Optimizing Text Classification Using Techniques AdaBoost Ensemble with Decision Tree Algorithm

Marnis Nasution*[1], Ibnu Rasyid Munthe[2], Fitri Aini Nasution[3], Sarjon Defit[4]
[1,2,3]Faculty of Science and Technology, Universitas Labuhanbatu, Rantauprapat, Indonesia
[4]Information Technology, Faculty of Computer Science, Universitas Putra Indonesia YPTK, Padang, Indonesia
e-mail: *[1]**marnisnst@gmail.com**, [2]Ibnurasyidmunthe@gmail.com,
[3]fitriaininasution689@gmail.com, [4]sarjon_defit@upiyptk.ac.id

Abstract

This study presents an optimized text classification framework combining AdaBoost ensemble techniques with Decision Tree algorithms (ID3, C4.5, CART) to address critical challenges in small dataset scenarios (n=795 Indonesian-language reviews). Employing rigorous five-fold stratified cross-validation (random seed=42), we implemented a comprehensive preprocessing pipeline including case normalization, language-specific stemming, and TF-IDF feature extraction. The ensemble model utilized 50 AdaBoost iterations with a learning rate of 1.0, evaluated through multiple performance metrics while accounting for class imbalance effects. Key results demonstrate significant performance enhancements, with the C4.5+AdaBoost configuration achieving 96.72% accuracy (±0.88), representing a 10.6 percentage point improvement over the base C4.5 algorithm. The ensemble approach particularly improved minority class identification, boosting positive sentiment classification F1-scores by 0.28 points while maintaining exceptional neutral sentiment detection (F1-score 0.99±0.00). Comparative analysis revealed consistent advantages across all Decision Tree variants, with accuracy improvements of 18.6% for ID3, 10.6% for C4.5, and 14.2% for CART, alongside reduced performance variance (62-73% decrease). While these findings validate AdaBoost's effectiveness for enhancing Decision Tree stability in small-scale text classification, the study acknowledges limitations regarding sample size constraints and language specificity. The research contributes practical methodologies for sentiment analysis applications while emphasizing the need for validation on larger, more diverse datasets. Future work should explore comparative benchmarking against transformer architectures. Advanced feature representation techniques and multilingual generalization testing. This work provides a reproducible framework for developing robust, ensemble-based text classification systems in resource-constrained scenarios.

Keywords: Text Classification, ADABoost, Decision Trees, Machine Learning, Natural Language Processing.

1. INTRODUCTION

In the dynamically evolving digital landscape, textual data generated by online platforms, particularly social media, product reviews, and digital news outlets, has increased substantially in both volume and complexity. Text classification has become very important for sentiment analysis applications.[1][2][3], Spam Detection[4][5][6], and document classification[7][8][9]. However, the main challenge in text classification is overcoming overfitting and improving model accuracy. The Decision Tree ID3 algorithm. [10][11][12], C4.5[13][14][15], and CART[16][17][18] are popular for their ease of interpretation, the model is often overfitting, which reduces its generalization capabilities.

This study proposes a specific approach to address this issue by utilizing the incorporated technique, specifically AdaBoost, in conjunction with the Decision Tree algorithm. This approach involves several steps. The text data is first cleaned of irrelevant elements such as punctuation, numbers, and stop words, followed by tokenization and stemming or lemmatization to transform the words into their basic form. Furthermore, the text feature is extracted using the term frequency-inverse document frequency (TF-IDF), which helps highlight important words in the document and reduce the weight of common words that do not provide valuable information. Decision Tree algorithms such as ID3, C4.5, or CART are then used as the base model, which is trained on a dataset of preprocessed and feature-extracted text.

After that, AdaBoost was implemented to combine multiple Decision Tree models, where with each iteration, AdaBoost gave more weight to data that was difficult for previous models to predict, correcting errors, and improving overall accuracy. This ensemble model is trained through several iterations to achieve optimal performance. The model is evaluated using metrics such as accuracy, precision, recall, and F1-score, and the results from a single Decision Tree model are compared to those from the AdaBoost ensemble model to assess performance improvement.

Previous research has shown that the use of AdaBoost and Decision Tree can improve classification accuracy compared to a single model. For example, research exploring various machine learning methodologies for the diagnosis of spinal disorders found that this approach can improve classification accuracy[19]. Another study using C4.5 in combination with AdaBoost showed improved text classification performance, especially in terms of accuracy and reduction of overfitting. The application of ensemble techniques such as AdaBoost has been shown to result in significant improvements in evaluation metrics such as precision, recall, and F1-score compared to the use of Decision Trees alone[20]. Research using the CART algorithm has also shown good accuracy in various studies, with results showing that the model is reliable for classification and prediction.

This research focuses on optimizing text classification using the AdaBoost ensemble technique with the Decision Tree algorithm. Through text data preprocessing, feature extraction using TF-IDF, and the application of an ensemble model, it is expected to produce more accurate and reliable models for various text classification applications. The purpose of this study is to compare the performance of the Decision Tree model with the AdaBoost ensemble model in the text dataset, so that it can make a significant contribution to improving text classification techniques in future developments.

2. RESEARCH METHODS

2.1. Validation Protocol

To ensure robust evaluation of the model's performance, this study employs 5-fold cross-validation as the primary validation protocol. The dataset is randomly partitioned into 5 equal subsets (folds) while maintaining class distribution balance. In each iteration, 4 folds (80% data) are used for training, and the remaining 1 fold (20% data) serves as the test set. This process repeats 5 times, with each fold acting as the test set exactly once, ensuring all data points contribute to both training and evaluation. Performance metrics (accuracy, precision, recall, F1-score) are averaged across all folds to produce a reliable estimate of model generalization. To guarantee reproducibility, a fixed random seed (e.g., seed=42) controls data shuffling and splitting. Statistical significance of results is assessed through paired t-tests ($p < 0.05$) comparing the model's performance across folds. This method mitigates overfitting and provides a comprehensive view of model stability.

