Journal of Renewable Energy, Electrical, and Computer Engineering
Volume 5, Number 1, March 2025. 9-17
eISSN 2776-0049
Research Original Article
DOI: https://doi.org/10.29103/jreece.v5i1.21227

Comparison of Random Forest Algorithm Classifier and Naïve Bayes Algorithm in Whatsapp Message Type Classification

Abdul Hadi✉[1], Mukti Qamal[2] , Yesy Afrillia[3]
[1]Informatics Engineering, Faculty of Engineering, Universitas Malikussaleh, Bukit Indah, Lhokseumawe, 24353, Indonesia, abdul.200170265@mhs.unimal.ac.id
[2]Informatics Engineering, Faculty of Engineering, Universitas Malikussaleh, Bukit Indah, Lhokseumawe, 24353, Indonesia, mukti.qamal@unimal.ac.id
[3]Informatics Engineering, Faculty of Engineering, Universitas Malikussaleh, Bukit Indah, Lhokseumawe, 24353, Indonesia, yesy.afrillia@unimal.ac.id
✉Corresponding Author: **abdul.200170265@mhs.unimal.ac.id | Phone: +6282215256067**

Received: December 13, 2024 **Revision: January 15, 2025** **Accepted: March 10, 2025**

Abstract

This study compares the effectiveness of Random Forest and Naïve Bayes algorithms in classifying WhatsApp messages into three categories: normal, promotional, and fraudulent messages. With over 2.78 billion active users worldwide and 90% of Indonesian internet users utilizing WhatsApp, the platform's end-to-end encryption creates challenges for automatic spam detection, necessitating machine learning approaches. A dataset of 300 messages, equally distributed across the three categories, underwent preprocessing including cleansing, case folding, stopword removal, normalization, and stemming before being converted to numerical form using TF-IDF vectorization. Experimental results demonstrated that Naïve Bayes outperformed Random Forest with higher accuracy (88.67% vs. 86.00%), precision (89.64% vs. 88.95%), recall (88.67% vs. 86.00%), and F1-score (88.61% vs. 85.99%). Cross-validation analysis with 10-fold validation further confirmed Naïve Bayes' superior consistency and stability across all evaluation metrics. Additionally, Naïve Bayes exhibited remarkable computational efficiency, requiring only 0.13 seconds for training compared to Random Forest's 3.65 seconds. Confusion matrix analysis revealed Naïve Bayes' particular effectiveness in distinguishing between normal and fraudulent messages, crucial for preventing users from falling victim to scams. The model successfully identified key fraud indicators such as "claim," "account," and "verification" while demonstrating precision in ambiguous cases. These findings contribute significantly to developing more effective spam detection systems for encrypted messaging platforms where traditional filtering mechanisms cannot be applied, ultimately enhancing user safety and experience through automated identification of potentially harmful content.

Keywords: Whatsapp Classification, Message Classification, Naïve Bayes, Random Forest, Text Mining

Introduction

The rapid development of information technology has changed the way people communicate, especially through instant messaging applications. WhatsApp has become one of the most popular communication platforms, with more than 2.78 billion active users worldwide and more than 90% of internet users in Indonesia utilising it as their primary communication medium (AlAfnan & Awad, 2024). The app offers various communication features, such as text messaging, voice and video calls, document sending, and groups with up to 1,024 members (Johns et al., 2023). In addition, the broadcast message and forwarding features allow users to spread information quickly and widely. However, this convenience also opens a gap for the spread of spam and hoax messages, which can annoy users and even pose a cybersecurity risk (Yanto, 2021).

WhatsApp has implemented several measures to reduce the spread of malicious messages, such as limiting message forwarding to only five contacts and a spam account reporting system (Sapitri et al., 2023). However, these efforts are still not fully effective in addressing the surge in spam messages, which often take the form of aggressive promotions, malicious links, and fraud modes that can harm users financially. Unlike SMS, which can still be filtered by mobile operators, WhatsApp uses end-to-end encryption, which while enhancing privacy, also makes it difficult to automatically detect malicious messages (Hasanah et al., 2023).

Spam message detection and classification is a crucial aspect in improving user safety and convenience. Machine learning methods, such as Naïve Bayes and Support Vector Machine (SVM), have been widely used in text classification, including spam detection on various communication platforms. Naïve Bayes is known as an efficient and fast probabilistic algorithm, while SVM excels in handling high-dimensional data and generating optimal decision boundaries (Dwiyansaputra et al., 2021).

Several previous studies have shown that Naïve Bayes performs well in spam classification, although SVMs also provide competitive results. However, with the increasing complexity of message patterns and variations in content, further studies are needed to compare these two algorithms in WhatsApp message classification. This study aims to analyse and compare the performance of Naïve Bayes and SVM in WhatsApp message type classification, by evaluating accuracy, precision, recall, and F1-score. The results of this study are expected to provide insight into algorithms that are more effective in handling WhatsApp message classification, as well as contribute to the development of a more accurate and reliable spam detection system (Herwanto et al., 2021).

