Enhancing Spam Filter Using Naive Bayes and Count Vectorizer

Jiachen Liang[1, *]

[1] Department of Computer Science, Pennsylvania State University, 201 Old Main, 16802, USA
*jpl6373@psu.edu

Abstract. This study delves into advancements in the realm of email spam filtration, a critical pillar in augmenting email security infrastructure. Given the unceasing challenges presented by unwarranted spam, the deployment of efficacious spam filtration methodologies remains imperative. Contemporary strategies encompass IP address filtering, rule-based filtering, and the employment of Naive Bayes algorithms. However, these methodologies often succumb to the continuously evolving spamming techniques. To counter these drawbacks, the current study proposes an enhanced spam filtering architecture anchored in machine learning techniques, which involves augmented spam data procurement, data processing, feature extraction via Term Frequency-Inverse Document Frequency (TF-IDF), and the implementation of machine learning models such as Naive Bayes and Support Vector Machines (SVM). This research conducts a comparative analysis of these machine learning classifiers and underscores the superior performance of the SVM Linear model in spam detection, achieving elevated accuracy levels while ensuring balanced precision and recall for both spam and non-spam emails. These findings underscore promising strides in the arena of email security. The study culminates by advocating for persistent research and the incorporation of advanced techniques to augment the accuracy and user-friendly nature of spam filtering systems.

Keywords: spam filter, Naive Bayes, Count Vectorizer, SVM.

1 Introduction

Email has become an essential tool for acquiring information and engaging in communication in today's information age [1]. Its significance extends to both personal and professional domains. As a result, enhancing email security has become increasingly important [2]. Email is susceptible to various threats in the network, including malware, spam, and the potential leakage of sensitive data. These issues can lead to the exposure of recipients' sensitive information or the annoyance caused by spam [3].

This research concentrates on enhancing spam filtering methods. Spam refers to unsolicited and bulk emails that are often distributed for marketing purposes or to carry out malicious activities while ham refers to legitimate emails [4]. Spammers continuously develop new techniques to evade spam filters and exploit email

P. Kar et al. (eds.), *Proceedings of the 2023 International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2023)*, Advances in Computer Science Research 108,
https://doi.org/10.2991/978-94-6463-300-9_58

distributions for their financial gain [5]. The widespread presence of spam poses significant challenges to individuals, persistently affecting email users. Spam messages, characterized by their unsolicited nature and unwanted content, are disseminated to a large number of recipients. These emails often serve malicious intentions such as promoting products or services, distributing malware, or spreading false information [6]. In response, organizations and email service providers employ effective spam filtering techniques to mitigate these threats. An efficient spam filtering system is crucial for safeguarding user privacy, ensuring the integrity of communication channels, and enhancing overall email security.

The research methodology involves a comprehensive analysis of existing spam filtering techniques, exploring their limitations, and proposing innovative approaches to enhance their effectiveness. The research contributes to the field by providing insights into the advancements in spam filtering and addressing the emerging challenges posed by spammers. The proposed methods offer enhanced accuracy in differentiating between spam and legitimate emails, reducing the likelihood of false positives and false negatives. Email users can enjoy a more secure and streamlined communication experience by improving the efficiency and reliability of spam filtering. In conclusion, this research presents an in-depth analysis of the advancements and implementation of an improved spam filtering system. The study aims to address the challenges posed by spam emails and enhance email security. By utilizing advanced techniques and algorithms, the proposed system offers enhanced accuracy and efficiency in filtering out spam messages. The research contributes to the field by providing valuable insights and solutions for combating the persistent issue of spam in email communication.

2 Literature Review

The issue of spam emails has been tackled in various fields through three common methods discussed in the literature. The first method involves filtering spam based on the sender's Internet Protocol (IP) address. By implementing whitelists, recipients can ensure the receipt of legitimate emails from trusted sources, while unconventional or non-whitelisted IP addresses are classified as spam. Users have the option to review and blacklist such emails, effectively blocking future emails from those sources [7, 8]. The second method utilizes rule-based filtering for spam detection. Users can establish specific rules that are automatically applied to incoming emails. When an email satisfies any of these predefined rules, it is redirected to the spam folder. Rules can be based on specific words or phrases in the email body or header. However, this approach has limitations, as the pattern of spam is often elusive to average users, and configuring and maintaining rules can be cumbersome [9]. The third methodology capitalizes on Naive Bayes algorithms, a highly utilized supervised machine learning approach often employed for text classification among other functions. Naive Bayes classifiers utilize Bayes' theorem under the premise of independent assumptions between features for class prediction. These classifiers model the distribution of input variables for each class and deliver efficient performance even with limited training

data. Naive Bayes classifiers have demonstrated high success rates in text classification, making them widely used in spam filtering. The algorithm calculates the probability of an email being spam or legitimate based on the occurrence of specific words or features [10, 11].