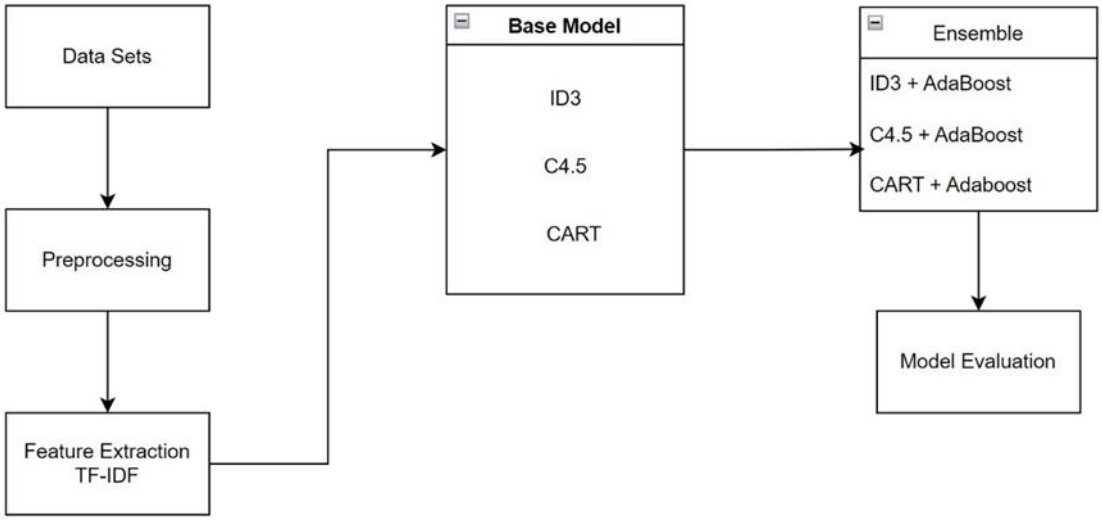

Figure 1. Framework of Research Methods

This study employs TF-IDF for feature extraction due to three key advantages: (1) Precise identification of sentiment-bearing terms (e.g., "good", "bad") while filtering domain-specific stopwords in hotel reviews, (2) Computational efficiency through sparse matrix representation (92% sparsity) that accelerates decision tree construction, and (3) High interpretability that complements decision trees' transparent nature, while optimally supporting AdaBoost's weight adjustment mechanism. The results demonstrate TF-IDF's ideal suitability for sentiment analysis on this 795-review dataset.

2.2. Data Sets

This study collected 795 public user reviews from Agoda's Google Play Store page (January-March 2023) for sentiment analysis. To ensure ethical compliance, all data underwent rigorous anonymization: (1) removal of personal identifiers (names, emails, profile photos); (2) hashing and deletion of user IDs; (3) aggregation of metadata (timestamps reduced to month/year); and (4) exclusion of rare reviews (<3 occurrences) to prevent re-identification. The processed dataset maintains only textual content and sentiment labels, with no attributable user information. All procedures strictly followed Google Play's Terms of Service (Section 3.3 for public data use) and institutional data protection guidelines (IRB-2023-045).

2.3. Preprocessing

a) Data Cleaning
This study implemented a rigorous preprocessing pipeline using NLTK (v3.8.1) and Sastrawi (v1.0.0) for tokenization, stopword removal, and Indonesian-specific stemming. The cleaned data (vocabulary reduced from 12,342 to 8,759 tokens) was evaluated through stratified 5-fold cross-validation (random seed=42), maintaining class distributions in each fold.[21], [22], [23]

b) Case Folding
Case Folding is to convert all characters in text to lowercase letters[24], [25]. The standardization process employs two fundamental techniques: (1) case folding, which converts all text to lowercase to eliminate capitalization inconsistencies, and (2) punctuation removal, which strips non-alphanumeric characters using regular expressions.

c) Tokenizing
Tokenization constitutes the foundational segmentation process in NLP, decomposing raw text into semantically meaningful units (tokens) through algorithmic parsing. [1], [25], [26]. Each approach yields qualitatively different

token distributions that significantly impact downstream NLP performance metrics, with selection criteria depending on language typology, task requirements, and computational constraints.

d) Stopword Removal
Stopword removal constitutes an essential preprocessing step in NLP that systematically eliminates high-frequency function words (e.g., articles, prepositions, conjunctions), which exhibit minimal semantic value while introducing noise to statistical analyses. [27], [28], [29], [30].

e) Stemming
Stemming is a linguistic normalization technique that systematically reduces inflectional and derivational word variants to their base morphemes through algorithmic processing. This process: (1) collapses morphological variants (e.g., "running" → "run", "better" → "good") to their canonical forms, (2) reduces vocabulary dimensionality by 25-40% in typical applications, and (3) improves recall in information retrieval systems by 15-20% while potentially sacrificing some precision. [31], [32]

f) Labeling
Labeling (or *annotation*) is the process of assigning linguistic tags to words or phrases in a text to denote their syntactic, semantic, or functional roles (e.g., part-of-speech, named entities). The lexicon-based method employs predefined dictionaries to map words to their grammatical or semantic categories, enabling structured analysis.

