

# Analysis of language identification algorithms for regional Indonesian languages

Herry Sujaini, Arif Bijaksana Putra

Department of Informatics, Faculty of Engineering, University of Tanjungpura, Pontianak, Indonesia

## Article Info

### Article history:

Received May 7, 2023

Revised Oct 28, 2023

Accepted Nov 15, 2023

### Keywords:

Algorithm comparison

K-nearest neighbors

Language identification

Naïve Bayes

N-gram feature

## ABSTRACT

Detecting local languages in Indonesia is essential for recognizing linguistic diversity, promoting intercultural understanding, preserving endangered languages, and improving access to education and services. By identifying and documenting these languages, we can support language preservation efforts, provide tailored resources for communities, and celebrate the unique cultural heritage of different ethnic groups. Ultimately, this encourages a more accepting and open-minded society, prioritizing various languages and cultural customs. This research aims to identify the most suitable algorithm for language detection in Indonesian regional languages and gain insights into their unique characteristics through n-gram analysis. By understanding language diversity, the study contributes to preserving Indonesia's cultural and linguistic heritage and improving language detection techniques. This study compares the performance of five algorithms (Naïve Bayes, K-nearest neighbors (KNN), least-squares, Kullback Leibler divergence, and Kolmogorov Smirnov test) to determine the most accurate and efficient method for language identification. Incorporating trigram features alongside unigrams and bigrams significantly improved the model's performance, with F1 scores increasing from 0.923 to 0.959. The study found that using more features leads to better accuracy, with Naïve Bayes and KNN emerging as the top-performing algorithms for language identification.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Herry Sujaini

Department of Informatics, Faculty of Engineering, University of Tanjungpura, Indonesia

Jl. Prof. Hadari Nawawi, Pontianak, Indonesia

Email: hs@untan.ac.id

## 1. INTRODUCTION

Indonesia is a country located in Southeast Asia, consisting of more than 17,000 islands and home to over 270 million people. It is the fourth most populous country in the world and has a rich cultural heritage with over 300 ethnic groups [1]. This diversity extends to the country's linguistic landscape, as Indonesia is home to more than 700 languages [2].

Indonesia recognizes only one national language, which is Bahasa Indonesia, based on the Malay language, and it serves as a common language for the country. However, it does not imply that the local languages are less significant. In reality, several Indonesians continue to use their local languages in their everyday lives, such as at home, at work, and in their communities. The linguistic diversity of Indonesia is due to its geographical and historical factors. The country's vast archipelago with numerous islands has created a natural obstacle that has contributed to the evolution of unique languages and dialects. Additionally, Indonesia's past colonization by various European powers and trading relationships with its neighboring countries have also influenced the development and usage of different languages.

Despite the rich linguistic diversity in Indonesia, many of the local languages are endangered [3]. The Indonesian government has recognized this issue and has taken steps to preserve and promote local languages through policies such as providing education in local languages and encouraging the use of local languages in media and literature. Indonesia's linguistic diversity is a reflection of its cultural richness and complex history. However, it is important to address the challenges that threaten the survival of local languages in the country to ensure that this diversity is preserved for future generations.

Detecting local languages in Indonesia is crucial for several reasons. First and foremost, it helps to recognize and acknowledge the linguistic diversity that exists within the country. By identifying and documenting local languages, we gain a better understanding of the cultural heritage and traditions of different communities in Indonesia. This knowledge can be used to promote intercultural understanding and respect among the different ethnic groups in the country. Furthermore, detecting local languages is important for language preservation efforts. Many of the local languages in Indonesia are endangered, and without proper documentation and recognition, they may be lost forever. Detecting and identifying these languages is a critical step toward promoting language preservation efforts, such as language revitalization and documentation programs.

In addition, detecting local languages is important for education and communication purposes. Local languages are often used in communities as the primary means of communication, particularly in rural areas. By detecting and identifying local languages, we can provide better language education and resources to communities that speak these languages, improving their access to education, healthcare, and other basic services. Finally, detecting local languages can help to promote linguistic diversity and cultural heritage. By recognizing and promoting local languages, we can celebrate the unique identities and traditions of different communities in Indonesia. This can help to foster a sense of pride and belonging among these communities and promote a more inclusive and tolerant society.

There are numerous techniques available for identifying the language of a given text, and these can be broadly categorized into different types of approaches. One popular category is statistical methods, which include techniques like Markov models and trigram frequency vectors [4]. Another category of methods is deep learning, which has emerged as a powerful tool for language identification in recent years [5]. Finally, there are alternative classification approaches that can be employed, such as decision trees, support vector machines (SVM), and k-nearest neighbor (KNN) algorithms. The choice of method will depend on various factors, including the size of the dataset, the complexity of the text data, and the computational resources available for the task.

The application of deep learning in languages with limited resources (LRLs) may not always be appropriate due to the insufficient amount of data that can be used for machine learning or other types of processing [6]. In such cases, the use of deep learning techniques may result in poor performance or even failure to produce meaningful results. This is because deep learning algorithms require a large amount of training data to learn and generalize patterns effectively. However, in LRLs, such data may not be available, making it challenging to train a deep-learning model. Therefore, alternative methods, such as statistical models, may be more suitable for language identification in LRLs. In this research, the Naïve Bayes algorithm, KNN, least-squares, Kullback Leibler divergence (K-L divergence), and Kolmogorov Smirnov test (K-S test) were used.

The primary objective of this research is to evaluate and compare several algorithms to identify the most suitable one for the task of language detection, particularly for regional languages in Indonesia that are classified as having limited resources. In addition to this, the study aims to gain insight into the unique characteristics of regional languages in Indonesia by analyzing their n-gram models. This research recognizes the importance of understanding the language diversity in Indonesia and aims to contribute to the development of effective language detection techniques that can be applied to identify and preserve the cultural and linguistic heritage of the country. The analysis of n-gram models of regional languages in Indonesia is expected to reveal distinct patterns and structures that can help in better understanding the linguistic features and characteristics of these languages. Overall, the research aims to provide valuable insights into the language diversity of Indonesia and contribute to the development of language detection algorithms that can effectively detect the language of the text in various applications.

In recent years, extensive research has been conducted in the area of language detection and identification, particularly in the context of multilingual and low resource settings such as Indonesia. A number of studies have investigated various methods and techniques for detecting and identifying regional languages in Indonesia. One common approach is based on deep learning and machine learning algorithms. For example: Utomo and Sibaroni [7] developed a text classification system to identify British and American English in sentences. The system achieved 96.53% accuracy using N-gram features, term frequency-inverse document frequency (TF-IDF) weighting, and a word dictionary with a 2.0 DF threshold. Babhulgaonkar and Sonavane [8] found that the SVM-based identifier achieved 89% accuracy, improving upon the traditional n-gram

approach by 18%, including language identification in machine translation improved translation quality. Dovbnia *et al.* [9] focused on identifying low-resource celtic languages. The goals were to collect a dataset, train a classification model, experiment with feature extraction methods, and evaluate performance on a reduced dataset. The study found that unsupervised features improved performance and were more robust to reduced labeled data. The best model achieved a 98% F1 score and 97% Matthews correlation coefficient (MCC) using dense neural networks.

