ILKOM Jurnal Ilmiah Vol. 18, No. 1, April 2026, pp. 1-16
Accredited 2nd by KEMDIKTISAINTEK No. 10/C/C3/DT.05.00/2025; E-ISSN 2548-7779 | P-ISSN 2087-1716

Optimization of Text Emotion Classification through the Combination of ITC Smoothed and Linear Models

Melki Garonga [a,1,*]; Mc Rore Rangga Punne [a,2]; Irene Devi Damayanti [a,3];
[a] *Univesitas Kristen Indonesia Toraja, Jl. Nusantara No. 12, Tana Toraja 91811, Indonesia*
[1] *melkigaronga@ukitoraja.ac.id;* [2] *mcroreranggapunne@gmail.com;* [3] *irenedamayanti@ukitoraja.ac.id;*
** Corresponding author*

Article history: Received September 06, 2025; Revised October 14, 2025; Accepted March 12, 2026; Available online April 20, 2026

Abstract

This research investigates four feature extraction techniques TF-IDF, Smoothed TF-IDF, Inverse Term Counting (ITC), and ITC Smoothed to determine how effectively they enhance text-based emotion classification when working with imbalanced datasets. The study also seeks to pinpoint the most effective pairing between feature extraction methods and classification algorithms. Its key contributions include a methodical side-by-side comparison of these lesser-examined TF-IDF variations and demonstrating empirically that linear models handle class imbalances with considerable resilience. The analysis drew upon an Indonesian Twitter dataset comprising 4,132 tweets, categorized into six unequally distributed emotional states: anger, fear, joy, love, sadness, and neutrality. These four feature extraction approaches were assessed using five distinct classifiers: Naive Bayes, Logistic Regression, SVM, Random Forest, and KNN. Performance was measured through accuracy, precision, recall, and F1-score. Findings indicate that linear classifiers, specifically Logistic Regression and SVM, delivered superior performance, achieving accuracy rates between 93.71% and 94.44%. These models consistently outperformed both probabilistic and distance-based algorithms regardless of the feature extraction method applied. Interestingly, the impact of smoothing proved context-dependent. While applying smoothing to both TF-IDF and ITC boosted the performance of linear models over their unsmoothed counterparts, it paradoxically reduced accuracy for the standard ITC method. This outcome questions the widely held belief that smoothing universally enhances model performance. The combination of Logistic Regression with the unITC Smoothed method yielded the peak accuracy of 94.44%. The study offers actionable guidance, suggesting the pairing of Logistic Regression with ITC as a highly effective strategy for text-based emotion classification. It also contributes theoretically by underscoring the particular aptitude of linear models for managing high-dimensional text data within imbalanced class contexts.

Keywords: Emotion Text Classification, Feature Extraction, Imbalanced Dataset, ITC, Linear Model, Smoothed TF-IDF, Text Mining.

Introduction

Textual data is an abundant source of information, especially from social media and the internet, so that sentiment analysis and text-based emotion classification are important fields for understanding public opinion, social behavior, and decision making [1]. The choice of representation features such as BoW, TF-IDF, and transformer embeddings directly affects the performance of emotion classification models [2]. Term Frequency-Inverse Document Frequency (TF-IDF), as a dominant text representation technique, focuses on emphasizing words with high informative value. However, various implementation approaches of TF-IDF show significant performance variations in the context of sentiment analysis and emotion classification [3]. Several studies indicate that parameter choices such as weighting schemes, normalization, and stopword removal can have a significant impact on classification results [4]. Therefore, a comparative analysis of various TF-IDF feature extraction method variations is necessary to identify the best configuration that can improve emotion classification accuracy.

In the context of text-based emotion classification, various studies have also shown that TF-IDF-based approaches can compete with or even outperform more complex methods like Word2Vec or transformers, especially in terms of efficiency and accuracy on large-scale datasets or those with limited computational resources [2]. A study using TF-IDF with Random Forest and SVM on a Shopee review dataset achieved 87.2% accuracy but did not explore other TF-IDF variants [3]. Meanwhile, another study proposed an improved TF-IDF for literary text classification but did not integrate it with an evaluation of various classification algorithms [4]. A comparative study between TF-IDF, Word2Vec, and BERT for emotion classification reported that TF-IDF with SVM achieved 89.3% accuracy; however,

http://dx.doi.org/10.33096/ilkom.v18i1.2954.1-16

smoothed variants and ITC were not tested in that study [2]. Previous research has provided an Indonesian-language emotion dataset from Twitter public opinion, but that research stopped at the dataset provision stage and did not perform classification modeling [5].

From the review of these studies, it was found that previous research generally only used standard TF-IDF without exploring variants such as Smoothed TF-IDF, Inverse Term Count (ITC), or ITC Smoothed in the context of emotion classification, even though earlier studies indicated that these variants have the potential to produce more discriminative feature representations [6], [7]. The absence of a study that systematically compares the performance of these four feature extraction variants on diverse classification algorithms with an Indonesian-language emotion dataset further strengthens the urgency of this research. Furthermore, analysis of model resilience to class imbalance, which is a common characteristic of public opinion datasets, in the context of feature extraction method variations, is still very limited. In addition, the interaction between smoothing techniques with term frequency-based methods (TF-IDF) versus term uniqueness-based methods (ITC) has never been studied comparatively, so it is not yet known empirically whether the smoothing effect is universal or depends on the basic characteristics of the feature extraction method used.

To address these research gaps, this study introduces a novel contribution a systematic comparative analysis of four feature extraction techniques TF-IDF, Smoothed TF-IDF, ITC, and ITC Smoothed all evaluated within a unified experimental framework for emotion classification, an approach not previously undertaken. In addition, the research tests the hypothesis that linear models, particularly Logistic Regression and SVM, consistently outperform other classifiers across all feature variants while maintaining robustness against class imbalance. This challenges the prevailing assumption that complex or ensemble methods are essential when dealing with imbalanced datasets. Ultimately, this study seeks to evaluate and compare the performance of these four feature extraction methods in combination with five classification algorithms, aiming to pinpoint the most effective configurations and offer actionable recommendations for building text-based emotion classification systems.

Method

The research methodology is a series of systematic stages designed to guide the research process conceptually and purposefully, in accordance with the objectives to be achieved. This approach aims to produce a system that has undergone testing stages and is expected to provide solutions to the identified problems. The overall research procedure and workflow are outlined in **Figure 1**:

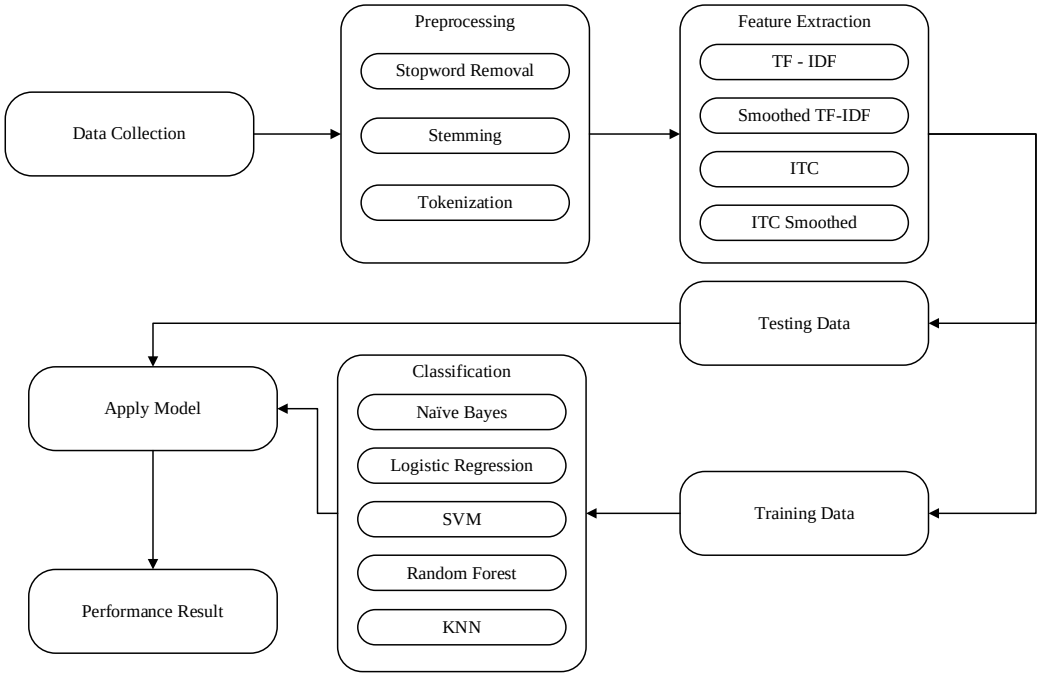

Figure 1. Procedure And Workflow

A. Data Collection

This study uses a dataset [5] by title "*Emotion Dataset from Indonesian Public Opinion*" which contains Indonesian-language tweets annotated into six emotion labels: anger, fear, joy, love, sad, and neutral. T To create this dataset, tweets were gathered using the Twitter API and annotated manually following the emotion framework proposed by Shaver and Parrott. Now publicly hosted on GitHub, the dataset serves as a resource for advancing emotion classification tasks specifically for Indonesian text.

The total data used in this study amounts to 4,132 tweets that have been reduced and annotated into six emotion categories. Each data is an Indonesian-language tweet (post) depicting public opinion on social media, particularly Twitter. The data distribution based on emotion labels is as follows Joy category with 944 data, Anger with 838 data, Sad with 838 data, Fear with 805 data, Neutral with 430 data, and Love with 277 data. **Figure 2** shows the dataset stored in .csv format.

Tweet	Label
pagi2 udah di buat emosi :)	Anger
kok stabilitas negara, memange 10 thn negara tdk aman, bahkan sby menyuburkan ormas2 radikal, intoleran, teroris, yg berafiliasi ke partai tertentu..narasi klhtn intelektual tp bodoh..	Anger
dah lah emosi mulu liat emyu	Anger
aib? bodoh benar! sebelum kata aib itu muncul, terlebih dahulu sudah ada tindakan. yakni kekejian! jangan kau sembunyikan caramu menelaah masalah. semisal anak perempuanmu ditempa	Anger
dih lu yg nyebelin bego	Anger
asli malu maluin org indo tolol yg rep latah "cilukba" pake huruf hijaiyah sm "ngntd" sama ganti huruf t pake salib, ada tiktok filipin lewat fyp aku dan repnya "ngtd" semua, dasar goblogg tren	Anger
drama abg tolol	Anger
masih emosi sih sama katla kemarin. mana keterangannya gini aja. ((hasil mengaci)) kzl.	Anger
bangsat tribute no.1, bencana no.2 mau ngalahin ini keknya	Anger
pengen pergi jauh terus teriak sambil nangis sekencang kencangnya nanti balik kalo gue udah lupa segalanya wkwj kn tolol mustahil banget	Anger
ya allah mau marah tpi gimana ya	Anger
silibus kita memang kbat gila babi. bodoh yang buat silibus ni. learning should be fun.	Anger
pagi-pagi dikerjain orang haha dikira gua bego, ya gua ladenin lahh liat sampe manaa	Anger
angka covid lagi naek. kalo masih dilanjut event offlinenya, asli bodoh sih ni brand.	Anger
bukan lebih ke sangar sih , kalo menurutku goblokk bin tolol . di ingetin untuk kebaikan malah acuh	Anger
mas ngelonte dulu mas, biar otaknya jadi seger lg, bagi org kafir seperti lu ngewe dgn lonte ibadah yg utama. punya tolol kok kebangetan bgt sih	Anger
1-10!! marah aku	Anger
tolol banget. mana ada bercinta. yg ada mimpi di kejar kejar setan semalem. asu bocil nih yang iya iya aja kalo bicara	Anger
izinin suami poligami buat menghindari zina? lol banget, heh mmngnya nabi saya poligami gegara mata keranjang trus sangean? trus ada niatan zina, tpi biar gak zina jdi dinikahin? ngelirik cew	Anger
gua nyerah joki aja kali ya tugas poem ya allah kenapa aku bego b ing	Anger
hah? kaya dendam bnaget sama ibu	Anger
huahh benci covid bangett. semoga berhenti di jimin tidak ada jungkook dan juga taehyung. no god pliss	Anger
benci banget sama diri sendiri, gampang nangis kalo ngobrol sama orang tua.	Anger
ni geng lgbt patutle. libtard, gay, bodoh, tahi	Anger
iya ni tvn kan suka memancing emosi penonton	Anger
sei ini ya gue baru bangun tidur udh bikin emosi aja	Anger
mau marah bangeett kenapa orang orang jahat bgt si	Anger
kadang suka emosi baca beginian tapi orang goblog emang susah kalo diajak ngobrol	Anger
si mutajir mancing emosi sih pagi2	Anger
udah gw baca panjang-panjang kirain analisisnya bakal boom. ternyta hanya hasil pikiran otak pendek. tolol	Anger
emosi bung	Anger
mvp pun kene marah tak capai kpi ke ?	Anger
opini lu jelek bro. menutup keburukan dg agama lu benarkan. poligami untuk menghindari zina? tolol! palkon banyak bacot ttg agama ya gini	Anger
kapan ya manajemen di rs diubah biar ga lempar lemparan gini anjimm orang ke rs tuh butuh penanganan cepet bukan bolak balik ribet. emosi kan gue ajgg	Anger
dari turunkan ayam 2 kupang baik buat polisi majukan industri sumber makanan negara. kita negara tropika kot apa jadah harap makanan ternakan import? bodoh sangat ke orang malaysia sar	Anger
unesco org arab hahaa kata yg benci anis	Anger
satu kata jangan mengundi , sebab semua bodoh	Anger
diam, adalah jawaban terbaik untuk menghadapi orang bodoh!	Anger

Figure 2. Dataset

B. Preprocessing

Preprocessing is a crucial initial stage in the text-based emotion classification pipeline, which aims to clean and prepare the data so that it can be optimally represented by the model [8]. The choice of words in each sentence has high expressive power and can convey emotions clearly. Therefore, the preprocessing stages in this study include various important steps. The application of systematic preprocessing has been proven to improve the quality of extracted features and directly contributes to increasing the accuracy of emotion classification models, especially those based on feature representation like TF-IDF [9].

- Stopword Removal.

