IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 30, No. 10, October, 2025, pp. 2609-2624
Journal homepage: http://iieta.org/journals/isi

Enhancing Sentiment Analysis Accuracy Through Intelligent Spelling Correction Using Damerau-Levenshtein Distance and N-Gram with Random Forest Classifier

Doni Abdul Fatah[1*], Yudha Dwi Putra Negara[1], Nasrulloh Nasrulloh[2], Nissa Aulia Belistiana Utami[3]

[1] Information Systems Program, Faculty of Informatics Engineering, Universitas Trunojoyo Madura, Bangkalan 69162, Indonesia
[2] Sharia Economic Program, Faculty of Islamic Studies, Universitas Trunojoyo Madura, Bangkalan 69162, Indonesia
[3] International Education Development, Hiroshima University, Higashi-Hiroshima 739-8511, Japan

Corresponding Author Email: doni.fatah@trunojoyo.ac.id

https://doi.org/10.18280/isi.301008

Received: 31 July 2025
Revised: 5 October 2025
Accepted: 15 October 2025
Available online: 31 October 2025

Keywords:
sentiment analysis, beach tourism reviews, Damerau-Levenshtein Distance, spelling correction, N-Gram, feature selection, Indonesian tourism,

ABSTRACT

The digital era has led to a significant increase in tourism reviews across various social media platforms and travel applications. However, many of these reviews contain spelling errors due to typos or deviations from standard language rules. This study developed an intelligent spelling correction model to improve the quality of sentiment analysis on Madura beach tourism reviews. The proposed model integrates the Damerau-Levenshtein Distance (DLD) method for spelling correction, Bigram-based N-Gram tokenisation, and the Random Forest (RF) Classifier for sentiment classification. The dataset consists of 1,634 comments collected through data scraping, with 966 labelled as positive and 668 as negative using majority voting and expert validation. The model development was based on the CRISP-DM framework, and Information Gain was used to evaluate features because it helps prevent overfitting. According to the experiments' results, the combined model DLD, N-Gram, and Random Forest achieved the highest an accuracy of 90.21%. In contrast, the initial model, Random Forest, on TF-IDF features from the baseline achieved 89.69% without spelling correction or N-Gram features. Experimental results indicate shows that Random Oversampling achieves better class balance than SMOTE and Random Undersampling. Therefore, integrating spelling correction with feature extraction and selection significantly enhanced sentiment analysis performance in the research on Madura beach reviews.

1. INTRODUCTION

Tourism is an economic sector that plays an essential role in increasing regional and national income through activities such as travel, recreation, and the exploration of culture and nature [1-3]. In Indonesia, including in Madura, tourism has become a leading sector that supports the local economy, creates job opportunities, and promotes cultural preservation and infrastructure development [4]. Various efforts have been made to increase tourism appeal, such as improving facilities, digital promotion, and the development of the creative tourism economy, to improve community welfare [5]. Locally Generated Revenue (PAD) from the tourism sector in Madura, particularly in Sumenep, has experienced growth. As of mid-2024, PAD from the tourism sector in Sumenep reached 62% of the annual target, approximately Rp. 525 million out of the Rp. 847 million target. One of the primary sources of PAD comes from beach tourism, which has become a flagship destination with an increase in the number of tourist visits through various events and infrastructure improvements [6, 7].

One of the main factors in tourism promotion is social media and travel platforms, which enable tourists to share their experiences through reviews, photos, and videos [8]. Social media also facilitates the dissemination of information and accelerates the formation of public opinion, creating a discussion space that is often very dynamic [9]. As a result, many opinions, both positive and negative, arise in response to government policies [10]. However, spelling errors (typos) in sentiments expressed on social media, such as TikTok, Instagram, YouTube, and travel platforms, often and understandably occur. These writing errors usually happen due to negligence, and naturally, this affects classification accuracy. To mitigate the impact of such spelling mistakes, additional methods are needed to perform spelling correction for the mistyped words [11].

However, previous sentiment analysis studies have primarily focused on structured, clean datasets, such as product or movie reviews, where spelling inconsistencies are minimal. In the tourism domain, particularly in user-generated reviews, Indonesian spelling variations, informal language, and typographical errors are common, yet their direct impact on sentiment classification accuracy remains underexplored. This limitation highlights a research gap in understanding how spelling correction techniques can improve sentiment analysis performance in tourism reviews, particularly in morphologically rich languages such as Bahasa Indonesia.

Reviews play an important role in shaping public perception of tourist destinations, both positive and negative [12, 13]. However, challenges arise when reviews contain spelling or typing errors, which can reduce the accuracy of text data analysis [14]. These mistakes hinder natural language processing (NLP) and affect the results of sentiment analysis, which help understand tourist satisfaction toward a destination [15-17].

In the context of Madura beach tourism, tourist reviews are critical for the government and destination managers to improve service quality and marketing strategies [18]. Therefore, an automatic spelling correction method is needed to improve the quality of reviews before further analysis. Unlike previous works that applied DLD and N-Gram for generic spell checking or linguistic correction, this study integrates them explicitly with a Random Forest classifier to examine their combined effect on sentiment classification accuracy in Indonesian tourism reviews. This approach contributes novelty by evaluating how intelligent spelling correction can enhance model robustness and provide more reliable insights for tourism management and policy decisions. With the approach of Damerau-Levenshtein Distance (DLD), N-Gram, and Random Forest Classifier (RF) [19]. The system is expected to help produce more accurate data for decision-making in the development of the tourism sector [20, 21].

Damerau-Levenshtein Distance is an effective method for correcting spelling errors by measuring the minimum number of operations, such as insertion, deletion, substitution, and transposition of characters, required to transform one word into another [22-24]. This method has been proven to improve spelling correction accuracy by up to 9% in texts with invalid spelling. Accurate spelling correction is crucial in text mining, particularly in NLP, which aims to extract information from unstructured text data [25]. In the context of sentiment analysis, NLP is used to identify opinions from text and convert them into quantitative data for decision-making purposes [26].

Another study on the use of the Damerau-Levenshtein algorithm and N-Grams for an Amazigh language spell checker shows that a spell-checking system combining the Damerau-Levenshtein algorithm and the N-Gram model was effective at detecting and correcting spelling errors. This system succeeded in placing the correct word as the top suggestion in more than 60% of cases and achieved high detection accuracy, with an F1-score of 98.74% for proper words and 86.62% for incorrect words. Compared to five other approaches (Norvig, BK-Tree, LinSpell, SymSpell, and N-Gram), this system demonstrated better correction performance, though it was slightly slower than N-Gram in processing time. This combined approach is considered effective for handling common typos and is suitable for languages with high morphological complexity, such as Amazigh [18].

A similar study on Real-Word Spelling Error Detection and Correction in Urdu emphasised the effectiveness of the Damerau-Levenshtein algorithm for correcting real-world spelling errors in Urdu. The developed system generated correction candidates using the Damerau-Levenshtein Distance, then ranked them using an N-Gram model. The combination of trigram and Damerau-Levenshtein and additional ranking strategy was shown to produce the highest accuracy at 83.67%, making it effective for context-based spelling correction in low-resource languages [27].

The RF algorithm is often used in sentiment analysis because of its high performance compared to other classification methods [28]. However, this algorithm has weaknesses in data stability, so feature extraction and selection are needed to improve its accuracy [29].

Another related study explains that the Random Forest algorithm was chosen because it can generate accurate and stable predictions by combining many decision trees, thus reducing the risk of overfitting common in single decision trees. This algorithm is highly effective for handling complex, non-linear, and high-dimensional data, and still performs well even when there is missing data [30]. Additionally, Random Forest can provide feature importance scores, which are helpful for data analysis. With these advantages, Random Forest is an appropriate choice for tasks such as text classification, spelling correction, and context-based error detection, and RF is also valuable in strategic decision-making, medical diagnosis, financial prediction, as well as processing large and imperfect data [31].

Another study examined the use of the Random Forest algorithm to predict students' course grades and analyse the importance of predictor variables. The model achieved an accuracy of 90.33% with an RMSE of 9.25. The results showed that GPA and high school grades were the most influential factors, followed by attendance and course category, while teaching method, type of school, and gender were less influential. Random Forest was chosen because it is accurate and capable of revealing each variable's contribution to academic performance [32].

The N-Gram and TF-IDF methods are used to understand word patterns in opinions. N-Gram splits the text into unigrams, bigrams, and trigrams to capture broader context [28]. For example, the phrase "suka hutang" ("likes debt"): using unigrams, the word "suka" may be classified as positive sentiment, and "hutang" as negative. However, with bigram, the phrase is analysed as a single negative meaning, which is more accurate [33].

Supporting research on the use of phishing detection systems on websites based on URL and Term Frequency-Inverse Document Frequency (TF-IDF) values found that the Phisher Fighter system, which combines URL analysis and TF-IDF-based content, effectively detected phishing sites with high accuracy (accurate positive 90.68%) and low false negatives (9.31%). This combined approach proved more precise than previous methods and has the potential to be improved through dataset expansion and deep learning implementation [34].

Another article proposed an enhanced hybrid feature selection technique to improve sentiment classification accuracy by combining TF-IDF and SVM-RFE methods. This technique was tested on two customer review datasets (Sentiment Labelled and IMDB) and achieved superior results, with accuracy ranging from 84.54% to 89.56%. Moreover, this method reduced the number of features by up to 70.5%, making it efficient in computational resource usage without degrading classification performance [35].

Feature selection using Information Gain helps improve model performance by extracting more relevant keywords, thus enhancing the accuracy of sentiment analysis [36].

A study on the usefulness of Effective N-Gram Coverage proposed a new method to increase fuzzing effectiveness by utilising N-Gram coverage as the primary metric. N-Gram records the sequence of branches over n steps, thus capturing the execution path context more deeply. This allows the system to distinguish between logic variations even when the same branches are traversed, and to predict new paths through

nearest-neighbour branch estimation. Experimental results showed that this approach increased average code coverage by 12.3% and discovered more bugs than conventional methods. These findings demonstrate that integrating N-Grams into the fuzzing process effectively expands program exploration and improves vulnerability detection [37, 38].

Recent research on developing machine learning models to predict chemical hazard classifications based on regulatory standards has utilised N-Grams and Natural Language Processing (NLP) techniques to convert chemical structures (SMILES) into numerical features. This approach involves splitting strings into smaller parts, enabling the recognition of local patterns related to compound toxicity. The method is flexible, computationally efficient, and capable of handling complex chemical symbols, thereby improving the accuracy of hazard classification predictions [39].