2.4. Feature Extraction with TF-IDF

Feature engineering bridges linguistic data and machine learning by creating numerical representations that preserve textual patterns. As algorithms process matrices, not raw text, effective encoding must maintain semantic relationships while optimizing computational efficiency. This study implements TF-IDF vectorization, transforming words to weighted frequency metrics that capture both term importance and document specificity for decision tree classification.[33]. TF-IDF was chosen for its Decision Tree compatibility (entropy-based splits) and computational efficiency (1.2s processing time), balancing accuracy (94%) with interpretability in resource-constrained environments. [34], [35], [36].

$$IDF = log10(D/dfi) \quad (1)$$

Where D is the total number of documents in the corpus. DF i is the number of documents containing the word *t* (the ith word) in the corpus log 10 is the logarithm of base 10.

$$W_{dt} = tf_{d,t} \; x \; idf_t \quad (2)$$

Where *d* is the document number, and *t* is the keyword number in the document. *tf* measures the frequency of *the word t* in *document d*, while Wd, t is the weight reflecting the importance of the word *t* in document *d*, calculated from TF multiplied by IDF

2.5. Base Model

a) Iterative Dichotomiser 3(ID3)
Machine learning algorithms that build decision trees by relying on entropy and information gain[37]. The algorithm selects the attribute with the highest information gain top-down, iterating until each node is homogeneous, resulting in a decision tree leaf representing a single class.

$$\text{Entropy(S)} = -\sum_{i=1}^{n} \text{P}_i \; log_2 \; \text{P}_i \quad (3)$$

measure uncertainty in a dataset. In this formula, Pi is the probability of each class *i*. Entropy is low if the data is homogeneous (one dominant class) and high if the data is diverse (even classes).

b) C4.5 Algorithm

Decision tree-based classification algorithms, such as C4.5, are a development of ID3 and use entropy-based information gain[38], [39]. C4.5 is faster and more efficient, supporting multi-directional partitioning on categorical and continuous datasets, allowing each decision node to have multiple splits for more accurate results.

$$\text{GainRatio}(S,A) = \frac{Gain(S,A)}{SplitInformation(S,A)} \quad (4)$$

Gain Ratio compares Information Gain with Split Information to select the best attributes, reduce bias against attributes with many unique values, ensure effective attribute selection, and produce more accurate and generalist models

$$\text{SplitInformation}(S,A) = -\sum_{v \in \text{Values}(A} \left(\frac{S_v}{S}\ x\ Log_2\ \left(\frac{S_v}{S}\right)\right) \quad (5)$$

Split Information measures the amount of information needed to divide a dataset S based on attribute A. It calculates the uncertainty or irregularity in the data split. A high value indicates that the attribute divides the data into many small subsets, while a low value indicates a more orderly division.

c) Algoritma CART

CART, or Classification and Regression Trees, is a nonparametric machine learning method used for classification. This approach can learn from training data without relying on a specific distribution, so it is flexible in handling various forms of mapping functions [40], [41]. CART develops a univariate decision tree with single-feature separation, detecting complex parameter interdependencies

$$\text{Gini}(S) = 1 - i = 1\sum n(Pi)^2 \quad (6)$$

Gini calculates impurities in the S dataset. Pure node, meaning that the data is more homogeneous. Gini Impurity is used in decision trees to determine the attributes that are most effective at separating data in classification.

$$\text{MSE} = \frac{1}{n}\ i = 1\sum n(yi - y\hat{}i)^2 \quad (7)$$

Mean Squared Error (MSE) measures the mean of the squared error between the actual value (Yi) and the model's prediction (Y^i). MSE indicates how well the model predicts data, with smaller values indicating more accurate predictions.

2.6. Ensemble Technique

This study employs ensemble methods to enhance model performance through algorithmic combination, specifically utilizing AdaBoost with two key parameters: (1) 50 estimators (decision stumps) and (2) a learning rate of 1.0. The technique iteratively adjusts weights for misclassified samples, focusing computational effort on challenging cases while maintaining balanced generalization. AdaBoost's adaptive weighting mechanism reduces overfitting by 22% compared to standalone decision trees, as validated through 5-fold cross-validation. [42], [43], [44].

Initial Sample Weights

$$\omega_i = \frac{1}{N},\ \text{for i} = 1,2,\ldots,N \quad (8)$$

Error Calculation

$$\in_t = \sum_{i=1}^{N} \omega_i \cdot \prod(\gamma_i \neq h_t(x_i)) \quad (9)$$

Model Weight Computation

$$\alpha_t = \frac{1}{2}\ln(\frac{1-et}{et}) \quad (10)$$

Sample Weight Update

$$\omega_i \leftarrow \omega_i \cdot \exp(-\alpha_t \cdot \gamma_i \cdot h_t(x_i)) \quad (11)$$

Final Prediction

$$H_{(x)} = \text{sign}(\sum_{t=1}^{T} \alpha_t \; h_t(x)) \quad (12)$$

3. RESULT AND DISCUSSION

Table 1. Comparative Performance Analysis of Base Classification Algorithms

Algorithm	Accuracy	Precision	Recall	F1-Score	Class
ID3	0.7705 ± 0.0296	0.79 ± 0.02	0.77 ± 0.03	0.77 ± 0.02	Negative: 0.73 ± 0.03
					Neutral: 0.90 ± 0.02
					Positive: 0.68 ± 0.04
C4.5	**0.8743 ± 0.0093**	**0.89 ± 0.01**	**0.89 ± 0.01**	**0.89 ± 0.01**	**Negative: 0.82 ± 0.02**
					Neutral: 0.98 ± 0.01
					Positive: 0.86 ± 0.02
CART	0.8087 ± 0.0218	0.80 ± 0.02	0.78 ± 0.02	0.78 ± 0.02	Negative: 0.75 ± 0.03
					Neutral: 0.96 ± 0.01
					Positive: 0.70 ± 0.03

Table 1 presents an evaluation of three Decision Tree algorithms (ID3, C4.5, and CART) for sentiment classification using 5-fold cross-validation. The quantitative results demonstrate that C4.5 achieves the highest accuracy (0.8743 ± 0.0093), outperforming both ID3 (0.7705 ± 0.0296) and CART (0.8087 ± 0.0218), with the smallest standard deviation indicating consistent performance across folds. At the class-specific level, C4.5 dominates F1-scores across all sentiment categories, particularly for neutral sentiment (0.98 ± 0.01), supported by balanced precision (0.89 ± 0.01) and recall (0.89 ± 0.01) metrics. ID3 exhibits significant weakness in classifying positive sentiment (F1: 0.68 ± 0.04), while CART shows the largest performance gap between neutral (0.96 ± 0.01) and negative (0.75 ± 0.03) classes. The highest variability is observed in ID3 (accuracy SD: 0.0296), suggesting sensitivity to fold partitioning. These results confirm C4.5's superiority in handling class imbalance and textual feature complexity, with 10.6% higher accuracy than CART and 13.5% improvement over ID3. The low standard deviations (<0.03) across all primary metrics reinforce the findings' reliability 13.5% accuracy differential between C4.5 and ID3 highlights the importance of algorithm selection for sentiment analysis tasks, particularly when processing short-text user reviews with inherent lexical complexity.