It is important to note that spammers continuously develop new techniques to evade detection, necessitating ongoing updates and refinements to spam filtering methods [12]. Currently, the most effective approach involves machine learning applied to email classification. By utilizing machine learning algorithms, spam can be effectively screened and detected within a large volume of emails. This process involves data processing, feature selection, training, and testing. Researchers and email service providers strive to enhance the performance and accuracy of spam filtering systems to keep up with the evolving nature of spam [13].

3 Methodology

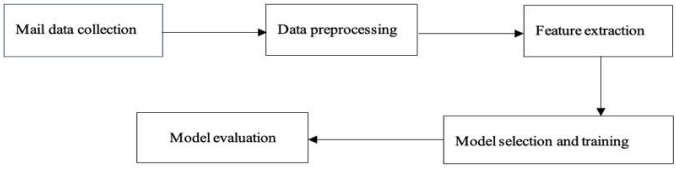

Fig. 1. Spam Filter Model process flow (Photo/Picture credit: Original)

Figure 1 presents the spam filter model pipeline. To begin with, it is essential to collect email data, followed by performing a series of preprocessing steps, including data cleaning, handling missing values, removing duplicates, and ensuring data consistency and quality. Subsequently, feature extraction is conducted to select relevant features from the email data, preparing it for further analysis. Ultimately, a suitable computational model is selected for the training phase, and the efficacy of this model is meticulously assessed utilizing pertinent evaluation metrics.

3.1 Mail data collection

The original dataset for spam filtering was modified by incorporating real-life examples of spam and removing redundant or improperly categorized data. This modification aims to enhance the effectiveness of the spam filtering system. There are two primary improvements compared to the original dataset: increased universality and improved accuracy. By including diverse examples of spam, the dataset expands to cover a wider range of spam characteristics. This enables spam filters to recognize and adapt to new variants of spam. A more diverse and representative dataset allows spam filtering models to learn robust patterns associated with spam, thereby improving classification accuracy and minimizing false positives and negatives. The modified dataset is further utilized to evaluate the system by appending three examples to the end of the code and using implemented functions to assess the spam

filtering capabilities. This evaluation process serves to validate the performance of the system and its effectiveness in identifying and classifying spam.

3.2 Data preprocessing

In the provided code snippet, a text preprocessing class is defined to facilitate the normalization of textual data. This class is part of a broader data preparation pipeline and performs a series of operations to refine raw input text. It begins by converting the text to lowercase, ensuring uniformity and preventing distinctions based on case differences alone. Punctuation marks are then removed, as they often carry little semantic meaning and can introduce noise. Next, stopwords, which are commonly occurring words such as "is," "the," and "and," are eliminated. This removal helps reduce dimensionality and computational complexity. It also employs the Porter stemming algorithm, which truncates words to their root form, further reducing dimensionality and consolidating word variations. This normalization process is crucial for machine learning and natural language processing tasks, as it transforms unstructured text into a more analyzable and consistent format. This format facilitates pattern extraction and enables the application of linguistic models.

3.3 Feature Extraction

Within the domain of language and text processing, the role of feature extraction is of paramount importance, as it transforms textual data into a structure that is amenable to the computational requirements of machine learning algorithms. Two commonly used methods for feature extraction in text data are Count Vectorizer (CV) and TF-IDF. The CV is a straightforward approach to representing text data numerically by creating a matrix of token counts from a collection of text documents. The underlying concept is that documents with similar content are likely to have similar counts of words. By analyzing the word frequencies, meaningful features can be derived and utilized for training machine learning models. On the other hand, TF-IDF is another method to convert a text document into a numeric vector. The approach encapsulates not solely the prevalence of a word within an individual document (captured by the term frequency) but also the rarity or uniqueness of the word across the entire corpus of documents, quantified by the inverse document frequency. This is particularly useful for identifying commonly occurring but less informative words like "is," "the," and "and".