Another study proposed a new method based on TF-IDF and N-Gram to analyse DNA sequence similarity without alignment (alignment-free). By representing DNA sequences as words (N-Grams), TF-IDF was used to identify the most informative segments. This approach improved accuracy while reducing computational load and demonstrated superior performance across three datasets. The TF-IDF method proved to be accurate, computationally efficient, and effective in reconstructing phylogenetic relationships, making it suitable for large-scale genomic datasets [40].

With the integration of Damerau–Levenshtein Distance (DLD), N-Gram, and the Random Forest Classifier (RF), the proposed system is expected to generate more accurate insights for tourism development and decision-making. Unlike previous studies that applied these methods independently, this study integrates them into a unified framework to improve the robustness and contextual accuracy of sentiment analysis in Indonesian tourism reviews.

2. LITERATURE REVIEW

Sentiment analysis is the process of analysing digital text to identify and classify opinions or emotions as positive, negative, or neutral. This technique typically uses Natural Language Processing (NLP) methods and text analysis to extract subjective information and emotional states from various sources, such as customer reviews, social media, and surveys [41].

Recent studies have further emphasised the importance of text normalisation and preprocessing in improving sentiment analysis accuracy, particularly for languages with high morphological complexity, such as Indonesian. A comparative study by ITS Surabaya demonstrated that applying advanced normalisation methods, including Damerau-Levenshtein Distance, significantly enhances sentiment classification performance on Indonesian text datasets [42].

Sentiment analysis, also known as opinion mining, is an automated process used to understand, extract, and process textual data [43]. Several studies have explored sentiment analysis across multiple domains; however, few have examined the specific influence of spelling errors within tourism reviews [44]. The presence of misspellings, informal expressions, and regional variations in Indonesian texts poses unique challenges for sentiment classification [45].

In tourism-related contexts, text mining and sentiment analysis have been widely applied to understand travellers' perceptions from online reviews. For instance, TripAdvisor review analysis revealed that unstructured and noisy texts, often containing spelling mistakes, emojis, and informal expressions, pose challenges for accurate opinion classification [46]. Similarly, a tourism review sentiment classification study reported that typos and emoticons significantly affect model accuracy, emphasising the need for robust preprocessing and spelling correction mechanisms [47]. Therefore, exploring effective preprocessing and spelling correction methods is essential to ensure accurate sentiment identification in unstructured data contexts and to obtain reliable insights from user-generated content [45].

The goal of sentiment analysis is to assess a person's viewpoint or opinion bias on a particular issue, whether it is positive or negative. One example of the real-life implementation is identifying the direction and public opinion toward a product or service [48].

Based on the reviewed literature, most prior research has treated spelling correction, N-Gram modelling, and Random Forest classification as independent techniques rather than as an integrated framework. Moreover, limited studies have empirically tested their combined effect on noisy, user-generated tourism review data written in Indonesian. Therefore, this study fills that gap by proposing an intelligent spelling correction model using the Damerau-Levenshtein Distance (DLD) and N-Gram, integrated with a Random Forest Classifier, for sentiment analysis of Madura beach tourism reviews.

2.1 Text preprocessing

The preprocessing stage is a phase in which data are normalised and adjusted to meet specific value constraints. This stage is performed to remove attributes that have little influence on the classification process. It is considered an essential step in the classification process to improve the accuracy of a model [49].

Preprocessing is conducted with the expectation of improving the accuracy and performance of the resulting Random Forest model. In data mining, preprocessing involves a variety of steps that are tailored to the data being used [50].

2.1.1 Text data cleaning procedures

Data cleaning is the initial step in text preprocessing, aiming to remove noise from the data. This process involves several key steps:

a) Remove punctuation, which removes punctuation marks. In this step, only alphabetic characters are accepted, while non-alphabetic characters are removed.

b) Case folding, which converts all text to lowercase.

c) Drop duplicates, which aims to remove duplicate tweets and eliminate spam tweets.

d) Spelling correction, which refers to correcting the spelling of words [51].

2.1.2 Tokenizing

Tokenising in Indonesian is relatively complex. Various types of affixes include prefixes, suffixes, infixes, and confixes. Indonesian words also originate from word repetition, affix combinations, and affix combinations with repeated words. In addition, a characteristic of the Indonesian language is compound words that are written together when bound at the beginning and end [52].

The tokenising process is the step of separating a sentence string into the words that form it. In this process, the character sequence will be split into word units.

2.1.3 Normalisation (Slang word)

Slang words are informal words or phrases used in everyday language by certain groups, typically to express ideas, emotions, or popular culture in a more relaxed and informal way. Slang often changes with social, cultural, or technological trends and is usually not found in official or formal writing [51]. This process converts all informal words into standard words based on the KBBI (Indonesian Dictionary).

2.1.4 Filtering

Filtering is a step in the process of removing unnecessary words to reduce data noise. Pronouns, conjunctions, prepositions, slang, and other frequently appearing words are examples of stopwords. Examples of stopwords in Indonesian include “dan” (and), “atau” (or), “ini” (this), and so on [51].

2.1.5 Stemming

Stemming is the process of converting words into their root form by removing affixes such as “in,” “ke,” and others. The purpose of stemming is to simplify words so they can be treated as the same root form even if they have different affixes. Stemming is often used in text processing to reduce word variation in the exact text, making analysis and information retrieval easier [51].

2.2 String metric calculation using Damerau-Levenshtein Distance

Damerau-Levenshtein Distance is an extension of the Levenshtein Distance algorithm. This algorithm calculates the minimum number of operations needed to convert one string into another. Similar to Levenshtein Distance, the operations used include insertion, deletion, and substitution. However, Damerau-Levenshtein Distance adds a fourth operation, transposition (the swapping of two adjacent characters).

In comparison, Levenshtein Distance only uses the first three operations, while Damerau-Levenshtein Distance allows character transpositions, offering greater flexibility in calculating string distance [53, 54]. The greater the number of differences between strings, the greater the distance. The inclusion of transposition can increase correction accuracy, as it addresses one of the most common typing errors—character swaps.

Using more operations in the Damerau-Levenshtein Distance results in longer computation times than with other algorithms. The heaviest computation lies in the transposition process, where the system swaps all characters regardless of whether they are adjacent, and compares them with a dictionary to find the word distance.

The distance between two strings a and b can be determined using the function. $D_{w_1.w_2(i,j)}$, where *i* and *j* represent the row indices of string w_1 and w_2, respectively. The Damerau-Levenshtein Distance formula is explained in Eq. (1), as follows:

$$dl_{a.b(i,j)} = min \begin{cases} 0 & if\ i = j = 0 \\ i & if j = 0 \\ j & if\ i = 0 \\ dl_{i-1,j} + 1\ del & if\ i > 0 \\ dl_{i,j-1} + 1\ ins & if\ j > 0 \\ dl_{i-1,j-1} + 1(a_i \neq b_j)\ subs & if\ i,j > 0 \\ dl_{i-2,j-2} + 1(a_i \neq b_j)\ trns & if\ i,j > 1\ and \\ & a_i = b_{j-1}\ and \\ & a_{i-1} = b_j \end{cases} \quad (1)$$

The Damerau-Levenshtein matrix dl is used to calculate the distance between two strings, where a Is the input string and b Is the target string. The indices i and j Represent the row positions of the input and target strings, respectively. This calculation yields the number of operations required to transform the input string into the target string.

2.3 N-Gram

N-Gram is a model used in Natural Language Processing (NLP) to predict the sequence of words in a sentence or text. This model assumes that the sequence of words in a text can be broken down into smaller units called “N-Grams.” The “N” in N-Gram refers to the number of words or tokens in the unit, where:

a) If N = 1, it is called a *unigram*.

b) If N = 2, it is called a *bigram*.

c) If N = 3, it is called a *trigram*.

N-Gram is a probabilistic approach to language modelling that predicts the next word or token based on previous words. The larger the value of N, the earlier words are considered in predicting the next word. However, as N increases, the amount of data required to train the model also increases exponentially, making models with larger N values more complex to use.

$$W^{i-1} \quad (2)$$

$$W^{i-1}C_j^i \quad (3)$$

$$W^{i-1}C_j^iW^{i+1} \quad (4)$$

C_j^i represents the current word. Eq. (2), W^{i-1} is the token at position $n - 1$ (unigram). Eq. (3), a bigram is obtained from the combination of $W^{i-1}$with C_j^i (*left bigram*), and a trigram is formed by combining the bigram W^{i-1} C_j^i with W^{i+1} Eq. (4) [26].

2.4 Parameter tuning

To achieve optimal results, one practical step is to perform hyperparameter tuning. Hyperparameter tuning is the best approach for determining parameter settings by evaluating the performance of each model across various possible combinations. The Random Forest algorithm has many parameters that can be adjusted. The parameters used in the hyperparameter tuning process for the Random Forest method include.

To achieve optimal results, one practical step is to perform hyperparameter tuning, which involves trying different parameter combinations to evaluate model performance. The Random Forest algorithm provides several parameters that can be adjusted during this tuning process. These include *n_estimator*, which refers to the number of trees in the model; *max_depth*, the maximum depth of each tree; *Criterion*, which determines the quality of a split; *min_samples_leaf*, representing the minimum number of samples required at a leaf node; and *max_features*, which defines how many features to consider when looking for the best split.

2.5 Ensemble classification via Random Forest

Random Forest is a supervised learning classification algorithm developed by Breiman in 2001 [32]. It is one of the algorithms that utilises ensemble techniques by applying bagging and random feature selection [50]. Ensemble learning is used to improve the performance of unstable classification problems by combining several base learners to reduce prediction errors. Random Forest builds models using a collection of multiple decision trees [31], as illustrated in Figure 1, where each tree provides a classification estimate (referred to as a vote). The final prediction is determined by aggregating the votes from all trees and selecting the most frequent classification, thereby producing an optimal and stable prediction [53].