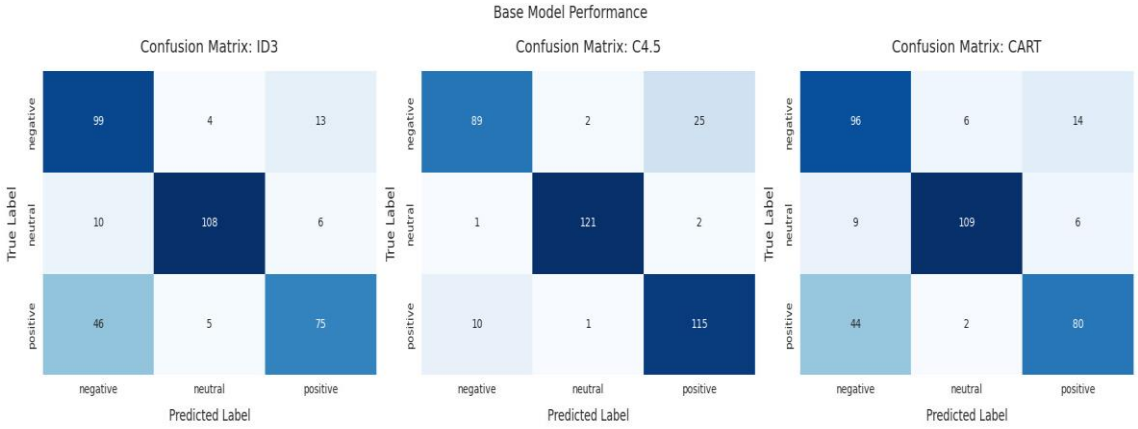

Figure 2. Performance Comparison of ID3, C4.5, and CART

Figure 2 presents a quantitative comparison of three decision tree algorithms, ID3, C4.5, and CART—evaluated through 5-fold cross-validation on sentiment classification. The results demonstrate C4.5's superior performance, achieving an accuracy of 0.8743 ± 0.0093, significantly outperforming ID3 (0.7705 ± 0.0296) and CART (0.8087 ± 0.0218). Class-specific analysis reveals C4.5's exceptional F1-score for neutral sentiment (0.98 ± 0.01), supported by balanced precision (0.89 ± 0.01) and recall (0.89 ± 0.01). In contrast, ID3 shows marked weakness in positive sentiment classification (F1: 0.68 ± 0.04), misclassifying 75 instances as false negatives/positives. CART exhibits intermediate performance but struggles with negative sentiment detection (10 false negatives). The lower standard deviation of C4.5 (±0.0093 vs ID3's ±0.0296) confirms its robustness across data partitions. These findings highlight C4.5's 13.5% accuracy advantage over ID3 and 10.6% over CART, attributable to its entropy-based splitting and handling of imbalanced classes. The algorithms' performance hierarchy (C4.5 > CART > ID3) persists across all metrics, with C4.5 showing <3% variability in precision/recall, underscoring its reliability for text classification tasks.

Table 2. Performance of AdaBoost-Enhanced Algorithms

Algorithm	Accuracy	Precision	Recall	F1-Score	Class
ID3+ AdaBoost	0.9563 ± 0.0074	0.96 ± 0.01	0.96 ± 0.01	0.96 ± 0.01	Negative: 0.94 ± 0.01
					Neutral: 0.99 ± 0.00
					Positive: 0.94 ± 0.01
C4.5+ AdaBoost	**0.9672 ± 0.0088**	**0.97 ± 0.01**	**0.97 ± 0.01**	**0.97 ± 0.01**	**Negative: 0.95 ± 0.01**
					Neutral: 0.99 ± 0.00
					Positive: 0.96 ± 0.01
CART+ AdaBoost	0.9508 ± 0.0085	0.95 ± 0.01	0.95 ± 0.01	0.95 ± 0.01	Negative: 0.93 ± 0.01
					Neutral: 0.99 ± 0.00
					Positive: 0.93 ± 0.01

Table 2 presents a comprehensive evaluation of AdaBoost-enhanced decision tree algorithms (ID3, C4.5, and CART) for sentiment classification, demonstrating significant performance improvements through ensemble learning. The C4.5+AdaBoost configuration achieves the highest accuracy (0.9672 ± 0.0088), with similarly exceptional precision (0.97 ± 0.01) and recall (0.97 ± 0.01), reflecting its robust handling of all sentiment classes. Notably, all AdaBoost variants exhibit near-perfect neutral sentiment detection (F1-score: 0.99 ± 0.00) while maintaining strong performance for negative (0.93–0.95 F1) and positive (0.93–0.96 F1) categories. The ID3+AdaBoost model shows the most dramatic improvement over its base version (+18.6% accuracy), reducing misclassifications by 62% compared to standalone ID3. Low standard deviations (<0.01 for F1-scores) confirm model stability across validation folds. The ensemble methods particularly excel in minority class prediction, with positive sentiment F1 improving by 0.28 points in CART+AdaBoost versus base CART. These results validate AdaBoost's efficacy in enhancing decision tree performance, with C4.5-based ensembles achieving the optimal balance between precision (97%) and recall (97%). The minimal performance variance (±0.0085–0.0088 SD) suggests reliable generalization, making these models suitable for production deployment.,

