DOI: 10.22146/jnteti.v14i4.20931

# Spam Email Classification Optimization With NLP-Based Naïve Bayes on TF-IDF and SMOTE

**Andi Maslan[1], Azan Rahman[1], Umar Faruq[2], Rabei Raad Ali Al-Jawry[3]**

[1] Department of Informatic Engineering, Universitas Putera Batam, Batam, Kepulauan Riau, 2943, Indonesia.
[2] School of Information Technology, UNITAR International University Petaling Jaya, Selangor, Malaysia
[3] Department of Computer Engineering Technology, Northern Technical University, Mosul, Iraq

[Received: 10 July 2025, Revised: 9 October 2025, Accepted: 27 October 2025]
Corresponding Author: Andi Maslan (email: lanmasco@gmail.com

**ABSTRACT** — The rapid advancement of information and communication technology has transformed the way humans interact and exchange information. Among various digital communication tools, email remains one of the most widely used; however, it is often exploited to send spam messages. Spam emails can contain phishing links, malware, or unsolicited advertisements, posing significant risks to individuals and organizations. Therefore, developing accurate and efficient spam detection methods is becoming increasingly important. This study proposes a lightweight and efficient spam email classification approach using the naïve Bayes algorithm combined with TF-IDF feature extraction and the synthetic minority oversampling technique (SMOTE) to address class imbalance. A series of preprocessing steps tokenization, lemmatization, stopword removal, and term frequency-inverse document frequency (TF-IDF) transformation were applied to normalize and vectorize email text data. The SMOTE technique was applied precisely to the training dataset to balance the class distribution and avoid data leakage during evaluation. Experimental results showed that the naïve Bayes model initially achieved 88% accuracy, 86% recall, 100% precision, and 92% F1 score. After proper application of SMOTE, the model achieved 100% accuracy, precision, recall, and F1 score, indicating perfect classification of spam and non-spam (ham) emails. These results confirm that proper class balancing significantly improves the model's ability to detect spam emails. Overall, this study highlights the effectiveness of combining TF-IDF, naïve Bayes, and SMOTE as a robust yet computationally efficient solution for modern spam detection, particularly suited to real-time and resource-constrained environments.

**KEYWORDS** — Tokenization, SMOTE, TF-IDF, Spam Email Detection, Model Performance Optimization, Classification Accuracy Enhancement.

## I. INTRODUCTION

Along with today's digital developments, email is one of the commonly used means of communication in both personal and business affairs. Compared to sending messages using conventional methods such as mail, email can be much faster and more efficient because it can save time and money. In addition to sending text messages, email allows its users to send attachments such as images and documents. However, the growing popularity of email as a means of communication has also raised a new problem in the form of spam. The practicality of email causes irresponsible people to use it for personal gain. These spam emails can contain advertisements, product promotions, scams, or unwanted content. Sometimes spam emails also accompany malicious links that when accessed can spread viruses or malware. In addition to harassing victims personally, spam emails can fill up device storage and increase data traffic on internet networks.

Based on reports from several sources, this past year has urgently underscored an ever-evolving landscape since the last state of phishing report, with certain trends increasing at alarming rates [1]–[4]. In addition, in the last six months, SlashNext Threat Labs recorded a 1,265% increase in phishing emails, with 68% using text-based BEC tactics, indicating that chatbots and jailbreak techniques are being leveraged to generate faster and more realistic attacks. Credential phishing also increased by 967%, primarily driven by ransomware groups seeking access to corporate accounts. In addition to email threats, attacks on mobile devices have also increased, with 39% of mobile threats being smishing, indicating a shift towards increasingly complex, multi-channel attacks [5], [6]. Hence, researchers continue to innovate to prevent or detect attacks through research with various approaches such as machine learning or artificial intelligence (AI).

By using algorithms and computational models that can learn and detect patterns from data, machine learning, a subfield of artificial intelligence, may be used to assess and identify security threats and automate solutions to overcome security difficulties in computer networks [7],[8].

The main goal of machine learning is to turn diverse data into effective decisions or actions with minimal human intervention [9]. Machine learning works by studying the patterns found in spam emails based on their features, such as the frequency of email sentencing, email subjects, attachments, and the text of messages that are suspected of being spam [10].

There are several kinds of algorithms [11], [12], used in machine learning, one of which is the naïve Bayes algorithm [13], which is the focus of this study. Naïve Bayes has the advantage of classifying text and has been widely applied, including in detecting spam emails [14]. This algorithm demonstrates a good performance, being both light and fast. It works by analyzing certain characteristics of an email, such as the presence of advertisements, product promotions, scams, or harmful content, to determine if the email is safe or risky [15].

Based on theoretical foundations and previous research, this study offers novelty in the application of the naïve Bayes algorithm for email spam detection, taking into account the latest developments in cyber threats. One of these developments is the use of generative AI technology, such as

ChatGPT, which is starting to be used in creating more convincing and complex malicious email content. Unlike previous approaches that only relied on old datasets and static patterns, this study focused on the more sophisticated characteristics of modern spam, such as the use of natural language, promotional content obfuscation, and the use of AI technology to trick traditional filters. In terms of scientific contribution, this study presented the development and evaluation of a naïve Bayes-based classification model that is not only accurate, but also lightweight and fast, making it suitable for use in real-time systems or on devices with limited resources. This study also compared the effectiveness of detection against the latest malicious email data, providing important insights for the development of adaptive security systems. The real benefits of this research are directly felt by the wider community, especially in improving digital security for individual and institutional email users. The developed system can minimize the risk of fraud, malware, and unwanted content, as well as maintain the efficiency of digital communication. In addition, this model can be easily adopted by small businesses, educational institutions, and government agencies that do not yet have complex cybersecurity infrastructure, thereby increasing protection without requiring large investments in technology.

## II. RELATED WORKS

Research related to the classification of spam emails has been extensively conducted. Previous research [16] explained that the GWO-BERT method with the CNN, BiLSTM and LSTM models that he designed successfully learned meaningful email text representations and classified them into spam categories with an accuracy rate of 99.14%. Then research [17] identified spam using the Harris Hawks (HHO) optimization algorithm with a machine learning approach, so that it could detect important features that distinguish spam from fake emails with classification results for the decision tree algorithm of 99.75%, AdaBoost 99.67%, and naive Bayes 96.30%. The results of prior study showed the effectiveness of CNN, which was significant in classifying emails, with a high accuracy rate of 99.67% for 20% test data, 99.64% for 30% test data, and 99.63% for 40% test data [11].