Literature Review

Text Classification

Text classification is an important part of natural language processing (NLP) that aims to categorise text documents into one or more classes based on their content. This process involves the automatic identification of the category that best fits the given text through linguistic content analysis (Lavanya & Sasikala, 2021). In the context of WhatsApp messages, text classification allows the system to distinguish between normal, promotional, and fraudulent messages based on the linguistic features contained in the message.

Text Preprocessing

Text preprocessing is the initial stage in the text mining process that focuses on cleaning data from noise, so that the data becomes more structured and concise (Gaur et al., 2023). There are several general stages in the text preprocessing process as follows:

1. Cleansing, This process may involve removing punctuation marks, numbers, non-ASCII special characters, URLs, as well as reducing excessive use of spaces (Samad et al., 2020).
2. Case Folding, Convert all letters to lowercase to standardise the text and reduce the feature dimension (Naseem et al., 2021).
3. Stopword Removal, Eliminating common words that appear repeatedly in language such as 'and', 'which', 'in', which usually do not carry significant information for classification (Kerner et al., 2020).
4. Stemming, Stemming involves reducing words to their base form by removing affixes, while lemmatisation involves transforming words to the base form present in the dictionary (Abidin & Junaidi, 2024).
5. Normalization, Normalization is the process of converting colloquial words or abbreviations into standard words according to KBBI (Big Indonesian Dictionary) (Mutiara et al., 2020).

Feature Extraction with TF-IDF (Term Frequency – Invers Document Frequency)

Feature extraction is a crucial stage in text classification that aims to transform raw text data into numerical representations that can be processed by machine learning algorithms. This process allows the algorithm to identify patterns and relationships in the text that are relevant for the classification task (Wang et al., 2020).

TF-IDF is a technique that gives weight to words or terms to determine their relevance to documents (Jalilifard et al., 2021). This method calculates the TF and IDF values for each word. The TF value will increase along with the frequency of occurrence of the word in the document. Meanwhile, the IDF value reflects how rarely a word appears throughout the document-the rarer the occurrence, the higher the IDF value (Sihombing et al., 2024).

Random Forest Algorithm

Random Forest is an ensemble algorithm that consists of many decision trees and combines the results of all trees to produce predictions. It overcomes the overfitting problem that often occurs with a single decision tree by training the tree on different subsets of data and features (Khan et al., 2020). Random Forest has the advantage of handling high-dimensional data and can provide information about feature importance that is useful for analysis (Quist et al., 2021).

$$y = mode(\{h_1\{x\},\{h_2\{x\} \ldots \{h_n\{x\}\}) \qquad (1)$$

Naïve Bayes Algorithm

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem assuming independence between features. Although this assumption is often unrealistic, Naive Bayes remains effective in text classification due to its simplicity, efficiency, and ability to work with small datasets (Mansoori et al., 2024). For text classification, a frequently used variant is Multinomial Naive Bayes which takes into account the frequency of occurrence of words in documents (Rezaeian & Novikova, 2020).

$$P(A|B) = \frac{P(B|A)\,.\,P(A)}{P(B)} \qquad (2)$$

Confussion Matrix

Confusion matrix is a matrix-shaped method used to measure the number of correct classifications in a particular class, taking into account the algorithm used (Qadrini et al., 2021). This matrix serves as a tool to evaluate the performance of classification models and provide a summary of the prediction results on a dataset (Setiyana, 2021). The confusion matrix consists of four main components: True Positive (TP), when the model accurately predicts a positive instance as positive; True Negative (TN), when the model successfully predicts a negative instance as negative; False Positive (FP), when the model incorrectly predicts a negative instance as positive; and False Negative (FN), when the model incorrectly predicts a positive instance as negative (Normawati & Prayogi, 2021). To evaluate the performance of classification algorithms, several common metrics are used:

1. Accuracy, Proportion of correct predictions out of total predictions.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (3)$$

2. Precision, The proportion of correct positive predictions out of total positive predictions, measures how precise the algorithm is in identifying positive classes.

$$Precision = \frac{TP}{TP+FP} \qquad (4)$$

3. Recall, The proportion of positive cases identified measures how complete the algorithm is in identifying the positive class.

$$Recall = \frac{TP}{TP+FN} \qquad (5)$$

4. F1-Score, The harmonic mean of precision and recall, provides a balance between the two metrics.

$$F1 - Score = 2 \times \frac{Precission \times Recall}{Precission + Recall} \qquad (6)$$

Previous Research

Previous research has investigated text classification using various machine learning algorithms (Putera et al., 2023) conducted a study on SMS spam classification using the K-Nearest Neighbor (K-NN) algorithm. The research aimed to minimize fraud cases by classifying SMS messages into three categories: normal, promotional, and fraudulent. The dataset consisted of 50 randomly selected messages, which underwent preprocessing and feature weighting using TF-IDF and Cosine Similarity before classification with K-NN. Another study by (Devita et al., 2018) compared the performance of Naïve Bayes and K-NN for classifying Indonesian-language articles. Using article data from journa2um.ac.id, the study applied preprocessing and feature weighting techniques before classification. The results showed that Naïve Bayes outperformed K-NN in terms of accuracy. Unlike these studies, the current research focuses on comparing the Naïve Bayes algorithm with the Random Forest Classifier in a different case study, aiming to determine which algorithm achieves higher accuracy for sentiment analysis.