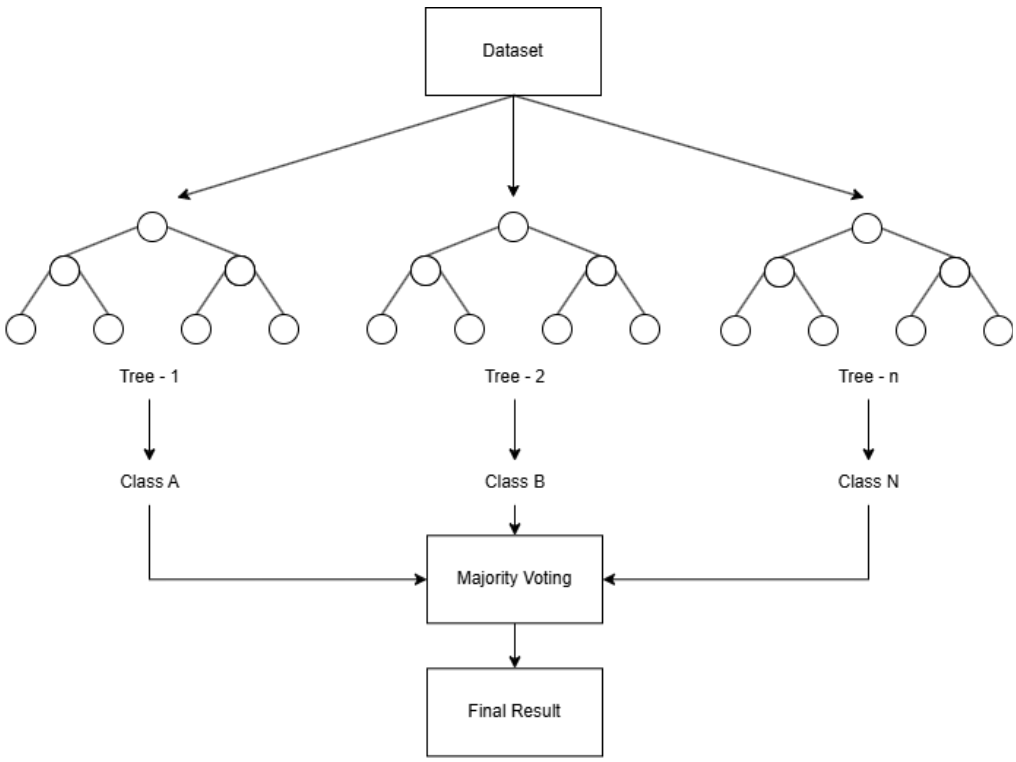

Figure 1. Random Forest modelling

In the Random Forest method, several processes can be described as follows:

a) Bootstrapping

This stage involves creating a subset by randomly sampling with replacement of size *n* from the dataset.

b) Random feature selection

In this stage, trees are built to their maximum size, and the splitting variable among the *m* predictor variables is selected randomly. The best splitter is then chosen based on these *m* predictors. Random Forest has several hyperparameters that must be manually tuned to improve system performance in this study. One such hyperparameter is the criterion, which measures the quality of each split. There are two available options for the criterion hyperparameter: gain and entropy. Gain uses impurity gain as the metric, while entropy measures quality based on information gain. The prediction for an observation is made by aggregating the results from *k* trees using a majority vote. In the process of constructing the decision tree, Random Forest selects features with the smallest Gini split to form the tree. Thus, not all features are used in a single tree. The feature with the smallest Gini index is chosen as the splitting feature. The steps for calculating the Gini index are provided in Eq. (5) [55].

$$Gini(S) = 1 - \sum_{i=1}^{m}(Pi)^2 \quad (5)$$

The explanation of the formula above is as follows: S represents the total number of data samples, m denotes the number of classes or data labels, and p_i refers to the probability of class i, which is calculated by dividing the number of data in class i by the total number of data samples. Subsequently, the Gini Split is calculated using Eq. (6), as follows:

$$Gini_{split}(S) = \frac{|S_1|}{|S|} Gini(S_1) + \frac{|S_2|}{|S|} Gini(S_2) \quad (6)$$

The explanation of the formula above is as follows: S refers to the dataset before the split, while S_1 and S_2 are the two data subsets after the split. $Gini(S)$ indicates the Gini Index value for the dataset S. The expression $\frac{|S_1|}{|S|}$ represents the proportion of samples in the subset S_1 relative to the total number of samples in the set S, and $\frac{|S_2|}{|S|}$ represents the proportion of samples in the subset S_2 to the total samples in S.

To obtain the prediction result from the Random Forest algorithm, the majority voting method is used across individual decision trees. In a Random Forest composed of N decision trees, this is described in Eq. (7).

$$l(y) = argmax_x(\sum_{n=1}^{N} I_{h_n(y)=c}) \quad (7)$$

In the formula, l is an indicator function, and h_n represents the output of the n-th decision tree in the Random Forest model [31].

Random Forest has an internal mechanism that allows for an estimate of the general error, known as the out-of-bag (OOB) error. During the construction of decision trees, only about two-thirds of the original data from the bootstrap samples are used. At the same time, the remaining one-third is used to test the model's performance using the trees built.

The OOB error estimation is the average prediction error for each training case, computed using only the trees that did not include that case in their bootstrap sample. Once the Random Forest is entirely constructed, the entire training process will involve each tree, and a proximity matrix is calculated for each case based on how often pairs of cases end up in the same terminal node [31].

3. RESEARCH METHODOLOGY

The research adopts the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology [56-58], which encompasses the research procedures, as illustrated in Figure 2.

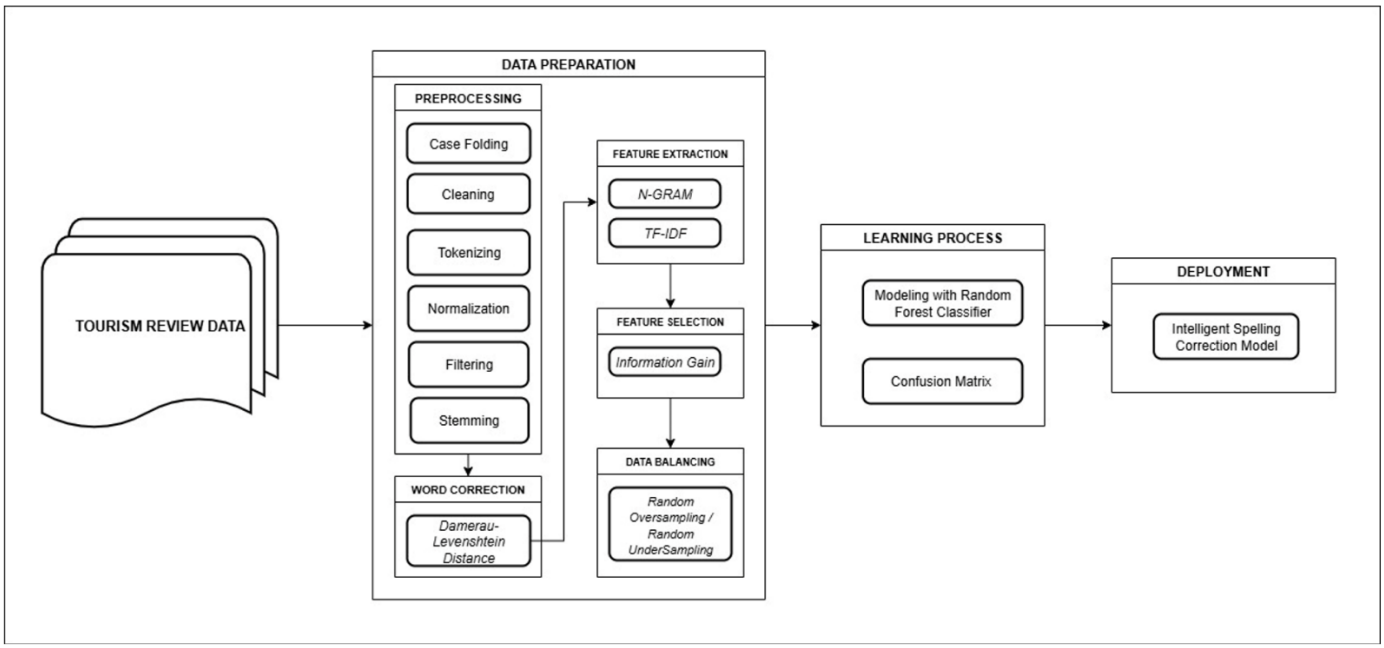

Figure 2. Diagram of research development

3.1 Research stages

This research employs a structured approach to develop an Intelligent Spelling Correction Model based on tourism review data. The research stages are designed systematically and consist of several key phases: data collection, data preparation, learning process, and deployment. The complete flow of the research process is illustrated in Figure 2.

3.1.1 Data collection

The initial stage involves collecting tourism review data from various digital sources, such as travel websites, social media, and user review platforms. This data is typically unstructured and contains numerous spelling errors or informal word forms, which require cleaning and correction.

3.1.2 Data preparation

This stage is a crucial part of text data analysis and includes the following processes:

a. Preprocessing: The goal of this step is to clean and normalise the text.
b. Case Folding, converting all letters to lowercase for consistency.
c. Cleaning, removing special characters, numbers, or irrelevant symbols.
d. Tokenising, splitting sentences into word units (tokens).

e. Normalisation, standardising informal or nonstandard words to their proper forms.
f. Filtering, removing stopwords or less meaningful words.
g. Stemming, reducing words to their root form using a stemming algorithm.

(1) Word Correction, after preprocessing, an automatic spelling correction process is applied using the Damerau-Levenshtein Distance algorithm.

The dictionary used for correction combines two sources: 1) a general Indonesian dictionary (KBBI) and 2) vocabulary extracted from the tourism review dataset, ensuring both linguistic accuracy and domain relevance.

(2) Feature Extraction, to convert the text data into numerical features suitable for classification, two feature extraction methods are applied:

a. N-Gram, the model uses unigram and bigram features (n = 1–2) to capture both individual words and short contextual phrases.
b. TF-IDF (Term Frequency-Inverse Document Frequency), TF-IDF parameters were set with min_df = 2, max_df = 0.9, and sublinear_tf = True to minimise noise from infrequent or overly common words.

(3) Feature selection is performed using the Information Gain method to choose the most relevant features for the target variable and reduce model complexity.

(4) Data Balancing, to prevent bias toward the majority class in the dataset, techniques such as random oversampling (adding more minority class samples) and random undersampling (reducing majority class samples) are applied to achieve a balanced class distribution.

To enhance reproducibility, detailed implementation settings are described as follows.

For the Damerau–Levenshtein Distance (DLD) algorithm, the spelling-correction dictionary was derived from the Kamus Besar Bahasa Indonesia (KBBI), the official Indonesian-language dictionary, ensuring consistency with standard Indonesian word forms.

The N-Gram tokenisation was configured to include both Bigram (n = 2) and Trigram (n = 3) combinations to capture contextual word dependencies in user-generated text.