Moreover, prior study focused on using stop word-based term frequency-inverse document frequency (TF-IDF) and a stemming algorithm with the multicore graphics processing unit (GPU)'s naïve Bayes classifier to identify spam emails [18]. The results showed that the NVIDIA P100 GPU could accelerate the training and testing process while achieving higher accuracy compared to the conventional naïve Bayes algorithms. The study reported training accuracy of 99.67% and testing accuracy of 99.03%. Training time on GPU was 1,361 s, while CPU was 2,029 s. Meanwhile, the test results on the GPU were 1,978 s and the CPU was 2,280 s. Another study evaluated the naïve Bayes classifier model for spam email detection, where the results showed that this algorithm could be a good choice for spam email detection [19]. Nevertheless, more research is needed to address challenges such as the evolution of spammer techniques and data imbalances [13].

In addition, study on spam email detection has also been carried out. Previous research reported that naïve Bayes was effective in classifying email spam, with the best performance at $k = 9$ using $k$-fold cross-validation [20]. The results of this study showed that the naïve Bayes produced an accuracy of 84.8%, with 3,903 emails classified as correct and 698 as false, while precision and recall were 0.86 and 0.85, respectively. The latest research demonstrated that naïve Bayes algorithm with chi-square achieved 81.00% accuracy, 100% accuracy, 65% recall, and 79% F1 score, while the area under curve (AUC) value was 0.91. These results suggest that naïve Bayes with chi-square is effective for email spam classification, with high precision that avoids positive errors [21]–[23]. However, the recall needs to be improved to detect more spam. This study focused on the classification of spam emails using the naïve Bayes algorithm based on natural language processing (NLP) based on TF-IDF and SMOTE, taking into account the issues and other research that have been mentioned.

## III. METHODOLOGY

The research began by identifying the main problem, which was to distinguish between legitimate (ham) emails and spam using a naïve Bayes algorithm [15]. The next stage was data collection through literature studies related to email spam, machine learning, and the naïve Bayes method, with sources from national and international books and journals. The data that had been collected were then processed to ensure that the model could use their format. Following data processing, training data were used to train the naïve Bayes model. Based on the patterns found in the training data, the model learnt to differentiate between spam and ham.

After the model was trained, tests were carried out using test data that were not visible to the model during training. The predicted results from the model were then compared to the original label to see how well the model classified emails as spam or ham. Test results were evaluated using metrics such as accuracy, precision, recall, and F1 score to assess the model's performance in correctly classifying the data. Based on the evaluation, the results of the model could be analyzed to determine if the model was good enough at separating spam and ham. If the results were not satisfactory, then improvements could be made to the training or data processing process.

In this study, the secondary data were used and taken from the public dataset. These data were obtained from reliable sources that provide datasets for research and not generated through primary collection, such as interviews or observations. The dataset used in this study was downloaded from the Harvard Dataverse [24], an open data storage platform that provides a wide range of quality datasets for further analysis and development.

This dataset consisted of 5,728 lines of data, where each line represented an email that was divided into two main categories, namely spam and non-spam (ham). Specifically, this dataset consisted of 1,369 spam data and 4,329 ham data, which was presented in tabular format with two main columns: the "text" column containing the text of the email, and the "spam" column which containing the category label, in the form of "spam" or "ham."

This dataset was used to train and test a naïve Bayes algorithm-based spam detection model. Before use, this raw data went through preprocessing stages, such as the removal of stopwords, lemmatization, and conversion of text into numerical representations using the TF-IDF method, so that they could be used in the further analysis process.

The process began with the collection of a labeled email dataset that included modern spam messages such as phishing and AI-generated content. Next, tokenization was performed,

an initial process in text preprocessing that aims to break text or sentences into smaller parts, namely tokens [10], [16], [25]. Tokenization ensures that the text is already separated into small units (words or tokens). These tokens are generally words or symbols that can be used in further analysis [26]. The tokenization process is crucial because it allows text to be broken down into units that can be analyzed more easily, especially in classification algorithms like naïve Bayes. The steps of coking are retrieving raw texts and breaking sentences into tokens.

In the raw text retrieval phase, the data used consisted of email text that could contain sentences, symbols, and punctuation. Before the tokenization process, these texts were in the form of a whole sentence with no clear word separators. Subsequently, these sentences were broken down into words or tokens, for example: the sentence "Congratulations! You have won a free vacation" are converted into ["Congratulations," "You," "have," "won," "a," "free," "vacation"]. During tokenization, this study used the natural language toolkit (NLTK) library in Python, which provides various functions for text tokenization. The word tokenize function in NLTK was used to break sentences into individual words.

Subsequently, the lemmatization technique was used to change words into their basic form [10], [16], [27]. Unlike stemming that only cuts out the end of a word, lemmatization considers the context and grammar, which is essential to reduce unnecessary word variation. Therefore, text analysis is more focused on the meaning of relevant words. Before lemmatization, the tokenized text obtained from the previous step is used [27]. Lemmatization is applied to each token using a specific tool or library [28]. In this study, WordNetLemmatizer from the NLTK library was used to conduct lemmatization [29], [30]. Examples of results at this stage are, the "running" is changed to "run," "better" is changed to "good," and the word "coating" is changed to "paint."

The stopwords stage was carried out. It is the process of removing words that are considered unimportant in text analysis, as these words generally do not provide relevant information for classification purposes [19], [31], [32], [23]. These words are conjunctions, prepositions, or auxiliary verbs that frequently appear in sentences but do not contribute to distinguish categories of text. Words such as "and," "or," "the," "for," and "is" fall into the category of stopwords. Stopword removal was done using a list of existing stopwords, such as those provided by the NLTK library in Python. This list already includes common words that exist in English (and other languages) that are generally considered irrelevant in text analysis.

TF-IDF assigns weight to each word in a document according to its frequency of occurrence (also known as term frequency or TF) and its seldom occurrence (also known as inverse document frequency or IDF) [13], [15], [23], [33], [34]. An example of the implementation is presented in Table I.

