INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Enhancing Spam Email Classification Using Effective Preprocessing Strategies and Optimal Machine Learning Algorithms

Pramod P Ghogare[1]*, Husain H Dawoodi[2], Manoj P Patil[3]

1 Research Scholar, School of Computer Sciences, Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon, Maharashtra, India
2 System Analyst, School of Computer Sciences, Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon, Maharashtra, India
3 Assistant Professor, School of Computer Sciences, Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon, Maharashtra, India

OPEN ACCESS

Received: 22-11-2023
Accepted: 06-03-2024
Published: 12-04-2024

Citation: Ghogare PP, Dawoodi HH, Patil MP (2024) Enhancing Spam Email Classification Using Effective Preprocessing Strategies and Optimal Machine Learning Algorithms. Indian Journal of Science and Technology 17(15): 1545-1556. https://doi.org/10.17485/IJST/v17i15.2979

* **Corresponding author**.

pramod.ghogare@yahoo.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objective: This article proposes a content-based spam email classification by applying various text pre-processing techniques. NLP techniques have been applied to pre-process the content of an email to get the optimal performance of spam email classification using machine learning. **Method**: Several combinations of pre-processing methods, such as stopping, removing tags, converting to lower case, removing punctuation, removing special characters, and natural language processing, were applied to the extracted content from the email with machine learning algorithms like NB, SVM, and RF to classify an email as ham or spam. The standard datasets like Enron and SpamAssassin, along with the personal email dataset collected from Yahoo Mail, are used to evaluate the performance of the models. **Findings**: Applying stemming in pre-processing to the RF classifier yielded the best results, achieving 99.2% accuracy on the SpamAssassin dataset and 99.3% accuracy on the Enron dataset. Lemmatization followed closely with 99% accuracy. In real-world testing on a personal Yahoo email dataset, the proposed method significantly improved accuracy from 89.82% to 97.28% compared to the email service provider's built-in classifier. Additionally, the study found that SVM performs accurately when stop words are retained. **Novelty:** This article introduces a unique perspective by highlighting the fine-tuning of pre-processing techniques. The focus is on removing tags and certain special characters, while retaining those that improve spam email classification accuracy. Unlike prior works that primarily emphasize algorithmic approaches and pre-defined processing functions, our research delves into the intricacies of data preparation, showcasing its significant impact on spam email classifiers. These findings emphasize the crucial role of pre-processing and contribute to a more nuanced understanding of effective strategies for robust spam detection.

Keywords: Spam; Classification; Pre-processing; NLP; Machine Learning

1 Introduction

The advent of spam emails poses a challenge to the seamless functioning of email as a communication channel. Spam emails often contain product details, advertisements, discount offers, and solicit donations for promotional purposes. The incessant intrusion of such unsolicited emails hampers user productivity. Filtering out spam emails is imperative to conserve resources, time, and money otherwise consumed by these unwanted messages. As the spam email industry evolves in complexity, the adaptability of spam filters becomes crucial in effectively countering these unforeseen challenges. In recent years researchers have extensively used machine learning for spam e-mail classification To work properly, machine learning classifiers rely heavily on well-prepared training data. In order to optimize output, it is critical to minimize noise and show input data in a refined manner. This strategy, known as pre-processing, greatly enhances the resilience, precision, and competency of machine learning classifiers when generalizing to unknown data. This key stage increases the overall performance of a machine learning model by increasing its ability to discover meaningful patterns within input information.

Choosing the best pre-processing mix is one of the most important decisions in spam email classification using machine learning. Instead of settling on a single preprocessing method, the authors of[(1)] suggested experimenting with different combinations. This is because different combinations may be beneficial or detrimental depending on factors such as the domain, dataset, machine learning method, and Bag-of-Words size. They used six different processing methods, including deleting HTML tags, removing punctuation marks, eliminating stop words, and performing spelling correction. All of these algorithms, however, were not evaluated on the standard email dataset; instead, they were tested on WebKB, R8, SMS spam collection, and sentiment-labeled words.

Diale et al. (2019) investigated excessive features in machine learning classifiers, which can harm performance. A preprocessing stage involving feature extraction and reduction is proposed to improve computation speed and classification accuracy. The study focuses on data transformation before applying classifiers such as Random Forests, Support Vector Machines, and C4.5 decision trees. The proposed feature representation preserves class separability in a lower-dimensional space, allowing for effective identification of spam or non-spam emails with a small feature size[(2)]. This research focuses on feature extraction and reduction. However, it does not go into detail about potential additional preparation stages such as text cleaning or tokenization, which could affect the overall results. The authors focus primarily on improving the kernel function to achieve optimal classifier performance. In addition, they use strategies such as converting words to lowercase, deleting stop words, and performing stemming analyses to assess the effectiveness of their feature selection methods.

The authors in [(3)] has found the application of machine learning and natural language processing (NLP) techniques proves highly effective in email spam classification. Employing supervised learning algorithms like Naive Bayes, Support Vector Machines, and KNN, coupled with preprocessing methods such as tokenization, stop-word removal, and stemming, enables the creation of precise and dependable spam filters. The versatility of these techniques extends to handling more intricate spamming strategies like phishing attacks and spear phishing. The integration of machine learning and NLP in email spam classification not only saves users valuable time and resources but also enhances the overall productivity and security of email communication. The model does not test the model using actual email content; instead, the author passes the test string to the model after training on a dataset and receiving the results as spam or ham for the entered string.