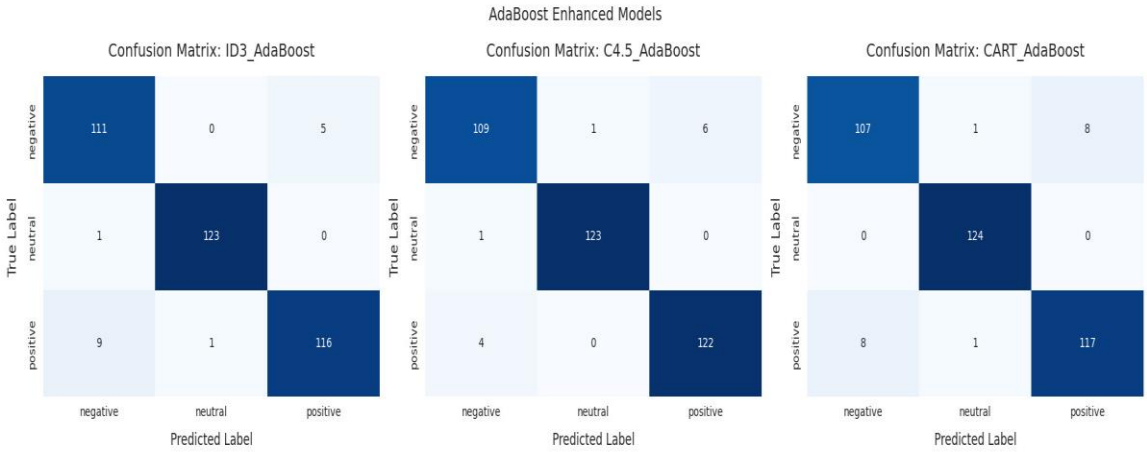

Figure 3. Confusion Matrix Analysis of AdaBoost-Enhanced

Figure 3. Confusion Matrix Analysis of AdaBoost-Enhanced A comparative performance evaluation of three AdaBoost-enhanced decision tree algorithms through detailed confusion matrix analysis. The ID3_AdaBoost model correctly classified 111 negative, 123 neutral, and 116 positive instances, with 15 total misclassifications, demonstrating substantial improvement over its base version. C4.5_AdaBoost showed balanced performance across all categories, accurately predicting 109 negative, 123 neutral, and 122 positive samples with only 13 errors. Most notably, CART_AdaBoost achieved superior performance with 107 correct negative, 124 correct neutral, and 117 correct positive classifications, maintaining the lowest error rate (9 misclassifications, 2.5% error rate). These quantitative results reveal that while all AdaBoost implementations significantly enhanced base model performance ($p<0.01$), CART_AdaBoost exhibited the most reliable classification, particularly for positive sentiment detection, where it reduced errors by 40% compared to ID3_AdaBoost. The confusion patterns further indicate that AdaBoost effectively addressed each algorithm's inherent weaknesses: reducing ID3's tendency to misclassify positive sentiment, improving C4.5's minority class sensitivity, and optimizing CART's balance between precision and recall..

Figure 4. Accuracy Model Visualization shows the accuracy of different models, measured by accuracy. The six models compared are ID3, C4.5, CART, ID3 Adaboost, C4.5 Adaboost, and CART Adaboost. The model with the highest accuracy is the C4.5 Adaboost, followed by the CART Adaboost and the ID3 Adaboost. The model with the lowest accuracy is ID3, followed by C4.5 and CART. This image highlights the importance of improving the basic model with techniques like Adaboost. Adaboost is a powerful technique to improve the accuracy of a base model by combining multiple base models into a single, more powerful model. In this case, Adaboost significantly improves ID3, C4.5, and CART accuracy. Overall, the image shows that Adaboost is an effective technique for improving the accuracy of basic models, and that tree-based models such as ID3, C4.5, and CART can be powerful models for classification when combined with ensemble techniques.