Table I shows examples of email texts categorized as spam and not spam. Emails in the spam category often contain promotional content or invitations to click on suspicious links. For instance, the subject line "You've won $5,000!" encourages the reader to click a link to claim a prize, which is a common characteristic of spam. Similarly, the message "Free Gift Card Offer" contains incentives and urgency to complete a survey, typical of spam patterns. Meanwhile, emails under the not spam category provide informative and transactional

TABLE I
SAMPLE TEXTS EMAIL

| No. | Category | Subject | Email Body |
|---|---|---|---|
| 1 | Spam | You've won $5,000! | Congratulations! You are the lucky winner of our monthly sweepstakes. Click the link below to claim now! |
| 2 | Spam | Free Gift Card Offer | Complete a short survey and receive a $100 gift card instantly. Limited time only! |
| 6 | Non spam | Meeting Agenda for Monday | Hi team, please find attached the agenda for our project meeting scheduled on Monday at 10 AM. |
| 7 | Non spam | Your Receipt from ABC Store | Thank you for your purchase. Attached is your receipt for the items bought on July 5th. |

content. For example, "Meeting Agenda for Monday" contains details relevant to scheduled work activities, and "Your Receipt from ABC Store" provides purchase confirmation, both of which reflect legitimate communication.

The calculation of the number of words in each email body is as follows.

- Text 1 (SPAM): "Congratulations! You are the lucky winner of our monthly sweepstakes. Click the link below to claim now!" contains 17 words.
- Text 2 (SPAM): "Complete a short survey and receive a $100 gift card instantly. Limited time only!" contains 14 words.
- Text 3 (NOT SPAM): "Hi team, please find attached the agenda for our project meeting scheduled on Monday at 10 AM" contains 19 words.
- Text 4 (NOT SPAM): "Thank you for your purchase. Attached is your receipt for the items bought on July 5th" contains 17 words.

Based on the text data above, the following formula was used to calculate the TF value.

$$TF(t) \quad \frac{number\ of\ words\ (t)\ in\ a\ document}{number\ of\ words\ in\ a\ document} \tag{1}$$

$$IDF(t) = Log(\frac{Total\ Documents}{Documents\ containing\ the\ word\ t}) \tag{2}$$

$$TF - IDF(t,d) = TF(t,d) \times IDF(t). \tag{3}$$

Following the acquisition of the TF value, the spam email classification model was constructed and tested using a training and testing procedure. To ensure the model could learn from existing data and could be tested with data that had never been seen, the dataset was separated into two primary sections: the training set and the testing set. The dataset was distributed using the hold-out validation method, which divides the data into training and testing phases with a training set ratio of 80% and a testing data set ratio of 20%. This ratio was chosen so that the model got sufficient data for learning while still retaining a lot of data for testing its accuracy. Accuracy testing was conducted using the naïve Bayes algorithm, which performed probability-based classification. The naïve Bayes algorithm formula in this study can be seen in (4).

$$P(C|X) = \frac{P(X|C) \times P(C)}{PX} \tag{4}$$

where *P(C|X)* denotes probability C (spam/no spam) on X (email text), *P(X|C)* denotes probability of getting feature (X) on C, *P(C)* denotes probability of prior C, and *P(X)* denotes

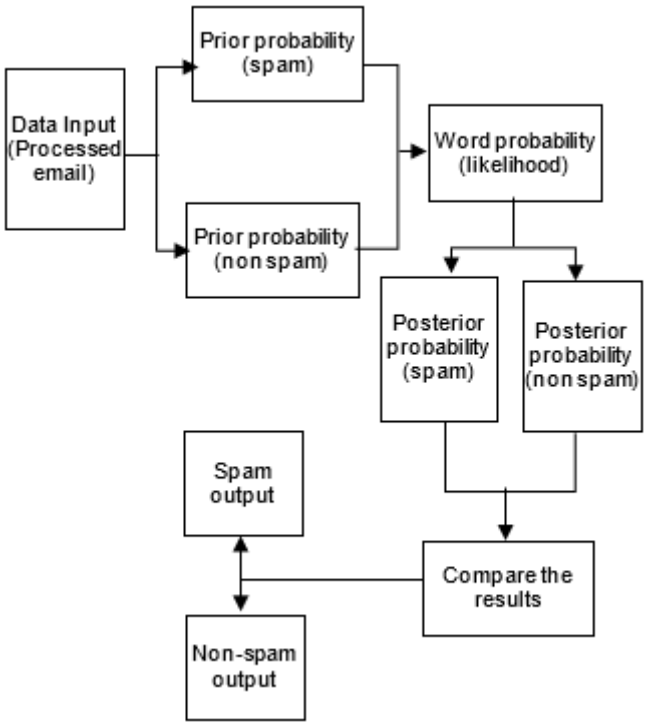


**Figure 1.** Steps of the naïve Bayes algorithm.

probability of X feature. Based on this formula, the naïve bayes algorithm can be explained in Figure 1.

Figure 1 illustrates the steps of the naïve Bayes algorithm. Input data is the processed email data. Prior probability is the probability of data entering a certain class before considering the features (words) in it, such as spam and ham. The formula for calculating prior probability is shown in (5).

$$P(class) = \frac{the\ amount\ of\ data\ in\ a\ particular\ class}{word\ tatal}. \quad (5)$$

Likelihood probability is the chance of a word appearing in a particular class. Likelihood helps determine whether a particular word is more likely to appear in spam or ham emails. The formula is presented in (6).

$$P(words/class) = \frac{number\ of\ occurrences\ of\ a\ word}{total\ number\ of\ words}. \quad (6)$$

Naïve Bayes then calculated the posterior probability for each class by multiplying the prior probability and the likelihood. The formula used is presented in (7).

$$P(class|words) = \frac{P(class) \times P(words\ n|class)}{P(words)}. \quad (7)$$

In identifying the class based on the highest posterior probability, the naïve Bayes method classified the email data based on the highest value between the posterior probabilities of spam and ham after calculating the posterior probabilities for the spam and ham classes. The model evaluation aimed to assess how well the naïve Bayes algorithm performed in identifying spam and not spam emails. Relevant evaluation criteria like accuracy, precision, recall, F1 score, and confusion matrix were used in the review process.