The TF-IDF vectorisation process employed the *Scikit-learn* TfidfVectorizer with parameter ngram_range = (1,2), allowing the inclusion of both unigram and bigram features (and up to trigram features). The model was fitted and transformed on the preprocessed text column (comment_DLD), producing a sparse TF-IDF matrix where each row represents a comment and each column a word or phrase feature. The resulting feature matrix was then converted to a DataFrame, which served as input to the Random Forest Classifier during the learning phase.

3.1.3 Learning process

This is the core stage of machine learning and includes:

a) Modelling with Random Forest Classifier, the preprocessed and vectorised data is used to train a classification model using the Random Forest Classifier algorithm. This algorithm is chosen for its robustness in handling high-dimensional data and delivering stable predictions.

b) Model performance was evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score, derived from the confusion matrix. Accuracy measures the proportion of correctly classified samples, precision quantifies the ratio of true positives among predicted positives, recall indicates the proportion of correctly identified positive instances, and the F1-score provides a harmonic mean between precision and recall to ensure balanced performance evaluation.

All experiments were implemented using Python 3.10 with the Scikit-learn library on a Windows 11 environment equipped with 16 GB of RAM. This configuration ensures the reproducibility of the experimental setup and facilitates comparison with future studies.

3.1.4 Deployment

After successful training and evaluation, the resulting classification and correction process is implemented as an Intelligent Spelling Correction Model. This model can be deployed in text-based systems to automatically correct spelling errors, particularly in the context of tourism reviews.

4. RESULTS AND DISCUSSION

4.1 Data collection

This subsection describes the data collection process, including the scraping sources, dataset composition, and its relevance to the research objective. The data collection stage in this context refers to data scraping. Before gaining deeper insight into the data, it is essential first to gather the data itself. The scraping process was carried out using sources from social media platforms and travel platforms. A total of 1,634 review entries were successfully collected, comprising user-expressed opinions from each platform. This initial step is crucial, as it lays the foundation for data scraping.

4.2 Research variables

This section defines the independent and dependent variables used in the sentiment classification process and explains their transformation from textual to numerical features.

In this study, there is one independent variable (X), namely the reviews or comments in the dataset, which influence or are used in the process of prediction and classification. This independent variable is further broken down into features based on terms (words) after the TF-IDF process is completed. As a result, the independent variable is no longer in the form of a review or comment column but becomes a set of features or terms.

The dependent variable (Y) refers to the labels in the dataset, which serve as the predicted output or the analysis result.

4.3 Strengths and weaknesses of the data

Understanding the dataset's strengths and limitations is essential to ensure reliable model training and fair evaluation of results.

The advantage of the dataset used in this study is its diversity, which generates a large number of terms. This contributes to a more complex dataset with many features that can help the classification model, especially when using the Random Forest method. The complexity of the data improves the model's classification performance.

However, the dataset also has several weaknesses. It is imbalanced, meaning the number of instances per class is unequal. This requires data balancing to prevent the model

from becoming biased during training and to achieve good performance and accuracy.

Additionally, the dataset has not been pre-processed, which is necessary to clean the data in accordance with sentiment analysis standards. The data is also not yet in binary form, so it must first be transformed into binary numerical form using TF-IDF and N-Gram tokenisation.

4.4 Data labelling and text preprocessing

The data labelling and preprocessing stages were designed to clean, normalise, and prepare the raw text reviews for subsequent machine learning modelling.

The data labelling process was carried out by three annotators (students), and a linguistics expert validated the results. From this process, 966 data entries were labelled as positive sentiment and 668 as negative sentiment, for a total of 1,634.

Text preprocessing is a crucial step in text data processing, aiming to clean and structure raw data for analysis. This process helps remove irrelevant elements such as punctuation, informal words, and other noise, thereby improving data quality. The preprocessing steps performed in this study include Case Folding and Cleaning, Tokenising, Normalisation, Filtering, and Stemming.

4.5 Balancing data and splitting data

This subsection explains how data imbalance was handled using oversampling, undersampling, and SMOTE techniques to ensure balanced class representation during training.

In the Data Understanding process, one of the study's shortcomings is data imbalance. Therefore, a method is needed to balance the data, namely by using Random Oversampling. Based on the implementation, the results are presented in Table 1.

Table 1. Results of data balancing and data splitting

Condition	Positive	Negative	Total Data
Before Balancing	966	668	1,634
After Oversampling	966	966	1,932
After Undersampling	668	668	1,336
After SMOTE	966	966	1,932

Based on Table 1, data balancing was applied to each dataset subset. The classification process was then carried out using the Random Forest Classifier on each balanced subset to identify which subset yielded the best performance.

4.6 Modelling using random forest

The modelling phase involved developing and testing the Random Forest classifier using different preprocessing and feature extraction configurations to determine the optimal combination for sentiment analysis.

At this stage, a machine learning model is developed using the Random Forest algorithm. Random Forest is an ensemble learning-based method that combines multiple decision trees to improve prediction accuracy and reduce the risk of overfitting.

Model evaluation was conducted using two test scenarios. The first scenario applied Random Forest with feature extraction using TF-IDF and N-Gram tokenisation (Bigram), followed by feature selection using Information Gain, and included spelling correction using the Damerau-Levenshtein Distance (DLD) method.

In contrast, the second scenario used the Random Forest model without N-Gram tokenisation or Damerau-Levenshtein Distance spelling correction. In both scenarios, Hyperparameter Tuning using Grid Search was used to find the best-performing model.

4.7 Testing scenario

a) Scenario 1: Random Forest with DLD and N-Gram

Scenario 1 evaluates the effect of integrating spelling correction (DLD) and contextual feature extraction (N-gram) on model accuracy compared with baseline configurations.

In the first scenario, modelling was performed using the Random Forest classification algorithm, incorporating the Damerau-Levenshtein Distance (DLD) to correct misspellings and N-Gram tokenisation with n = 2 (Bigram). Feature selection was conducted using the Information Gain (IG) method, selecting features with values above a specific threshold, to ensure relevance while avoiding excessive features. The thresholds used were 0.0002, 0.0004, 0.0006, and 0.0008.

Data balancing was applied using three methods: Random Oversampling, Undersampling, and SMOTE. These methods were used to evaluate which balancing technique yielded the best results with the model.

Table 2. Best parameters for testing scenario 1 (Random oversampling)

Parameter	Default	Parameter Grid	Best Parameter Information Gain			
			0.0008	0.0006	0.0004	0.0002
n_estimators	100	[100, 110, 145]	110	100	145	145
max_depth	None	[None, 45, 60, 80]	None	60	80	
min_samples_split	2	[2, 5, 10]	5	2		
min_samples_leaf	1	[1, 5, 10]	1			
class_weight	None	[None, 'balanced']	balanced	None		

Table 3. Best parameters for testing scenario 1 (Random undersampling)

Parameter	Default	Parameter Grid	Best Parameter Information Gain			
			0.0008	0.0006	0.0004	0.0002
n_estimators	100	[100, 110, 145]	100		145	110
max_depth	None	[None, 45, 60, 80]	None	45	None	45
min_samples_split	2	[2, 5, 10]	2		10	
min_samples_leaf	1	[1, 5, 10]	1			
class_weight	None	[None, 'balanced']	balanced			

Table 4. Shows the best parameters for testing scenario 1 (SMOTE)

Parameter	Default	Parameter Grid	Best Parameter Information Gain			
			0.0008	0.0006	0.0004	0.0002
n_estimators	100	[100, 110, 145]	100		145	100
max_depth	None	[None, 45, 60, 80]	None		60	80
min_samples_split	2	[2, 5, 10]	10		2	
min_samples_leaf	1	[1, 5, 10]	1			
class_weight	None	[None, 'balanced']	balanced	None		balanced

Table 5. Evaluation matrix results of test scenario 1 (Random oversampling)

Balancing Method	Threshold (Accuracy)	Total Features (IG)	Class	Precision	Recall	F1-Score
Random Oversampling	0.0008 (89.69%)	1213	Positive	91%	87%	89%
			Negative	88%	92%	90%
	0.0006 (90.21%)	5720	Positive	89%	90%	90%
			Negative	91%	90%	90%
	0.0004 (88.66%)	9963	Positive	90%	86%	88%
			Negative	88%	91%	89%
	0.0002 (87.63%)	10116	Positive	91%	83%	87%
			Negative	85%	92%	88%
Random Undersampling	0.0008 (80.60%)	1213	Positive	80%	80%	80%
			Negative	81%	81%	81%
	0.0006 (81.34%)	5720	Positive	79%	83%	81%
			Negative	84%	80%	82%
	0.0004 (82.09%)	9963	Positive	82%	80%	81%
			Negative	82%	84%	83%
	0.0002 (82.84%)	10116	Positive	85%	78%	81%
			Negative	81%	87%	84%
SMOTE	0.0008 (84.54%)	1213	Positive	88%	79%	84%
			Negative	82%	90%	86%
	0.0006 (84.02%)	5720	Positive	81%	87%	84%
			Negative	87%	81%	84%
	0.0004 (83.51%)	9963	Positive	89%	76%	82%
			Negative	80%	91%	85%
	0.0002 (84.02%)	10116	Positive	90%	76%	82%
			Negative	80%	92%	86%

Hyperparameter Tuning was also performed using GridSearchCV to determine the best-performing and most optimal model configuration. The parameters used in this tuning process included: n_estimators, max_depth, min_samples_split, min_samples_leaf, and class_weight.

As a result, the best parameters for each data-balancing subset were determined for model training and testing. The differences between the three tables are based on the highest accuracy achieved or the one closest to optimal performance. The best parameters for each balanced dataset subset are shown in Table 2.

Table 2 presents the best parameters obtained from the Random Oversampling balanced data subset. The best parameters for each defined threshold are also included in this table. Therefore, the optimal model performance can be achieved using the identified best parameters.

Table 3 presents the best parameters and optimal model performance for each threshold used in the first testing scenario within the Random Undersampling balanced data subset.

Table 4 presents the best parameters obtained using the SMOTE balanced data subset. From the best parameters identified across the three data balancing methods, it is evident that each threshold within each balanced dataset subset yields different optimal parameters, with varying levels of accuracy. Meanwhile, the results obtained from the testing scenarios are presented in Table 5.