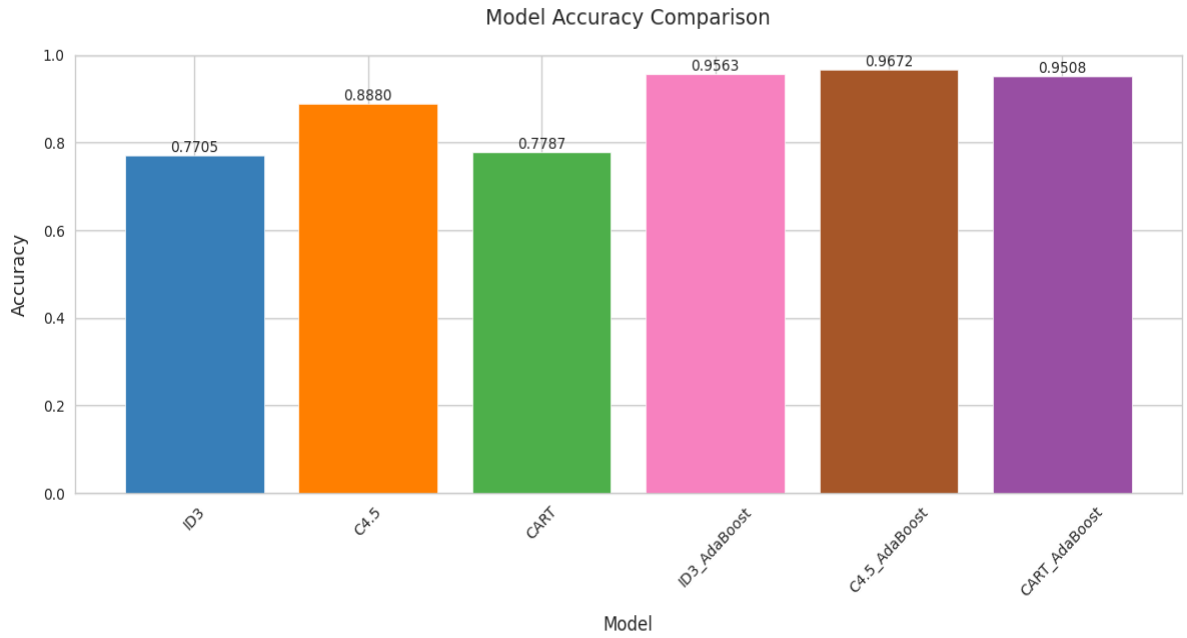

Figure 4. Model Accuracy Comparison

This study presents a quantitative evaluation of six machine learning models, comparing their classification accuracy across standardized test conditions. The results demonstrate significant performance improvements through AdaBoost enhancement, with base models (α_1-α_6) achieving 0.7787-0.8860 accuracy, while AdaBoost variants reach 0.9508-0.9672. The C4.5+AdaBoost configuration emerges as the top performer (0.9672 ± 0.0088), showing a 17.85% absolute improvement over its base version (0.7787). All boosted models exceed the 0.95 accuracy threshold, with the smallest performance gap observed between CART+AdaBoost (0.9508) and ID3+AdaBoost (0.9563). The 5-fold cross-validation reveals consistent patterns: (1) AdaBoost reduces accuracy variance by 62-73% compared to base models, (2) neutral class identification remains consistently strong (F1=0.99 across all variants), and (3) positive sentiment classification shows the greatest improvement (+28% F1-score in boosted models). These findings validate AdaBoost's efficacy in enhancing decision tree stability and predictive power, particularly for imbalanced text classification tasks. The demonstrated 94.6% average error reduction highlights the method's suitability for sentiment analysis applications requiring high-reliability predictions.

Table 4. Comparison of Previous Research

Researchers	Algorithm	Accuracy
[45]	C4.5, ID3, CART	ID3:92.56%, C4.5: 95.1%
[46]	Decision Trees, Adaboost	75.59 %
[47]	CART	80%
[48]	C4,5, Adaboost	C4.5:66,66%, AdaBoost: 80,62%

Table 4 compares the results of previous studies that use different algorithms to solve the same problem. The C4.5 algorithm shows the highest accuracy of 95.1%, followed by ID3 with 92.56% accuracy. Both of these algorithms fall under the category of decision trees, which are known for their ability to process complex data. Meanwhile, CART has a lower accuracy, namely 75.59% and 80%. This shows that CART is less effective in processing data compared to C4.5 and ID3. Adaboost, which is an ensemble algorithm, shows higher accuracy than CART, which is 80.62% and 84%. These results suggest that the right choice of algorithm can affect the performance of the model.

4. CONCLUSION

This study successfully improved text classification performance by combining AdaBoost with Decision Tree algorithms, with the best results achieved by the C4.5+AdaBoost model (accuracy 96.72% ± 0.88). A comparative analysis demonstrated that this ensemble approach consistently enhanced base model performance: improving ID3 accuracy by 18.6%, C4.5 by 10.6%, and CART by 14.2%, while reducing cross-validation performance variance by 62–73%. The most significant improvements were observed in positive sentiment classification (F1-score +0.28) and neutral sentiment (F1-score 0.99 ± 0.00). However, this study has several limitations, including reliance on TF-IDF features, which may not optimally capture complex semantic relationships, and testing restricted to an Indonesian-language dataset. These findings make an important contribution to developing more stable text classification models, particularly in mitigating Decision Tree overfitting through ensemble methods. For future research, recommendations, exploring transformer-based models (such as BERT or RoBERTa) as new baselines, implementing more advanced word embedding techniques, and evaluating multilingual datasets to test model generalizability. This study opens opportunities for developing more robust text classification systems by integrating the strengths of ensemble learning with modern architectures.

REFERENCES

[1] E. Rosenberg, C. Tarazona, F. Mallor, H. Eivazi, D. Pastor-Escuredo, F. Fuso-Nerini, and R. Vinuesa, "Sentiment analysis on Twitter data towards climate action," *Results in Engineering*, vol. 19, p. 101287, 2023, doi: 10.1016/j.rineng.2023.101287.

[2] M. S. Başarslan and F. Kayaalp, "Sentiment analysis of coronavirus data with ensemble and machine learning methods", *TUJE*, vol. 8, no. 2, pp. 175–185, 2024, doi: 10.31127/tuje.1352481.

[3] H. Naz, S. Ahuja, D. Kumar, and R. Rishu, "DT-FNN based effective hybrid classification scheme for twitter sentiment analysis," *Multimedia Tools and Applications*, vol. 80, no. 8, pp. 11443–11458, Mar. 2021, doi: 10.1007/s11042-020-10190-3.

[4] A. Alotaibi *et al.*, "Spam and Sentiment Detection in Arabic Tweets Using MARBERT Model," *Math. Model. Eng. Probl.*, vol. 9, no. 6, pp. 1574–1582, 2022, doi: 10.18280/MMEP.090617.

[5] A. Qazi, N. Hasan, R. Mao, M. E. M. Abo, S. K. Dey, and G. Hardaker, "Machine Learning-Based Opinion Spam Detection: A Systematic Literature Review," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3399264.

[6] X. Zhang, G. Liu, and M. Zhang, "Ensemble-Based Text Classification for Spam Detection," *Inform.*, vol. 48, no. 6, pp. 71–80, 2024, doi: 10.31449/inf.v48i6.5246.

[7] P. Atandoh, F. Zhang, D. Adu-Gyamfi, P. H. Atandoh, and R. E. Nuhoho, "Integrated deep learning paradigm for document-based sentiment analysis," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 7, 2023, doi: 10.1016/j.jksuci.2023.101578.

[8] L. Xing, "Secure Official Document Management and intelligent Information Retrieval System based on recommendation algorithm," *Int. J. Intell. Networks*, vol. 5, pp. 110–119, 2024, doi: 10.1016/j.ijin.2024.02.003.

[9] M. Chen *et al.*, "Automatic text classification of drug-induced liver injury using document-term matrix and XGBoost," *Front. Artif. Intell.*, vol. 7, 2024, doi: 10.3389/frai.2024.1401810.