Local language detection in Indonesia has been a research topic among several researchers. These researchers have conducted various studies to develop systems that can detect and identify local languages spoken in different regions of Indonesia. For instance, Saputri and Adriani [10] developed a spoken language identification system for Indonesian local languages using three features combined on the hidden layer of deep neural network (DNN). The system achieved high accuracy with an F1 score of 87.85%, 93.46%, and 96.73% for speech data with 3 seconds, 10 seconds, and 30 seconds duration respectively. Martadinata *et al.* [11] developed a language identification tool to automatically identify social media posts in Indonesian, Javanese, Sundanese, and Minangkabau. The statistical method is found to be the most effective among N-grams, statistical models, and the small words technique. The experiments show that the tool achieves the best results when trained on internet articles and tested on our constructed social media data.

The identification of language is a vital preprocessing step for many natural language processing (NLP) tasks, as it involves identifying the language of a given text. Language detection employs two main steps, which involve generating a document model for the text and generating a language model for each known language. Various techniques exist for language detection, including statistical methods, machine learning algorithms, and rule-based systems [12]–[14].

Statistical methods use statistical models to identify the language of a given text by computing profiles that consist of n-grams, which are sequences of n consecutive letters that appear in each known language [15]. Examples of statistical methods include Markov models, trigram frequency vectors, and n-gram based text categorization [16]. Machine learning algorithms employ training data to learn patterns in the text that can be used to identify its language, and examples include Naive Bayes, SVM, and neural networks [17], [18]. Rule-based systems use a set of rules based on linguistic features such as character sets, word frequency distributions, and grammatical structures to identify the language of a given text [19].

The applications of language detection in NLP are widespread, as it is commonly used to process datasets that contain documents in different languages or to determine the language of a dataset before running further algorithms on it. Additionally, web crawlers can use language detection to find pages that are potentially written in multiple languages. In this research, we aim to detect local languages in Indonesia using machine learning algorithms, despite the limited availability of language resources. This is a challenging task, as the number of local languages in Indonesia is vast, and the resources available for each language are often limited. To address this challenge, we will use several machine learning algorithms that are known to work well with limited resources.

## 2. METHOD

### 2.1. Data and algorithms

In this study, we utilized 8 regional languages that are spoken in Indonesia, namely Javanese (Kromo), Batak, Sundanese, Bugis, Malay (Pontianak), Dayak (Taman), Madurese, and Minang. To gather data for our study, we obtained a dataset of 1,000 sentences for each of the aforementioned languages from NusaCrowd. The NusaCrowd project is a collaborative effort aimed at gathering and consolidating resources for the Indonesian language, including previously inaccessible resources. As part of this project, the authors have compiled 137 datasets and developed 117 standardized data loaders. The quality of these datasets has been evaluated using both manual and automatic methods, and their effectiveness has been demonstrated through various experiments [20]. Before conducting the training phase, we performed several data preprocessing steps. This included performing case folding, removing all punctuation marks, eliminating any numeric characters, and performing tokenization. Out of the total 8,000 sentences in our dataset, we randomly selected 20% to use as test data and reserved the remaining 80% for training purposes.

In our study, we compared the performance of two algorithms, Naïve Bayes and KNN along with additional experimentation using analytical algorithms. Considering our work's distribution-like nature, it might be questioned whether a classical analytical algorithm can efficiently predict languages. Limiting ourselves to a fixed n (for n-Grams) would produce an actual distribution, but this approach is not suitable for the mixture case, where occurrences of 'e', 'n', and 'en' are not independent. To circumvent technical issues, we will convert all inputs to a relative form. The analytical algorithms we used consist of least-squares, K-L divergence, and K-S test [21]–[24].

Naïve Bayes algorithm: Naïve Bayes is a probabilistic algorithm that is commonly used for classification tasks. It works by calculating the conditional probability of a class given a set of features,

assuming that the features are conditionally independent of each other. Despite its simplicity and the strong assumption of feature independence, Naïve Bayes has shown to be effective in many real-world applications, such as spam detection, sentiment analysis, and text classification. The KNN algorithm works by finding the  $k$  language models that are closest to the input text in terms of feature similarity. The algorithm then assigns the input text to the language that has the majority of these nearest neighbors.

Analytical algorithms, on the other hand, are rule-based methods that use specific linguistic patterns and structures to identify the language of the input text. These algorithms take into account the unique characteristics of each language and apply a series of predefined rules to classify the text. The least-squares method is a mathematical approach used in various fields, such as statistics, data fitting, and optimization. It aims to find the best-fitting curve or line that minimizes the sum of the squared differences between the observed data points and the corresponding points on the fitted curve or line. In other words, it tries to minimize the overall squared error between the observed data and the model's predictions. This method is often used in regression analysis, where it helps determine the parameters of a linear or non-linear model that best fits the given data. The K-L divergence is a measure of the difference between two probability distributions, in our case, the language models. This algorithm calculates the divergence between the input text's feature distribution and each language model's distribution, assigning the text to the language with the lowest divergence value. Lastly, the K-S test is a non-parametric method that compares the cumulative distribution functions of the input text's features and the language models. The algorithm assigns the input text to the language with the smallest distance between the two distributions.

## 2.2. Research strategy

In this study, we conducted several variations on the experimental strategy, specifically focusing on the use of algorithms to identify the most effective one(s) for the given task. Firstly, we used the Naïve Bayes algorithm in combination with several variations of features, including unigram features and a combination of unigrams, bigrams, and trigrams. The aim was to determine which feature set would produce the best results when used with Naïve Bayes.

In the second variation, we carried out an experiment in which we took into consideration the top  $n$  features of each language. This variation aimed to determine whether a smaller subset of features would be sufficient for producing accurate results or whether more features were required. Lastly, we used the best features from the first and third experiments in conjunction with other algorithms, such as KNN, least-squares, K-L divergence, and K-S test. The purpose of using different algorithms was to compare their effectiveness and determine which one(s) produced the most accurate results. Overall, our study involved testing different variations of algorithms and features to determine which combination would yield the most accurate results. By experimenting with different algorithms and features, we aimed to gain a better understanding of how these factors impact result accuracy and to identify the best approach for the given task.