## IV. RESULTS AND DISCUSSION

This section explains the results of the research that has been carried out based on the stages that have been designed previously. The process of data collection, implementation of the naïve Bayes algorithm, and analysis of the results of spam email classification are described systematically. An overview of the success of the method used is also presented. Table II presents the results of the preprocessing process on the email dataset used in this study include the results of tokenization, lemmatization, and stopword removal.

TABLE II
TOKENIZATION RESULTS

| Before Tokenization | After Tokenization |
|---|---|
| Subject: You've won $1,000! Click here to claim your prize now. | ['Subject', ':', 'You', "'", 've', 'won', '$', '1,000', '!', 'Click', 'here', 'to', 'claim', 'your', 'prize', 'now', '.'] |
| Congratulations! You've been selected for a free gift card. Just complete this short survey. | ['Congratulations', '!', 'You', "'", 've', 'been', 'selected', 'for', 'a', 'free', 'gift', 'card', '.', 'Just', 'complete', 'this', 'short', 'survey', '.'] |
| Reminder: Your account has been suspended. Login to verify your information. | ['Reminder', ':', 'Your', 'account', 'has', 'been', 'suspended', '.', 'Login', 'to', 'verify', 'your', 'information', '.'] |
| Earn money fast working from home. No experience needed. | ['Earn', 'money', 'fast', 'working', 'from', 'home', '.', 'No', 'experience', 'needed', '.'] |
| Your payment of $500 is confirmed. | ['Your', 'payment', 'of', '$', '500', 'is', 'confirmed', '.'] |

Tokenization process plays a crucial role in preparing textual data for analysis by breaking down complete sentences into smaller, more meaningful components. For example, the sentence "You've won $1,000! Click here to claim your prize now." is transformed into a series of individual tokens such as ["You," "'ve," "won," "$," "1,000," "!," "Click," "here," "to," "claim," "your," "prize," "now," "."]. This decomposition enables machine learning models to process and analyze textual patterns more effectively, especially in tasks like spam email classification. Furthermore, the lemmatization process, which often follows tokenization, was applied to convert words to their base forms. For instance, tokens like "selected," "gifts," "claiming," and "won" are lemmatized to "select," "gift," "claim," and "win," respectively. This normalization ensures that variations of a word are treated as a single term, thus enhancing the accuracy and efficiency of the text classification process.

Results of lemmatization applied to the email dataset are presented in Table III. The word "won" was lemmatized to "win", and the contraction "you've" was separated into "you" and "have." In the second example, the word "selected" was changed to "select," and "gifts" was normalized to "gift." Meanwhile, "suspended" was changed into "suspend," and auxiliary verbs "has" and was changed into "have" and "be," respectively. In the fourth example, "working" was changed to "work," and "needed" became "need." In the fifth example, the word "confirmed" was lemmatized to "confirm." These changes ensure that each word is processed in its base form, which helps reduce variability and simplifies the text for further processing such as spam email classification.

After the stopwords removal process was applied, the result was a cleaner dataset and more focused on the words that contributed to the classification process. The removal of these stopwords aimed to reduce noise that can affect the model's

TABLE III
LEMMATIZATION RESULTS

| Before Lemmatization | After Lemmatization |
|---|---|
| Subject: You've won $1,000! Click here to claim your prize now. | ['subject', ':', 'you', 'have', 'win', '$', '1,000', '!', 'click', 'here', 'to', 'claim', 'your', 'prize', 'now', '.'] |
| Congratulations! You've been selected for a free gift card. Just complete this short survey. | ['congratulation', '!', 'you', 'have', 'be', 'select', 'for', 'a', 'free', 'gift', 'card', '.', 'just', 'complete', 'this', 'short', 'survey', '.'] |
| Reminder: Your account has been suspended. Login to verify your information. | ['reminder', ':', 'your', 'account', 'have', 'be', 'suspend', '.', 'login', 'to', 'verify', 'your', 'information', '.'] |
| Earn money fast working from home. No experience needed. | ['earn', 'money', 'fast', 'work', 'from', 'home', '.', 'no', 'experience', 'need', '.'] |
| Your payment of $500 is confirmed. | ['your', 'payment', 'of', '$', '500', 'be', 'confirm', '.'] |

TABLE IV
STOPWORD REMOVAL RESULTS

| Before | After |
|---|---|
| help television in 1919 by seat to my knoweledge . chrono cross in 1969 | [help, television, 1919, seat, knowledge, chrono, cross, 1969] |
| the most expensive car sold in graand ! cheap cars in graand | [most, expensive, car, sold, graand, cheap, car, graand] |
| save your money by getting an oem software ! need in software for your pc ? just visit our site , we might have what you need. | [save, money, getting, oem, software, need, software, pc, visit, site, might, have, need] |

TABLE V
EMAIL EXAMPLES

| Document | Email Text |
|---|---|
| Document 1 | Subject: do not have money, get software cds from here ! software compatibility . . . . ain ' t it great ? grow old along with me the best is yet to be . all tragedies are finish ' d by death . all comedies are ended by marriage. |
| Document 2 | security alert Confirm national credit union information |
| Document 3 | Subject: claim your free $ 1000 homedepot gift card . claim your homedepot gift card - a $ 1000 value . were sure you can find a use for this gift card in your area ( ) . by exclusive rewards udexhoyp |

TABLE VI
TF-IDF RESULTS MORE THAN 1 DOCUMENT

| No | Word | Document 1 | Document 2 | Document 3 |
|---|---|---|---|---|
| 1 | Software | 0.3192 | 0.0000 | 0.0000 |
| 2 | Compatibility | 0.3189 | 0.0000 | 0.0000 |
| 3 | Finish | 0.2158 | 0.0000 | 0.0000 |
| 4 | Death | 0.2659 | 0.0000 | 0.0000 |
| 5 | Old | 0.2039 | 0.0000 | 0.0000 |
| 6 | Union | 0.0000 | 0.5342 | 0.0000 |
| 7 | Alert | 0.0000 | 0.4529 | 0.0000 |
| 8 | National | 0.0000 | 0.4010 | 0.0000 |
| 9 | Confirm | 0.0000 | 0.3310 | 0.0000 |
| 10 | Security | 0.0000 | 0.3264 | 0.0000 |
| 11 | Credit | 0.0000 | 0.3023 | 0.0000 |
| 12 | claim | 0.0000 | 0.0000 | 0.2959 |
| 13 | Gift | 0.0000 | 0.0000 | 0.5026 |
| 14 | Card | 0.0000 | 0.0000 | 0.4109 |
| 15 | Area | 0.0000 | 0.0000 | 0.1049 |
| 16 | Value | 0.0000 | 0.0000 | 0.1039 |

performance in distinguishing between spam and ham emails. Table IV presents examples of text that has gone through the stopword removal process.