Materials & Methods

This section describes the methods used in the research, including the data collection process, preprocessing stage, algorithm implementation, and model evaluation. This research uses WhatsApp messages as the dataset, which is collected and processed through several stages before being applied in the machine learning model. Each step is described systematically to ensure replicability and validity of the research. The following process diagram illustrates the flow of steps performed in this research.

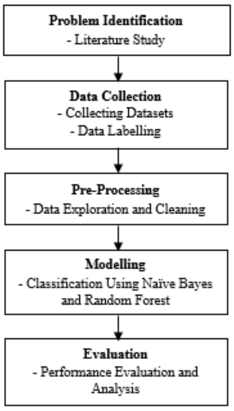

Figure 1. Schematic of Research

The dataset used in this research is in the form of WhatsApp messages collected manually through scraping from various sources. The dataset consists of 300 messages that have been categorised into three main classes: normal messages, promo messages, and scam messages, each with 100 data. This class distribution is done to ensure balance in the classification process. The data collection process was conducted with ethical aspects of the research in mind. All messages have been anonymised to protect the privacy of senders and recipients, by removing sensitive information such as names, phone numbers and personal links.

Before being used in modelling, the data goes through a preprocessing stage to improve quality and reduce noise. This process starts with cleaning, which removes irrelevant elements such as emojis, URLs, and special characters. Next, the text is broken down into words through tokenisation, followed by stopword removal to remove common words that do not contribute to the classification. To make it more uniform, stemming or lemmatisation is performed to convert words to their base form. After that, the text features are converted to numerical format using the Term Frequency-

Inverse Document Frequency (TF-IDF) method, which represents the weight of words in the document.

Two machine learning algorithms are applied in this study, namely Multinomial Naïve Bayes (MNB) and Random Forest Classifier. MNB was chosen for its effectiveness in probability-based text classification, while Random Forest was used as a more complex ensemble learning method with a combination of multiple decision trees. The model was developed using the Scikit-learn library, with parameters optimised through validation to improve performance.

Model evaluation is performed using accuracy, precision, recall, and F1-score metrics to measure prediction effectiveness. In addition, the confusion matrix was used to analyse the distribution of prediction errors. To improve the reliability of the results, the k-fold cross-validation method with k = 10 was applied. Statistical significance testing was also conducted to evaluate the difference in performance of the two algorithms in the classification of WhatsApp messages.

Results and Discussion

In this chapter, the stages involved in the sentiment analysis process are explained comprehensively, starting from data preparation to the application of the classification model. The first stage is the **dataset overview**, which provides an understanding of the data source, dataset structure, and the distribution of sentiments within the data. Next, **data preprocessing** is carried out, which involves a series of text cleaning and transformation processes to make the data more suitable for processing by the model. Once the data is processed, it is **converted into numerical form using TF-IDF Vectorizer**, allowing the text to be represented as vectors. The subsequent step is **building the classification model**, where the algorithms used in this study—Naïve Bayes and Support Vector Machine (SVM)—are applied to perform sentiment classification. After the model is built, **model performance evaluation** is conducted using evaluation metrics such as accuracy, precision, recall, and f1-score. Finally, **result analysis** is performed to understand the model's performance and interpret the classification results obtained.

Dataset Overview

The dataset used in this study consists of WhatsApp messages categorized into three main classes: normal messages, promo messages, and scam messages. The data was collected manually through scraping from various sources, such as community groups, promotional messages from businesses, and messages suspected of containing fraud or spam. This dataset comprises 300 messages, with an equal number of data points in each category to ensure the model is not biased toward any single class. The following is the distribution of messages in the dataset :

Table 1. Sample of Datasets

Whatsapp Message	Category	Amount of Data
“Coba siapa yg lagi di prodi? Punten liatin jadwal sidang. Ada tulisan suruh kumpul jam brp gitu gak? Nuhun”	Normal	100 Data
“MEGA ELEKTRONIK SALE LED TV 32" cuma 1,5jt Kulkas 2 pintu 2,8jt Mesin cuci 10kg 3,5jt * khusus member ELEKTRONIK JAYA	Promotion	100 data
“Selamat! Anda mendapatkan bonus saldo GoPay senilai 500rb. Segera klaim sebelum kedaluwarsa: [bit.ly/gopaybonus]”	Fraud	100 Data

Normal Messages are everyday messages such as personal chats or group discussions, containing greetings, reminders, or coordination of activities. Promo Messages contain product/service promotions, often with links to business sites. Whereas Scam Messages are suspicious, containing scams such as false prize claims, blocking threats, or requests for personal information, and should be watched out for.