## 3. RESULTS AND DISCUSSION

In this experiment, we have incorporated various scripts developed by Martin Kleine Kalvelage, who has generously shared his work on the Kaggle platform for public use. These scripts contribute to the overall methodology and analysis, enabling researchers to leverage and build upon the existing work to enhance the study's outcomes. The data set used in this study consists of 8,000 sentences, with 1,000 sentences for each language. The data set was carefully selected to ensure that it is representative of a wide range of languages and that the distribution of sentences across languages is balanced. In addition, the data set is large enough to provide sufficient data for training and testing machine learning models. To provide a clearer understanding of the linguistic diversity and characteristics of each language in our study, we have prepared an example sentence for every language in Table 1.

Table 1. Sentence for every language

Language	Sentence
Malay	akhermye, kelakuan dari hak atas bunge tabongan yang bede-bede kedaerah
Dayak	Aika ingki ' harus maniang reservasi?
Bugis	Cerita appabottinggenna menuru ade to rioloona tau ogi Telo' Pakedai.
Javanese	Pathokan nglatinaken (transliterasi) Serat Centhini kados ing ngandhap punika
Batak	Angkal tersebut laos ris manghalabahon sagi sikkola
Minang	dek ulah sifaik iduik jadi sansaro
Sunda	Balukarna, perlakuan ngeunaan hak luhur kembang tabungan kasebat benten-benten antardaerah
Madurese	temmahna, papajuhan nyamber ha atas bhunga tabungan sasebbhut abhidhah antardaerah.

The number of tokens in the data set is 88,380. Tokens are individual units of text, such as words or punctuation marks, and this information is important for understanding the characteristics of the data set. The length of the shortest word in the data set is 1, while the longest word is 24. The average word length in the data set is 11.05, providing additional insights into the linguistic features of the data set.

To prepare the data set for analysis, it was randomly divided into training and test sets. The training set comprises 80% of the data set, or 6,400 sentences, while the test set comprises 20% of the data set or 1,600 sentences. This split ensures that the models are trained and tested on separate data sets, allowing for an unbiased assessment of their performance. The size of the data set and the division between training and test sets were carefully chosen to ensure that the models are trained on sufficient data while still allowing for a meaningful evaluation of their performance.

The training data calculations showed that the number of unigrams for each language in the dataset is as follows: Madurese has 29, Malay has 27, Minang has 29, Sundanese has 26, Buginese has 26, Batak has 26, Dayaks has 25, and Java has 26. This information can be used to understand the distribution of unigrams across different languages and develop more accurate models for NLP tasks. These results provide valuable information about the distribution of unigrams across different languages in the dataset. By understanding the number of unigrams for each language, researchers can develop more effective models for analyzing and classifying text in these languages. Furthermore, knowing the specific unigrams present in each language can help to identify linguistic patterns and features that are unique to each language. Overall, the results of these calculations provide important insights into the linguistic properties of the dataset, which can be used to guide further analysis and modeling efforts.

Figure 1 displays the unigram variations for each language in the dataset, with specific characters as columns and languages as rows. The first empty column shows a space character, while the next column shows an alphabetic character. Each column presents the percentage of occurrences of characters in that language. Figure 1 illustrates that the appearance of characters in each language has varying degrees of occurrence. For instance, the character 'a' has the highest percentage of occurrence in the Minang and Madurese languages, while the character 'e' has the highest percentage of occurrence in Malay. Conversely, some characters, such as 'f' and 'w', have low occurrences in almost all languages in the dataset.

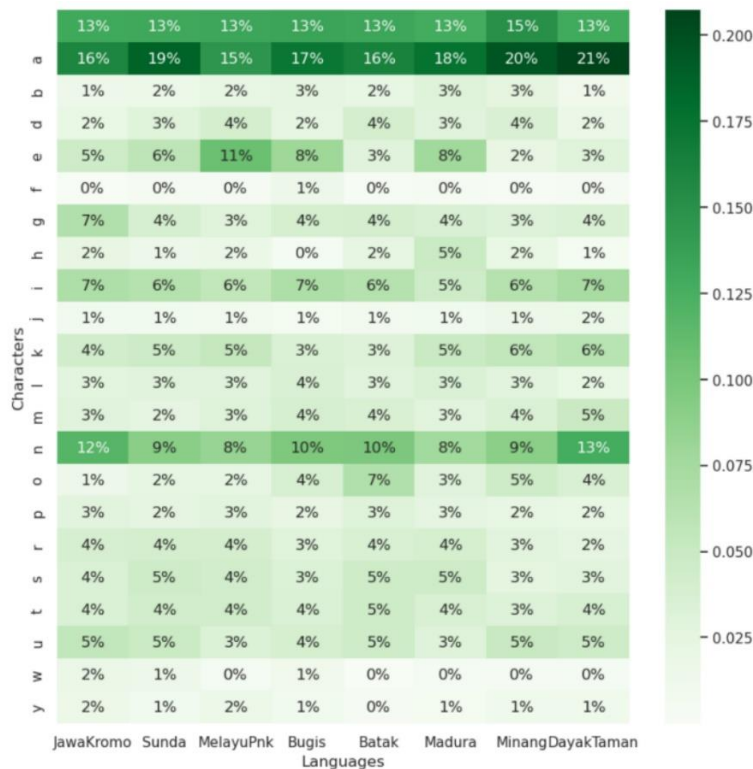


Figure 1. Unigram characteristics of each language

Furthermore, the table shows that several languages have similar variations of unigrams, such as Javanese, Kromo, and Sundanese, which have nearly identical percentages of occurrences for most of the

characters displayed. However, certain characters have a significantly different percentage of appearance in one language than another. By examining the figure, it is evident that the Malay language has several prominent unigram characteristics. The character 'e' has a relatively high percentage of occurrence, around 11%. This indicates that the character 'e' appears with greater frequency in Malay compared to the other languages listed in Figure 1.

Moreover, the unigram variation data indicates that the Dayak language has several characteristics that distinguish it from the other languages in the dataset. This characteristic can be observed from the proportion of the use of certain characters in the Dayak language text. One of the most striking characteristics is the high use of the letter "n", which equals 0.13. This proportion is higher than in other languages, except for Javanese, Kromo, and Batak. Additionally, the Dayak language also has a relatively high proportion of the use of the letters "a" and "u" compared to other languages.

Furthermore, the character 'o' also has a relatively high percentage of occurrence in Malay, around 2%. The occurrence percentage of the character 'o' in Malay is higher than that in languages such as Javanese, Sundanese, and Dayak. Finally, by considering the variations in the unigrams in the table, further analysis can be conducted to evaluate the effectiveness of the language classification algorithm used in the dataset. For example, it can be examined whether certain characters that appear significantly different in each language can make a significant contribution to distinguishing one language from another.

