Spam Email Detection using Naïve Bayes classifier

Liansong Wang

BASIS International School Park Lane Harbour, 516000 Shenzhen, China

Abstract. Spam email detection is still a considerable and ongoing challenge in today's online environment, as the number of unsolicited emails keeps growing exponentially. Various algorithms such as the tree-based model, support vector machine Algorithm, and Convolutional Neural Network have been explored in prior research to tackle this challenge. This research specifically examines the effectiveness of the Naïve Bayes classifier for identifying and filtering spam emails. By delving into the fundamental principles of this classifier, its practical implementation, and the comprehensive evaluation of its performance on a combined dataset, its strengths and limitations in distinguishing spam from ham messages are revealed. The result of the study demonstrates an overall accuracy of 97.82%, showcasing the Naïve Bayes classifier's high efficiency and stability in identifying spam. With consistently high metrics score throughout both classes, the Naïve Bayes classifier has proven to be an exceptionally reliable tool for spam email detection, underscoring its suitability for numerous real-world applications.

1 Introduction

With the proliferation of digital communication, email has become an essential component of everyday life, serving as a predominant medium for both personal and professional interactions. However, this surge in usage has also given rise to an equally significant challenge: the overwhelming prevalence of unsolicited bulk emails, commonly known as spam. Spam emails not only clutter inboxes, impeding the efficient communication process but also pose serious risks, such as exposure to malicious content and the potential for financial loss through phishing attacks [1, 2]. As a result, an effective spam detection mechanism is crucial for enhancing user experience and upholding the integrity of email systems.

Among various strategies developed to combat spam, machine-learning methods have proven to be effective for improving detection accuracy and efficiency [3]. Several algorithms have been effectively utilized for spam classification, including Support vector machine (SVM) [4], Genetic Decision Tree [5], particle swarm optimization [6], and deep learning techniques like Convolutional Neural Networks (CNNs) [7]. Each of them provides a different level of accuracy and robustness. Within these, the Naïve Bayes classifier is particularly prominent due to its simplicity, efficiency, and capability to handle huge datasets

Corresponding author: Liansong.wang42567-biph@basischina.com

[8]. This classifier operates on the principle of conditional independence, allowing it to effectively categorize emails by examining the occurrence of each words related to both spam and ham content [9]. By leveraging historical data, the Naïve Bayes classifier can be trained to identify patterns that distinguish spam from ham messages, thereby automating the filtering process.

This paper focuses on applying the Naïve Bayes algorithm to identify spam emails using a combined dataset. By conducting a detailed analysis of the algorithm's performance and effectiveness, this study seeks to evaluate the Naïve Bayes classifier in the context of spam filtering and provide insights into strategies for identifying spam. The findings are expected to inform future advancements in spam detection technologies, ultimately enhancing user experience in digital communications.

2 Methodology

The methodology used in this study is designed to train model for identifying spam emails using the Naïve Bayes approach. The methodology involves two major sections: data preprocessing and model training.

2.1 Naïve Bayes Classifier

The Naïve Bayes classifier is a family of linear statistical classifier that makes predictions based on Bayes' rule. Like other text classifiers, the Naïve Bayes classifier can compute a range of probabilities by counting the occurrence and arrangement of words in a specific dataset [10]. But differently, it calculates probabilities by assuming all the words are conditionally independent, which multiplies the likelihoods of each individual feature given the class label [8]. This assumption might sound unreliable since it ignored all the interactions between each term. In fact, it allows the classifier to easily process large datasets without complicated iterative parameter estimations. Surprisingly, it works well in terms of text classification.

2.2 Data Preparation and Preprocessing

To process text messages, the Naïve Bayes classifier requires a vector that captures how often each word appears within the emails, acting as the representation of the data. To ensure the vector format is suitable and unified, data preprocessing is required. The preprocessing step usually contains three tasks: text cleaning, tokenization and feature extraction [11].

1. Text Cleaning: First, to keep the data clean, all the characters that are not relative (such as punctuation, special characters, and HTML tags) or not contributing to the classification (such as stop words that carry little meaning, like "and" or "the") need to be removed. Additionally, all words must be converted to lowercase to ensure consistency.

2. Tokenization: Next, the processed text is split into single words or tokens, which converts the data into features that the classifier can analyze, allowing it to count the frequency of each word.

3. Feature Extraction: Finally, the attributes are derived from the tokens by using the Bag of Words (BoW) model. This model can count the frequency of each word occurring in the email and store it as a vector for the use of a text classifier.

2.3 Training the Naïve Bayes Model

The Naïve Bayes classifier determines the likeliness of an email being categorized as spam using Bayes' theorem. The formula for Bayes' theorem is presented below:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \quad (1)$$

In this formular, P(c|x) represents the final probability of a message to classified as spam, P(x|c) shows the likelihood of each word occurring in spam emails, P(c) is the prior probability of any particular email being spam or ham, and P(x) is the marginal probability of the individual words across all emails. Since the prior probabilities remain constant throughout classification, the resulting probability of an email's category is only affected by the class prior and the likelihood of each individual word.