Data Preprocessing

In the data preprocessing stage, a series of steps are carried out to clean and prepare the text before it is used for modeling. The first step is case folding, which involves converting all letters in the text to lowercase to eliminate differences between the same words due to capitalization variations. Following this, tokenization is performed, breaking the text into individual words for further processing.

Next, stopword removal is conducted, which involves eliminating common words that do not carry significant meaning for classification, such as "dan" (and), "di" (in), "ke" (to), "yang" (that), and others. This step aims to reduce words that do not provide important information for the model. After stopwords are removed, the next step is stemming, which reduces words to their base forms. Stemming is used to minimize variations of words with the same meaning, such as converting "mendapatkan" (to get) to its root form "dapat" (can).

Additionally, preprocessing also includes the removal of URLs, emojis, and special characters. URLs, which often appear in messages, such as promotional or phishing links, are removed because they do not provide meaningful information for classification. Emojis and other special characters are also eliminated to ensure the model focuses solely on the main text.

Table 2. Preprocessing Proccess

Preprocessing Proccess	Word Before Preprocessing	Word After Preprocessing
Cleansing & Casefolding	“「Bjir!」 gw nemu video lo di twitter 🔞 Kok bisa nyebar gini ya? Ini linknya: bit.ly/vidtwt22c 30rb views udah padahal baru upload td pagi 😱”	“bjir gw nemu video lo di twitter kok bisa nyebar gini ya ini linknya views udah padahal baru upload td pagi”
Normalization	“bjir gw nemu video lo di twitter kok bisa nyebar gini ya ini linknya views udah padahal baru upload td pagi”	"bjir saya menemukan video kamu di twitter kok bisa menyebar begini ya ini linknya 30000 tayangan sudah padahal baru unggah tadi pagi"
Stopword	"bjir saya menemukan video kamu di twitter kok bisa menyebar begini ya ini linknya 30000 tayangan sudah padahal baru unggah tadi pagi"	menemukan video twitter menyebar linknya 30000 tayangan unggah pagi
Stemming	menemukan video twitter menyebar linknya 30000 tayangan unggah pagi	temu video twitter sebar link tayang unggah pagi

TF-IDF (Term Frequency - Inverse Document Frequency)

After the preprocessing stage is completed, the text data needs to be converted into numerical form so that it can be used as input for the classification model. In this study, the **TF-IDF (Term Frequency - Inverse Document Frequency)** technique is used to represent the text as numerical vectors based on the weight of words in the document. TF-IDF assigns higher values to words that appear frequently in a specific document but rarely in the overall dataset, thereby reflecting words that are significant in distinguishing sentiment classes.

The conversion process is carried out by applying the **TF-IDF Vectorizer** to the processed data. Each document in the dataset is represented as a feature vector with dimensions equal to the number of unique words in the entire corpus. The weight of each word is calculated based on its frequency in a document (Term Frequency) and its presence in other documents (Inverse Document Frequency). Once applied, the result of this process is a **sparse matrix** with dimensions (number of documents × number of word features) containing the TF-IDF weights for each word in the dataset. This matrix is then used as input for the classification model. Here are some words in the dataset with the highest value:

Table 3. TF-IDF Results

Word	TF-IDF Score
“diskon”	10.075570
“klaim”	9.824948
“akun”	9.638531
“hindar”	8.676186
“aman”	8.137318

Modelling

After the text data is converted into numerical representation using the TF-IDF method, the next stage in this study is to build and evaluate classification models to analyze the sentiment of WhatsApp messages. Two machine learning algorithms used in this study are Naïve Bayes and Random Forest. The selection of these two models is based on their characteristics in handling text data. Naïve Bayes, as a probabilistic model, is often used in text classification due to its ability to handle data with large and highly sparse features. Meanwhile, Random Forest, as an ensemble-based model, excels in addressing overfitting issues and can capture complex relationships between features in the data.

Before the model training process begins, the dataset that has undergone preprocessing is divided into two subsets: a training set and a testing set, with proportions of 80% for training and 20% for testing, respectively. This division is performed using stratification, ensuring that the class distribution in both subsets remains balanced.

Random Forest

Random Forest is an ensemble-based model composed of a collection of decision trees that work collectively to improve prediction accuracy. This algorithm builds multiple decision trees on random subsets of the training data and combines their results to produce a final prediction based on a majority voting mechanism. The key steps in implementing Random Forest for WhatsApp message classification are:

1. The dataset is trained by constructing multiple decision trees using various subsets of data and features.
2. Each tree makes a prediction regarding the sentiment of the tested message.
3. The final result is determined based on the majority vote from all the trees.

Random Forest excels in handling overfitting, as combining multiple decision trees makes the model more stable and less reliant on any single subset of data.