  Stopword removal is a process in NLP to remove common words such as "*dan*", "*di*", "*ke*", "*yang*", which are considered to have no significant meaning in text analysis. Stopwords are common words that appear often but add minimal value to understanding a document's main message or context. Removing them helps NLP models achieve better performance and precision in tasks ranging from information extraction to text classification and sentiment detection [10].

- Stemming.

  Stemming is a fundamental NLP technique that strips words down to their base or root form. This normalization reduces lexical diversity, allowing algorithms to recognize related words as a single entity and improving overall efficiency in information processing. For example, the words "*berlari*", "*lari*", and "*pelari*" will be mapped to the base form "*lari*". The stemming process is very important in the text data pre-processing stage, such as in document classification, information retrieval, and sentiment analysis [11].

- Tokenization.

  The tokenization process segments raw text into smaller components whether words, subwords, or characters depending on the chosen granularity. As the entry point to most NLP workflows, tokenization creates the fundamental units that algorithms rely on for tasks like understanding context, extracting meaning, and performing further linguistic analysis [12].

C. Feature Extraction

Feature extraction transforms raw textual data into compact vector representations through techniques ranging from TF-IDF to contextual embeddings like BERT. This process reduces dimensionality and computational overhead while carefully preserving the semantic and syntactic information that machine learning models depend on for tasks such as classification, clustering, or regression [13]. The feature extraction methods used in this study are:

- TF-IDF (Term Frequency - Inverse Document Frequency)

  At its core, TF-IDF is a numerical method for quantifying how significant a word is to a document within a larger collection. It combines two metrics: the frequency of a term in a given document (TF) and its inverse frequency across all documents (IDF). Words that appear often in a single document but seldom elsewhere are assigned greater weight, making this approach particularly effective at identifying unique, document-specific terminology [14]. The mathematical formulation of TF-IDF is given in equations (1) and (2) below:

$$IDF = log\ N/d \tag{1}$$

$$TF\text{-}IDF = TF\ x\ IDF \tag{2}$$

- ITC (Inverse Term Count).

  TC (Inverse Term Count) is a feature weighting scheme inspired by the logic of IDF, though it diverges in how it measures term rarity. Instead of counting document occurrences, ITC generally contrasts the total vocabulary size the number of unique terms in the corpus with a word's overall frequency across all texts. The result is a weighting system that, like IDF, diminishes the influence of ubiquitous terms and amplifies less frequent, potentially more meaningful words [6]. The mathematical formulation of ITC is given in equations (3) and (4) below:

$$TF = 1+log\ (TF) \tag{3}$$

$$ITC = TF\ x\ IDF \tag{4}$$

- Smoothed TF-IDF.

  The smoothed TF-IDF method enhances the original formulation by introducing a smoothing parameter into the IDF calculation. This modification serves two important purposes it eliminates the risk of division by zero when a term appears in no documents, and it provides more balanced weight distribution for words that occur sparsely across the corpus. By guaranteeing non-zero values for all terms, this approach creates a more stable and reliable feature representation [15]. The mathematical formulation of Smoothed TF-IDF is given in equations (5) and (6) below:

$$IDF = log[(1+n)/\ 1 + DF] + 1 \tag{5}$$

$$TF\text{-}IDF = TF\ x\ IDF \tag{6}$$

- ITC Smoothed

  This hybrid weighting method synthesizes ITC and Smoothed TF-IDF principles. It features two key components: logarithmic normalization of term frequencies, plus additive smoothing in the IDF computation (achieved by incrementing the numerator by 1). The result is a refined approach to term weighting that balances frequency impact while ensuring numerical stability [7]. The mathematical formulation of ITC Smoothed is given in equations (7), (8) and (9) below:

$$TF = 1+log\ (TF) \tag{7}$$

$$IDF = log[(1+n)/1 + DF] + 1 \quad (8)$$

$$ITC\ Smoothed = TF\ x\ IDF \quad (9)$$

D. Split Data

The practice of dividing data into training and testing sets is fundamental to machine learning workflows. The training subset teaches the model to recognize patterns, while the testing subset kept completely separate evaluates its performance on new data. This approach yields trustworthy estimates of real-world model behavior and flags potential overfitting [16]. Our implementation followed an 80/20 split, with 80% of instances used for training and 20% reserved for testing.

E. Clasification

In machine learning, classification refers to the supervised learning task of assigning input data to predefined categories or classes. The process involves training algorithms on labeled datasets, where they learn to recognize patterns that link specific features to corresponding class labels. After this learning phase, the resulting model can generalize its knowledge to predict the correct class for previously unseen instances [17]. The algorithms used in this research are:

- Naive Bayes

  Naive Bayes is a probabilistic classification algorithm that predicts data categories based on statistical calculations, adopting the assumption of independence between features. Although this simplistic assumption is rarely met in real data, this algorithm still shows significant effectiveness, especially in text processing such as sentiment analysis and document classification. Its main advantage lies in its high computational efficiency when handling high-dimensional data, while maintaining competitive accuracy performance, making it a practical solution for various NLP applications [18], [19], [20].

- Logistic Regression

  Logistic Regression has emerged as a popular choice for text classification tasks due to its capacity to handle high-dimensional feature spaces while delivering interpretable probability-based outputs. The algorithm excels at assigning documents or sentences to predefined categories such as spam detection or sentiment analysis by leveraging text representations like TF-IDF vectors. Its advantages include rapid training times, strong performance on linearly separable data, and adaptability to various preprocessing techniques including feature selection and modern text embeddings. Multiple studies have confirmed its effectiveness in domains ranging from social media comment classification to sentiment analysis [21], [22].

- Support Vector Machine (SVM)

  SVM distinguishes itself as a classification method that seeks to create the widest possible margin between class boundaries. By identifying the hyperplane that best separates categories, it achieves strong generalization performance. The algorithm's flexibility is enhanced through kernel tricks, which enable it to handle non-linear patterns by projecting data into expanded feature spaces. This capability makes SVM particularly effective for text classification, where feature spaces are inherently high-dimensional [23]. Research has consistently demonstrated SVM's strengths in this domain: superior accuracy with text representations [25], adaptability to various document types and lengths, and seamless integration with modern methodologies including kernel mapping and ensemble strategies [24].

- Random Forest

  Random Forest represents a powerful ensemble learning approach that aggregates the outputs of numerous decision trees to achieve superior classification performance while guarding against overfitting. The algorithm introduces randomness at two critical junctures it constructs distinct training subsets for each tree using bootstrap sampling, and it randomly selects features when determining node splits. The integration of advanced feature selection with the Random Forest classifier enhances prediction performance in high-dimensional data. By leveraging the ensemble mechanism, where multiple decision trees are trained on randomly sampled subsets and varied feature sets, the model delivers robust and accurate classification while controlling overfitting [25].

In text classification applications, Random Forest has demonstrated particular strengths, including exceptional accuracy in identifying spam, phishing attempts, and deceptive content with one study reporting 96% accuracy in phishing email detection [26], the algorithm also exhibits robust performance when confronted with irrelevant features or class imbalances, capabilities that can be further enhanced through thoughtful feature selection and optimization of tree count parameters [27].