The number of bigrams is 539. Below are the 0.01 significant bigrams for each language:

- Malay [' b', ' d', ' k', ' m', ' p', ' s', 'ak', 'an', 'ar', 'at', 'da', 'de', 'di', 'e', 'el', 'en', 'er', 'i', 'ka', 'ke', 'la', 'n', 'ng', 'pe', 'ra', 'se', 'ta']
- Dayak [' a', ' i', ' k', ' m', ' t', ' a', 'ak', 'am', 'an', 'ar', 'at', 'g', 'i', 'in', 'ka', 'ma', 'n', 'na', 'ng', 'o', 'pa', 'sa', 'ta', 'un', 'ya']
- Bugis [' a', ' m', ' s', ' a', 'an', 'ba', 'e', 'en', 'g', 'i', 'la', 'ma', 'na', 'ng', 'nn', 'o']
- Javanese [' a', ' k', ' m', ' p', ' s', ' a', 'an', 'ar', 'en', 'g', 'ga', 'i', 'in', 'ka', 'la', 'ma', 'n', 'ng', 'pa', 'ra', 'sa', 'un']
- Batak [' d', ' m', ' p', ' s', ' t', ' a', 'an', 'ar', 'as', 'at', 'ba', 'da', 'do', 'g', 'ga', 'ho', 'i', 'la', 'ma', 'n', 'na', 'ng', 'on', 'pa', 'sa', 'si', 't']
- Minang [' a', ' b', ' d', ' k', ' m', ' p', ' s', ' t', ' a', 'ah', 'ak', 'al', 'am', 'an', 'ar', 'ba', 'da', 'di', 'g', 'h', 'i', 'in', 'k', 'ka', 'la', 'ma', 'n', 'na', 'ng', 'o', 'pa', 'ra', 'sa', 'ta', 'ua']
- Sunda [' a', ' d', ' k', ' p', ' s', ' a', 'an', 'ar', 'at', 'di', 'eu', 'i', 'ka', 'la', 'n', 'na', 'ng', 'ra', 'sa', 'ta', 'u']
- Madurese [' a', ' b', ' d', ' k', ' p', ' s', ' a', 'ab', 'ah', 'an', 'ar', 'as', 'ba', 'dh', 'e', 'el', 'en', 'gh', 'h', 'ha', 'hi', 'i', 'ka', 'ke', 'la', 'n', 'ng', 'pa', 'ra', 'sa', 'se', 'ta']

The most significant bigram with a value of 0.02 is as follows,

- Malay [' d', 'an', 'e', 'n']
- Dayak [' m', 'a', 'an', 'in', 'ka', 'ma', 'n', 'na', 'ng']
- Bugis [' m', 'a', 'an', 'e', 'en', 'i', 'na', 'ng']
- Javanese [' a', 'an', 'g', 'in', 'n', 'ng']
- Batak [' a', 'an', 'n', 'ng', 'on']
- Minang ['an', 'k', 'n', 'ng', 'o']
- Sunda [' a', 'an', 'n', 'ng', 'sa']
- Madurese [' a', 'an', 'n']

Malay has a high frequency of bigram occurrences in the characters 'an' and 'ak', indicating the common use of words ending in '-an' and '-ak'. Dayak language has the highest frequency of bigram occurrences in the characters 'an', 'ka', and 'ma', indicating the frequent use of words starting with 'k' and 'm', as well as words ending in '-an'. Bugis language has the highest frequency of bigram occurrences in the characters 'an', 'ba', and 'en', indicating the frequent use of words with the prefix 'ba-'. Javanese Kromo and Batak have similar characteristics with the highest frequency of bigram occurrences in the characters 'an', 'ar', and 'ga', indicating the frequent use of words ending in '-an' and starting with 'g'. Minang language has the highest frequency of bigram occurrences in the characters 'an', 'ak', and 'al', indicating the frequent use of words ending in '-an' and '-ak', and words ending in 'al-'. Sundanese has the highest frequency of bigram occurrences in the characters 'an', 'at', and 'eu', indicating the frequent use of words ending in '-an' and '-at', and words ending in 'eu-'. The Madurese language has the highest frequency of bigram occurrences in the characters 'an', 'ar', and 'ng', indicating the frequent use of words ending in '-an' and starting with 'ng'.

The combined number of features from unigrams and bigrams is 573. Among the languages, Madurese has the highest number of combined features at 407, while Bugis has the lowest at 340. Javanese has 343 features, Batak has 367, Sunda has 393, Malay has 389, Dayak has 336, and Minang has 365. This information highlights the variation in the number of features across the different languages, which can impact the performance of language classification algorithms that rely on these features.

### 3.2. Evaluation of the language detection algorithms

F1 (micro), F1 (macro), F1 (weighted), recall, and precision are commonly used evaluation metrics in machine learning for classification tasks. Precision is the proportion of true positive instances over the total number of instances that the algorithm has classified as positive. It measures the ability of the algorithm to correctly identify the positive instances without incorrectly classifying negative instances as positive. The recall is the proportion of true positive instances over the total number of actual positive instances in the dataset. It measures the ability of the algorithm to identify all the positive instances in the dataset, without missing any. F1 score (micro) is a variant of F1 score that aggregates the individual scores for each class by computing a single score across all classes. It gives equal weight to each instance in the dataset and is commonly used for imbalanced datasets. F1 score (macro) is a variant of F1 score that computes the average score across all classes, without taking into account the class imbalance in the dataset. It gives equal weight to each class and is useful when all classes are of equal importance. F1 score (weighted) is a variant of F1 score that computes the average score across all classes, taking into account the class imbalance in the dataset. It gives higher weight to the classes with more instances and is useful when the classes are of unequal importance. All of these metrics take values between 0 and 1, with higher values indicating better performance.

The performance evaluation metrics for the Naïve Bayes model were used with unigram features and achieved the following results: an F1 score of 0.698 for micro-average, 0.697 for macro-average, and 0.697 for weighted-average, a recall score of 0.701, and a precision score of 0.705. Figure 2 provides a visual representation of the matrix, showcasing the relationship between the actual number of sentences and the sentences predicted by the system for each language under investigation. This matrix allows for a clear comparison of the system's predictions and the true distribution of sentences within each language, thus enabling researchers to evaluate the performance of the chosen algorithm in accurately identifying and classifying languages. For instance, in the test data, out of the sentences in the Batak language, 161 were correctly predicted, while 3 were incorrectly predicted as Bugis, 9 as Dayak, 5 as Jawa, 3 as Madura, 0 as Melayu, 11 as Minang, and 11 as Sunda. However, none of the sentences were predicted as Malay.