Similarly, Urmi et al. (2022) investigated SMS spam detection using machine learning and discovered that the Random Forest classifier is ideal for SMS spam classification and can also be used for spam email classification. The authors tested five machine learning classifiers, and RF produced the highest accuracy[4].

Chakraborty et al. created a web application that uses supervised machine learning to classify spam emails. The model, which includes Training and Testing phases, uses a dataset of approximately 5735 emails and requires data cleaning for quality assurance. Input messages are processed using NLTK, which includes tokenization, sentence consideration, and vector conversion. Feature extraction produces word clouds, which visually highlight frequent words in spam and legitimate categories. During the dynamic testing phase, the model computes probabilities for spam and non-spam, then selects the result based on the highest probability category. Deployed as a user-friendly web application, it allows input for real-time evaluation and classification, demonstrating the model's adaptability and practical application in spam email classification. In the developed web application the predefined NLTK library was used for pre-processing[5].

Moutafis et al. (2023) propose using various machine learning classifiers to classify raw emails as spam or benign. A CSV file with email attributes is preprocessed to create a training set for 10 classifiers, including Support Vector Machines, k-nearest Neighbors, Naive Bayes, and Neural Networks[6]. The model presented in this article differs from the one developed in[6], which is based on externally generated CSV files and produces a full data frame with sender details, subject, content, and sender country information. In contrast, the approach presented here creates a data frame directly from each email file in the dataset, focusing only on the email body. Origin-based data that is not considered includes email subjects, sender information, and country names.

Goyal et al. used three supervised machine learning methods: Gaussian Naïve Bayes (GNB), Multinomial Naïve Bayes, and Bernoulli Naïve Bayes to detect fake reviews. The GNB classifier outperforms other models in terms of accuracy and F1-score metric, as well as identifying deceptive reviews[7]. The authors used the NLTK library to clean up the review data, which is a predefined functionality, so the pre-processing methodology is not novel. Custom methods cannot be implemented in any library due to its predefined functionality. The results of the predefined library function may differ on different types of datasets, so it is critical to use a customised pre-processing methodology. This article can help you decide which type of pre-processing methodology to use based on the type of data you have, as multiple methods have been tested on different datasets.

The authors of[8] concluded that preprocessing precision can significantly improve categorization results. They used Naive Bayes and SVM to investigate the effects of noise reduction, stop-word removal, stemming, lemmatization, and term frequency on classification results. To improve accuracy, each classifier should undergo specific preprocessing steps. The combination of stop-word removal and stemming, for example, improved results for the Nave Bayes classifier, whereas the SVM classifier did not require extensive preprocessing. The authors argued for more research into different preprocessing methods and classifier type combinations in the domain of email spam detection, emphasizing the importance of larger datasets for robust evaluation.

The primary goal of developing an anti-spam strategy is to identify the most effective combination of pre-processing algorithms for categorising spam emails based solely on their content, without regard for origin or sender information. The HTML tags, special characters, and formatting artifacts are examples of excessive or noisy elements in emails. Furthermore, the inclusion of various word forms, such as plurals and verb conjugations, could increase the dataset's dimensionality. The proposed methodology enhances pre-processing by combining it with natural language processing techniques, thereby improving spam email classification accuracy. We test the effectiveness of this strategy with a variety of machine learning classifiers and datasets, as well as real-world scenarios, to determine its practical applicability. The appropriate pre-processing workflow has also been thoroughly investigated. Precision in pre-processing has significantly helped to improve classification outcomes.

2 Material and Methods

2.1 Proposed framework:

The proposed architecture for machine learning-based spam email categorization is depicted in Figure 1. Initially, a dataset is created, the text is retrieved from the email files from each dataset to gather data for pre-processing, and several pre-processing techniques are used. The captured text is modified before being sent to machine learning algorithms. Finally, three machine-learning approaches are used to classify emails.

2.2 Data Collection

The email dataset was sourced from various origins, including SpamAssassin, Enron, and emails extracted from personal Yahoo mailboxes, with specific details provided in Table 1. The dataset consists of email files that contain the complete email content in text format, organized systematically in directories.

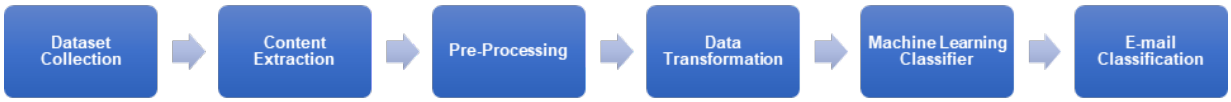

Fig 1. Proposed Framework for Spam E-mail Classification

Table 1. Dataset Details

Dataset	Period	Type of E-mail	Number of E-mails	Total E-mails
SpamAssassin (https://spamassassin.apache.org/old/publiccorpus/)	29-06-2004 to 11-03-2005	Spam	2398	9349
		Non-spam	6951	
Enron (https://www.cs.cmu.edu/~enron/)	10-12-1999 to 06-09-2005	Spam	20988	46932
		Non-spam	31087	
Emails from personal Yahoo mailbox	11-06-2008 to 10-02-2023	Spam	7800	17480
		Non-Spam	9680	

2.3 Text Extraction

The email contains two parts: the header and the body. The header contains metadata about the email, such as tracing information, sender information, the email's unique ID, the format of the email body, and so on. The header part is not considered for classification in this study. The classification is based on the content/body of the email. The body of the e-mail contains the actual data that describes the purpose of the e-mail. Content-based filters extract the email's body, which is then processed and passed to the classifier for classification. The e-body mail's contains data in the form of characters, words, HTML tags, special symbols, and punctuation. The body of the email is extracted and routed to the next stage of pre-processing.