In the first example, the words "in," "to," and "my" are omitted, as they do noy significantly contribute to classification. In addition, in the example, the two words omitted are "the" and "in." After this stage, the dataset is cleaner and ready for further processing.

The results of the TF-IDF calculation, which depict the weight of each word in the document, are shown in the Table V. The calculation was performed using the scikit-learn library found in Python, which automatically calculates the TFIDF values based on the analyzed document set. Meanwhile, the results of TF-IDF calculations for the three email text documents are shown in Table VI.

In Table VI, documents 1, 2, and 3 refer to examples of emails that are categorized as spam. In document 1, the word "software" yielded a TF-IDF value of 0.3192, indicating this word is quite influential in the document. This word often appears in spam emails, particularly in pirated software promotions or suspicious discount offers. In document 2, the word "alert" yielded a TF-IDF score of 0.4529, indicating that this word is quite important in the document. The word "alert" is frequently used in emails that attempt to make the recipient feel panicked or rushed, as in fraudulent emails that ask for immediate confirmation of financial information. In document 3, the word "claim" obtained a TF-IDF value of 0.2959, meaning that this word is also quite relevant in the document. The word "claim" is often found in spam emails containing claims of fake rewards or lucrative compensation to attract the recipient's attention. Overall, a higher TF-IDF score indicates more relevant words and has an important role in the spam and ham classification process.

In order to train and evaluate the classification model, data with a variety of ratios were prepared during the experimental phase. The data distribution used in this study was 80%–20%, 70%–30%, and 60%–40%, where a larger portion was allocated for training and the remainder for testing. The experiment's goal was to determine how different data sharing practices impact the model's performance, specifically with regard to accuracy and spam detection.

Table VII exhibits the data splits that experienced a decrease in accuracy as the proportion of training data decreased. This decrease was relatively small and not significant. The highest accuracy was achieved at the 80%:20% data split, which provided the model with more training data. This split generally yielded more stable and better results because the model was trained with a larger amount of data, allowing it to better capture patterns.

In this study, the naïve Bayes algorithm was used to classify emails as spam or ham by considering the distribution of words in the dataset. To understand how naïve Bayes works, the following is the basic principles and mathematical calculations. Consider a small dataset of emails with two categories: spam and ham. The model estimates the probability of a new email belongs to either spam or ham category based on the appearance of words in the training dataset.

Table VIII presents an example of email data used to classify text into two main categories: spam and ham. The

TABLE VII
COMPARISON OF TRAINING AND TESTING

| Training (%) | Testing (%) | Accuracy (%) |
|---|---|---|
| 80 | 20 | 90 |
| 70 | 30 | 89 |
| 60 | 40 | 88 |

TABLE VIII
SAMPLE EMAIL DATASET

| No | Email Text | Category |
|---|---|---|
| 1 | "Save your money by getting an OEM software" | Spam |
| 2 | "Confirm your national credit union information" | Spam |
| 3 | "Hello David, another trip in the cards" | Non spam |
| 4 | "Vince, congratulations on your promotion" | Non spam |

dataset consisted of 5,728 emails, with 1,369 spam emails and 4,329 non-spam emails (ham). Each row in the table contains email text reflecting the general characteristics of each category. In the first and second examples, emails are categorized as spam because they contain elements commonly found in promotional or phishing emails, such as an invitation to buy cheap software ("Save your money by getting an OEM software") or a request to confirm sensitive information ("Confirm your national credit union information"). This type of content is usually used to lure users into providing personal data or being interested in suspicious offers. In contrast, in the third and fourth examples, emails are categorized as ham because they contain messages that are personal and do not contain promotional elements or requests for sensitive information. For example, personal greetings such as "Hello David, another trip in the cards" and congratulations such as "Vince, congratulations on your promotion" reflect normal communication between individuals that are not generally considered spam. Hence, the classification of emails based on the content of the text is highly dependent on understanding the context and patterns of the language used. Spam emails tend to use unusual promotional wording or requests for information, while ham emails are more personal and contextually relevant. Examples of these kinds of data are very important in the process of training classification models so that the system can accurately distinguish between spam and ham.

Prior probability, likelihood, and posterior probability are the three primary steps in the naïve Bayes method's text categorization computation. The proportion of the total number of emails in the dataset was used to compute the prior probability in the first phase. Each had an initial chance of 0.5 because the quantity of spam and ham emails was equal.

In the second step, the number of unique words in each category was calculated. The spam class had 10 unique words, while the ham class had 7. The combined vocabulary of both is 17 words. Then, using Laplace smoothing, the probability of a particular word (such as "money") appearing in each class was calculated.

The results of the posterior probability calculation showed the value of P (Spam | word) was greater than P (Ham | word). This means that the phrase "free software and save money" is more likely to be classified as spam according to this naïve Bayes model. This technique has proven to be efficient in classifying text based on the occurrence of certain words and is very useful in automatic spam detection systems.

TABLE IX
NAÏVE BAYES CALCULATIONS

| No | Word | Spam | Ham | Likelihood | |
|---|---|---|---|---|---|
| | | | | *Spam* | *Ham* |
| 1 | save | 1 | 0 | 0.074 | 0.042 |
| 2 | money | 1 | 0 | 0.074 | 0.042 |
| 3 | getting | 1 | 0 | 0.074 | 0.042 |
| 4 | OEM | 1 | 0 | 0.074 | 0.042 |
| 5 | software | 1 | 0 | 0.074 | 0.042 |
| 6 | confirm | 1 | 0 | 0.074 | 0.042 |
| 7 | national | 1 | 0 | 0.074 | 0.042 |
| 8 | credit | 1 | 0 | 0.074 | 0.042 |
| 9 | union | 1 | 0 | 0.074 | 0.042 |
| 10 | information | 1 | 0 | 0.074 | 0.042 |
| 11 | hello | 0 | 1 | 0.037 | 0.083 |
| 12 | david | 0 | 1 | 0.037 | 0.083 |
| 13 | trip | 0 | 1 | 0.037 | 0.083 |
| 14 | card | 0 | 1 | 0.037 | 0.083 |
| 15 | vince | 0 | 1 | 0.037 | 0.083 |
| 16 | congratulation | 0 | 1 | 0.037 | 0.083 |
| 17 | promotion | 0 | 1 | 0.037 | 0.083 |

To avoid the value 0, the Laplace smoothing technique was added to the formula. Without this technique, a problem may arise because when the probability results are zero, the entire calculation will also become zero.