The prior probability of each class is determined by taking the total messages within that category and dividing it by the total messages in the dataset:

$$P(c) = \frac{Number\ of\ messages\ in\ class}{Total\ number\ of\ messages} \quad (2)$$

The likelihoods of each word are calculated by tallying the frequencies of each term in the emails of every category. To reduce bias, the Laplace smoothing technique (adding a small constant) is applied to handle words that might not appear in the training data, which avoids zero probabilities [12]:

$$P(word|class) = \frac{Count\ of\ word\ in\ class+1}{Total\ number\ of\ words\ in\ class+Vocabulary\ size} \quad (3)$$

The total likelihood of an email's classification is computed by multiplying the likelihood of each individual word in the message given to the class:

$$P(x|c) = \prod_{i=n}^{n} P(word_i|class) \quad (4)$$

Once the posterior probability is computed, the Naïve Bayes classifier will make a prediction by comparing the resulting probabilities of each category. The category with the highest resulting probability will be chosen as the predicted category for the message.

3 Result and Discussions

3.1. Dataset

The primary dataset used in this study is a combined collection of spam emails sourced the 2007 TREC Public Spam Corpus and the Enron-Spam Dataset. This collection consists of 83,446 email entries marked as either spam ('1') or ham ('0'), as shown in Fig. 1, with 43,910 instances of spam (52.62%) and 39,538 instances of ham (47.38%).

	label	text
0	1	ounce feather bowl hummingbird opec moment ala...
1	1	wulvob get your medircations online qnb ikud v...
2	0	computer connection from cnn com wednesday es...
3	1	university degree obtain a prosperous future m...
4	0	thanks for all your answers guys i know i shou...
...	...	...
83443	0	hi given a date how do i get the last date of ...
83444	1	now you can order software on cd or download i...
83445	1	dear valued member canadianpharmacy provides a...
83446	0	subscribe change profile contact us long term ...
83447	1	get the most out of life ! viagra has helped m...

Fig. 1. Examples of the first five and last five email samples in the dataset.

The dataset was split randomly into training and testing subsets with a ratio of 80% to 20%. The training subset has a total of 66,756 emails, including 35,128 spam and 31,630 ham, while the test subset contains 16,690 emails, with 8,782 spam and 7,908 ham. The overall proportions of spam and ham remain the same after the split.

3.2. Confusion Matrix

A confusion matrix is used to analyze the performance of the trained model on the test subset. It visualizes the model's effectiveness in predicting the emails' categories by displaying the counts of the correctly identified spam emails (true positives, TP), the non-spam emails incorrectly classified as spam (false positives, FP), the correctly identified non-spam emails (true negatives, TN), and the spam emails that were misclassified as ham (false negatives, FN).

Through the confusion matrix, four key metrics can be calculated to evaluate the effectiveness of the spam email detection model. As summarized in Table 1, these metrics include accuracy, recall, precision, and F1-score [13]. Among them, accuracy indicates the proportion of emails correctly classified into their respective classes; recall reflects the model's ability to identify actual spam messages; precision measures the accuracy of emails predicted as spam; while the F1-score demonstrates the harmonic mean of precision and recall, offering a comprehensive assessment in terms of correctly detecting spam while reducing false positives

Table 1. The assessment metrics for spam email classifier.

Evaluation Measure	Evaluation Function
Accuracy	$Acc = \frac{TP + TN}{TP + FP + TN + FN}$
Recall	$r = \frac{TP}{TP + FN}$
Precision	$p = \frac{TP}{TP + FP}$
F1-score	$F1 = \frac{2pr}{p + r}$

3.3. Model Evaluation Results

In this research, a 5-fold cross-validation method is employed to determine the best-performing model. This method assesses models by dividing the dataset into multiple subsets (folds) and testing the model's performance across various subset combinations [14]. As the result of the cross-validation, the accuracies for each fold are as follows: Fold 1 achieved a score of 0.9733, Fold 2 got 0.9766, Fold 3 got 0.9762, Fold 4 got 0.9759, and Fold 5 got 0.9755. The average cross-validation score is approximately 0.9751, suggesting that the model performs stable across all subsets, with minimal variance in accuracy among the different data partitions.

The model trained in Fold 2 was selected as the best-performing model for spam email detection. After training on the complete training dataset and evaluating it with the test subset, it reached a total accuracy of 97.82%. The confusion matrix values of the prediction outcomes are illustrated in Fig. 2.