2.4 Data Pre-processing Methods

Typically, data in e-mail content contains noise and missing values. To effectively classify email as spam or ham, quality data must be mined effectively. To reduce processing time and storage requirements, e-mails must be translated prior to classification tasks, and e-mails from corpora are pre-processed to transmute and make them suitable for classification. Pre-processing consists of several steps, including data cleansing, amalgamation, makeover, and data reduction. During this preparation phase, punctuation and null characters are removed and the text is converted to lowercase. Special symbols such as []'> ():'/! # % @ & * _ + = -; have been eliminated to prevent excessive processing time. In this paper stop-word removal, stemming, and lemmatization are used as pre-processing techniques, these techniques are explained as follows.

2.4.1 Stopping

Stopping is the process of deleting Stop words from a language. Stopwords in the English language, for example, are "the", "a", "is", "are", and so on. By removing less valuable words from the text, the powerful words may be highlighted appropriately. The elimination of stopwords decreases future sets, which aids in keeping models at a manageable size.

2.4.2 Stemming

Words are reduced to their stems in the stemming procedure. For example, stemming reduces the terms "consulting", "consultanr", and "consultants" to the root word "consult". Over-stemming occurs when too much of a word is removed, resulting in meaningless stems and the loss of the term's meaning. Words like "universal", "universities", "university", and "universe" are all derived from the basic word "univers", which has no meaning. Under stemming, one word has several forms, and all of these forms should stem from the same word, however, this does not happen.

2.4.3 Lemmatization

It is the process of finding a word's lemma based on its intended meaning. It is contingent on correctly detecting the intended meaning of a word in a phrase, neighboring sentence, or document. For example, the verb 'to talk' might occur as 'speaking', 'speak', 'speaks', and the word 'speak' which is known as the lemma. Lemmatization is the process of identifying the root form of words. After applying the lemmatization to the e-mail content, numerous words were discovered to be changed into the base

form. For example, 'allocating' becomes 'allocate' and 'affected' becomes 'affect'. These word bases can help the machine learning classifier do better classification. Lemmatization is like stemming in that the goal is to eliminate variants and transform a word into its root form. Lemmatization does more than merely chop things off; it also changes words to their roots. For example, "best" would be mapped to "good".

To inspect the consequence of stopping, stemming, and lemmatization we studied four different combinations of preprocessing.

Without Pre-processing (WPre) – In this method, no preprocessing was applied to the content extracted from the e-mails. Content is directly forwarded to classification algorithms.

Pre-processing/Stopping (Pre) – The content extracted from e-mail files and stop words, special characters, symbols as ~, []'<>():{'}\/!#%^@&*()_+=-;. and new line characters were removed from the content without applying natural language processing.

Preprocessing with Lemmatization (Pre+L) – With the preprocessed content, lemmatization was applied. Lemmatization is the process that converts the term into its base form.

Pre-processing with Stemming (Pre+S) - With the pre-processed content stemming was applied. Stemming is the method that reduces words to short, and it is just a minor form of the word.

2.5 Data Transformation

Machine learning classifiers require data in a specified format; pre-processed material is included in the data frame per classifier requirements. The data frame stores the data in two columns: the first column contains the pre-processed text, and the second column contains the e-mail's label as non-spam and spam. This data frame is fed into machine learning algorithms with various parameter values for spam e-mail classification.

2.6 Classification

Several preprocessing methods were used to create the data frame prior to spam e-mail classification. The classifiers listed below are trained on 80% of emails and tested on 20% of emails. Machine learning is critical in spam email classification because it trains on large datasets to differentiate spam from legitimate emails, automating the filtering process and allowing for efficient real-time identification. The machine learning techniques are explored to improve spam classification, reducing false positives and false negatives, thereby enhancing user experience and email security[9]. These algorithms can automatically learn patterns and features of spam emails from extensive datasets, enabling them to make precise predictions on new, unfamiliar emails.

2.6.1 Naïve Bayes (NB)

The Naive Bayes classifier method is founded on the Bayesian theorem. The Naïve Bayes calculates the conditional possibilities of classes for a given instance and predicates the class with a higher probability. The basic concept of NB is considered those tokens present in the content, according to the tokens present in the e-mail classification as spam or non-spam. The evidence obtained from the training phase is used to estimate the probability. According to the probability, the testing e-mail is classified as spam or non-spam.

$$P(c|\overrightarrow{x}) = \frac{P(c) \bullet P(\overrightarrow{x}|c)}{P(\overrightarrow{x})} \tag{1}$$

For token vector $\overrightarrow{x} = \{x_1, x_2, x_3, \ldots, x_n\}$ of an e-mail, where $x_1, x_2, x_3, \ldots, x_n$ are the values of features, and n is the number of e-mails in the dataset. Every attribute is considered as a token present or not. Let c signify the class of e-mail to be predicted. The $P(c|\overrightarrow{x})$is the chance that $\overrightarrow{x}$ fits class c is as shown in Equation (1) where $P(\overrightarrow{x})$ means the probability of an arbitrarily chosen e-mail represented by the vector $\overrightarrow{x}$. $P(c)$ is a prior possibility of class c. $P(\overrightarrow{x}|c)$ denotes the probability of a randomly picked e-mail with class c has $\overrightarrow{x}$ as its representation, this classifier is called naïve Bayes because 'naïve' means each term is expected to befall self-sufficiently from the other.