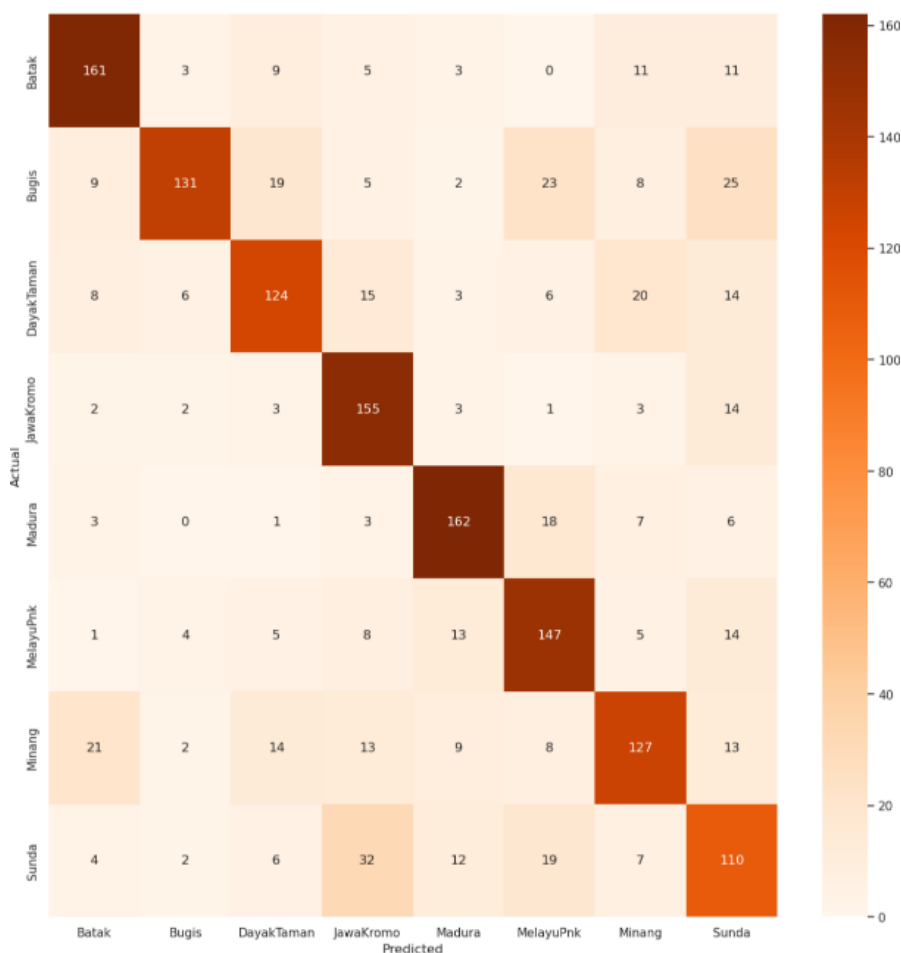


Figure 2. Evaluation metrics for the Naïve Bayes model with unigram features

These scores indicate the overall effectiveness of the Naïve Bayes model in classifying the data based on the unigram features. The micro-average F1 score suggests that the model has a relatively balanced performance in predicting all classes. The macro-average F1 score takes the average of the F1 score of each class, which indicates that the model has a slightly better performance in predicting some classes compared to others. The weighted-average F1 score takes the average of the F1 score of each class, weighted by the number of samples in each class, which indicates that the model has a better overall performance in predicting the classes with more samples. The recall score indicates the proportion of correctly classified samples for each class, while the precision score indicates the proportion of truly positive samples among all positive predictions for each class.

Figure 3 presents a visual representation of the matrix, illustrating the relationship between the actual number of sentences and the sentences predicted by the system for each language investigated using a combination of unigram and bigram features. The performance evaluation metrics for the Naïve Bayes model using combined features of unigram and bigram showed promising results. The F1 score for micro-averaging was 0.923, indicating high accuracy of the model in predicting the correct language for each text instance. The macro-averaged F1 score was 0.923, indicating that the model performs well in all classes. The weighted F1 score was 0.923, indicating that the model has a balanced performance in each language class. The recall score was 0.924, which indicates that the model can correctly identify most of the instances of each language class. The precision score was 0.923, indicating that the model can accurately predict most of the instances it identifies as a particular language class. Overall, these results show that the combination of unigram and bigram features is more effective than using only unigram features in predicting the language of a given sample text using the Naïve Bayes algorithm.