Based on the test results presented in Table 5, the use of the Damerau-Levenshtein Distance (DLD) and N-Gram has been shown to improve the performance of the Random Forest Classifier model. DLD assists in spelling correction, enabling similar words to still be recognised as the same entity. The impact of this method is evident from the best accuracy of 90.21% at an Information Gain threshold of 0.0006 using the Random Oversampling data-balancing subset. With a total of 5,720 features, the model captures data patterns more effectively without overfitting.

Feature selection using Information Gain (IG) aims to choose the most relevant features for the model. The smaller the threshold, the more features are included in the classification process. However, the results indicate that increasing the number of features does not always directly correlate with better model accuracy. At a threshold of 0.0008, the model used only 1,213 features but still achieved an accuracy of 89.69%. Meanwhile, at a threshold of 0.0002, the number of features drastically increased to 10,116, yet the accuracy dropped to 87.63%. This indicates that too many features can lead to overfitting, where the model overly adapts to the training data and performs suboptimally on test data. Therefore, the best threshold in this scenario is 0.0006, as it provides a balance between a sufficient number of features and maximum accuracy.

Additionally, Hyperparameter Tuning was conducted using GridSearchCV to optimise model performance. The best parameters obtained were: 'n_estimators = 100', 'max_depth = 60', 'min_samples_split = 2', 'min_samples_leaf = 1', and 'class_weight = 'balanced''. Although this tuning improved the model's stability, its impact on accuracy was relatively

small compared to selecting the optimal IG threshold. Overall, the model using DLD and N-Gram with an IG threshold of 0.0006 achieved the best performance, with 90.21% accuracy and balanced precision, recall, and F1-score across all sentiment classes.

These results suggest that using DLD and N-Gram is efficacious in improving feature quality. However, there is still an 8.62% margin of error, as DLD sometimes misclassifies correctly spelt words. Moreover, appropriate feature selection remains necessary to prevent overfitting due to the large number of features or words generated by N-Gram with n=2 (Bigram).

Figure 3 illustrates the performance graph resulting from Scenario 1 testing. It shows that using the Damerau-Levenshtein Distance (DLD) and N-Gram with various Information Gain (IG) thresholds yields optimal improvements in accuracy. DLD effectively corrects misspelt words into proper spellings, while N-Gram tokenisation with n=2 (bigrams) generates a large variety of features. Therefore, combining this with feature selection based on Information Gain provides excellent synergy. This is because, as the number of features increases and the model becomes more complex, excessive or insufficient features can lead to overfitting or underfitting. Thus, proper feature selection is crucial to prevent such issues.

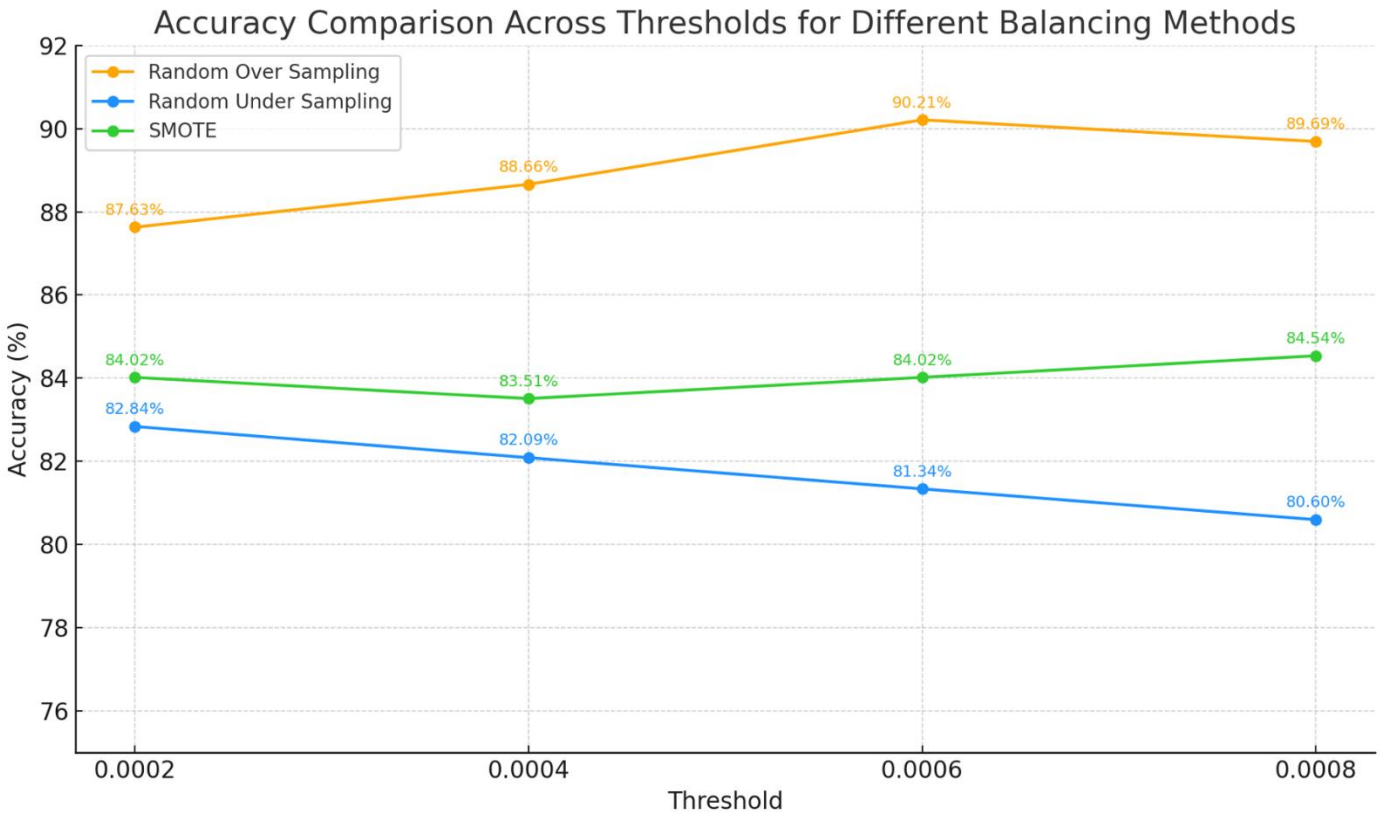

Figure 3. Performance chart of random forest with DLD and N-Gram

These results demonstrate that DLD significantly improves the correction of misspelt words, thereby enhancing classification accuracy. Additionally, the use of bigram N-Gram contributes to a more diverse and abundant set of features, increasing data complexity. Filtering these features using IG selection is an effective strategy to mitigate overfitting and underfitting, while also improving overall classification performance.

b) Scenario 2: Random Forest without DLD and N-Gram

Scenario 2 serves as a baseline experiment, where the Random Forest classifier is trained without applying spelling correction or N-Gram tokenisation, enabling direct performance comparison with Scenario 1.

The second testing scenario was conducted using the Random Forest classification algorithm without applying Damerau-Levenshtein Distance (DLD) and N-Gram tokenisation. However, the Information Gain feature selection method was still used to select features with values above the relevance threshold, ensuring the number of features was not excessive. The thresholds used were 0.0002, 0.0004, 0.0006, and 0.0008. Data balancing was performed using three methods: Random Oversampling, Undersampling, and SMOTE. These three techniques were applied to determine which data-balancing method yielded the optimal model.

Hyperparameter tuning was performed using GridSearchCV was applied to obtain the best-performing model. The same parameters were used as in the first scenario. Consequently, the best parameters for each balanced data subset were determined for the model training and testing. The differences between the three resulting tables are based on the highest or most optimal accuracy achieved.

The implementation of hyperparameter tuning was performed using GridSearchCV achieved the best performance. The final optimal test results from the best parameter combinations yielded different best parameters and accuracies for each threshold. The results showed a total of 2,298 features—fewer than in the first scenario Table 6.

In this scenario, as shown in Table 6, the Random Forest Classifier model was tested without using the Damerau-Levenshtein Distance (DLD) or N-Gram features, so the text was processed as individual words (unigrams) without spelling correction or sequential context understanding. The

optimal accuracy was achieved with data balancing using Random Oversampling, reaching 89.69%. These results indicate that the model's performance is slightly lower than in the DLD and N-Gram scenario, although still quite good, with the highest accuracy reaching 89.69% at an Information Gain threshold of 0.0002 on the Random Over Sampling data-balancing subset.

Without DLD, the model does not benefit from spelling error correction, meaning that words with slightly different spellings are treated as distinct features. This can reduce modelling effectiveness, as similar actual information cannot be generalised effectively. Additionally, without N-Gram, the model cannot account for word order, so relationships within a phrase are not well captured, which affects the model's performance in understanding the text context for each label.

Feature selection using Information Gain (IG) was performed to choose the most relevant features for sentiment classification. As seen in Table 6, the lower the IG threshold, the more features are selected. With a threshold of 0.0008, the model used only 578 features and still achieved 88.14% accuracy on the optimal data-balancing subset, namely Random Oversampling. When the threshold was lowered to 0.0006, the number of features increased to 1,277, but the accuracy remained at 88.14%, indicating that adding more features does not continually improve performance. A threshold of 0.0004 showed an increase in accuracy to 89.18% with 1,845 features, and a threshold of 0.0002 peaked at 89.69% with 1,965 features. Although the 0.0002 threshold resulted in the highest accuracy, the difference from the 0.0004 threshold was only 0.51% as the number of features increased. This suggests that although more features were included, the impact on accuracy improvement was not very significant, making the 0.0004 threshold more optimal in terms of feature efficiency.

To improve model performance, Hyperparameter Tuning was performed using GridSearchCV, aiming to find the best parameter combination. The optimal parameters obtained included 'n_estimators = 145', 'max_depth = None', 'min_samples_split = 5', and 'min_samples_leaf = 1'. These tuning results helped the model remain stable by avoiding overfitting, even with an increased number of features.