Notably, Naïve Bayes exhibits fast convergence, quickly learning from limited training data and making accurate predictions. Its computational efficiency makes it practical for real-time or online spam email filtering. Despite its simplicity, Naïve Bayes often performs remarkably well, especially when the dataset is well-prepared, and features are thoughtfully selected. Hence, it remains a popular choice due to its ease of implementation and reliable real-world performance[10].

2.6.2 Support Vector Machine (SVM)

Support Vector Machines (SVM) is a supervised machine learning method used for classification. The SVM treats data as points, and these points are arranged so that the variance between adjacent points is extreme. The testing instances are then arranged on the graph based on which side of the margin can be identified. The best separation has more remoteness among nearby data points, whereas a larger margin indicates a minor generalisation error. Testing points are classified into classes based on their margin.

Support Vector Machines (SVMs) are preferred for spam email classification due to their ability to handle high-dimensional data, with each feature representing terms within emails. Their ability to model non-linear decision boundaries is critical for handling complex patterns in spam emails and ensuring accurate classification. SVMs improve generalisation performance by increasing the margin between classes, making them less prone to overfitting. Furthermore, SVMs show resilience in handling imbalanced datasets commonly encountered in spam classification, prioritising the definition of decision boundaries over the overall class distribution.

2.6.3 Random Forest (RF)

It is an ensemble learning classifier that can handle problems with data grouping into classes. During the training phase, numerous decision trees are built, and those trees are used for prediction by taking the most voted classes from all of the individual trees. Random Forest emerges as an ensemble learning method for spam email classification, combining the predictions of multiple decision trees to support the overall model's efficacy and resilience to overfitting and data noise. This collective approach improves the model's robustness by reducing overfitting by smoothing out data noise and outliers. Their ability to handle imbalanced datasets, a common challenge in spam email classification, ensures a balanced and accurate prediction by taking into account the distribution of spam and non-spam classes.

2.7 Evaluation metrics

The performance of spam e-mail classification is measured using the following performance metrics.

Table 2. Confusion matrix

Classifier	Spam	Non-spam
Spam	Spam e-mails are classified as spam. (True-Positive)	Spam e-mails are classified as non-spam. (False-Negative)
Non-spam	Non-spam e-mails are classified as spam. (False-Positive)	Non-spam e-mails are classified as non-spam. (True-Negative)

$$Accuracy = \frac{TP + TN}{TP+FP+FN+TN} \times 100 \tag{2}$$

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

$$F-Score = \frac{2.Precision.Recall}{Precision+Recall} \tag{5}$$

The classification accuracy is a metric to test the classifier's performance, defined as the proportion of the total number of correctly classified spam and non-spam e-mails to the total number of emails. Accuracy fails to talk about misclassification, and f-score is used to avoid this problem. The f-score is the harmonic sum of recall and precision. Misclassification is a severe issue in spam e-mail classifiers, especially when a non-spam is misclassified as spam e-mail. A spam e-mail is misclassified as legitimate and shown in the inbox that can be simply detected and rectified. The lower FPR confirms the maximum accurate classification.

3 Results and Discussion

The following section discusses the experimental results of spam email classification on SpamAssassin and Enron datasets using pre-processing techniques.