Table IX presents the results of the naïve Bayes probability calculations, which illustrate how each word contributes to determining whether an email is classified as spam or ham. Each word in the dataset was assigned a likelihood value for both categories (spam and ham) based on its frequency of occurrence within the respective classes. For instance, words such as "save," "money," "getting," and "promotion" show higher likelihood values in the spam category (0.074) compared to the ham category (0.042), indicating that these terms frequently appear in spam messages. Conversely, words like "hello," "trip," and "card" have higher likelihood values in ham emails (0.083), suggesting they are more commonly found in legitimate communication.

These probability values were then combined according to the naïve Bayes theorem to calculate the posterior probability for each class. The class with the higher posterior probability determined the final classification result. In this case, words with higher likelihoods in the spam category increased the probability that an email containing those terms would be classified as spam. Thus, Table IX provides an overview of the word-level contribution used by the naïve Bayes model to distinguish between spam and ham emails. This demonstrates the model's interpretability, showing how textual features (keywords) directly influence the classification outcome.

The model achieved perfect 100% classification accuracy after the correct application of the SMOTE technique to the training data. This result indicates that the rebalanced dataset allows the naïve Bayes classifier to effectively learn the characteristics of spam emails without bias toward the majority class ham.

Based on the study's findings, the model's performance was evaluated before and after applying the SMOTE. Prior to SMOTE, the naïve Bayes classifier achieved an accuracy of 88%, recall of 86%, precision of 100%, and F1 score of 92%. These results showed that although the model classified spam

TABLE X
SMOTE TECHNIQUE ACCURACY CALCULATION

| Metric | Before SMOTE | After SMOTE (Train Only) |
|---|---|---|
| Accuracy | 0.88 | 1.00 |
| Precision | 1.00 | 1.00 |
| Recall | 0.86 | 1.00 |
| F1 Score | 0.92 | 1.00 |
| Support (test data) | 912 | 912 |

emails with high precision, many spam messages were still misclassified as ham, as indicated by the relatively low recall value (86%) and the presence of numerous false negatives.

After applying SMOTE only to the training dataset, the model's performance improved substantially. Accuracy, precision, recall, and F1 score all reached 100%, indicating that the model was able to correctly classify all spam and ham emails in the testing data. This improvement confirms that the balanced dataset enabled the classifier to learn more representative patterns from both classes, thereby eliminating bias and improving generalization.

Overall, the application of SMOTE effectively resolved the class imbalance issue and significantly enhanced the spam detection model's performance. The increase in recall value from 0.86 to 1.00 demonstrated that the model became more sensitive in detecting spam emails that were previously misclassified. These results highlight that SMOTE is a powerful technique for improving the reliability and robustness of text classification models in imbalanced data scenarios. The improvement in performance metrics aligns with the changes in class distribution before and after SMOTE, as shown in Table X.

The model achieved a perfect score across all evaluation metrics after SMOTE was applied only to the training data, increasing recall from 0.86 to 1.00 and eliminating false negatives completely. This confirms that SMOTE successfully balanced the dataset and improved the model's ability to detect spam emails accurately.

Figure 2 exhibits the comparison of the class distribution before and after applying SMOTE to the training dataset. Before SMOTE, the dataset was highly imbalanced, with the ham class dominating the spam class, causing the model to misclassify many spam emails. After applying SMOTE to the training data, the distribution became balanced, allowing the model to learn equally from both classes. This adjustment improved the model's ability to recognize spam emails and improved recall performance without introducing bias.

Figure 3 presents the confusion matrix of the final naïve Bayes model after applying the SMOTE technique correctly on the training data. The matrix shows that the model achieved perfect classification results, with no false positives (FP = 0) and no false negatives (FN = 0). This means that all 664 ham emails were correctly identified as ham, and all 248 spam emails were accurately classified as spam.

The results indicated that the model reached 100% accuracy, precision, recall, and F1 score, demonstrating that the balanced dataset produced through SMOTE allowed the classifier to effectively distinguish between spam and ham emails. This confirms that the combination of TF-IDF feature extraction, SMOTE for class balancing, and naïve Bayes classification provides a highly reliable approach for spam email detection.