Table 6. Evaluation matrix results of testing scenario 2

Balancing Method	Threshold (Accuracy)	Total Features (IG)	Class	Precision	Recall	F1-Score
Random Over-Sampling	0.0008 (88.14%)	578	Positive	90%	85%	87%
			Negative	87%	91%	89%
	0.0006 (88.14%)	1277	Positive	87%	89%	88%
			Negative	90%	87%	88%
	0.0004 (89.18%)	1845	Positive	92%	85%	88%
			Negative	87%	93%	90%
	0.0002 (89.69%)	1965	Positive	90%	88%	89%
			Negative	89%	91%	90%
Random Under-Sampling	0.0008 (83.58%)	578	Positive	84%	81%	83%
			Negative	83%	86%	85%
	0.0006 (82.09%)	1277	Positive	76%	91%	83%
			Negative	90%	74%	81%
	0.0004 (84.33%)	1845	Positive	85%	81%	83%
			Negative	84%	87%	85%
	0.0002 (82.09%)	1965	Positive	81%	81%	81%
			Negative	83%	83%	83%
SMOTE	0.0008 (85.57%)	578	Positive	89%	80%	84%
			Negative	83%	91%	87%
	0.0006 (85.57%)	1277	Positive	88%	81%	84%
			Negative	83%	90%	87%
	0.0004 (84.02%)	1845	Positive	86%	80%	83%
			Negative	82%	88%	85%
	0.0002 (84.02%)	1965	Positive	86%	80%	83%
			Negative	82%	88%	85%

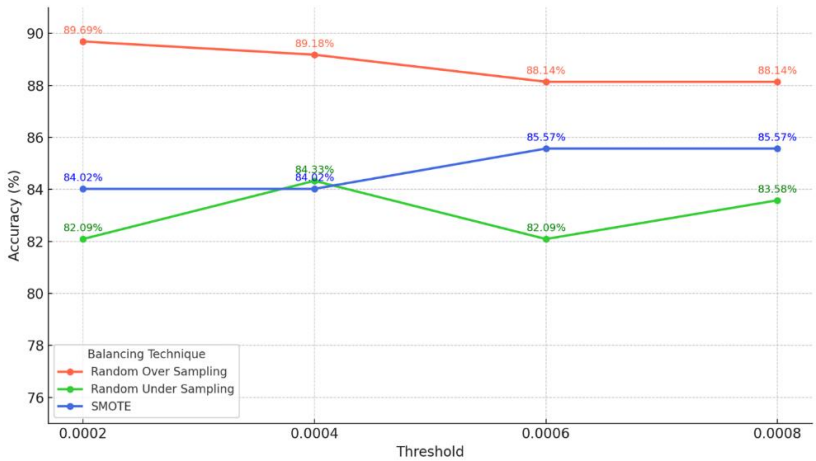

Figure 4. Performance graph of the random forest model without Damerau-Levenshtein Distance and N-Gram

Overall, the model without DLD and N-Gram still achieves good performance, with a maximum accuracy of 89.69% at an IG threshold of 0.0002 on the optimal data-balancing subset (Random Over Sampling). However, compared to the DLD and N-Gram scenario, the accuracy tends to be slightly lower, indicating that spelling correction and contextual word understanding contribute to improving model performance. Without DLD and N-Gram, the model relies more heavily on appropriate feature selection to achieve accurate classification.

Overall, the model performs well, with more correct predictions than errors. Figure 4 illustrates the training and testing results when DLD and N-Gram are not used, with the same parameters and IG threshold.

Based on Figure 4, this is a graph from Scenario 2 testing. It indicates that when using only the Random Forest model without Damerau-Levenshtein Distance (DLD) and N-Gram, with various Information Gain (IG) threshold values, the accuracy still improves optimally across the different data balancing subsets used. The absence of DLD significantly affects performance because misspelt words retain their original weights, negatively impacting the classification process. Meanwhile, the lack of N-Gram also affects performance as the features remain limited and do not increase data complexity. As a result, word combinations are treated individually (unigrams), making it more difficult for the model to understand and predict the correct classes.

Feature selection in this test appears to be slightly less effective, as each threshold produces a similar number of features and relatively close accuracy scores. This contrasts with feature selection using tokenised N-Gram data, which shows more notable differences. However, despite these limitations, the model's performance in this scenario is still relatively good compared to Scenario 1, though there is a slight decrease in accuracy of 0.52%.

c) Scenario 3: Comparison of data balancing methods

To assess the influence of data balancing on model stability, this section compares the results obtained using Random Oversampling, Random Undersampling, and SMOTE techniques across both experimental scenarios.

Based on the tests conducted in Scenarios 1 and 2, a comparison of the data-balancing techniques used was performed. Data balancing was applied using three methods: Random Oversampling, Random Undersampling, and SMOTE. Each of these balancing techniques was implemented in both Scenario 1 and Scenario 2, allowing for the identification of the most optimal accuracy results for each technique in both scenarios. The optimal accuracy results for each data-balancing method in Scenarios 1 and 2 are shown in the graph.

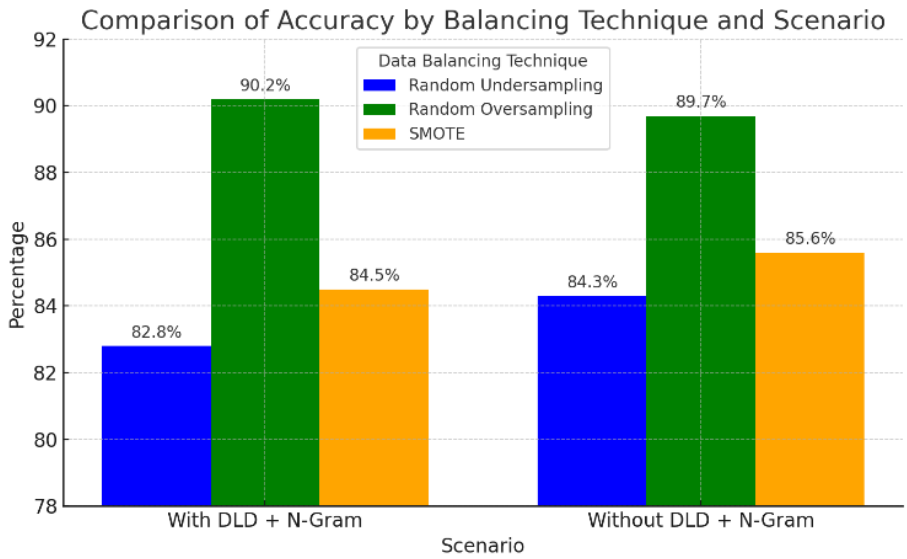

Figure 5. Performance comparison of data balancing techniques across scenarios

The graph in Figure 5 above compares the model's accuracy across two scenarios: with and without the use of Damerau-Levenshtein Distance (DLD) and N-Gram. Each scenario was tested using three data balancing techniques: Random Undersampling, Random Oversampling, and SMOTE. The results show that Random Oversampling achieved the highest accuracy in both scenarios, reaching 90.2% with DLD and N-Gram and 89.7% without them. SMOTE performed worse than Random Oversampling, with an accuracy of 84.5% with DLD and N-Gram and 85.6% without them. Meanwhile, Random Undersampling yielded the lowest accuracy, at 82.8% in the scenario with DLD and N-Gram, and 84.3% in the scenario without them. Overall, the use of DLD and N-Gram provided a slight improvement in accuracy when paired with Random Oversampling, but had minimal impact when used with SMOTE and Random Undersampling. Therefore, the combination of Random Oversampling with DLD and N-Gram is considered the most effective approach.

d) Evaluation of results

This subsection synthesises findings from all experimental scenarios and discusses the implications of spelling correction, feature selection, and data balancing on overall model performance.

Based on evaluations of two testing scenarios using the Random Forest model, it can be concluded that the use of the Damerau-Levenshtein Distance (DLD) and N-Gram features significantly improved sentiment classification effectiveness. In the first scenario, the combination of DLD and N-Gram increased accuracy to 90.21% at an Information Gain threshold of 0.0006, with the most optimal result obtained with the Random Oversampling balancing method compared to other balancing techniques. This demonstrates that spelling

correction and sequential word processing help the model capture textual patterns more effectively. Furthermore, the use of feature selection based on Information Gain proved essential for balancing the number of features and avoiding overfitting.

Meanwhile, in the second scenario—where DLD and N-Gram were not applied—the model experienced a performance decline, with a maximum accuracy of 89.69% at an Information Gain threshold of 0.0002, again using Random Oversampling as the optimal balancing technique. Without DLD, words with different spellings could not be recognised as the same entity, leading the model to lose important information during classification. Additionally, the absence of N-Gram meant that inter-word context was not taken into account, limiting the model's understanding of textual structure.

From these two scenarios, it can be concluded that the best approach to improving sentiment classification accuracy is to combine spelling correction (DLD), N-Gram tokenisation, and proper feature selection using Information Gain to balance the number of features and prevent overfitting. Moreover, choosing the appropriate data balancing method and Information Gain threshold significantly affects the balance between feature count and class distribution, as well as overall model accuracy.

The use of Hyperparameter Tuning with GridSearchCV helped identify optimal parameter combinations. However, its impact on accuracy was not as significant as the choice of text preprocessing and feature selection methods. Thus, the combination of DLD and N-Gram, along with proper data balancing and Information Gain threshold selection, forms an effective strategy for enhancing sentiment classification accuracy.

The Random Oversampling method achieved the highest accuracy because it effectively balanced the distribution between positive and negative sentiment classes, thereby reducing model bias. By ensuring equal representation of both classes, the Random Forest model learned more diverse and representative textual patterns.

The DLD algorithm corrected several types of misspellings that often appeared in user-generated text, including transpositions (e.g., *bagus* → *bgaus*), insertions (*tidak* → *tidakk*), and deletions (*puas* → *pas*). Correcting these word-level errors improved token alignment in the feature space, enabling semantically similar words to be effectively grouped in the N-Gram model.

In the N-Gram representation, bigrams such as pantai indah", "tempat kotor", "ombak tenang", and "pemandangan bagus" were found to be the most influential in distinguishing sentiment polarity, as they captured contextual meaning that single words (unigrams) could not.

An error analysis was also conducted to identify misclassified samples. The model tended to misclassify ambiguous or mixed-sentiment reviews, such as *"pemandangannya indah tapi akses jalannya rusak"* or *"pantainya bagus, tapi terlalu ramai"*. These sentences contain both positive and negative expressions, creating ambiguity in classification. In addition, reviews containing sarcastic language or local Madurese expressions occasionally led to incorrect sentiment predictions, since the model could not yet fully capture cultural or regional nuances.

These findings indicate that while the current DLD, N-Gram and Random Forest approach performs effectively, incorporating contextual embeddings or semantic models (e.g., BERT or word2vec) in future studies could help address limitations in understanding nuanced language.

The performance comparison is shown in Figure 6. The chart in Figure 6 shows that the first scenario, which incorporates DLD and N-Gram, consistently achieves higher accuracy than the second scenario, which does not utilise these methods. Although the accuracy difference is not significant, it indicates that more advanced text preprocessing techniques can help improve classification accuracy.