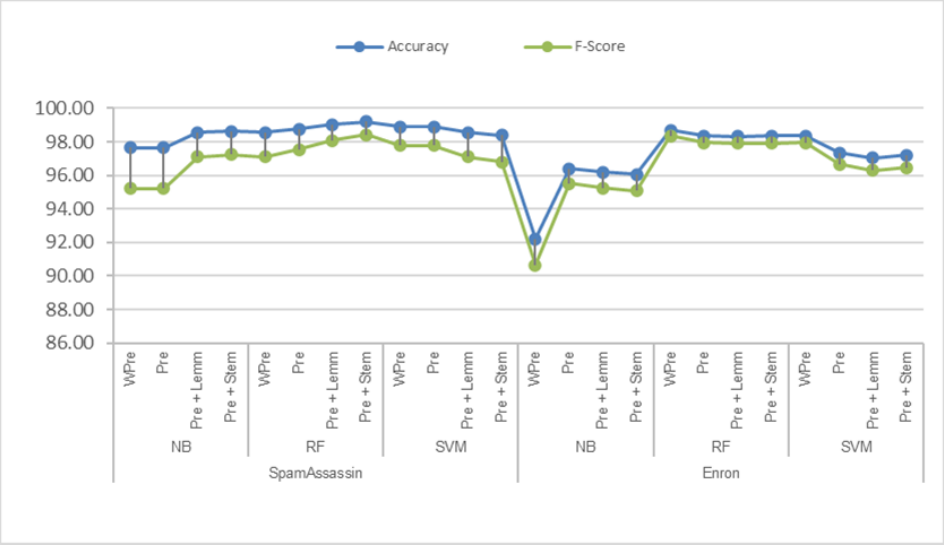

Fig 2. Comparison of Spam Assassin and Enron Datasets

3.1 SpamAssassin Dataset

Table 3. Classification Results on SpamAssassin Dataset

Pre-Processing Method	Classifier	Accuracy	Precision	Recall	F-score
	NB	97.65	98.64	91.98	95.20
Wpre	SVM	98.88	98.50	97.05	97.77
	RF	98.56	98.91	95.36	97.10
	NB	97.65	98.64	91.98	95.20
Pre	SVM	98.88	98.50	97.05	97.77
	RF	98.77	98.92	96.20	97.54
	NB	98.56	98.06	96.20	97.12
Pre+L	SVM	98.56	98.27	95.99	97.12
	RF	99.04	98.93	97.26	98.09
	NB	98.61	98.28	96.20	97.23
Pre+S	SVM	98.40	98.26	95.36	96.79
	RF	99.20	99.14	97.68	98.41

Figure 2 and Table 3 show the results on the SpamAssassin dataset by applying various text pre-processing techniques and carrying out spam email classification using NB, RF, and SVM.

When analyzing email content, the performance of Support Vector Machines (SVM) is notably effective in the presence of stop words and a diverse range of word types. This effectiveness stems from the fact that stop words are infrequently found in spam emails. The SVM strategically transforms the input space into a linearly separable one by eliminating irrelevant features while retaining pertinent stop words. In the context of non-probabilistic classifiers for classification, it becomes essential to preserve certain stop words in the text to enhance the efficacy of spam filtering. Despite their inherent lack of significance, some stop words are uncommon in spam emails, contributing to improved classification. Consequently, stemming and stop

word removal are rarely employed as pre-processing steps for SVM.

On the other hand, Natural Language Processing (NLP) emerges as a preferred choice for pre-processing when employing the Naive Bayes classifier. This preference is attributed to the effectiveness of stemming and lemmatization, processes that reduce words to their fundamental forms. The conversion of multiple word forms into a singular form enhances the frequency and probability of specific phrases. In the experimental setup, Multinomial Naive Bayes, considering word frequency, outperforms Bernoulli Naive Bayes, which focuses solely on the presence or absence of individual words.

As the preprocessing phase advances, the Random Forest (RF) classifier exhibits improved accuracy. The integration of NLP preprocessing proves beneficial for RF, contributing to enhanced accuracy and F-score. The RF model generates an internally balanced forecast by addressing combined errors during forest cultivation, mitigating mistakes in population class prediction. This internal balancing mechanism is pivotal in ensuring higher accuracy and minimizing misclassifications. NLP plays a crucial role in this process by eliminating various word forms, non-essential letters, and symbols, thereby assisting RF in rectifying errors during the forest farming process.

3.2 Enron Dataset

Table 4. Classification Results on Enron Dataset

Pre-processing Method	Classifier	Accuracy	Precision	Recall	F-Score
	NB	92.22	86.85	94.81	90.66
Wpre	SVM	98.36	98.00	97.88	97.94
	RF	98.69	98.76	97.95	98.35
	NB	96.41	94.92	96.12	95.52
Pre	SVM	97.35	96.85	96.48	96.66
	RF	98.37	98.21	97.68	97.94
	NB	96.19	94.43	96.09	95.25
Pre+L	SVM	97.05	96.58	96.00	96.29
	RF	98.33	98.06	97.73	97.90
	NB	96.05	94.18	96.02	95.09
Pre+S	SVM	97.19	96.75	96.16	96.45
	RF	98.35	98.04	97.80	97.92

Figure 2 and Table 4 showcase the classification outcomes for spam email classification on the Enron dataset. When preprocessing is conducted without Natural Language Processing (NLP), the accuracy of the Naive Bayes (NB) classifier improves due to a reduction in the number of words. The ”Pre” method proves suitable for NB classifiers, contributing to enhanced accuracy and diminished misclassification. Conversely, the Support Vector Machine (SVM) classifier exhibits superior performance with the ”WPre” method. Despite stemming and lemmatization reducing the dimension of content, they do not yield significant performance improvements. However, the utilization of the ”Pre” procedure leads to a noteworthy enhancement in performance.

Ensemble-based methods, such as RF, demonstrate their efficacy in classifying spam emails by leveraging a larger dataset and employing diverse pre-processing methods. This approach results in higher accuracy, reduced misclassification, and minimal variation. The observed gap underscores the disparities in accuracy and f-score between smaller and larger datasets. Leveraging the larger dataset proves advantageous in training the classifier for consistent and improved accuracy and f-score outcomes. The thoughtful selection of pre-processing techniques, tailored to the specific dataset and language, holds the potential to substantially elevate classification accuracy. Conversely, certain demonstrated pre-processing combinations may lead to a decline in accuracy. For example, employing lowercase conversion contributes to enhanced accuracy and dimension reduction. However, it's essential to recognize that there is no universally effective combination of pre-processing steps that guarantees optimal classification results across all datasets.

From Tables 3 and 4, the observed differences in precision between pre-processing methods can be attributed to how each method influences the quality and relevance of the features used by machine learning algorithms. In the context of spam email classification, the pre-processing method appears to be the most precise for NB. This suggests that the transformations used during preprocessing improve the algorithm's ability to correctly identify and classify spam emails, resulting in a higher precision rate.