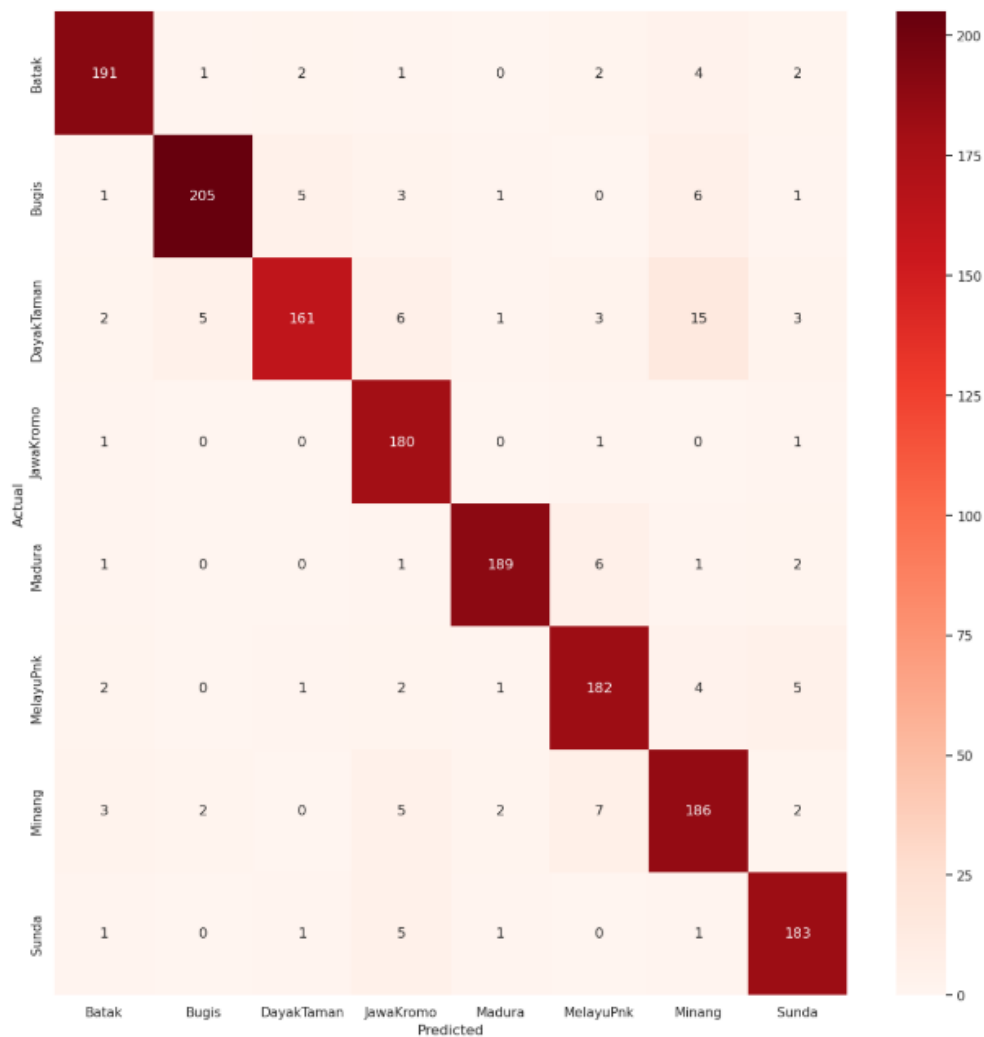


Figure 3. Evaluation metrics for the Naïve Bayes model with unigram+bigram features



We experimented with the inclusion of a trigram feature in our analysis, which expands the features to include unigrams, bigrams, and trigrams. The results indicate a significant improvement in the model's performance, with an F1 score of 0.959 (micro), 0.958 (macro), 0.958 (weighted), a recall score of 0.959, and a precision score of 0.960. In comparison to the previous results, which did not include trigrams, the F1 score has increased from 0.923 to 0.959. This suggests that the inclusion of trigram features has enhanced the model's ability to accurately classify and predict the target variable. Figure 4 presents a graphical illustration of the confusion matrix, which displays the correlation between the true number of sentences in each language and the sentences predicted by the system using a combination of unigrams, bigrams, and trigrams as features. This visualization allows researchers to easily assess the accuracy and performance of the language identification system by comparing the actual and predicted sentences for each language being studied. It also highlights any potential misclassifications or areas where the system may struggle to correctly identify a specific language, thereby providing valuable insights for further refinement of the algorithms and features used.

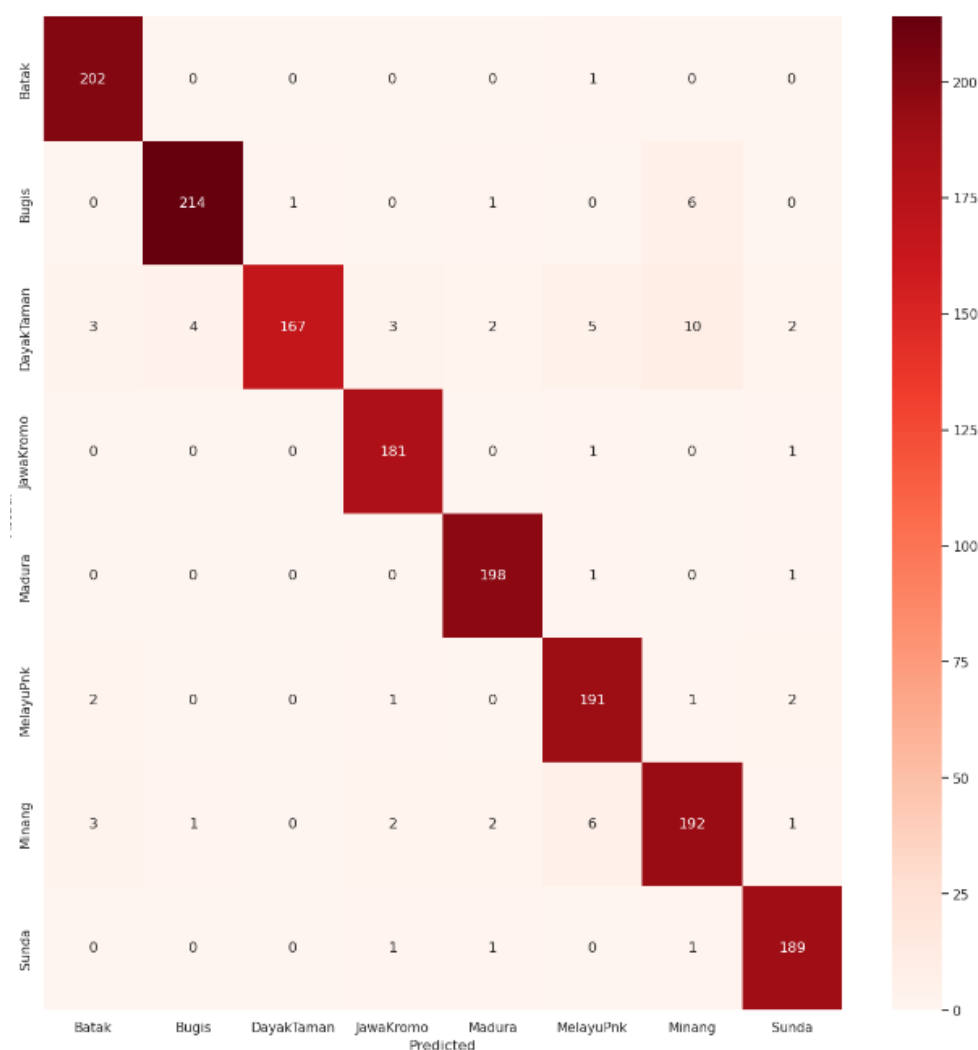


Figure 4. Evaluation metrics for the Naïve Bayes model with unigram+bigram+trigram features

In addition to the earlier description, we aimed to assess the performance of our language detection model under various experimental settings. To achieve this, we used a combination of unigrams and bigrams as features, as mentioned before, which resulted in a total of 573 features. To evaluate the model's efficacy, we conducted experiments using different sets of features by selecting the top n features for each language. Specifically, we explored the effects of n=50, 60, 70, 80, and 100 features on the accuracy of the model. The results of these experiments were analyzed using the naïve Bayes algorithm and are presented in Table 2. Through this evaluation, we aimed to identify the most effective set of features that can significantly enhance the accuracy of the language detection model.