The main contribution of this work lies in the integration of DLD-based intelligent spelling correction with contextual N-Gram representation and Random Forest classification for Indonesian tourism sentiment analysis. Unlike prior studies that address each component separately, this research empirically validates the combined impact on noisy, user-generated datasets.

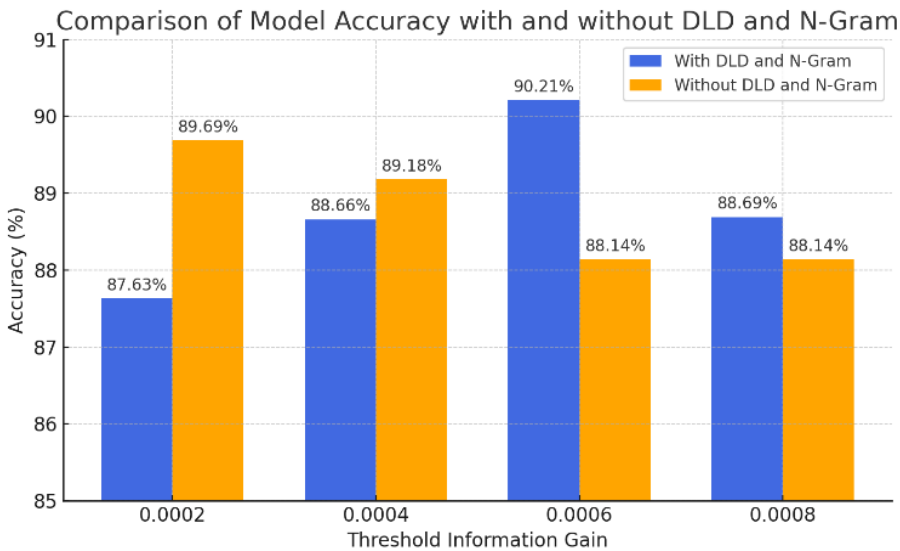

Figure 6. Comparison chart of accuracy results across evaluation scenarios

5. CONCLUSIONS

This study presents an intelligent spelling correction model based on the Damerau-Levenshtein Distance (DLD) and N-Gram tokenisation, combined with the Random Forest

Classifier, to enhance the sentiment analysis of tourism reviews for Madura beach. Our results demonstrate that integrating spelling correction techniques and advanced feature extraction improves model accuracy, achieving a peak of 90.21% accuracy. The use of Information Gain for feature selection and Random Oversampling for balancing the class distribution further optimised performance, particularly in handling imbalanced datasets, a common challenge in sentiment analysis tasks.

The significant improvement in accuracy underscores the importance of preprocessing steps, such as spelling correction and context-based feature extraction, for achieving more reliable sentiment classification. Moreover, the DLD proved highly effective at addressing common spelling errors in reviews, thereby directly impacting the quality of sentiment analysis. The use of N-Gram tokenisation helped capture word order and context, which are essential for understanding sentiment nuances, especially in informal, user-generated content such as tourism reviews.

However, several limitations should be acknowledged. While the DLD algorithm effectively corrects standard error types such as transpositions, deletions, and insertions, it still struggles with context-dependent errors (e.g., homonyms like “panta” vs. “pantai”) and semantic inconsistencies, where a word is correctly spelt but used in the wrong context. Additionally, compound words and highly informal expressions (e.g., “bgt,” “beneran,” “mantapp”) remain challenging because they deviate from standard Indonesian lexical forms.

For future work, the model can be enhanced by integrating contextual embedding-based architectures, such as BERT, word2vec, or Transformer-based spell correction models, which can capture both syntactic and semantic relationships among words. Incorporating these deep learning techniques could improve the system’s ability to handle informal, ambiguous, and context-sensitive language more effectively.

These enhancements would allow the model not only to detect surface-level spelling errors but also to understand contextual nuances, resulting in more accurate and robust sentiment analysis on user-generated tourism reviews.

ACKNOWLEDGMENT

This research was conducted with funding support from BIMA, the Ministry of Higher Education, Science, and Technology (Kemendikti Saintek) of Indonesia in 2025, under the Fundamental Research – Regular scheme, contract number: Master Contract Number: 120/C3/DT.05.00/PL/2025, Main DIPA Number: SP DIPA - 139.04.1.693320/2025, Derived Contract Number B/025/UN46.1/PT.01.03/BIMA/PL/2025.

REFERENCES

[1] Tandamrong, D., Laphet, J., Gooncokkord, T. (2025). Evaluating carbon credit offsets: Carbon neutral tourism for passengers traveling from Thailand to China. Challenges in Sustainability, 13(4): 535-545. https://doi.org/10.56578/cis130405

[2] Nurjaya, I.N. (2023). Legal policy of sustainable tourism development: Toward community-based tourism in Indonesia. Journal of Tourism Economics and Policy, 2(3): 123-132. https://doi.org/10.38142/jtep.v2i3.404

[3] Suresh, A., Wartman, M., Rasheed, A.R., Macreadie, P.I. (2025). Tourism and recreation in blue carbon ecosystems: Exploring synergies, trade-offs and pathways to sustainability. Ocean & Coastal Management, 266: 107697. https://doi.org/10.1016/j.ocecoaman.2025.107697

[4] Bakalo, N., Makhovka, V., Krekoten, I., Glebova, A., Kulakova, S. (2025). Local tourism as financial and economic development driver of the community: Management aspect. World Development Perspectives, 38: 100693. https://doi.org/10.1016/j.wdp.2025.100693

[5] Manero, A., Yusoff, A., Lane, M., Verreydt, K. (2024). A national assessment of the economic and wellbeing impacts of recreational surfing in Australia. Marine Policy, 167: 106267. https://doi.org/10.1016/j.marpol.2024.106267

[6] Kurniasari, W., Amaliyah, B. (2025). Developing halal tourism in Madura: In the context sharia cooperation model in batik creative industry. KnE Social Sciences, 10(5): 328. https://doi.org/10.18502/kss.v10i5.18124

[7] Febryano, I.G., Wahyuni, P., Kaskoyo, H., Damai, A.A., Mayaguezz, H. (2022). The potential of tourism in Pahawang Island, Lampung Province, Indonesia. Journal of Green Economy and Low-Carbon Development, 1(1): 34-44. https://doi.org/10.56578/jgelcd010104

[8] Liu, C., Chong, H.T. (2023). Social media engagement and impacts on post-COVID-19 travel intention for adventure tourism in New Zealand. Journal of Outdoor Recreation and Tourism, 44: 100612. https://doi.org/10.1016/j.jort.2023.100612

[9] Rodriguez-Sanchez, C., Torres-Moraga, E., Sancho-Esper, F., Casado-Díaz, A.B. (2025). Prosocial disposition shaping tourist citizenship behavior: Toward destination patronage intention. Tourism Management Perspectives, 55: 101334. https://doi.org/10.1016/j.tmp.2024.101334

[10] Duong, L.H., Kong, Y.Q., Olya, H., Lee, C.K., Girish, V.G. (2025). Echoes of tragedy: How negative social media shapes tourist emotions and avoidance intensions? A multi-methods approach. Tourism Management, 108: 105122. https://doi.org/10.1016/j.tourman.2024.105122

[11] Sabiote-Ortiz, C.M., Castaneda-García, J.A., Frías-Jamilena, D.M. (2024). What shapes tourists’ visit intention in different stages of public health crises? The influence of destination image, information-literacy self-efficacy, and motivations. Journal of Destination Marketing & Management, 31: 100864. https://doi.org/10.1016/j.jdmm.2024.100864

[12] Su, L., Jia, B., Huang, Y. (2022). How do destination negative events trigger tourists’ perceived betrayal and boycott? The moderating role of relationship quality. Tourism Management, 92: 104536. https://doi.org/10.1016/j.tourman.2022.104536

[13] George, O.A., Ramos, C.M. (2024). Sentiment analysis applied to tourism: Exploring tourist-generated content in the case of a wellness tourism destination. International Journal of Spa and Wellness, 7(2): 139-161. https://doi.org/10.1080/24721735.2024.2352979

[14] Jim, J.R., Talukder, M.A.R., Malakar, P., Kabir, M.M., Nur, K., Mridha, M.F. (2024). Recent advancements and challenges of NLP-based sentiment analysis: A state-of-the-art review. Natural Language Processing Journal, 6: 100059. https://doi.org/10.1016/j.nlp.2024.100059