Similarly, the use of lemmatization and stemming refines the text data, assisting in the extraction of important information and potentially reducing noise. These nuanced linguistic pre-processing techniques help to improve precision rates, as evidenced by the slightly lower but still significant precision values for NB. In contrast, the Without Pre-processing method produces the lowest precision, indicating that raw or minimally processed data may contain noise or irrelevant information, impairing the algorithm's ability to accurately distinguish spam from non-spam emails.

Pre-processing methods' effectiveness is influenced by the specific characteristics of the datasets, as well as the nature of the machine learning algorithms. In the case of SVM and RF, different pre-processing strategies, such as the RF, contribute to increased precision. This emphasises the importance of tailoring pre-processing steps to the specific requirements and characteristics of the data and algorithms used, resulting in optimal classification performance for spam emails. Pre-processing strategies, such as lemmatization and stemming, are highlighted for their importance in improving the overall effectiveness of machine learning algorithms in spam email classification tasks.

3.3 Comparative Analysis

The proposed models are compared to state-of-the-art methodologies used by various researchers in literature.

3.3.1 SpamAssassin Dataset

Table 5. Comparison of the SpamAssassin Dataset

Paper	Year	Pre-processing	NB		SVM		RF	
			Accuracy	F-score	Accuracy	F-score	Accuracy	F-score
Bindu et al.[11]	2016	Pre	87.8	75.0	98.1	96.0	99.9	97.0
Trivedi et al.[12]	2018	Pre+L	96.5	96.6	97.8	97.8	97.6	97.6
Mourafis et al.[6]	2023	Pre	91.3	-	99.1	-	98.2	-
Proposed framework	2023	WPre	97.6	95.2	98.8	97.7	98.5	97.1
		Pre	97.6	95.2	98.8	97.7	98.7	97.5
		Pre+L	98.5	97.1	98.5	97.1	99.0	98.0
		Pre+S	98.6	97.2	98.4	96.7	99.2	98.4

Table 5 provides a comparative assessment of spam email classification outcomes, underscoring the effectiveness of the proposed framework in contrast to other studies. The proposed framework consistently delivers superior performance across various pre-processing approaches. Furthermore, when incorporating pre-processing methods such as lemmatization and stemming, the proposed framework continues to excel, achieving a 98.5% accuracy and a 97.1% F-score for pre-processing and lemmatization and a 98.6% accuracy and a 97.2% F-score for pre-processing and stemming.

Additionally, when considering the Random Forest (RF) algorithm, the proposed framework consistently demonstrates exceptional performance across various pre-processing methods. Without pre-processing, it achieves an impressive accuracy of 98.5% and an F-score of 97.1%, Furthermore, by incorporating pre-processing techniques like lemmatization and stemming, the framework's accuracy and F-scores are further enhanced, with pre-processing with stemming yielding an exceptional accuracy of 99.2% and an F-score of 98.4%. These findings underscore the substantial efficiency and advancement of the proposed framework in spam email classification, positioning it as a highly effective approach, particularly when utilizing the RF algorithm, compared to earlier research endeavors.

3.3.2 Enron Dataset

Table 6 illustrates the comparative analysis of classification outcomes between the framework proposed in this study and the prior research efforts conducted on the Enron dataset. Notably, the accuracy and f-score achieved by the novel architecture surpass those reported in previous works. About[13], Mohammad et al. introduced an approach called Ensemble-based Lifelong Classification using Adjustable Dataset Partitioning (ELCADP), which exhibited superior results to individual Naïve Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF) methods. Nevertheless, the framework presented in Section 2 of this study outperforms ELCADP, suggesting that dataset partitioning might adversely affect SVM accuracy. While NB with dataset partitioning can marginally outperform the proposed framework, it's important to note that the framework's results are

achieved without feature selection.

Table 6. Comparison of the Enron Dataset

Paper	Year	Pre-prcessing	NB		SVM		RF	
			Accuracy	F-score	Accuracy	F-score	Accuracy	F-score
Trivedi et al.[12]	2018	Pre+L	92.8	92.8	93.7	93.7	93.8	93.8
Gaurav et al.[14]	2019	Pre+L	55.1	68.8	-	-	92.7	85.0
Mohammad et al.[13]	2020	WPre	92.0	92.0	89.9	90.7	-	-
Mourafis et al.[6]	2023	Pre	98.3	-	99.3	-	97.2	-
Proposed framework	2023	WPre	92.2	90.6	98.3	97.9	98.6	98.3
		Pre	96.4	95.5	97.3	96.6	98.7	97.5
		Pre+L	96.1	95.2	97.0	96.2	99.3	98.0
		Pre+S	96.0	95.0	97.1	96.4	98.3	97.9

In terms of pre-processing strategies, the proposed study indicates a preferable approach that greatly improves performance on both datasets. To improve accuracy, it is advised to systematically investigate a wide range of pre-processing approaches in conjunction with extensive classification results. Rather than focusing on a single type of pre-processing, it is best to evaluate several combinations because their efficacy might vary depending on factors such as dataset characteristics, machine learning approaches, and feature dimensions. These results clearly indicate the effectiveness of the Proposed Framework in surpassing the performance of[12] model when employing Lemmatization as a pre-processing method in conjunction with the RF classifier.