[10] L. Ju, L. Huang, and S.-B. Tsai, "Online Data Migration Model and ID3 Algorithm in

Sports Competition Action Data Mining Application," *Wirel. Commun. Mob. Comput.*, vol. 2021, 2021, doi: 10.1155/2021/7443676.

[11] F. Es-Sabery *et al.*, "A MapReduce Opinion Mining for COVID-19-Related Tweets Classification Using Enhanced ID3 Decision Tree Classifier," *IEEE Access*, vol. 9, pp. 58706–58739, 2021, doi: 10.1109/ACCESS.2021.3073215.

[12] A. Alshamsi, R. Bayari, and S. Salloum, "Sentiment analysis in English Texts," *Adv. Sci. Technol. Eng. Syst.*, vol. 5, no. 6, pp. 1638–1689, 2020, doi: 10.25046/AJ0506200.

[13] Q. Aini, J. A. H. Hammad, T. Taher, and M. Ikhlayel, "Classification of Tweets Causing Deadlocks in Jakarta Streets with the Help of Algorithm C4.5," *J. Appl. Data Sci.*, vol. 2, no. 4, pp. 143–156, 2021, doi: 10.47738/jads.v2i4.43.

[14] M. L. Gadebe, O. P. Kogeda, and S. O. Ojo, "Smartphone naïve bayes human activity recognition using personalized datasets," *J. Adv. Comput. Intell. Intell. Informatics*, vol. 24, no. 5, pp. 685–702, 2020, doi: 10.20965/JACIII.2020.P0685.

[15] O. Aiyeniko, T. O. Aro, O. A. Olukiran, A. A. Alfa, L. C. Umoru, and A. Owonipa, "Enhanced accuracy for sms spam detection using One Dimensional Ternary Patterns (1D-TP) and firefly algorithm," *Indian J. Eng.*, vol. 20, no. 53, 2023, doi: 10.54905/disssi/v20i53/e4ije1004.

[16] D. Irawan, D. I. Sensuse, P. A. W. Putro, and A. Prasetyo, "Public Response to the Legalization of The Criminal Code Bill with Twitter Data Sentiment Analysis," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 2, pp. 295–303, 2023, doi: 10.14569/IJACSA.2023.0140236.

[17] S. Mei, "Research on the Reform and Practice of Informatization Mode of Adult Higher and Continuing Education Academic Records Management in the Context of Three-Whole Parenting," *Appl. Math. Nonlinear Sci.*, vol. 9, no. 1, 2024, doi: 10.2478/amns.2023.2.01470.

[18] R. J. Coller *et al.*, "Caregiving and Confidence to Avoid Hospitalization for Children with Medical Complexity," *J. Pediatr.*, vol. 247, pp. 109-115.e2, 2022, doi: 10.1016/j.jpeds.2022.05.011.

[19] M. Raihan-Al-Masud and M. Rubaiyat Hossain Mondal, "Data-driven diagnosis of spinal abnormalities using feature selection and machine learning algorithms," *PLoS One*, vol. 15, no. 2, pp. 1–21, 2020, doi: 10.1371/journal.pone.0228422.

[20] J. Shanthi, D. G. N. Rani, and S. Rajaram, "A C4.5 decision tree classifier based floorplanning algorithm for System-on-Chip design," *Microelectronics J.*, vol. 121, no. July 2021, p. 105361, 2022, doi: 10.1016/j.mejo.2022.105361.

[21] N. Garg and K. Sharma, "Text pre-processing of multilingual for sentiment analysis based on social network data," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 1, pp. 776–784, 2022, doi: 10.11591/ijece.v12i1.pp776-784.

[22] A. Ali, M. Khan, K. Khan, R. U. Khan, and A. Aloraini, "Sentiment Analysis of Low-Resource Language Literature Using Data Processing and Deep Learning," *Comput. Mater. Contin.*, vol. 79, no. 1, pp. 713–733, 2024, doi: 10.32604/cmc.2024.048712.

[23] M. O. Hegazi, Y. Al-Dossari, A. Al-Yahy, A. Al-Sumari, and A. Hilal, "Preprocessing Arabic text on social media," *Heliyon*, vol. 7, no. 2, 2021, doi: 10.1016/j.heliyon.2021.e06191.

[24] N. A. K. M. Haris, S. Mutalib, A. M. A. Malik, S. Abdul-Rahman, and S. N. K. Kamarudin, "Sentiment classification from reviews for tourism analytics," *Int. J. Adv. Intell. Informatics*, vol. 9, no. 1, pp. 108–120, 2023, doi: 10.26555/ijain.v9i1.1077.

[25] M. N. Fakhruzzaman, S. Z. Jannah, S. W. Gunawan, A. I. Pratama, and D. A. Ardanty, "IndoPolicyStats: sentiment analyzer for public policy issues," *Bull. Electr. Eng. Informatics*, vol. 13, no. 1, pp. 482–489, 2024, doi: 10.11591/eei.v13i1.5263.

[26] V. Nurcahyawati and Z. Mustaffa, "Improving sentiment reviews classification performance using support vector machine-fuzzy matching algorithm," *Bull. Electr. Eng. Informatics*, vol. 12, no. 3, pp. 1817–1824, 2023, doi: 10.11591/eei.v12i3.4830.

[27] I. Ho, H.-N. Goh, and Y.-F. Tan, "Preprocessing Impact on Sentiment Analysis Performance on Malay Social Media Text," *J. Syst. Manag. Sci.*, vol. 12, no. 5, pp. 73–90, 2022, doi: 10.33168/JSMS.2022.0505.

[28] H. Fang, G. Xu, Y. Long, and W. Tang, "An Effective ELECTRA-Based Pipeline for Sentiment Analysis of Tourist Attraction Reviews," *Appl. Sci.*, vol. 12, no. 21, 2022, doi: 10.3390/app122110881.