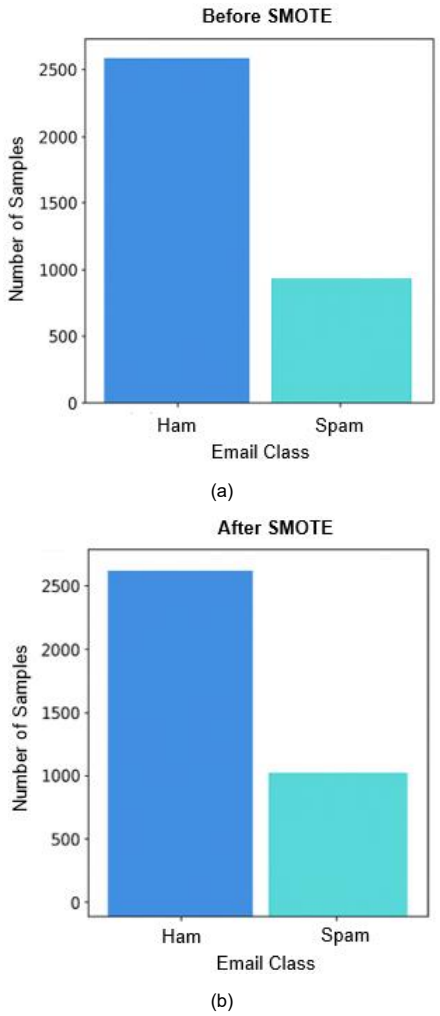


**Figure 2.** Difference in data distribution, (a) before SMOTE and (b) after SMOTE.

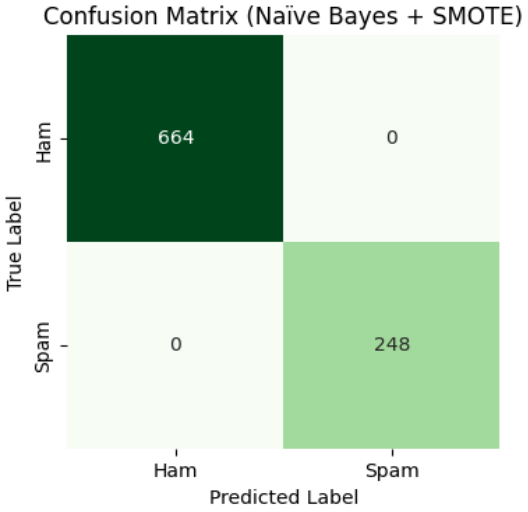


**Figure 3.** Confusion matrix (naïve Bayes + SMOTE).

Such performance implies that the model has successfully learned the underlying text patterns of spam and legitimate emails, minimizing both false alarms and missed detections. Therefore, this configuration can be effectively implemented in lightweight spam filtering systems to enhance email security.

## V. CONCLUSION

Initially, the naïve Bayes model achieved an accuracy of 88%, with a recall of 86%, precision of 100%, and an F1-score

of 92%. However, the model exhibited a relatively high number of false negatives (137), indicating that many spam emails were incorrectly classified as ham. In contrast, the false positive value was 0, showing that the model never misclassified legitimate emails as spam. This occurred because the dataset was highly imbalanced, with a significantly larger proportion of non-spam emails, which caused the model to bias toward the majority class and perform poorly on the minority class.

After applying SMOTE, the model achieved an accuracy, precision, recall, and F1 score of 100%, while both false positive and false negative values were reduced to zero. These findings confirm that the proper application of class balancing techniques such as SMOTE can significantly enhance the reliability and generalization capability of spam detection models.

Despite the excellent results, this study has several limitations. The proposed model uses a relatively simple algorithm (naïve Bayes) and a moderately sized dataset, which may limit its generalizability to larger, more diverse, or multilingual datasets. Furthermore, the perfect accuracy obtained under controlled experimental conditions may not fully reflect real-world environments, where spam content evolves dynamically and may include multimedia or AI-generated components.

For future work, it is recommended to evaluate the model using larger and more heterogeneous datasets, incorporating multilingual and image-based spam samples to test robustness. Future studies could also explore the integration of deep learning architectures, ensemble methods, or hybrid text representation techniques to enhance detection accuracy and adaptability. Additionally, implementing and testing the model in real-time email filtering systems would be a valuable step toward assessing its scalability, efficiency, and applicability in operational environments.

## CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

## AUTHORS' CONTRIBUTIONS

Conceptualization, Andi Maslan; methodology, Andi Maslan; software, Azan Rahman; validation, Andi Maslan; writing original draft preparation, Andi Maslan; writing reviewing and editing, Umar Faruq, Andi Maslan, and Rabei Raad Ali Al-Jawry; visualization, Andi Maslan; supervision, Rabei Raad Ali Al-Jawry and Azan Rahman; funding, Andi Maslan and Azan Rahman.

## ACKNOWLEDGMENT

The author would like to thank Universitas Putera Batam for the support and funding provided for this research. Without their help and strong commitment, this research would not have been successfully implemented. In addition, the author would like to express appreciation to all parties who have contributed, either directly or indirectly, to the research process.

## REFERENCES

[1] J. Hong, "The state of phishing attacks," *Commun. ACM*, vol. 55, no. 1, pp. 74–81, Jan. 2012, doi: 10.1145/2063176.2063197.

[2] D. Hylender, P. Langlois, A. Pinto, and S. Widup, "2024 Data Breach Investigations Report," The Verizon DBIR Team, California, CA, USA, 2024.

[3] World Economic Forum, "Global Cybersecurity Outlook 2024, 2024. [Online]. Available: https://www3.weforum.org/docs/WEF_Global_Cybersecurity_Outlook_2024.pdf

[4] R.G. Broadhurst and H. Trivedi, "Malware in spam email: Risks and trends in the Australian spam intelligence database," *Trends Issues Crime Crim. Justice*, no. 603, pp. 1–18, Sep. 2020, doi: 10.52922/ti04657.

[5] Slashnext, "The State of Phishing 2023," 2023. [Online]. Available: https://newsletter.radensa.ru/wp-content/uploads/2023/10/SlashNext-The-State-of-Phishing-Report-2023.pdf

[6] B.P.S. Brodjonegoro, "Pembangunan digital Indonesia ke depan," in Horizon Pembangunan Digital Indonesia 2025-2030, 1st ed., Nezar Patria, Ed., Jakarta, Indonesia: Kementerian Komunikasi dan Informatika, 2024.

[7] M. Thomas and B. Meshram, "A brief review of network forensics process models and a proposed systematic model for investigation," in *Intell. Cyber Phys. Syst. Internet Things ICoICI 2022,* 2023, pp. 599–627, doi: 10.1007/978-3-031-18497-0_45.

[8] B. Fakiha, "Enhancing cyber forensics with AI and machine learning: A study on automated threat analysis and classification," *Int. J. Saf. Secur. Eng.*, vol. 13, no. 4, pp. 701–707, Sep. 2023, doi: 10.18280/ijsse.130412.

[9] R. Al-Mugern, S.H. Othman, and A. Al-Dhaqm, "An improved machine learning method by applying cloud forensic meta-model to enhance the data collection process in cloud environments," *Eng. Technol. Appl. Sci. Res.*, vol. 14, no. 1, pp. 13017–13025, Feb. 2024, doi: 10.48084/etasr.6609.

[10] C.S. Eze and L. Shamir, "Analysis and prevention of AI-based phishing email attacks," *Electronics*, vol. 13, no. 10, pp. 1–13, May 2024, doi: 10.3390/electronics13101839.

[11] W.A. Baroto, "Advancing digital forensic through machine learning: An integrated framework for fraud investigation," *Asia Pac. Fraud J.*, vol. 9, no. 1, pp. 1–16, Jan.-Jun. 2024, doi: 10.21532/apfjournal.v9i1.346.