Table 2. F1, recall, and precision with top N features

Top N	F1 Score (Micro)	F1 Score (Macro)	F1 Score (Weighted)	Recall	Precision
50	0.861	0.860	0.860	0.861	0.861
60	0.889	0.888	0.888	0.889	0.889
70	0.898	0.897	0.897	0.898	0.897
80	0.901	0.900	0.900	0.902	0.901
90	0.904	0.903	0.904	0.905	0.904
100	0.906	0.905	0.906	0.906	0.906

Table 2 shows the F1 score, recall, and precision for different top N values when using a micro-average method to evaluate the Naïve Bayes model. As the value of N increases, the F1 score improves, indicating that the model is better at correctly predicting the language of a given text instance. The highest F1 score is achieved when the top N is 100, with a micro-average F1 score of 0.906, a macro-average F1 score of 0.905, and a weighted-average F1 score of 0.906. The recall and precision scores also increase as the value of N increases, indicating that the model becomes more accurate as it has access to more features.

### 3.3. Comparative analysis of algorithms

In this research, we aim to compare the performance of various algorithms in identifying languages based on text input. The algorithms chosen for comparison are Naïve Bayes, KNN, least-squares, K-L divergence, and K-S test. Each of these algorithms is applied to a combined feature set consisting of unigram, bigram, and trigram frequencies derived from the text data. To create the feature set, we first extract the top 100 most frequent unigrams, bigrams, and trigrams for each language in our dataset.

We evaluate the performance of these algorithms using standard metrics such as accuracy, precision, recall, and F1 score. The results of our experiments are presented in Table 3. This table provides a comprehensive comparison of the algorithms' performance, highlighting their strengths and weaknesses in language identification. Based on the results, we can determine the most suitable algorithm for this task and provide insights into areas where further research is required to improve language identification methods.

Based on the data provided in Table 3, we can observe the performance of the five algorithms used in the task of identifying local languages in Indonesia. The compared algorithms include Naïve Bayes, KNN, least-squares, K-L divergence, and K-S test. The performance of each algorithm is measured using F1 score metrics (Micro, Macro, and Weighted), recall, and precision. Based on the analysis of experimental results, Naïve Bayes showed the best performance with the highest F1 score (0.918) and almost identical recall (0.919) and precision (0.918) values. On the other hand, kNN demonstrated lower performance compared to Naïve Bayes, but still quite good, with an F1 score of 0.852 and a slightly higher precision (0.857) than recall (0.852), indicating that kNN is more selective in classifying languages. Meanwhile, the least-squares showed a decrease in performance compared to Naïve Bayes and KNN, but the precision (0.791) and recall (0.788) values remained quite balanced with an F1 score of 0.789. K-L divergence had better performance than least-squares, with an F1 score of 0.830, recall of 0.830, and precision of 0.849, but still lower than Naïve Bayes and KNN. Finally, K-S test exhibited the lowest performance among the five algorithms, with F1 scores (0.500), recall (0.445), and precision (0.451) which were much lower than the other algorithms.

Table 3. F1, recall, and precision of algorithm

Algorithm	F1 Score (Micro)	F1 Score (Macro)	F1 Score (Weighted)	Recall	Precision
Naïve Bayes	0.918	0.918	0.918	0.919	0.918
kNN	0.852	0.850	0.851	0.852	0.857
Least-Squares	0.789	0.789	0.790	0.788	0.791
K-L Divergence	0.828	0.830	0.830	0.830	0.849
K-S Test	0.500	0.445	0.500	0.445	0.451

The experimental results provide valuable insights into the performance of different algorithms for language identification tasks using combined unigram, bigram, and trigram features. Among the five algorithms tested, Naïve Bayes displayed the best performance. This outcome can be attributed to the algorithm's ability to handle the high-dimensional feature space and the independence assumption, which simplifies the model's complexity. On the other hand, KNN, despite showing lower performance than Naïve Bayes, still performed relatively well in the language identification task. The algorithm's selectiveness in classifying languages is evident from its slightly higher precision compared to recall. This characteristic of KNN can be useful when high precision is desired in applications such as text classification or sentiment analysis.

The least-squares showed a noticeable decrease in performance when compared to Naïve Bayes and KNN. While precision and recall values remained quite balanced, the overall F1 score was lower, suggesting that this algorithm might not be as well-suited for language identification tasks using the selected features. K-L divergence outperformed the least-squares, but its performance still lagged behind Naïve Bayes and KNN. The higher precision value for K-L divergence suggests that it might be effective in situations where false positives need to be minimized, such as spam detection or content filtering. K-S test, however, exhibited the weakest performance among the five algorithms. With significantly lower F1 scores, recall, and precision, it may not be the best choice for language identification tasks using unigram, bigram, and trigram features.

In summary, Naïve Bayes and KNN emerged as the top-performing algorithms for language identification in this study. Despite their strong performance in this study, Naïve Bayes and KNN algorithms may not always outperform other algorithms in every language identification task. For instance, in detecting gender in Arabic, both algorithms have demonstrated suboptimal performance [25]. This highlights the importance of considering the specific language and dataset involved when selecting an appropriate algorithm for language identification tasks. Future research can focus on refining these algorithms or exploring other machine-learning techniques to improve the performance of language identification tasks. Additionally, experimenting with other feature sets or incorporating advanced NLP techniques may lead to even better results.

#### 4. CONCLUSION

In conclusion, our research provides a comprehensive comparison of five different algorithms for language identification based on text input: Naïve Bayes, KNN, least-squares, K-L divergence, and K-S test. By evaluating their performance using accuracy, precision, recall, and F1 score metrics, we found that Naïve Bayes outperformed the other algorithms, demonstrating the highest F1 score, recall, and precision. The KNN algorithm also showed promising results, with its performance being slightly lower than Naïve Bayes. The least-squares exhibited a moderate performance level, while K-L divergence and K-S test had lower performance levels compared to the other three methods. These findings indicate that Naïve Bayes is the most suitable algorithm for language identification tasks in the context of our research. However, it is essential to acknowledge that the performance of these algorithms may vary depending on the specific languages and text data involved. Further research and experimentation are necessary to enhance the performance of language identification methods, taking into account different language characteristics, feature sets, and algorithmic approaches. By continuously refining these techniques, we can improve the accuracy and efficiency of language identification systems, which will benefit various applications in linguistics, NLP, and beyond.