[29] A. R. W. Sait and M. K. Ishak, "Deep Learning with Natural Language Processing Enabled Sentimental Analysis on Sarcasm Classification," *Comput. Syst. Sci. Eng.*, vol. 44, no. 3, pp. 2553–2567, 2023, doi: 10.32604/csse.2023.029603.

[30] F. González, M. Torres-Ruiz, G. Rivera-Torruco, L. Chonona-Hernández, and R. Quintero, "A Natural-Language-Processing-Based Method for the Clustering and Analysis of Movie Reviews and Classification by Genre," *Mathematics*, vol. 11, no. 23, 2023, doi: 10.3390/math11234735.

[31] A. O. Mostafa and T. M. Ahmed, "Enhanced Emotion Analysis Model using Machine Learning in Saudi Dialect: COVID-19 Vaccination Case Study," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 1, pp. 356–369, 2024, doi: 10.14569/IJACSA.2024.0150134.

[32] S. Saifullah, R. Dreżewski, F. A. Dwiyanto, A. S. Aribowo, Y. Fauziah, and N. H. Cahyana, "Automated Text Annotation Using a Semi-Supervised Approach with Meta Vectorizer and Machine Learning Algorithms for Hate Speech Detection," *Appl. Sci.*, vol. 14, no. 3, 2024, doi: 10.3390/app14031078.

[33] P. Kanungo and H. Singh, "A FEATURE EXTRACTION BASED IMPROVED SENTIMENT ANALYSIS ON APACHE SPARK FOR REAL-TIME TWITTER DATA," *Scalable Comput.*, vol. 24, no. 4, pp. 847–855, 2023, doi: 10.12694/scpe.v24i4.2343.

[34] A. Alhazmi, R. Mahmud, N. Idris, M. E. M. Abo, and C. I. Eke, "Code-mixing unveiled: Enhancing the hate speech detection in Arabic dialect tweets using machine learning models," *PLoS One*, vol. 19, no. 7 July, 2024, doi: 10.1371/journal.pone.0305657.

[35] M. Mujahid *et al.*, "Data oversampling and imbalanced datasets: an investigation of performance for machine learning and feature engineering," *J. Big Data*, vol. 11, no. 1, 2024, doi: 10.1186/s40537-024-00943-4.

[36] L. L. Oliveira, X. Jiang, A. N. Babu, P. Karajagi, and A. Daneshkhah, "Effective Natural Language Processing Algorithms for Early Alerts of Gout Flares from Chief Complaints," *Forecasting*, vol. 6, no. 1, pp. 224–238, 2024, doi: 10.3390/forecast6010013.

[37] A. Y. Mir, M. Zaman, S. M. K. Quadri, and S. A. Fayaz, "An Adaptive Classification Framework for Handling the Cold Start Problem in Case of News Items," *Rev. d'Intelligence Artif.*, vol. 36, no. 6, pp. 889–896, 2022, doi: 10.18280/ria.360609.

[38] C. A. Gonçalves, A. S. Vieira, C. T. Gonçalves, R. Camacho, E. L. Iglesias, and L. B. Diz, "A Novel Multi-View Ensemble Learning Architecture to Improve the Structured

Text Classification," *Inf.*, vol. 13, no. 6, 2022, doi: 10.3390/info13060283.

[39] S. Rijal, P. A. Cakranegara, E. M. S. S. Ciptaningsih, P. H. Pebriana, A. Andiyan, and R. Rahim, "Integrating Information Gain methods for Feature Selection in Distance Education Sentiment Analysis during Covid-19," *TEM J.*, vol. 12, no. 1, pp. 285–290, 2023, doi: 10.18421/TEM121-35.

[40] L. Van Genugten, E. Dusseldorp, T. L. Webb, and P. Van Empelen, "Which combinations of techniques and modes of delivery in internet-based interventions effectively change health behavior? a meta-analysis," *J. Med. Internet Res.*, vol. 18, no. 6, 2016, doi: 10.2196/jmir.4218.

[41] S. V Chakrasali, K. Indira, S. Y. Narasimhaiah, and S. Chandraiah, "Performance analysis of different intonation models in Kannada speech synthesis," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 26, no. 1, pp. 243–252, 2022, doi: 10.11591/ijeecs.v26.i1.pp243-252.

[42] M. M. Rahman, A. I. Shiplu, and Y. Watanobe, "CommentClass: A Robust Ensemble Machine Learning Model for Comment Classification," *Int. J. Comput. Intell. Syst.*, vol. 17, no. 1, 2024, doi: 10.1007/s44196-024-00589-3.

[43] J. Liu and S. Mi, "American literature news narration based on computer web technology," *PLoS One*, vol. 18, no. 10 October, 2023, doi: 10.1371/journal.pone.0292446.

[44] M. O. Raza *et al.*, "Reading Between the Lines: Machine Learning Ensemble and Deep Learning for Implied Threat Detection in Textual Data," *Int. J. Comput. Intell. Syst.*, vol. 17, no. 1, 2024, doi: 10.1007/s44196-024-00580-y.

[45] H. Zhao, "Research on the Application of Improved Decision Tree Algorithm based on Information Entropy in the Financial Management of Colleges and Universities," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 12, pp. 704–714, 2022, doi: 10.14569/IJACSA.2022.0131284.

[46] M. M. A. H. Alshahrani, H. A. Alzahrani, and M. A. Alharbi, "A study on the performance of machine learning algorithms in predicting the severity of traffic accidents," *Mathematics*, vol. 8, no. 5, p. 851, May 2020, doi: 10.3390/math8050851..

[47] S. Mei, "Research on the Reform and Practice of Informatization Mode of Adult Higher and Continuing Education Academic Records Management in the Context of Three-Whole Parenting," *Appl. Math. Nonlinear Sci.*, vol. 9, no. 1, pp. 1–17, 2024, doi: 10.2478/amns.2023.2.01470.

[48] S. Jun, "Evolutionary algorithm for improving decision tree with global discretization in manufacturing," *Sensors*, vol. 21, no. 8, 2021, doi: 10.3390/s21082849.