Naïve Bayes

Naïve Bayes is a probabilistic model that assumes independence between features in the data. In this study, Multinomial Naïve Bayes is used, which is commonly employed in text classification due to its ability to handle the distribution of words in documents effectively. The working process of this model can be explained as follows:

1. The model calculates the probability of word occurrences within each sentiment category.
2. Each WhatsApp message is evaluated based on the probabilities of the words it contains.
3. The model then determines the sentiment category based on the highest probability value.

Naïve Bayes has advantages in terms of computational speed and effectiveness on datasets with a large number of features, such as text data that has undergone the TF-IDF process.

Model Parameter

To achieve optimal performance, several important parameters of both models were adjusted. For Multinomial Naïve Bayes, the _alpha_ parameter (Laplace smoothing) was set to 1.0 to avoid zero probabilities for infrequently occurring words. Meanwhile, for Random Forest, the number of trees (_n_estimators_) was set to 100, _max depth_ was set to _None_ to allow the model to build decision trees without depth limitations, and the _criterion_ was set to _Gini Impurity_ to determine the best split at each tree node. Model training was conducted using the scikit-learn library, with computational time compared to evaluate the differences in efficiency between the two algorithms.

Evaluation

This evaluation aims to determine the effectiveness of the models in classifying WhatsApp message sentiment based on various metrics, such as accuracy, precision, recall, and F1-score. The performance results of the two models compared in this study are presented in the following table:

Table 4. Performance of the Algorithm with Mean and STD

Metrics	Naive Bayes (Mean ± Std)	Random Forest (Mean ± Std)
Accuracy	88.67% ± 4.76%	86.00% ± 4.90%
Precision	89.64% ± 4.70%	88.95% ± 3.66%
Recall	88.67% ± 4.76%	86.00% ± 4.90%
F1-Score	88.61% ± 4.83%	85.99% ± 4.76%

The test results indicate that the Naïve Bayes model performs better than Random Forest in classifying WhatsApp message sentiment. Naïve Bayes achieves an average accuracy of 88.67%, meaning the model correctly classifies data 88.67% of the time, with a variation of approximately ±4.76%. Meanwhile, Random Forest achieves an average accuracy of 86.00% ±4.90%, indicating slightly less stable performance.

In other metrics, Naïve Bayes records a precision of 89.64% ±4.70%, meaning 89.64% of all positive predictions are correct, with minimal variation in results. Its recall reaches 88.67% ±4.76%, indicating the model can correctly identify 88.67% of positive data, while the F1-score is 88.61% ±4.83%, reflecting a balance between precision and recall.

Random Forest has a precision of 88.95% ±3.66%, which is more stable but not significantly different from Naïve Bayes. However, its recall is lower at 86.00% ±4.90%, meaning the model is less capable of identifying all positive data. The F1-score of 85.99% ±4.76% is also lower compared to Naïve Bayes. Overall, Naïve Bayes excels in accuracy and precision-recall balance, while Random Forest demonstrates lower and less stable performance in sentiment classification.

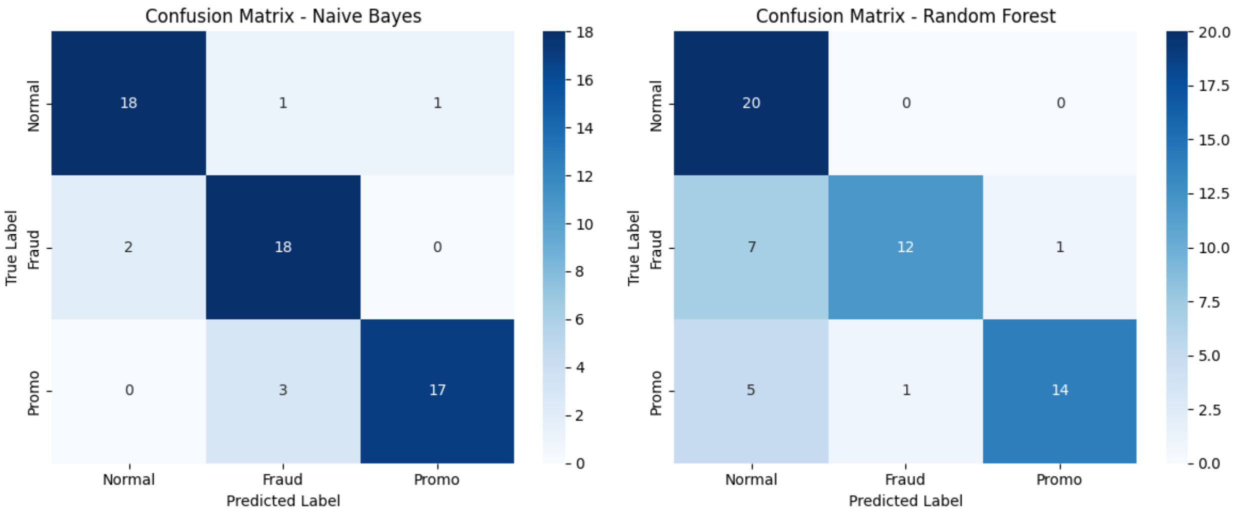

Figure 1. Confussion Matrix

A more detailed analysis of the classification report reveals that Naïve Bayes excels in detecting neutral and negative classes, achieving a precision of 94% for the negative class. In contrast, Random Forest struggles to distinguish between neutral and negative classes, with lower recall, particularly for the neutral class (60%) and the negative class (70%).

