accuracy and a higher rate of misclassification.

After applying SMOTE, both algorithms experienced a significant improvement. Naïve Bayes recorded 1,144 correct predictions out of 1,248 test samples an increase of 244 compared to the initial condition. KNN also showed a major performance boost with 1,048 correct predictions, an increase of 253. This indicates that SMOTE is highly effective in improving class distribution, especially for KNN, which was more affected by data imbalance.

In the scenario without SMOTE but with tuning, the number of correct predictions increased but not as much as with SMOTE. Naïve Bayes rose to 929, while KNN reached 817. The best results occurred when SMOTE and tuning were applied together: Naïve Bayes achieved 1,163 correct predictions, and KNN slightly surpassed it with 1,166. These results show that KNN's performance heavily relies on data processing and parameter tuning, whereas Naïve Bayes maintains consistent and stable performance across different conditions.

This study also presents several limitations that should be acknowledged. First, part of the dataset was generated with the assistance of ChatGPT and subsequently verified by language experts; however, data validation could be strengthened by conducting inter-annotator agreement or reliability checks among multiple experts. Second, the analysis mainly emphasized model accuracy without addressing linguistic challenges inherent in Indonesian figurative language, such as metaphorical ambiguity or irony, which could enrich the discussion. Third, the explanation of hyperparameter tuning remains general; specific configurations, such as the smoothing (α) value in Naïve Bayes and the choice of *k* and distance metrics in KNN, should be reported in detail to ensure reproducibility. Fourth, the benchmark was limited to only two algorithms (Naïve Bayes and KNN), whereas incorporating additional models such as SVM or Random Forest would provide more competitive insights. Lastly, while the tables and figures clearly present performance results, the analysis remains descriptive; further discussion is needed to explain why Naïve Bayes tends to be more stable on imbalanced data, while KNN benefits more significantly from SMOTE and hyperparameter tuning.

4. Conclusions

This study evaluated the performance of Naïve Bayes and K-Nearest Neighbor (KNN) in classifying Indonesian figurative language styles under four experimental conditions: without SMOTE or tuning, with SMOTE only, with tuning only, and with both SMOTE and tuning. The comparative design was intended to assess how class balancing through SMOTE and parameter

optimization through tuning influenced the effectiveness of both algorithms. By testing these scenarios systematically, the study was able to capture not only the overall accuracy of the models but also their stability across different data distributions and preprocessing strategies.

The results indicate that Naïve Bayes consistently achieved strong performance across most conditions, reaching up to 93.19% accuracy. This shows that Naïve Bayes, as a probabilistic classifier, is relatively robust to class imbalance and does not rely heavily on additional preprocessing. On the other hand, KNN demonstrated weaker results in baseline conditions but achieved the highest overall accuracy of 93.43% when both SMOTE and tuning were applied. This confirms that KNN is highly sensitive to data distribution and parameter settings, benefiting significantly from balanced training data and optimized hyperparameters. Moreover, SMOTE played a critical role in improving minority class recognition, with its impact being more pronounced on KNN than on Naïve Bayes.

These findings underscore the importance of data balancing and parameter optimization in figurative language classification for Indonesian texts. The study contributes by providing a comparative benchmark between probabilistic (Naïve Bayes) and instance-based (KNN) approaches under different preprocessing conditions, offering valuable insights for researchers and practitioners working with imbalanced datasets. For future research, it is recommended to explore additional machine learning and deep learning models such as Support Vector Machines (SVM), Random Forests, or Transformer-based architectures (e.g., BERT, IndoBERT), which have shown strong performance in recent NLP studies (Devlin et al., 2019; Ranasinghe & Zampieri, 2021). Furthermore, the integration of word embeddings and contextualized language models could further enhance the detection of figurative language in Indonesian and other low-resource languages, paving the way for more accurate and context-aware classification systems.