In comparison, the Proposed Framework with[6], these results, while slightly lower in accuracy than SVM-based approach, demonstrate the competitiveness of the Proposed Framework, especially considering the inclusion of F-scores, which provide a more comprehensive evaluation of classifier performance. However, it's important to note that their approach was applied to a limited subset of the available datasets, whereas our proposed classifiers underwent across the entire available dataset.

In summary, the comparative analysis emphasizes the significant progress made in spam email classification by the Proposed Framework. It achieves substantially higher accuracy, and F-score results than the earlier studies. Additionally, the Proposed Framework's comprehensive evaluation using F-scores provides a more holistic view of its classification performance, further highlighting its effectiveness in the field of spam email classification.

To assess the practical applicability of the proposed method, the proposed models were subjected to a trial run on personal Yahoo mailbox dataset described in section 2.2. Figure 3 provides the classification results, specifically focusing on email content analysis. These results provide a detailed assessment of the classification outcomes, highlighting the strengths and weaknesses of each classifier in the context of email content analysis.

Yahoo in-built Classifier achieved an accuracy rate of 89.82%, indicating its incapability to correctly categorize emails. However, it exhibited a relatively lower recall value of 78.92%, implying that it may have missed some spam emails. On the other hand, its precision rate was noteworthy at 97.86%, suggesting minimal false positives. Consequently, the F-score, a measure balancing precision and recall, reached a moderate 87.38%, indicating an overall decent performance. In contrast, the proposed method displayed remarkable results, emphasizing the advantages of pre-processing. It achieved an impressive accuracy of 97.28% and an exceptionally high recall rate of 95.30%, showcasing its proficiency in identifying spam emails. Moreover, it demonstrated exceptional precision at 98.70%, resulting in an impressive F-score of 96.97%. These metrics collectively underscore the substantial advantages of pre-processing techniques, particularly in the context of the "Proposed Classifier," which excelled in accurately distinguishing between spam and non-spam emails while maintaining a notably high level of precision.

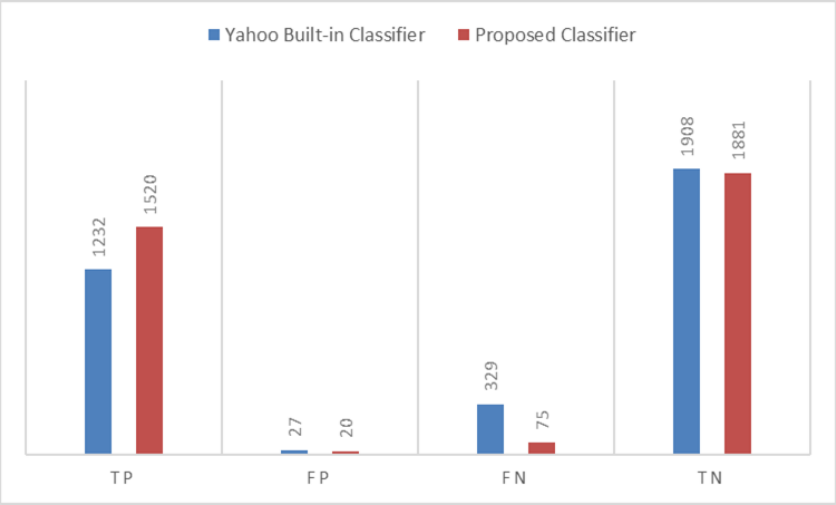

Fig 3. Yahoo built-in Classifier Vs Proposed Classifier

4 Conclusion

Addressing the issue of identifying spam emails is crucial, and the utilization of machine learning algorithms is vital for effective spam email classification. This research extensively evaluates three machine learning approaches for this purpose: Random Forest (RF), Support Vector Machine (SVM), and Naïve Bayes (NB). The proposed method, incorporating fine-tuned pre-processing with Natural Language Processing (NLP), enhances classification accuracy when compared to results obtained without pre-processing. The NLP Pre-processing technique with stemming applied to Random Forest classifier demonstrated best results achieving 99.2% accuracy on the SpamAssassin dataset, and 99.3% accuracy on the Enron dataset followed by pre-processing techniques of lemmatization having 99.0% accuracy. The proposed technique's customized pre-processing method is novel; previous research used predefined libraries to perform. Not only is customized pre-processing used, but it is also demonstrated which pre-processing method is most appropriate for the type of dataset. Conducting experiments on a personal email dataset reveals the model's efficacy in classifying spam emails, providing a promising indication of its suitability for real-world email environments. The text pre-processing techniques have greatly enhanced the accuracy of the spam email classification as compared to the results when pre-processing is not applied. While Yahoo's built-in classifier achieves an accuracy of 89%, the proposed technique demonstrated a remarkable 97% accuracy. In the future, this study can be extended by incorporating deep learning techniques to further enhance the overall accuracy of spam email classification.

5 Compliance with Ethical Standards

This research involved the use of both personal and publicly available datasets and strictly adhered to ethical principles and legal standards governing the handling of personal data. We meticulously addressed the following ethical considerations:

Informed Consent: This study involved the utilization of a personal email dataset belonging to the author for research purposes. The author, as the owner of the personal email dataset, provided explicit informed consent for its use in this research. The data within the personal email dataset were shared voluntarily by the author for research purposes and were handled with utmost care, privacy, and confidentiality.