Furthermore, the confusion matrix analysis highlights that Naïve Bayes is more consistent in predicting positive and negative classes, with fewer misclassifications compared to Random Forest. From the confusion matrix, it is evident that Naïve Bayes only misclassifies a few neutral samples as negative, whereas Random Forest has a higher rate of misclassification, especially in predicting neutral samples as negative.

Naïve Bayes requires 0.13 seconds for training and 0.00 seconds for prediction, making it very fast. In contrast, Random Forest takes 3.65 seconds for training and 0.10 seconds for prediction, indicating this model is more complex in data processing.

Comparison of Methods with Cross-Validation

Based on the test results, Naïve Bayes demonstrates better performance compared to Random Forest. The model achieves an average accuracy of 88.67%, which is higher than Random Forest's average accuracy of 86.00%. Additionally, Naïve Bayes also excels in precision (89.64%), recall (88.67%), and F1-score (88.61%), while Random Forest records precision (88.95%), recall (86.00%), and F1-score (85.99%). A concise comparison of the evaluation results can be seen in the following table:

Table 5. Performance Results on Each Fold

Fold	Accuracy (NB)	Precision (NB)	Recall (NB)	F1-Score (NB)	Accuracy (RF)	Precision (RF)	Recall (RF)	F1-Score (RF)
1	93.33%	93.81%	93.33%	93.39%	80.00%	83.96%	80.00%	79.41%
2	86.67%	89.11%	86.67%	86.94%	80.00%	86.53%	80.00%	80.79%
3	86.67%	89.36%	86.67%	86.31%	96.67%	96.89%	96.67%	96.60%
4	90.00%	91.33%	90.00%	90.04%	83.33%	88.54%	83.33%	83.61%
5	86.67%	86.75%	86.67%	86.23%	90.00%	91.89%	90.00%	89.65%
6	90.00%	90.11%	90.00%	89.89%	86.67%	88.25%	86.67%	86.58%
7	83.33%	83.74%	83.33%	83.39%	90.00%	91.40%	90.00%	89.59%
8	96.67%	97.00%	96.67%	96.68%	86.67%	88.21%	86.67%	86.48%
9	93.33%	94.44%	93.33%	93.45%	83.33%	89.74%	83.33%	83.90%
10	80.00%	80.71%	80.00%	79.81%	83.33%	84.08%	83.33%	83.33%

From the table, it can be concluded that Naïve Bayes has more consistent performance compared to Random Forest. Additionally, the smaller standard deviation in the evaluation metrics of Naïve Bayes indicates that this model is more stable across folds compared to Random Forest, which shows higher variability in results across certain folds.

When examining the results per fold, Naïve Bayes maintains more stable accuracy, with accuracy ranging between 80.00% and 96.67%, while Random Forest exhibits greater fluctuations, with accuracy ranging between 80.00% and 96.67%. This indicates that Naïve Bayes is more reliable across various testing scenarios.

Classification results of whatsapp messages

Below are the classification results from both algorithms, Naïve Bayes and Random Forest, based on several data samples. This table displays the original text, the prediction results using each algorithm, and the actual category:

Table 6. Classification Results (Sample)

Text	Actual Label	Predicted Label (NB)	Predicted Label (RF)
"Selamat! Anda terpilih sebagai pemenang undian berhadiah mobil dari PT. Sejahtera Abadi. Segera klaim hadiah Anda dengan menghubungi 0812-6745-3209 sebelum 24 jam. Jangan lewatkan kesempatan ini! 🎉 🚗 "	Fraud	Fraud	Fraud
“Shopee Pay: Akun anda terindikasi pelanggaran kebijakan. Verifikasi diperlukan. Login: shopee-secure02.my.id atau saldo akan ditarik kembali”	Fraud	Fraud	Fraud
"DANA x MCDELIVERY: Bayar McD pake DANA diskon 35rb. No min purchase. Berlaku 1x per akun. Sampai 5 Maret"	Promotion	Promotion	Promotion

"Pesan GoFood min. 100rb, diskon 50rb. KODE: GOFOODHEMAT. Berlaku sampai jam 17.00 ini"	Promotion	Promotion	Promotion
"Yaudah gausah babakaran atuh, yg pake kompor aja. Lagian ribet kan nyalain arengnya"	Normal	Normal	Normal
"Kalo kata raditya dika tuh cara pandang org ttg cinta akan berubah setelah mengalami patah hati terhebat. Haha"	Normal	Normal	Normal