This study is limited by the nature of its dataset, the lack of deeper linguistic analysis, the general description of hyperparameter tuning, and the restricted algorithm benchmark. Future research should address these issues by improving dataset validation (e.g., inter-annotator agreement), incorporating linguistic perspectives in figurative language analysis, reporting detailed hyperparameter configurations for reproducibility, and extending the benchmark to include algorithms such as SVM, Random Forest, or deep learning methods.

References

Annur, H. (2018). Classification of poor communities using the Naïve Bayes method. *August, 10*(2).

Annur, M. (2018). Text classification using Naïve Bayes algorithm. *Journal of Physics: Conference Series*.

Arifadilah, F. (2023). Comparison of hyperparameter optimization: Population Based Training, Random Search, and Bayesian Optimization in radicalism sentiment analysis.

Arifadilah, R. (2023). Optimization of hyperparameters in text classification models. *Procedia Computer Science*.

Busiarli, N., Aditya, L. A., & Andika, A. Y. (2016). Application of the Naïve Bayes algorithm and natural language processing for classifying types of news in news archives. *Proceedings of the National Seminar on Information and Communication Technology*, 6–7. http://www.kopertis3.or.id/html/wp-

Choudhary, S., Gupta, R., & Kumar, A. (2023). Advances in text classification: A comprehensive review. *Expert Systems with Applications, 216,* 119541. https://doi.org/10.1016/j.eswa.2022.119541

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT 2019: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. https://aclanthology.org/N19-1423

Dewi, P. S., Sastradipraja, C. K., & Gustian, D. (2021). Decision support system for job promotion using the Naïve Bayes classifier algorithm method. *Jurnal Teknologi dan Informasi (JATI).*

Dewi, R., et al. (2021). Comparative analysis of Naïve Bayes performance in text classification.

Farizki, H. (2023). The effect of Synthetic Minority Oversampling Technique (SMOTE) in sentiment analysis using the Support Vector Machine (SVM) algorithm.

Farizki, R. (2023). Data balancing with SMOTE for text classification tasks.

Fatiya, N. (2021). Synthetic oversampling in NLP for imbalanced datasets.

Fatiya, R. (2021). The effect of SMOTE (Synthetic Minority Oversampling Technique) to overcome data imbalance in sentiment analysis using the K-Nearest Neighbors algorithm.

Hendriyanto, M. D., & Sari, N. (2022). Application of the K-Nearest Neighbor algorithm in classifying hoax news titles.

Hendriyanto, A., & Sari, R. (2022). Evaluating KNN for Indonesian text categorization.

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L. E., & Brown, D. (2019). Text classification algorithms: A survey. *Information, 10*(4), 150. https://doi.org/10.3390/info10040150

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning for text classification: A survey. *ACM Computing Surveys, 54*(3), 1–40. https://doi.org/10.1145/3439726

Putri, A., et al. (2023). KNN for multi-class text classification in Indonesian datasets.

Putri, T. A. E., Widiharih, T., & Santoso, R. (2023). Hyperparameter tuning using RandomSearchCV on Adaptive Boosting for predicting survival of heart failure patients. *Jurnal Gaussian, 11*(3), 397–406. https://doi.org/10.14710/j.gauss.11.3.397-406

Ranasinghe, T., & Zampieri, M. (2021). Multilingual offensive language identification with transformer models. *Proceedings of the ACL Workshop*. https://doi.org/10.18653/v1/2021.woah-1.20

Saleh, A. (2015). Implementation of the Naïve Bayes classification method to predict household electricity usage.

Saleh, M. (2015). Implementation of TF-IDF weighting in Indonesian text classification.

Sharfina, N., & Ramadhan, N. G. (2023). SMOTE analysis in Hepatitis C classification using Random Forest and Naïve Bayes. *Jurnal Teknologi dan Sistem Komputer, 7*(1). https://doi.org/10.14710/jtsiskom.7.1

Sharfina, Z., & Ramadhan, R. (2023). SMOTE applications in handling imbalanced Indonesian text data.