- K-Nearest Neighbors (KNN)

  KNN's classification performance depends heavily on two core parameters: the carefully chosen k value and the distance metric employed. Getting these right is essential, as inappropriate selections can lead to poor predictions and unreliable models [28]. When applied to text classification, however, KNN offers the advantage of working effectively with multiple distance metrics. Among these, cosine similarity has emerged as particularly effective studies show it achieves the highest accuracy for classifying user opinions and text-based sentiment [29].

The selection of these five algorithms represents four main classification paradigms: probabilistic (Naive Bayes), linear (Logistic Regression and SVM), ensemble (Random Forest), and distance-based (KNN). This allows for a comprehensive comparative analysis of the interaction between feature extraction variants and the characteristics of each algorithm.

F. Model Validation and Optimization

Cross-validation with five folds was applied to the training data as a safeguard against overfitting and to verify model generalization capacity. During each of the five validation rounds, models learned from four folds and were tested on the one held out. The aggregated results from all iterations produced dependable performance estimates. For parameter tuning, we leveraged Grid Search coupled with 5-fold cross-validation to exhaustively explore the hyperparameter space and select the best-performing settings for each algorithm. Readers are referred to **Table 1** for a comprehensive overview of the parameter ranges considered.

Table 1. Configuration in Grid Search Process

No	Algorithm	Hyperparameters	Search Values
1	Naïve Bayes	α (alpha)	0.1, 0.5, 1.0, 1.5
		fit_prior	True, False
2	Logistic Regression	C	0.01, 0.1, 1, 10, 100
		Solver	lbfgs, liblinear
		Penalty	l2
3	Support Vector Machine	C	0.01, 0.1, 1, 10
		Kernel	linear, rbf
		Gamma	scale, auto
4	Random Forest	n_estimators	50, 100, 200
		max_depth	None, 10, 20, 30
		min_samples_split	2, 5, 10
5	K-Nearest Neighbors	n_neighbors	3, 5, 7, 9
		Weights	uniform, distance
		Distance Metric	euclidean, manhattan

To guarantee reproducible outcomes across all experiments, we set a fixed random state of 42 for every algorithm that accepts this parameter. The grid search procedure was conducted independently for each pairing of feature extraction method and classifier resulting in 4 feature variants combined with 5 algorithms, for a total of 20 distinct experimental configurations. Each of these scenarios underwent rigorous evaluation through 5-fold cross-validation coupled with grid-based hyperparameter tuning.

G. Apply Model

In machine learning practice, model evaluation is performed by testing its capabilities on a dataset not used during training. The goal is to ensure the model performs well on new data and does not simply memorize patterns from the training data (overfitting). This process also helps gauge the model's readiness for real-world applications. Common

challenges include selecting relevant evaluation metrics, handling imbalanced datasets, and using cross-validation techniques to improve evaluation accuracy and reliability [30].

H. Performance Result

In machine learning, performance results encompass the various metrics employed to assess a trained model's effectiveness when applied to data particularly unseen test data. These quantitative measures reveal how accurately the model can predict or classify new instances. Key performance indicators typically include:

- Accuracy

  Accuracy measures how often the model's predictions match the actual outcomes. It represents the percentage of correct predictions out of all predictions made. The accuracy formula is provided in equation (9):

$$Accuracy = \frac{TP + TN}{(TP + FP + TN + FN)} x\ 100 \quad (9)$$

- Precision

  Precision evaluates the model's exactness by determining how many of the instances predicted as positive truly belong to that class. This metric is especially important when the cost of false positives is high. Equation (10) shows how precision is calculated:

$$Precision = \frac{TPi}{(TPi + FPi)} x\ 100 \quad (10)$$

- Recall

  Recall, also referred to as sensitivity or true positive rate, assesses the model's ability to detect all actual positive instances within the dataset. This metric is crucial when missing positive cases has significant consequences. The recall formula is presented in equation (11):

$$Recall = \frac{TPi}{(TPi + FNi)} x\ 100 \quad (11)$$

- F Measure (F1-Score)

  The F-measure, specifically the F1-score, combines both precision and recall into a single metric by calculating their harmonic mean. This provides a balanced evaluation, particularly useful when dealing with imbalanced class distributions. Equation (12) displays the F1-score formula:

$$F_1 = 2x \frac{Precision * Recall}{Precision + Recall} x\ 100 \quad (12)$$

Results and Discussion

The experiments detailed in this chapter aim to assess how different TF-IDF feature extraction methods influence the performance of text-based emotion classification systems. By applying various TF-IDF configurations to emotional text data, we analyze their effects on several classification algorithms. The evaluation focuses on standard metrics accuracy, precision, recall, and F1-score for each feature extraction variant. Beyond raw performance numbers, this chapter also investigates the representational effectiveness of each method, how preprocessing choices affect outcomes, and the specific advantages and drawbacks each TF-IDF variation brings to the task of accurately identifying and categorizing emotions.

A. Data Exploration

At this stage, an initial exploration of the dataset was carried out to understand the distribution of emotion labels and the characteristics of words in each emotion category. The data used in this study consisted of several emotion categories that have gone through preprocessing stages, such as punctuation removal, letter normalization, stopword removal, and tokenization. The distribution of data quantity per emotion category is visualized in **Figure 3**.

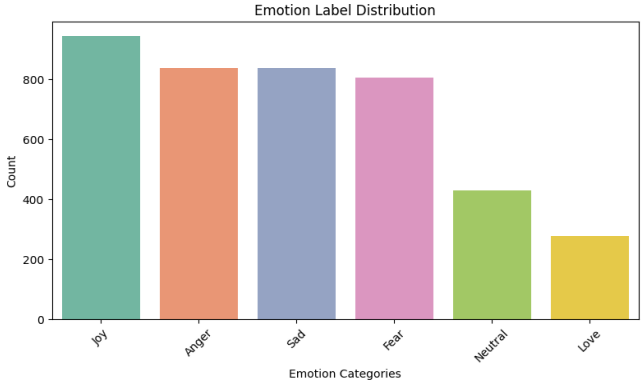

Figure 3. Emotion Label Distribution

To gain deeper insight into the linguistic characteristics of each emotion category, we employed WordCloud analysis to visualize word frequency patterns. The word clouds were generated from cleaned text data, with stopwords and non-alphabetic characters removed, revealing the prominent terms that define each emotional category. WordCloud helps provide an initial overview of the dominant context or keywords for each emotion, and can serve as an initial reference before the TF-IDF feature extraction process, as shown in **Figure 4**. Through this exploration, it was found that several words have a strong association with certain emotion categories. This finding strengthens the hypothesis that text representation through word frequency has strong potential in the text-based emotion classification process.

Figure 4. Word Occurrence Pattern

B. Classification and Accuracy

As depicted in the experimental framework in **Figure 1**, cleaned text data processed through stopword removal, stemming, and tokenization was converted into numerical feature spaces using four feature extraction variants TF-IDF, Smoothed TF-IDF, ITC, and ITC Smoothed. These feature representations were then fed into five different classification algorithms Naive Bayes, Logistic Regression, SVM, Random Forest, and KNN.

The classification phase involved training models on each feature-classifier combination and subsequently testing them on unseen data to gauge real-world performance. Model effectiveness was quantified using accuracy as the primary metric, supplemented by precision, recall, and F1-score to capture different dimensions of classification quality. By aggregating and comparing results across all method combinations, we were able to pinpoint the optimal configuration for Indonesian text-based emotion classification—balancing feature representation choices with algorithmic strengths to achieve the best possible predictive performance.