The table shows the classification results of WhatsApp messages into three categories: Fraud, Promotion, and Normal. The data in the table includes the original message text, the actual labels, and the prediction results using the Naïve Bayes (NB) and Random Forest (RF) algorithms. From the displayed results, both algorithms are able to classify the messages effectively, with the predicted labels by NB and RF matching the actual labels. For example, messages containing signs of fraud (Fraud), such as fake prize giveaways and ShopeePay account verification, were successfully detected as Fraud by both algorithms. Similarly, promotional messages from food delivery services were classified as Promotion, and casual conversation messages were categorized as Normal. These results indicate that both models perform quite well in grouping messages based on their content and purpose.

Conclusions

This study has successfully evaluated and compared the performance of Naive Bayes and Random Forest algorithms in classifying WhatsApp messages into three categories: normal, promotional, and fraud messages. The experimental results demonstrate that the Naive Bayes algorithm outperforms Random Forest across all evaluation metrics with an average accuracy of 88.67% compared to Random Forest's 86.00%. The Naive Bayes model also excels in precision (89.64%), recall (88.67%), and F1-score (88.61%), indicating its superior ability to correctly identify and categorize WhatsApp messages.

The cross-validation analysis further confirms the consistency and stability of Naive Bayes, as evidenced by smaller standard deviations in performance metrics across all folds. This consistency is particularly valuable in real-world applications where reliable performance is essential. Additionally, the Naive Bayes algorithm demonstrates significant computational efficiency, requiring only 0.13 seconds for training compared to Random Forest's 3.65 seconds, making it more suitable for deployment in resource-constrained environments or applications requiring real-time message classification.

The confusion matrix analysis reveals that Naive Bayes is particularly effective in distinguishing between normal and fraud messages, which is crucial for preventing users from falling victim to scams or phishing attempts. Both algorithms successfully classified obvious fraud patterns containing keywords like "claim," "account," and "verification," but Naive Bayes showed greater precision in ambiguous cases.

These findings contribute to the development of more effective spam detection systems for encrypted messaging platforms like WhatsApp, where traditional filtering mechanisms cannot be applied due to end-to-end encryption. The implementation of Naive Bayes-based classification models could significantly enhance user safety and experience by automatically identifying potentially harmful messages. Future research should focus on expanding the dataset with more diverse message patterns, incorporating more features such as message length and structural characteristics, and exploring hybrid approaches that combine the strengths of both algorithms to further improve classification performance.

Acknowledgments

The completion of this research would not have been possible without the invaluable guidance and support of several individuals. I would like to express my sincere gratitude to my thesis supervisors for their expert guidance, constructive feedback, and unwavering support throughout this research process. Their profound knowledge and insightful suggestions have significantly enhanced the quality of this work and shaped my understanding of machine learning applications in text classification.

I am also grateful to my colleagues and fellow researchers in the computer science department for their stimulating discussions, technical assistance, and moral support that helped me overcome various challenges encountered during this study. The collaborative environment fostered by my peers contributed significantly to refining the research methodology and interpretation of results.

Special appreciation goes to my friends who provided encouragement and were always willing to help with proofreading and offering fresh perspectives on my research. Their continuous motivation kept me focused and determined to complete this work to the best of my ability.

Finally, I would like to extend my heartfelt thanks to my family for their patience, understanding, and emotional support throughout my academic journey. Their unwavering belief in my capabilities and constant encouragement have been a source of strength and inspiration, especially during challenging times. This accomplishment would not have been possible without their love and support.

References

Abidin, Z., & Junaidi, A. (2024). Text Stemming and Lemmatization of Regional Languages in Indonesia: A Systematic Literature Review. *Journal of Information Systems Engineering and Business Intelligence*, *10*(2), 217–231.

AlAfnan, & Awad, M. (2024). Social Media Personalities in Asia: Demographics, Platform Preferences, and Behavior Based Analysis. *Studies in Media and Communication*, *12*(3), 349–363.

Devita, R. N., Herwanto, H. W., & Wibawa, A. P. (2018). Perbandingan kinerja metode naive bayes dan k-nearest neighbor untuk klasifikasi artikel berbahasa indonesia. *J. Teknol. Inf. Dan Ilmu Komput*, *5*(4).

Dwiyansaputra, R., Nugraha, G. S., Bimantoro, F., & Aranta, A. (2021). Deteksi SMS Spam Berbahasa Indonesia menggunakan TF-IDF dan Stochastic Gradient Descent Classifier. *Jurnal Teknologi Informasi, Komputer, Dan Aplikasinya (JTIKA)*, *3*(2), 200–207.