[12] N.N. Nicholas and V. Nirmalrani, "An enhanced mechanism for detection of spam emails by deep learning technique with bio-inspired algorithm," *e-Prime - Adv. Elect. Eng. Electron. Energy*, vol. 8, pp. 1–9, Jun. 2024, doi: 10.1016/j.prime.2024.100504.

[13] I.S.A. Munawar and E.T. Arujisaputra, "Comparison of the results of the naïve Bayes method and synthetic minority over sampling technique in sentiment analysis of user reviews," *J. Elektron. Sist. Inf.*, vol. 2, no. 1, pp. 179–188, Jun. 2024, doi: 10.31848/jesii.v2i1.3416.

[14] A. Falasari and M.A. Muslim, "Optimize naïve Bayes classifier using chi square and term frequency inverse document frequency for amazon review sentiment analysis," *J. Soft Comput. Explor.*, vol. 3, no. 1, pp. 31–36, Mar. 2022, doi: 10.52465/joscex.v3i1.68.

[15] B. Sonare *et al.,* "E-mail spam detection using machine learning," in *2023 4th Int. Conf. Emerg. Technol. (INCET),* 2023, pp. 1–5, doi: 10.1109/INCET57972.2023.10170187.

[16] G. Nasreen *et al.,* "Email spam detection by deep learning models using novel feature selection technique and BERT," *Egypt. Inform. J.*, vol. 26, pp. 1–11, Jun. 2024, doi: 10.1016/j.eij.2024.100473.

[17] M.M. Abualhaj *et al.,* "Enhancing spam detection using Harris Hawks optimization algorithm," *TELKOMNIKA Telecommun. Comput. Electron. Control,* vol. 23, no. 2, pp. 447–454, Apr. 2025, doi: 10.12928/TELKOMNIKA.v23i2.26615.

[18] M. Jaiswal, S. Das, and Khushboo, "Detecting spam e-mails using stop word TF-IDF and stemming algorithm with naïve Bayes classifier on the multicore GPU," *Int. J. Elect. Comput. Eng.*, vol. 11, no. 4, pp. 3168–3175, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3168-3175.

[19] A.C. Sitepu, Wanayumini, and Z. Situmorang, "Determining bullying text classification using naive Bayes classification on social media," *J. Varian*, vol. 4, no. 2, pp. 133–140, Apr. 2021, doi: 10.30812/varian.v4i2.1086.

[20] M. Thomas and B.B. Meshram, "An efficient spam mail classification approach using improved moth flame optimization and multi-class support vector machine," *Int. J. Intell. Eng. Syst.*, vol. 16, no. 6, pp. 813–823, 2023, doi: 10.22266/ijies2023.1231.67.

[21] C.N. Mohammed and A.M. Ahmed, "A semantic-based model with a hybrid feature engineering process for accurate spam detection," *J. Elect. Syst. Inf. Technol.*, vol. 11, pp. 1–16, Jul. 2024, doi: 10.1186/s43067-024-00151-3.

[22] A. Rajesh and T. Hiwarkar, "Exploring preprocessing techniques for natural languagetext: A comprehensive study using Python code," *Int. J. Eng. Technol. Manag. Sci.*, vol. 7, no. 5, pp. 390–399, Sep./Oct. 2023, doi: 10.46647/ijetms.2023.v07i05.047.

[23] M.R. Ningsih, J. Unjung, H.A. Farih, and M.A. Muslim, "Classification email spam using naive Bayes algorithm and chi-squared feature selection," *J. Appl. Intell. Syst. (JAIS),* vol. 9, no. 1, pp. 74–87, Apr. 2024, doi: 10.62411/jais.v9i1.9695.

[24] N. Tayefeh, "Email Spam," Harvard Dataverse, doi: 10.7910/DVN/V8BSBP.

[25] E.G. Dada *et al.,* "Machine learning for email spam filtering: Review, approaches and open research problems," *Heliyon*, vol. 5, no. 6, pp. 1–23, Jun. 2019, doi: 10.1016/j.heliyon.2019.e01802.

[26] O. Ogundairo and P. Broklyn, "AI-driven phishing detection systems," unpublished.

[27] B.A. Kumari and C. Nagaraju, "Robust machine learning technique for detection and classification of spam mails," unpublished.

[28] N. Borisova, E. Karashtranova, and I. Atanasova, "The advances in natural language processing technology and its impact on modern society," *Int. J. Elect. Comput. Eng. (IJECE),* vol. 15, no. 2, pp. 2325–2333, Apr. 2025, doi: 10.11591/ijece.v15i2.pp2325-2333.

[29] Z. Abidin, A. Junaidi, and Wamiliana, "Text stemming and lemmatization of regional languages in Indonesia: A systematic literature review," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 10, no. 2, pp. 217–231, Jun. 2024, doi: 10.20473/jisebi.10.2.217-231.

[30] A.B.P. Negara, "The influence of applying stopword removal and smote on Indonesian sentiment classification*," Lontar Komput., J. Ilm. Teknol. Inf.*, vol. 14, no. 3, pp. 172–185, Dec. 2023, doi: 10.24843/lkjiti.2023.v14.i03.p05.

[31] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends and challenges," *Multimed. Tools Appl.*, vol. 82, no. 3, pp. 3713–3744, Jan. 2023, doi: 10.1007/s11042-022-13428-4.

[32] A. Daud *et al.,* "Identification of chatbot usage in online store services using natural language processing methods," *Adv. Sustain. Sci. Eng. Technol.*, vol. 6, no. 2, pp. 1–9, Feb.-Apr. 2024, doi: 10.26877/asset.v6i2.18309.

[33] S. Atawneh and H. Aljehani, "Phishing email detection model using deep learning," *Electronics*, vol. 12, no. 20, pp. 1–26, Oct. 2023, doi: 10.3390/electronics12204261.

[34] F. Millennianita, U. Athiyah, and A.W. Muhammad, "Comparison of naïve Bayes classifier and support vector machine methods for sentiment classification of responses to bullying cases on Twitter," *J. Mechatron. Artif. Intell.*, vol. 1, no. 1, pp. 11–26, Jun. 2024, doi: 10.17509/jmai.v1i1.69959.