1. TF-IDF Results

After transforming the text data into TF-IDF feature vectors, we proceeded to train and evaluate five different classifiers: Naive Bayes, Logistic Regression, SVM, Random Forest, and KNN. Model performance was measured on unseen test data using confusion matrices and multiple evaluation metrics. Logistic Regression emerged as the top performer with an accuracy of 93.83%, closely followed by SVM at 93.71% and Random Forest at 92.87%.

Meanwhile, Naive Bayes and KNN lagged behind, achieving only 84.76% and 83.07% accuracy, respectively. The performance gap between these groups of algorithms is noteworthy. The superior results achieved by Logistic Regression and SVM both linear models suggest that TF-IDF feature representations create linearly separable emotion categories that these classifiers can effectively exploit. Conversely, the weaker performance of Naive Bayes and KNN indicates that probabilistic assumptions and distance-based approaches may be less suited to the characteristics of TF-IDF features in this particular emotion classification context. **Figure 5** below provides a visual comparison of the accuracy rates across all five classification algorithms.:

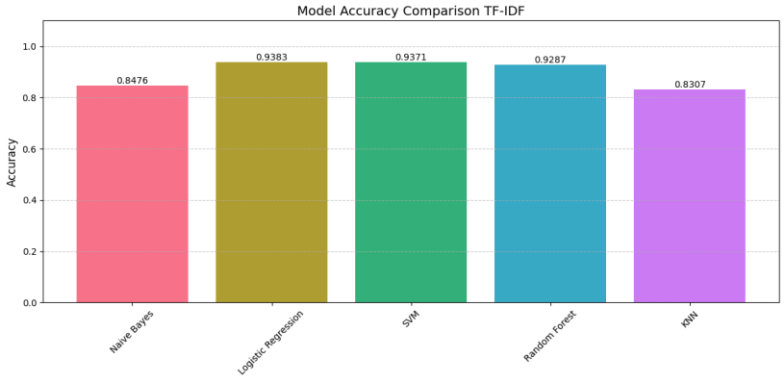

Figure 5. TF-IDF Accuracy Visualization

Figure 6 presents the confusion matrix obtained from the optimal configuration TF-IDF feature extraction combined with the Logistic Regression classifier, which achieved the highest performance among all tested combinations.

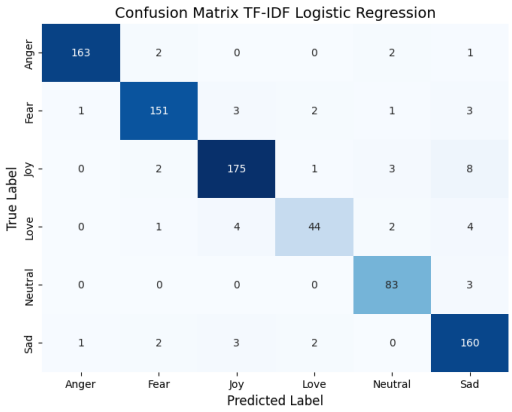

Figure 6. Confusion Matrix TF-IDF With Logistic Regression

When evaluating the five emotion classification models trained on TF-IDF features, Logistic Regression emerged as the clear leader, achieving the highest accuracy at 93.83% and uniquely dominating across all supplementary metrics precision, recall, and F1-score. SVM secured second place with 93.71% accuracy, delivering nearly balanced scores across all evaluation measures. Random Forest followed closely with 92.87% accuracy, demonstrating competitive performance, though slightly trailing the two linear models. In contrast, Naive Bayes (84.76%) and KNN (83.07%) exhibited substantially lower performance across every metric, highlighting their limitations when applied to TF-IDF-based emotion classification tasks. These comprehensive results are summarized in **Table 2**.

Table 2. Performance Model TF-IDF

Model	Accuracy	Precision	Recall	F1-Score
Naive Bayes	84.76 %	86.3 %	84.76 %	83.98 %
Logistic Regression	93.83 %	93.91 %	93.83 %	93.82 %
SVM	93.71 %	93.83 %	93.71 %	93.71 %
Random Forest	92.87 %	92.9 %	92.87 %	92.81 %

Model	Accuracy	Precision	Recall	F1-Score
KNN	83.07 %	83.34 %	83.07 %	82.40 %

2. ITC Results

Following the application of ITC feature extraction, Logistic Regression recorded the highest classification accuracy at 94.44%, marginally ahead of SVM (94.20%) and Random Forest (93.47%). Naive Bayes achieved 85.49%, significantly lower than the linear models. These results demonstrate that while Naive Bayes can perform reasonably well, linear classifiers consistently excel with ITC-based features, delivering both top accuracy and competitive balance across models. The underperformance of Random Forest relative to linear algorithms in this context is particularly noteworthy. A full visualization of these accuracy comparisons is displayed in **Figure 7**.

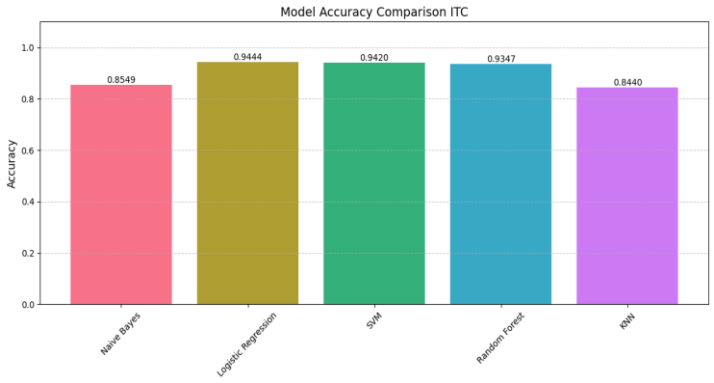

Figure 7. ITC Accuracy Visualization

Figure 8 presents the confusion matrix obtained from the optimal configuration ITC feature extraction combined with the Logistic Regression classifier, which achieved the highest performance among all tested combinations.

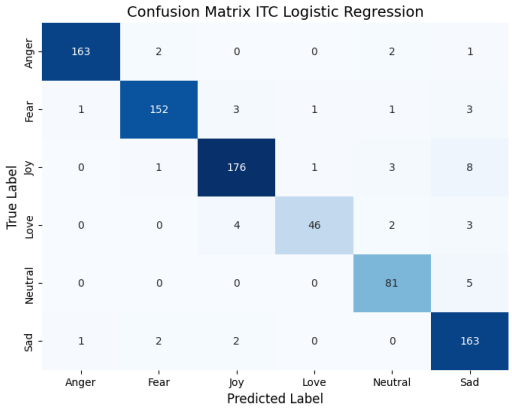

Figure 8. Confusion Matrix ITC With Logistic Regression

When evaluating the five emotion classification models trained on ITC-extracted features, Logistic Regression emerged as the top performer, achieving 94.44% accuracy alongside consistently high scores across all supplementary metrics precision (94.59%), recall (94.44%), and F1-score (94.44%). This performance edged out SVM, which attained 94.20% accuracy, and Random Forest at 93.47%. In contrast, Naive Bayes and KNN lagged considerably behind, with accuracies of 85.49% and 84.40%, respectively. These results reinforce that linear classifiers particularly Logistic Regression and SVM are best equipped to leverage ITC feature representations for accurate emotion classification, as detailed in **Table 3**.