Transparency and Accountability: Our research process prioritizes transparency and accountability. Within this article, we provide a comprehensive account of the methods, algorithms, and techniques employed in the experiment, ensuring transparency for the research community. We encourage further research and collaboration in this domain using publicly available data and the personal dataset, which is accessible upon reasonable request.

6 Competing Interests

The authors affirm the absence of conflicts of interest that could potentially impact the objectivity, methodology, or outcomes of this research. This study was conducted with the primary goal of advancing the field of spam email classification and enhancing email security.

7 Research Data Policy and Data Availability Statements

While a significant portion of the data is publicly accessible, the personal dataset supporting the findings of this study can be made available upon reasonable request to the corresponding author.

References

1) Hacohen-Kerner Y, Miller D, Yigal Y. The influence of preprocessing on text classification using a bag-of-words representation. *PLOS ONE*. 2020;15(5):1–22. Available from: https://doi.org/10.1371/journal.pone.0232525.
2) Diale M, Van Der Walt C, Celik T, Modupe A. Feature selection and support vector machine hyper-parameter optimisation for spam detection. In: 2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech). IEEE. 2017. Available from: https://doi.org/10.1109/RoboMech.2016.7813162.
3) Reddy A, Reddy KH, Abhishek A, Manish M, Dattu GVS, Ansari NM. Email Spam Detection Using Machine Learning. *Journal of Survey in Fisheries Sciences*. 2023;10(1):2658–2664. Available from: http://sifisheriessciences.com/index.php/journal/article/view/1249.
4) Urmi AS, Ahmed MT, Rahman M, Islam AT. A Proposal of Systematic SMS Spam Detection Model Using Supervised Machine Learning Classifiers. In: Computer Vision and Robotics. Algorithms for Intelligent Systems;Singapore. Springer. 2022;p. 459–471. Available from: https://doi.org/10.1007/978-981-16-8225-4_35.
5) Chakraborty A, Das UK, Sikder J, Maimuna M, Sarek KI. Content Based Email Spam Classifier as a Web Application Using Naïve Bayes Classifier. In: International Conference on Intelligent Computing & Optimization: ICO 2022;vol. 569 of Lecture Notes in Networks and Systems. Springer, Cham. 2023;p. 389–398. Available from: https://doi.org/10.1007/978-3-031-19958-5_36.
6) Moutafis I, Andreatos A, Stefaneas P. Spam Email Detection Using Machine Learning Techniques. In: Proceedings of the 22nd European Conference on Cyber Warfare and Security, ECCWS 2023;vol. 22, No. 1 of European Conference on Cyber Warfare and Security. Academic Conferences International Ltd. 2023;p. 303–310. Available from: https://papers.academic-conferences.org/index.php/eccws/article/view/1208.
7) Goyal NK, Pal A, Keswani B, Goyal DK, Gupta MK. A Novel Hybrid Feature Extraction Technique and Spam Review Detection using Ensemble Machine Learning Algorithm by Web Scrapping. *Indian Journal Of Science And Technology*. 2023;16(29):2261–2268. Available from: https://doi.org/10.17485/IJST/v16i29.1500.
8) Ruskanda FZ. Study on the Effect of Preprocessing Methods for Spam Email Detection. *Indonesian Journal on Computing (Indo-JC)*. 2019;4(1):109–118. Available from: https://doi.org/10.21108/INDOJC.2019.4.1.284.
9) Sohrab H, Abtahee A, Kashem I, Hoque MM, Sarker IH. Crime Prediction Using Spatio-Temporal Data. In: International Conference on Computing Science, Communication and Security, COMS2 2020;vol. 1235 of Communications in Computer and Information Science. Singapore. Springer. 2020;p. 277–289. Available from: https://doi.org/10.1007/978-981-15-6648-6_22.
10) Rusland NF, Wahid N, Kasim S, Hafit H. Analysis of Naïve Bayes Algorithm for Email Spam Filtering across Multiple Datasets. In: International Research and Innovation Summit (IRIS2017) ;vol. 226 of IOP Conference Series: Materials Science and Engineering. IOP Publishing. 2017;p. 1–9. Available from: https://iopscience.iop.org/article/10.1088/1757-899X/226/1/012091.
11) V B, Thomas C. Performance evaluation of classifiers for spam detection with benchmark datasets. In: 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE). IEEE. 2016. Available from: https://doi.org/10.1109/SAPIENCE.2016.7684121.
12) Trivedi SK, Dey S. A Modified Content-based Evolutionary Approach To Identify Unsolicited Emails. *Knowledge and Information Systems*. 2019;p. 1427–1451. Available from: https://doi.org/10.1007/s10115-018-1271-1.
13) Mohammad RMA. A lifelong spam emails classification model. *Applied Computing and Informatics*. 2024;20(1/2):35–54. Available from: https://www.emerald.com/insight/content/doi/10.1016/j.aci.2020.01.002/full/pdf?title=a-lifelong-spam-emails-classification-model.
14) Gaurav D, Tiwari SM, Goyal A, Gandhi N, Abraham A. Machine intelligence-based algorithms for spam filtering on document labeling. *Soft Computing*. 2020;24(13):9625–9638. Available from: https://doi.org/10.1007/s00500-019-04473-7.