#### REFERENCES

- [1] P. K. M. Soesilo and F. Rahman, "The pillars of survival in the COVID-19 Pandemic: The case of Indonesia," in *Community, Economy and COVID-19*, Cham: Springer, 2022, pp. 267–289, doi: 10.1007/978-3-030-98152-5\_13.
- [2] Z. Sakhiyya and N. Martin-Anatias, "Reviving the language at risk: a social semiotic analysis of the linguistic landscape of three cities in Indonesia," *International Journal of Multilingualism*, vol. 20, no. 2, pp. 290–307, 2023, doi: 10.1080/14790718.2020.1850737.
- [3] Y. Sewell, "Linguistic pragmatism, lingua francae, and language death in Indonesia," *Journal of Language Teaching*, vol. 2, no. 11, pp. 15–19, 2022, doi: 10.54475/jlt.2022.015.
- [4] M. Padró and L. Padró, "Comparing methods for language identification," *Procesamiento del lenguaje natural*, vol. 33, pp. 1-7, 2004.
- [5] H. S. Das and P. Roy, "A deep dive into deep learning techniques for solving spoken language identification problems," *Intelligent Speech Signal Processing*, pp. 81–100, 2019, doi: 10.1016/B978-0-12-818130-0.00005-2.
- [6] A. Karakanta, J. Dehdari, and J. van Genabith, "Neural machine translation for low-resource languages without parallel corpora," *Machine Translation*, vol. 32, no. 1–2, pp. 167–189, 2018, doi: 10.1007/s10590-017-9203-5.
- [7] M. R. A. Utomo and Y. Sibaroni, "Text classification of british english and American english using support vector machine," *2019 7th International Conference on Information and Communication Technology, ICoICT 2019*, 2019, pp. 1-6, doi: 10.1109/ICoICT.2019.8835256.
- [8] A. Babhulgaonkar and S. Sonavane, "Language identification for multilingual machine translation," *Proceedings of the 2020 IEEE International Conference on Communication and Signal Processing, ICCSP 2020*, pp. 401–405, 2020, doi: 10.1109/ICCSP48568.2020.9182184.
- [9] O. Dovbnia, W. Sosnowski, and A. Wróblewska, "Automatic language identification for Celtic Texts," *Communications in Computer and Information Science*, vol. 1793 CCIS, pp. 264–275, 2023, doi: 10.1007/978-981-99-1645-0\_22.
- [10] M. S. Saputri and M. Adriani, "Identifying Indonesian local languages on spontaneous speech data," *2019 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2019*, pp. 247–254, 2019, doi: 10.1109/ICACSIS47736.2019.8979939.
- [11] P. Martadinata, B. D. Trisedya, H. M. Manurung, and M. Adriani, "Building Indonesian local language detection tools using Wikipedia data," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9442, pp. 113–123, 2016, doi: 10.1007/978-3-319-31468-6\_8.
- [12] N. Saquib and A. Rahman, "Application of machine learning techniques for real-time sign language detection using wearable sensors," *MMSys 2020 - Proceedings of the 2020 Multimedia Systems Conference*, pp. 178–189, 2020, doi: 10.1145/3339825.3391869.




- [13] D. Shetty, H. Sarojadevi, U. Shakeel, S. Sanjana, G. M. Aishwarya, and P. Nupur, "An approach to identify indic languages using text classification and natural language processing," *MysuruCon 2022 - 2022 IEEE 2nd Mysore Sub Section International Conference*, 2022, pp. 1-6, doi: 10.1109/MysuruCon55714.2022.9972371.
- [14] B. Oussama, "Automatic Algerian offensive language detection in social media networks," Master Accademic dissertation, Department of Electronics and Telecommunications, University of Guelma, Guelma, Aljazair, 2021.
- [15] A. Joshi, J. T. Halseth, and P. Kanerva, "Language geometry using random indexing," in *Quantum Interaction*, Cham: Springer, 2017, pp. 265–274, doi: 10.1007/978-3-319-52289-0\_21.
- [16] K. Hornik, P. Mair, J. Rauch, W. Geiger, C. Buchta, and I. Feinerer, "The textcat package for n-gram based text categorization in R," *Journal of Statistical Software*, vol. 52, no. 6, pp. 1–17, 2013, doi: 10.18637/jss.v052.i06.
- [17] M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. AbdelMajeed, and T. Zia, "Abusive language detection from social media comments using conventional machine learning and deep learning approaches," *Multimedia Systems*, vol. 28, no. 6, pp. 1925–1940, 2022, doi: 10.1007/s00530-021-00784-8.
- [18] H. Venkatesan, T. Varun Venkatasubramanian, and J. Sangeetha, "Automatic language identification using machine learning techniques," *Proceedings of the 3rd International Conference on Communication and Electronics Systems, ICCES 2018*, pp. 583–588, 2018, doi: 10.1109/CESYS.2018.8724070.
- [19] K. Shaalan, "Rule-based approach in Arabic natural language processing," *International Journal on Information and Communication Technologies*, vol. 3, no. 3, pp. 11-19, 2010.
- [20] S. Cahyawijaya et al., "NusaCrowd: Open source initiative for Indonesian NLP resources," *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 13745–13819, 2023, doi: 10.18653/v1/2023.findings-acl.868.
- [21] Z. M. Yasin, N. A. Salim, N. F. A. Aziz, Y. M. Ali, and H. Mohamad, "Long-term load forecasting using grey wolf optimizer-least-squares support vector machine," *IAES International Journal of Artificial Intelligence*, vol. 9, no. 3, pp. 417–423, 2020, doi: 10.11591/ijai.v9.i3.pp417-423.
- [22] L. Yang, S. McClean, M. Donnelly, K. Burke, and K. Khan, "Process duration modelling and concept drift detection for business process mining," *Proceedings - 2021 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Internet of People, and Smart City Innovations, SmartWorld/ScalCom/UIC/ATC/IoP/SCI 2021*, pp. 653–658, 2021, doi: 10.1109/SWC50871.2021.00097.
- [23] K. Tekbyk, A. R. Ekti, G. K. Kurt, A. Gorcin, and S. Yarkan, "Modeling and analysis of short distance Sub-Terahertz communication channel via mixture of Gamma distribution," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 2945–2954, 2021, doi: 10.1109/TVT.2021.3063209.
- [24] W. Li, "Supporting database constraints in synthetic data generation based on generative adversarial networks," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 2875–2877, 2020, doi: 10.1145/3318464.3384414.
- [25] E. AlSukhni and Q. Alequr, "Investigating the use of machine learning algorithms in detecting gender of the Arabic tweet author," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 7, 2016, doi: 10.14569/ijacsa.2016.070746.

## BIOGRAPHIES OF AUTHORS



**Herry Sujaini**    graduated from a bachelor's degree in the Electrical Engineering Department, University of Tanjungpura. He got his master and a doctoral degree from STEI, Bandung Institute of Technology. Since 1997, he has become a lecturer at Informatics Department, Engineering Faculty, University of Tanjungpura. His research interest is on machine translation and machine learning. He can be contacted at email: hs@untan.ac.id.



**Arif Bijaksana Putra**    earned a doctorate in Electrical Engineering, followed by a master's and a doctoral degree from STEI, Bandung Institute of Technology. He has been working as a lecturer at the University of Tanjungpura's Informatics Department in the Engineering Faculty since 1999. His research focuses on text-to-speech and data science. He can be contacted at email: arifbpn@untan.ac.id.