Table 3. Performance Model ITC

Model	Accuracy	Precision	Recall	F1-Score
Naive Bayes	85.49 %	86.84 %	85.49 %	84.85 %
Logistic Regression	94.44 %	94.59 %	94.44 %	94.44 %
SVM	94.20 %	94.32 %	94.20 %	94.20 %
Random Forest	93.47 %	93.56 %	93.47 %	93.43 %
KNN	84.40 %	85.18 %	84.40 %	83.70 %

3. Smoothed TF-IDF Result

Following the application of Smoothed TF-IDF feature extraction, Logistic Regression recorded the highest classification accuracy at 93.95%, marginally ahead of SVM (93.71%) and Random Forest (93.23%). Naive Bayes achieved 84.64%, demonstrating reasonably strong performance despite its probabilistic nature, while KNN brought up the rear with 82.83% accuracy. These results underscore that linear classifiers consistently excel with Smoothed TF-IDF-based features, delivering top accuracy while maintaining competitive balance across models. A full visualization of these accuracy comparisons is displayed in **Figure 9**.

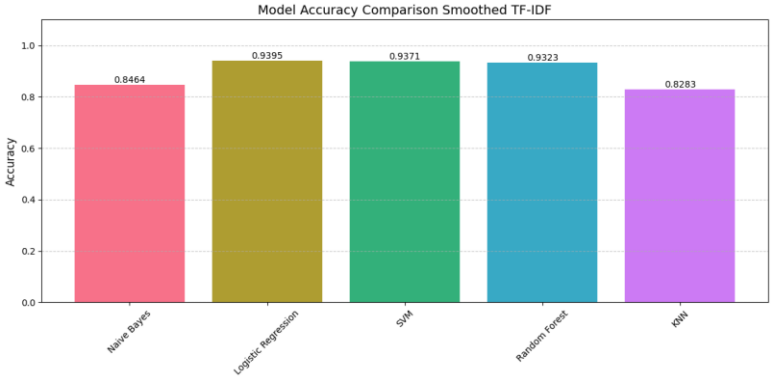

Figure 9. Smoothed TF-IDF Accuracy Visualization

Figure 10 presents the confusion matrix obtained from the optimal configuration Smoothed TF-IDF feature extraction combined with the Logistic Regression classifier, which achieved the highest performance among all tested combinations.

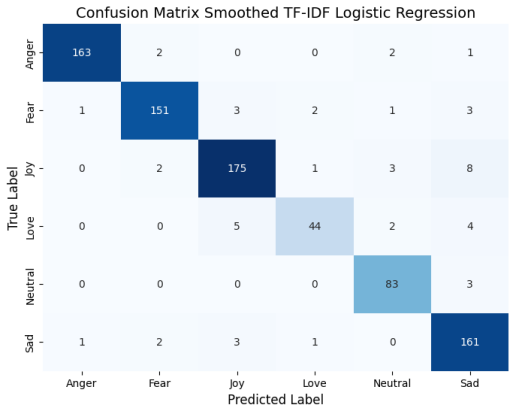

Figure 10. Confusion Matrix Smoothed TF-IDF With Logistic Regression

With Smoothed TF-IDF feature extraction, Logistic Regression achieved the highest classification accuracy at 93.95%, maintaining exceptional consistency across precision (94.05%), recall (93.95%), and F1-score (93.94%). SVM followed closely at 93.71%, while Random Forest recorded 93.23% accuracy. Naive Bayes and KNN trailed significantly at 84.64% and 82.83%, respectively. This performance hierarchy, illustrated in **Table 4**, confirms that linear models especially Logistic Regression and SVM most effectively capitalize on Smoothed TF-IDF feature representations for emotion classification tasks.

Table 4. Performance Model Smoothed TF-IDF

Model	Accuracy	Precision	Recall	F1-Score
Naive Bayes	84.64 %	86.19 %	84.64 %	83.86 %
Logistic Regression	93.95 %	94.05 %	93.95 %	93.94 %
SVM	93.71 %	93.81 %	93.71 %	93.71 %
Random Forest	93.23 %	93.30 %	93.23 %	93.19 %
KNN	82.83 %	83.28 %	82.83 %	82.13 %

4. ITC Smoothed Results

Following the application of ITC Smoothed feature extraction, Logistic Regression recorded the highest classification accuracy at 94.32%, marginally ahead of SVM (93.95%) and Random Forest (93.35%). Naive Bayes achieved 85.25%, while KNN brought up the rear with 84.89% accuracy. These findings demonstrate that linear classifiers consistently excel with smoothed feature representations, delivering top accuracy while maintaining only slight advantages over ensemble methods like Random Forest. However, their superiority over Naive Bayes and KNN is pronounced and consistent. A full visualization of these accuracy comparisons is displayed in **Figure 11**.

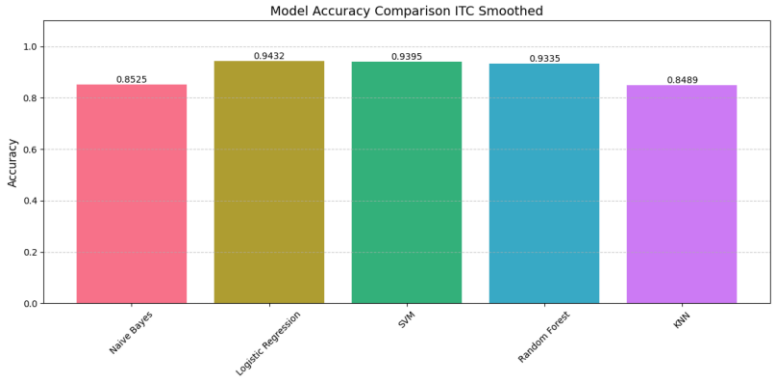

Figure 11. ITC Smoothed Accuracy Visualization

Figure 12 presents the confusion matrix obtained from the optimal configuration ITC Smoothed feature extraction combined with the Logistic Regression classifier, which achieved the highest performance among all tested combinations.

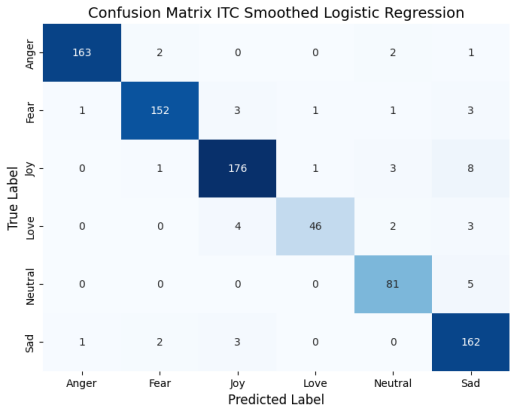

Figure 12. Confusion Matrix ITC Smoothed With Logistic Regression

With ITC Smoothed feature extraction, Logistic Regression achieved the highest classification accuracy at 94.32%, maintaining exceptional consistency across precision (94.46%), recall (94.32%), and F1-score (94.32%). SVM followed closely at 93.95%, while Random Forest recorded 93.35% accuracy. Naive Bayes and KNN trailed significantly at 85.25% and 84.89%, respectively. This performance hierarchy, illustrated in **Table 5**, confirms that

linear models especially Logistic Regression and SVM most effectively capitalize on ITC Smoothed feature representations for emotion classification tasks.