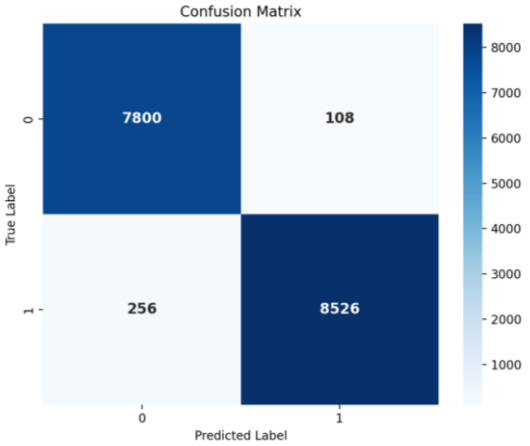

Fig. 2. The values of the confusion matrix for the prediction results.

Among the 7,908 ham emails tested, 7,800 of them were accurately classified as ham, while 108 were misclassified as spam. This results in an accuracy of 98.64% for ham emails.

For the 8,782 spam emails tested, 8,526 were accurately classified as spam, while 256 were misclassified as ham, resulting in an accuracy of 97.08%.

The comprehensive performance metrics of the model is presented in Table 2.

Table 2. Detailed classification performance metrics of the model.

	Precision	Recall	F1-score	Support
Spam	.97	.99	.98	7908
Ham	.99	.97	.98	8782
Accuracy			.98	16690
Macro avg	.98	.98	.98	16690
Weighted avg	.98	.98	.98	16690

These results from both ham and spam classes exhibit high precision and recall, demonstrating how the model could effectively distinguish between the two categories. With consistently high performance throughout the model, the Naïve Bayes model could be considered as a well-suited approach for this dataset. This emphasizes how the model is capable to achieve the intended outcome, making it a promising option for identifying spam emails.

4 Conclusion

In conclusion, the Naïve Bayes classifier exhibits excellent performance in differentiating between ham and spam emails. It emerges as a powerful tool for spam email detection, offering a solid foundation for upcoming research and practical uses in message filtering systems. Its ability to maintain high performance across diverse email types makes it invaluable for enhancing user experience and improving overall email management.

Looking ahead, future research could explore hybrid models that combine Naïve Bayes with other machine learning techniques such as support vector machines or deep learning approaches. This could leverage the strengths of different algorithms and hopefully improve its accuracy. Additionally, incorporating contextual and semantic analysis may deepen the understanding of email content, allowing it to identify intent and relevance better. By

addressing these areas, the Naïve Bayes classifier can further enhance its contribution to the advancement of spam detection systems.

References

1. A. H. Al-Ghushami, D. Syed, A. Zainab, A. Abdelshahid, H. Al-Eshaq, F. Alsayed, & R. Alkuwari, Email security: Concept, formulation, and applications, In 2022 14th International Conference on Computational Intelligence and Communication Networks (CICN), IEEE, (2022), 825-829
2. S. Rao, A. K. Verma, & T. Bhatia, A review on social spam detection: Challenges, open issues, and future directions. Expert Systems with Applications. **186**, 115742 (2021)
3. P. Teja Nallamothu & M. Shais Khan, Machine learning for SPAM detection. Asian Journal of Advances in Research. **6**, 167-179 (2023)
4. S. O. Olatunji, Improved email spam detection model based on support vector machines. Neural Computing and Applications. **31**, 691-699 (2019)
5. S. S. Ismail, R. F. Mansour, R. M. Abd El-Aziz, & A. I. Taloba, Efficient E-Mail Spam Detection Strategy Using Genetic Decision Tree Processing with NLP Features. Computational Intelligence and Neuroscience. **2022**, 7710005 (2022)
6. T. Alkhdour, R. Alrawashdeh, M. Almaiah, R. Alali, S. Salloum, T. H. Aldahiyani, A new technique for detecting email spam risks using LSTM-particle swarm optimization algorithms. Journal of Theoretical and Applied Information Technology. **102**, 5482-5499 (2024)
7. K. Debnath & N. Kar, Email spam detection using deep learning approach, In 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON), IEEE, (2022), 37-41
8. K. U. Santoshi, S. S. Bhavya, Y. B. Sri, & B. Venkateswarlu, Twitter spam detection using naïve bayes classifier, In 2021 6th International Conference on Inventive Computation Technologies (ICICT), IEEE, (2021), 773-777
9. I. Wickramasinghe & H. Kalutarage, Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. Soft Computing. **25**, 2277-2293 (2021)
10. A. Gasparetto, M. Marcuzzo, A. Zangari, & A. Albarelli, A survey on text classification algorithms: From text to predictions. Information. **13**, 83 (2022)
11. M. M. Kodabagi, Efficient data preprocessing approach for imbalanced data in email classification system, In 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), IEEE, (2020), 338-341
12. E. R. Setyaningsih & I. Listiowarni, Categorization of exam questions based on bloom taxonomy using naïve bayes and laplace smoothing, In 2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT), IEEE, (2021), 330-333
13. J. Liang, Confusion matrix: Machine learning. POGIL Activity Clearinghouse. **3**, 1 (2022)
14. P. Misra & A. S. Yadav, Improving the classification accuracy using recursive feature elimination with cross-validation. Int. J. Emerg. Technol. **11**, 659-665 (2020)