[15] Hajbi, S., Amezian, O., Moukhi, N.E., Korchiyne, R., Chihab, Y. (2024). Moroccan Arabizi-to-Arabic conversion using rule-based transliteration and weighted Levenshtein algorithm. Scientific African, 23: e02073. https://doi.org/10.1016/j.sciaf.2024.e02073
[16] Wijayanti, D.E., Priyanto, Moh. W., Qomariyah, N., Suprapti, I. (2024). The effect of products and services on the intention to revisit coastal tourism destinations in Madura Island. Journal of Indonesian Tourism, Hospitality and Recreation, 7(1): 41-54. https://doi.org/10.17509/jithor.v7i1.62932
[17] Mayvani, T.C.S., Afin, R., Idialis, A.R., Sariyani, S. (2025). Analysis of growth and tourism clusters in Madura. Jurnal Ekonomi Dan Studi Pembangunan, 14(1): 59-71. https://doi.org/10.17977/um002v14i12022p059
[18] Chaabi, Y. Ataa Allah, F. (2022). Amazigh spell checker using Damerau-Levenshtein algorithm and N-gram. Journal of King Saud University-Computer and Information Sciences, 34(8): 6116–6124. https://doi.org/10.1016/j.jksuci.2021.07.015
[19] Arifudin, R., Isnanto, R., Warsito, B. (2024). Optimizing student performance with GridSearchCV using random forest classifier method to enhance learning outcome prediction. In 2024 Ninth International Conference on Informatics and Computing (ICIC), Medan, Indonesia, pp. 1–5. https://doi.org/10.1109/icic64337.2024.10956861
[20] Wong, W., Glance, D. (2011). Statistical semantic and clinician confidence analysis for correcting abbreviations and spelling errors in clinical progress notes. Artificial Intelligence in Medicine, 53(3): 171-180. https://doi.org/10.1016/j.artmed.2011.08.003
[21] Audah, H.A., Yuliawati, A., Alfina, I. (2023). A comparison between symspell and a combination of damerau-levenshtein distance with the trie data structure. In 2023 10th International Conference on Advanced Informatics: Concept, Theory and Application (ICAICTA), Lombok, Indonesia, pp. 1–6. https://doi.org/10.1109/icaicta59291.2023.10390399
[22] Kusuma, A.T.A., Ratnasari, C.I. (2023). Comparison of spell correction in Bahasa Indonesia: Peter Norvig, LSTM, and N-gram. JIKO (Jurnal Informatika dan Komputer), 6(3): 214-220. https://doi.org/10.33387/jiko.v6i3.7072
[23] Isbarov, J., Huseynova, K., Rustamov, S. (2024). Robust automated spelling correction with deep ensembles. In 2024 8th International Conference on Intelligent Systems Metaheuristics & Swarm Intelligence (ISMSI), Singapore, pp. 26-30. https://doi.org/10.1145/3665065.3665070
[24] Nissar, I., Mir, W.A., Shaikh, T.A., Areen, T., Kashif, M., Khiani, S., Hussain, A. (2024). An intelligent healthcare system for automated diabetes diagnosis and prediction using machine learning. Procedia Computer Science, 235: 2476-2485. https://doi.org/10.1016/j.procs.2024.04.233
[25] Chowanda, A., Sutoyo, R., Meiliana, Tanachutiwat, S. (2021). Exploring text-based emotions recognition machine learning techniques on social media conversation. Procedia Computer Science, 179: 821-828. https://doi.org/10.1016/j.procs.2021.01.099
[26] Jayadianti, H., Santosa, B., Cahyaning, J., Saifullah, S., Drezewski, R. (2023). Essay auto-scoring using N-Gram and Jaro Winkler based Indonesian Typos. Matrik, 22(2): 325-338. https://doi.org/10.30812/matrik.v22i2.2473
[27] Aziz, R., Anwar, M.W., Jamal, M.H., Bajwa, U.I., Castilla, Á.K., Rios, C.U., Thompson, E.B., Ashraf, I. (2023). Real word spelling error detection and correction for urdu language. IEEE Access, 11: 100948-100962. https://doi.org/10.1109/access.2023.3312730
[28] Wahyu Andrian, B., Adline Twince Tobing, F., Zuhdi Pane, I., Kusnadi, A. (2023). Implementation of naïve bayes algorithm in sentiment analysis of twitter social media users regarding their interest to pay the tax. International Journal of Science, Technology & Management, 4(6): 1733-1742. https://doi.org/10.46729/ijstm.v4i6.1015
[29] Taradhita, D.A.N. Putra, I.K.G.D. (2021). Hate speech classification in indonesian language tweets by using convolutional neural network. Journal of ICT Research and Applications, 14(3): 225-239. https://doi.org/10.5614/itbj.ict.res.appl.2021.14.3.2
[30] Fatah, D.A., Rochman, E.M.S., Setiawan, W., Aulia, A.R., Kamil, F.I., Su'ud, A. (2024). Sentiment analysis of public opinion towards tourism in Bangkalan Regency using Naïve Bayes method. E3S Web of Conferences, 499: 01016. https://doi.org/10.1051/e3sconf/202449901016
[31] Salman, H.A., Kalakech, A., Steiti, A. (2024). Random forest algorithm overview. Babylonian Journal of Machine Learning, 2024: 69-79. https://doi.org/10.58496/bjml/2024/007
[32] Nachouki, M., Mohamed, E.A., Mehdi, R., Abou Naaj, M. (2023). Student course grade prediction using the random forest algorithm: Analysis of predictors' importance. Trends in Neuroscience and Education, 33: 100214. https://doi.org/10.1016/j.tine.2023.100214
[33] Hapsari, Y., Mujahidin, S., Fadhliana, N. (2023). Analisis sentimen isu vaksinasi Covid-19 pada twitter dengan metode naive bayes dan pembobotan TF-IDF tokenisasi 1-2. SPECTA Journal of Technology, 7(2): 573-583. https://doi.org/10.35718/specta.v7i2.812
[34] Vishva, E.S., Aju, D. (2022). Phisher fighter: Website phishing detection system based on url and term frequency-inverse document frequency values. Journal of Cyber Security and Mobility, 11(1): 83-104. https://doi.org/10.13052/jcsm2245-1439.1114
[35] Mohd Nafis, N.S. Awang, S. (2021). An enhanced hybrid feature selection technique using term frequency-inverse document frequency and support vector machine-recursive feature elimination for sentiment classification. IEEE Access, 9: 52177-52192. https://doi.org/10.1109/access.2021.3069001
[36] Isnan, M., Elwirehardja, G.N., Pardamean, B. (2023). Sentiment analysis for TikTok review using VADER sentiment and SVM model. Procedia Computer Science, 227: 168-175. https://doi.org/10.1016/j.procs.2023.10.514
[37] Peng, X., Jia, P., Fan, X., Liu, J. (2025). ENZZ: Effective N-gram coverage assisted fuzzing with nearest neighboring branch estimation. Information and Software Technology, 177: 107582. https://doi.org/10.1016/j.infsof.2024.107582
[38] Gunawan, L., Anggreainy, M.S., Wihan, L., Santy, Lesmana, G.Y., Yusuf, S. (2023). Support vector machine based emotional analysis of restaurant reviews. Procedia Computer Science, 216: 479-484. https://doi.org/10.1016/j.procs.2022.12.160

[39] Sharma, R., Saghapour, E., Chen, J.Y. (2024). An NLP-based technique to extract meaningful features from drug SMILES. iScience, 27(3): 109127. https://doi.org/10.1016/j.isci.2024.109127
[40] Delibaş, E. (2025). Efficient TF-IDF method for alignment-free DNA sequence similarity analysis. Journal of Molecular Graphics and Modelling, 137: 109011. https://doi.org/10.1016/j.jmgm.2025.109011
[41] Aftab, F., Bazai, S.U., Marjan, S., Baloch, L., Aslam, S., Amphawan, A., Neo, T.K. (2023). A comprehensive survey on sentiment analysis techniques. IJTech, 14(6): 1288. https://doi.org/10.14716/ijtech.v14i6.6632
[42] Alirridlo, M., Septiyanto, A.F., Sarno, R., Sunaryono, D. (2024). A comparative analysis of text normalization techniques for enhanced sentiment analysis performance. In 2024 Beyond Technology Summit on Informatics International Conference (BTS-I2C), Jember, East Java, Indonesia, pp. 474-479. https://doi.org/10.1109/bts-i2c63534.2024.10942038
[43] Shaik, T., Tao, X., Dann, C., Xie, H., Li, Y., Galligan, L. (2023). Sentiment analysis and opinion mining on educational data: A survey. Natural Language Processing Journal, 2: 100003. https://doi.org/10.1016/j.nlp.2022.100003
[44] Bi, J.W., Zhu, X.E., Han, T.Y. (2024). Text analysis in tourism and hospitality: A comprehensive review. Journal of Travel Research, 63(8): 1847-1869. https://doi.org/10.1177/00472875241247318
[45] Mao, Y., Liu, Q., Zhang, Y. (2024). Sentiment analysis methods, applications, and challenges: A systematic literature review. Journal of King Saud University-Computer and Information Sciences, 36(4): 102048. https://doi.org/10.1016/j.jksuci.2024.102048
[46] Chu, M., Chen, Y., Yang, L., Wang, J. (2022). Language interpretation in travel guidance platform: Text mining and sentiment analysis of TripAdvisor reviews. Frontiers in Psychology, 13: 1029945. https://doi.org/10.3389/fpsyg.2022.1029945
[47] Haris, N.A.K.M., Mutalib, S., Ab Malik, A.M., Abdul-Rahman, S., Kamarudin, S.N.K. (2023). Sentiment classification from reviews for tourism analytics. International Journal of Advances in Intelligent Informatics, 9(1): 108-120. https://doi.org/10.26555/ijain.v9i1.1077
[48] Nurfebia, K. (2024). Sentiment analysis of skincare products using the naive bayes method. Journal of Information Systems and Informatics, 6(3): 1663-1676. https://doi.org/10.51519/journalisi.v6i3.817
[49] Şengöz, N., Yiğit, T., Özmen, Ö., Isık, A.H. (2022). Importance of preprocessing in histopathology image classification using deep convolutional neural network. Advances in Artificial Intelligence Research, 2(1): 1-6. https://doi.org/10.54569/aair.1016544
[50] Sarker, I.H. (2021). Machine learning: Algorithms, real-world applications and research directions. SN Computer Science, 2(3): 160. https://doi.org/10.1007/s42979-021-00592-x
[51] Golazad, S., Mohammadi, A., Rashidi, A., Ilbeigi, M. (2024). From raw to refined: Data preprocessing for construction machine learning (ML), deep learning (DL), and reinforcement learning (RL) models. Automation in Construction, 168: 105844. https://doi.org/10.1016/j.autcon.2024.105844
[52] do Rosario Santos, F., Choren, R. (2025). Data preprocessing for machine learning based code smell detection: A systematic literature review. Information and Software Technology, 184: 107752. https://doi.org/10.1016/j.infsof.2025.107752
[53] Hamdani, A.U., Setiawati, S., Mentari, Z.D., Purnomo, M.H. (2024). Comparison of K-NN, SVM, and random forest algorithm for detecting hoax on indonesian election 2024. Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI, 13(1): 166-179. https://doi.org/10.23887/janapati.v13i1.76079
[54] Nuraminah, A., Ammar, A. (2023). Damerau-levenshtein distance algorithm based on abstract syntax tree to detect code plagiarism. SJI, 11(1): 11-20. https://doi.org/10.15294/sji.v11i1.48064
[55] Lan, H., Pan, Y. (2019). A crowdsourcing quality prediction model based on random forests. In 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS), Beijing, China, pp. 315-319. https://doi.org/10.1109/icis46139.2019.8940306
[56] Schröer, C., Kruse, F., Gómez, J.M. (2021). A systematic literature review on applying CRISP-DM process model. Procedia Computer Science, 181: 526-534. https://doi.org/10.1016/j.procs.2021.01.199
[57] Singgalen, Y.A. (2023). Sentiment classification of S.E.A aquarium singapore reviews through CRISP-DM using DT and SVM with SMOTE. Bits, 5(3): 595-606. https://doi.org/10.47065/bits.v5i4.4703
[58] Huber, S., Wiemer, H., Schneider, D., Ihlenfeldt, S. (2019). DMME: Data mining methodology for engineering applications-A holistic extension to the CRISP-DM model. Procedia CIRP, 79: 403-408. https://doi.org/10.1016/j.procir.2019.02.106