Table 5. Performance Model Smoothed TF-IDF

Model	Accuracy	Precision	Recall	F1-Score
Naive Bayes	85.25 %	86.68 %	85.25 %	84.53 %
Logistic Regression	94.32 %	94.46 %	94.32 %	94.32 %
SVM	93.95 %	94.05 %	93.95 %	93.96 %
Random Forest	93.35 %	93.48 %	93.35 %	93.34 %
KNN	84.89 %	85.43 %	84.89 %	84.11 %

The experimental outcomes demonstrate that classification performance depends heavily on both the feature extraction technique employed and the algorithm selected. Across all experimental conditions, linear models notably Logistic Regression and SVM proved superior to probabilistic (Naive Bayes), distance-based (KNN), and ensemble (Random Forest) approaches. This consistent pattern suggests that TF-IDF and ITC feature representations create decision spaces that are particularly well-suited to linear separation.

When examining the standard TF-IDF condition specifically, Logistic Regression and SVM achieved accuracy above 93%, dramatically outperforming both Naive Bayes and KNN. This disparity reflects fundamental differences in how these algorithms process high-dimensional data. Linear models naturally exploit the sparse structure of TF-IDF features, while KNN's distance-based logic breaks down as dimensions increase a well-understood limitation known as the curse of dimensionality that explains its weaker performance in text classification tasks.

The comparison between TF-IDF and ITC shows that ITC provides consistent performance improvement, especially in linear models. ITC emphasizes term uniqueness at the corpus level more aggressively than standard IDF, thus being able to suppress the influence of common words that appear frequently across emotions. This has a positive impact on the class separation process, particularly for emotions that have distinctive vocabulary but relatively low frequency. This finding strengthens the indication that an approach based on term uniqueness is more effective for emotion classification tasks compared to a pure frequency-based approach.

The smoothing effect appears increasingly significant when applied to both feature extraction methods. Smoothed TF-IDF and ITC Smoothed consistently produce higher accuracy compared to their versions without smoothing. The smoothing technique plays a role in stabilizing the weights of rarely occurring terms and reducing model sensitivity to uneven word distribution. This impact is highly relevant considering the dataset used has an imbalanced class distribution, where some emotions like love and neutral have far fewer data points compared to dominant classes like joy and anger.

The combination of ITC Smoothed with Logistic Regression produced the best overall performance with the highest accuracy of 94.44%. The advantage of this combination demonstrates a synergy between feature weighting that emphasizes term uniqueness, the stabilizing effect of smoothing, and the ability of linear models to maximize the separating margin between classes in a high-dimensional feature space. These results also indicate that performance improvement does not always depend on model complexity, but rather on the suitability between the feature representation and the learning mechanism used.

From the perspective of resilience to class imbalance, the experimental results show that Logistic Regression and SVM maintained high precision, recall, and F1-scores across all feature variants. This indicates that these two models have good generalization ability even when faced with uneven class distributions. Conversely, Naive Bayes and KNN showed more significant performance degradation, indicating their limitations in handling class imbalance in text feature representations.

Overall, the results of this study confirm that the choice of feature extraction method is equally as important as the choice of classification algorithm. Smoothed variants, especially ITC Smoothed, prove to make a real contribution to improving model performance, especially when combined with linear algorithms. These findings not only strengthen the results of previous studies, but also provide new empirical evidence that simple but appropriately configured approaches can outperform more complex methods in the scenario of Indonesian-language text-based emotion classification.

C. Conclusion

This research successfully compared four variants of feature extraction methods TF-IDF, Smoothed TF-IDF, ITC, and ITC Smoothed on five classification algorithms for the task of Indonesian language text emotion classification. The main findings show that linear models (Logistic Regression and SVM) consistently outperformed probabilistic, ensemble, and distance-based models across all feature extraction variants, with the highest accuracy reaching 94.44% in the combination of ITC and Logistic Regression. A particularly notable discovery concerns the conditional nature of smoothing effects: while smoothing enhanced TF-IDF performance, it paradoxically diminished ITC effectiveness due to excessive smoothing. Additionally, linear models demonstrated remarkable robustness to class imbalance, maintaining stable performance on minority emotion categories—challenging the conventional wisdom that complex methods are essential for handling imbalanced datasets.

This research has several limitations that need to be noted. The dataset only originates from Indonesian-language Twitter, so generalization across platforms and languages requires further verification. This research also did not test modern feature extraction methods like Word2Vec or IndoBERT, did not conduct computational cost analysis, and did not apply specific imbalance handling techniques like SMOTE. These limitations do not reduce the significance of the findings, but rather open opportunities for further research development.

Recommended directions for future research include replication across platforms and languages, comparison with deep learning methods, development of adaptive smoothing based on term distribution characteristics, integration of ITC with imbalance handling techniques, and more comprehensive computational cost analysis. Thus, this research not only provides a direct contribution to the development of text emotion classification systems but also opens up new, broader research agendas in the field of feature extraction and text classification modeling.

Acknowledgement

Special thanks are extended by the authors to UKI Toraja for the chance to conduct this research. Furthermore, we acknowledge with appreciation the contributions of everyone who assisted in bringing this research to fruition.

References

[1] L. P. Hung and S. Alias, “Beyond Sentiment Analysis : A Review of Recent Trends in Text Based Sentiment Analysis and Emotion Detection,” *J. Adv. Comput. Intell. Intell. Informatics*, vol. 27, no. 1, 2023, doi: 10.20965/jaciii.2023.p0084.

[2] J. T. Black and M. Z. Shakir, “Emotion on the edge : An evaluation of feature representations and machine learning models,” *Nat. Lang. Process. J.*, vol. 10, no. January, p. 100127, 2025, doi: 10.1016/j.nlp.2025.100127.

[3] S. Suswadi and M. Erkamim, “Sentiment Analysis of Shopee App Reviews Using Random Forest and Support Vector Machine,” *Ilk. J. Ilm.*, vol. 15, no. 3, pp. 427–435, 2023, doi: 10.33096/ilkom.v15i3.1610.427-435.

[4] L. Xiang, “Application of an Improved TF-IDF Method in Literary Text Classification,” *Adv. Multimed.*, vol. 2022, 2022, doi: 10.1155/2022/9285324.

[5] Riccosan, K. E. Saputra, G. D. Pratama, and A. Chowanda, “Emotion dataset from Indonesian public opinion,” *Data Br.*, vol. 43, no. July, 2022, doi: 10.1016/j.dib.2022.108465.

[6] N. S. Mohd Nafis and S. Awang, “An Enhanced Hybrid Feature Selection Technique Using Term Frequency-Inverse Document Frequency and Support Vector Machine-Recursive Feature Elimination for Sentiment Classification,” *IEEE Access*, vol. 9, pp. 52177–52192, 2021, doi: 10.1109/ACCESS.2021.3069001.