Fhonna, R. P., Afrillia, Y., Aqmal, J., & Abadi, S. (2023). Klasifikasi Penentuan Jenis Tanah yang Sesuai Terhadap Tanaman Pangan Sebagai Solusi Ketahanan Pangan di Kabupaten Pidie Jaya menggunakan Metode Random Forest. *Jurnal Informasi Dan Teknologi*, 12–18.

Gaur, P., Vashistha, S., & Jha, P. (2023). Twitter sentiment analysis using naive bayes-based machine learning technique. In *Sentiment Analysis and Deep Learning: Proceedings of ICSADL 2022* (pp. 367–376). Springer.

Hasanah, A. N. R., Krestianti, R. A., & Wati, S. (2023). Implementasi Algoritma Regresi Logistik untuk Binary Classification dalam Spam SMS dan WhatsApp. *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, *7*(1), 80–93.

Herwanto, Chusna, N. L., & Arif, M. S. (2021). Klasifikasi SMS Spam Berbahasa Indonesia Menggunakan Algoritma Multinomial Naïve Bayes. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *5*(4), 1316–1325.

Jalilifard, A., Caridá, V. F., Mansano, A. F., Cristo, R. S., & da Fonseca, F. P. C. (2021). Semantic sensitive TF-IDF to determine word relevance in documents. *Advances in Computing and Network Communications: Proceedings of CoCoNet 2020, Volume 2*, 327–337.

Johns, A., Matamoros-Fernández, A., & Baulch, E. (2023). *WhatsApp: From a one-to-one messaging app to a global communication platform*. John Wiley & Sons.

Kerner, Y. H., Miller, D., & Yigal, Y. (2020). The influence of preprocessing on text classification using a bag-of-words representation. *PloS One*, *15*(5), e0232525.

Lavanya, P. M., & Sasikala, E. (2021). Deep learning techniques on text classification using Natural language processing (NLP) in social healthcare network: A comprehensive survey. *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, 603–609.

Mutiara, A. B., Wibowo, E. P., & Santosa, P. I. (2020). The Crowdsourcing Method to Normalize "Bahasa Alay", a Case of Indonesian Corpus. *2020 Fifth International Conference on Informatics and Computing (ICIC)*, 1–5.

Naseem, U., Razzak, I., & Eklund, P. W. (2021). A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter. *Multimedia Tools and Applications*, *80*, 35239–35266.

Normawati, D., & Prayogi, S. A. (2021). Implementasi Naïve Bayes classifier dan confusion matrix pada analisis sentimen berbasis teks pada Twitter. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, *5*(2), 697–711.

Putera, A. W., Suriati, S., & Lestari, Y. D. (2023). Klasifikasi Sms Spam Menggunakan Algoritma K-Nearest Neighbor. *Jurnal Ilmu Komputer Dan Sistem Komputer Terapan (JIKSTRA)*, *5*(1), 43–55.

Qadrini, L., Seppewali, A., & Aina, A. (2021). Decision tree dan adaboost pada klasifikasi penerima program bantuan sosial. *Jurnal Inovasi Penelitian*, *2*(7), 1959–1966.

Qamal, M. (2021). Analisis Sentimen Toko Online Menggunakan Algoritma Naive Bayes Classifier. *Jurnal Teknologi Terapan and Sains 4.0*, *2*(3), 641–650.

Quist, J., Taylor, L., Staaf, J., & Grigoriadis, A. (2021). Random forest modelling of high-dimensional mixed-type data for breast cancer classification. *Cancers*, *13*(5), 991.

Rezaeian, N., & Novikova, G. (2020). Persian text classification using naive bayes algorithms and support vector machine algorithm. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, *8*(1), 178–188.

Samad, M. D., Khounviengxay, N. D., & Witherow, M. A. (2020). Effect of text processing steps on twitter sentiment classification using word embedding. *ArXiv Preprint ArXiv:2007.13027*.

Sapitri, I. A., Yusra, Y., & Fikry, M. (2023). Pengklasifikasian Sentimen Ulasan Aplikasi Whatsapp Pada Google Play Store Menggunakan Support Vector Machine. *Jurnal Tekinkom (Teknik Informasi Dan Komputer)*, *6*(1), 1–7.

Setiyana, T. B. (2021). *ANALISIS SENTIMEN PADA REVIEW APLIKASI KESEHATAN HALODOC MENGGUNAKAN METODE MAXIMUM ENTROPY*. Muhammadiyah University, Semarang.

Sihombing, J. J., Arnita, A., Al Idrus, S. I., & Niska, D. Y. (2024). Implementation of text summarization on indonesian scientific articles using textrank algorithm with TF-IDF web-based. *Journal of Soft Computing Exploration*, *5*(3), 310–319.

Wang, D., Su, J., & Yu, H. (2020). Feature extraction and analysis of natural language processing for deep learning English language. *IEEE Access*, *8*, 46335–46345.

Yanto, O. (2021). Hoax As A Cyber Crime In The Whirlpool Of Information Technology. *International Journal of Education and Sosiotechnology (IJES)*, *1*(3), 13–23.