[7] M. R. Punne, Indrabayu, and I. Nurtanio, “Mood classification from song lyrics using the Naive Bayes Algorithm, Support Vector Machine (SVM) and XGBoost,” *Proc. 2024 IEEE Int. Conf. Ind. 4.0, Artif. Intell. Commun. Technol. IAICT 2024*, pp. 162–167, 2024, doi: 10.1109/IAICT62357.2024.10617452.

[8] N. Umaira, C. Mohd, and N. A. Shafie, “Performance of TF-IDF for Text Classification Reviews on Google Play Store : Shopee,” *ournal Comput. Res. Innov.*, vol. 9, no. 2, 2024, doi: 10.24191/jcrinn.v9i2.410.

[9] Y. Setiawan, D. Gunawan, and R. Efendi, “Feature Extraction TF-IDF to Perform Cyberbullying Text Classification: A Literature Review and Future Research Direction,” *2022 Int. Conf. Inf. Technol. Syst. Innov.*

ICITSI 2022 - Proc., pp. 283–288, 2022, doi: 10.1109/ICITSI56531.2022.9970942.

[10] S. Chanda and S. Pal, "The Effect of Stopword Removal on Information Retrieval for Code-Mixed Data Obtained Via Social Media," *SN Comput. Sci.*, vol. 4, no. 5, 2023, doi: 10.1007/s42979-023-01942-7.

[11] A. S. Rizki, N. M. Aristi, N. Ridha, A. F. Zulfahri, and D. A. Wibowo, "Implementation of The Indonesian Language Stemming Algorithm in Twitter Data Preprocessing. Case Study: Twitter Wargabanua and Instakalsel," *Fidel. J. Tek. Elektro*, vol. 5, no. 3, pp. 175–183, 2023, doi: 10.52005/fidelity.v5i3.170.

[12] R. Friedman, "Tokenization in the Theory of Knowledge," *Encyclopedia*, vol. 3, no. 1, pp. 380–386, 2023, doi: 10.3390/encyclopedia3010024.

[13] J. T. Pintas, L. A. F. Fernandes, and A. C. B. Garcia, "Feature selection methods for text classification: a systematic literature review," *Artif. Intell. Rev.*, vol. 54, no. 8, pp. 6149–6200, 2021, doi: 10.1007/s10462-021-09970-6.

[14] A. B. Nassif, A. Elnagar, I. Shahin, and S. Henno, "Deep learning for Arabic subjective sentiment analysis: Challenges and research opportunities," *Appl. Soft Comput.*, vol. 98, 2021, doi: 10.1016/j.asoc.2020.106836.

[15] S. I. Manzoor, J. Singla, and Nikita, "Fake news detection using machine learning approaches: A systematic review," *Proc. Int. Conf. Trends Electron. Informatics, ICOEI 2019*, pp. 230–234, 2019, doi: 10.1109/ICOEI.2019.8862770.

[16] A. A. Shujaaddeen, F. Mutaher Ba-Alwi, A. T. Zahary, and A. Sultan Alhegami, "A Model for Measuring the Effect of Splitting Data Method on the Efficiency of Machine Learning Models: A Comparative Study," *4th Int. Conf. Emerg. Smart Technol. Appl. eSmarTA 2024*, pp. 269–277, 2024, doi: 10.1109/eSmarTA62850.2024.10639022.

[17] D. M. Abdullah and A. M. Abdulazeez, "Machine Learning Applications based on SVM Classification: A Review," *Qubahan Acad. J.*, vol. 1, no. 2, pp. 81–90, 2021, doi: 10.48161/qaj.v1n2a50.

[18] P. J. B. Pajila, B. G. Sheena, A. Gayathri, J. Aswini, M. Nalini, and R. Siva Subramanian, "A Comprehensive Survey on Naive Bayes Algorithm: Advantages, Limitations and Applications," *Proc. 4th Int. Conf. Smart Electron. Commun. ICOSEC 2023*, pp. 1228–1234, 2023, doi: 10.1109/ICOSEC58147.2023.10276274.

[19] M. Sindhuja, K. S. Nitin, and K. S. Devi, "Twitter Sentiment Analysis using Enhanced TF-DIF Naive Bayes Classifier Approach," *Proc. - 7th Int. Conf. Comput. Methodol. Commun. ICCMC 2023*, pp. 547–551, 2023, doi: 10.1109/ICCMC56507.2023.10084106.

[20] J. C. Tesoro, "A Semantic Approach of the Naïve Bayes Classification Algorithm," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 3, pp. 3287–3294, 2020, doi: 10.30534/ijatcse/2020/125932020.

[21] M. Özbay Karakuş and O. Er, "A comparative study on prediction of survival event of heart failure patients using machine learning algorithms," *Neural Comput. Appl.*, vol. 34, no. 16, pp. 13895–13908, 2022, doi: 10.1007/s00521-022-07201-9.

[22] A. Zaidi and A. S. M. Al Luhayb, "Two Statistical Approaches to Justify the Use of the Logistic Function in Binary Logistic Regression," *Math. Probl. Eng.*, vol. 2023, no. 1, 2023, doi: 10.1155/2023/5525675.

[23] D. Ogaga and A. Olalere, "Evaluation and Comparison of SVM, Deep Learning, and Naïve Bayes Performances for Natural Language Processing Text Classification Task," no. November, 2023, doi: 10.20944/preprints202311.1462.v1.

[24] P. Saigal and V. Khanna, "Multi-category news classification using Support Vector Machine based classifiers," *SN Appl. Sci.*, vol. 2, no. 3, 2020, doi: 10.1007/s42452-020-2266-6.

[25] A. Yaqoob *et al.*, "SGA-Driven feature selection and random forest classification for enhanced breast cancer diagnosis : A comparative study," *Sci. Rep.*, pp. 1–23, 2025, doi: 10.1038/s41598-025-95786-1.

[26] R. Rajoju, V. Sathvika, G. N. S. Smaran, C. Tejashwini, and G. A. Reddy, "Text Phishing Detection System using Random Forest Algorithm," *Proc. 3rd Int. Conf. Appl. Artif. Intell. Comput. ICAAIC 2024*, pp. 1332–1339, 2024, doi: 10.1109/ICAAIC60222.2024.10575110.

[27] N. Jalal, A. Mehmood, G. S. Choi, and I. Ashraf, "A novel improved random forest for text classification using feature ranking and optimal number of trees," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 2733–2742, 2022, doi: 10.1016/j.jksuci.2022.03.012.

[28] M. Suyal and P. Goyal, "A Review on Analysis of K-Nearest Neighbor Classification Machine Learning Algorithms based on Supervised Learning," *Int. J. Eng. Trends Technol.*, vol. 70, no. 7, pp. 43–48, 2022, doi: 10.14445/22315381/IJETT-V70I7P205.

[29] N. Kalcheva, M. Todorova, and I. Penev, "Study of the K-Nearest Neighbors Method with Various Features for Text Classification in Machine Learning," *Int. Conf. Autom. Informatics, ICAI 2023 - Proc.*, pp. 37–40, 2023, doi: 10.1109/ICAI58806.2023.10339061.

[30] C. Chai, J. Wang, Y. Luo, Z. Niu, and G. Li, "Data Management for Machine Learning : A Survey," *IEEE Trans. Knowl. Data Eng.*, vol. 4347, no. 2, 2022, doi: 10.1109/TKDE.2022.3148237.