The TF-IDF is computed as the product of two distinct terms: the Term Frequency (TF) and the Inverse Document Frequency (IDF). The TF component quantifies the prevalence of a word within a singular document, whilst the IDF component gauges the significance of the word within the entire corpus of documents. Formula 1 defines the calculation for obtaining the TF-IDF value of a word, it is determined by multiplying these two quantities together:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) * \text{IDF}(t) \quad (1)$$

This research report primarily focuses on the utilization of TF-IDF for feature extraction. While a CV is simpler to implement and perform, TF-IDF is particularly suitable for handling large datasets with documents of varying lengths. It is especially effective in identifying words that are frequent in a specific document but infrequent in others. TF-IDF can be applied to various tasks, including document classification, clustering, and topic modeling.

3.4 Model Selection and Training

The study focuses on selecting machine learning algorithms suitable for text categorization. Leading alternatives encompass methodologies such as Naive Bayes, SVM, and ensemble techniques including, but not limited to, Random Forests and Gradient Boosting. The dataset is bifurcated into distinct subsets for training and testing purposes. CV is utilized for feature extraction due to its efficiency with large datasets and emphasis on word occurrences, aligning with the objective of constructing a simple yet effective model. For classification, two algorithms are employed: Naive Bayes and SVM. Naive Bayes, a supervised learning algorithm, assumes feature independence and calculates the probability of emails being classified as spam or ham. The Multinomial NB model from scikit-learn is utilized for this purpose. The SVM methodology seeks to establish an optimal hyperplane within the feature space that maximizes the margin of separation between the two classes of emails, namely spam and non-spam (often referred to as "ham"), leveraging the Linear Support Vector Classification (SVC) implementation as provided by the scikit-learn library. The kernel is modified from Radial Basis Function (RBF) to linear based on its superior performance and efficacy in handling linear separability of features.

3.5 Model evaluation

The textual data encompassed within the messages will be transmuted into vector representations via the bag-of-words model, facilitating their interpretability by machine learning algorithms. This transformation will be executed utilizing TF-IDF Vectorizer, a technique that converts an assembly of text documents (the SMS corpus) into a two-dimensional matrix. Within this matrix, one dimension signifies the documents, while the other signifies each unique term within the SMS corpus.

If a specific term, denoted as "t," manifests "p" times in the "m" document, the corresponding position (m, n) in the matrix encapsulates the TF-IDF value for that term. The TF quantifies how frequently a term recurs within a document, computed as the occurrence count of the term "t" within the document "p," normalized by the total count of terms within that document. The classifiers are subsequently scrutinized on the testing set using various performance indicators, including accuracy, precision, recall, and F1 scores. A confusion matrix is formulated to provide a visual delineation of true positives, true negatives, false positives, and false negatives. Upon achieving satisfactory preliminary performance, these classifiers are prepared for deployment in

real-world applications, with the objective of predicting the spam or non-spam nature of actual emails.

4 Experiment and Results

The experiment aimed to evaluate three different machine learning models for email classification, including SVM RBF, SVM Linear, and Naive Bayes. Figure 2 illustrates the dataset used in the experiment for email classification. The dataset comprises a collection of labeled emails, with each email being classified as either spam or ham. It reveals that the dataset contains a significant number of spam emails, with a count of 700 or more, while the count of ham emails reaches 4000 or more. This distribution of spam and ham emails provides a diverse and representative dataset for evaluating the performance of the classification models.

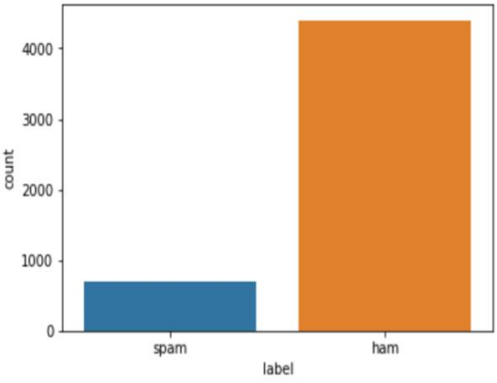

Fig. 2. Classified mail dataset size (Photo/Picture credit: Original)

4.1 Data Preparation

The data corpus was partitioned into two distinct subsets: a training set and a testing set. The partitioning followed a proportional distribution of 70:30, such that 70% of the data was allocated for training purposes and the remaining 30% reserved for testing, thereby ensuring an optimal balance between model learning and subsequent performance evaluation. The emails were preprocessed by removing any special characters, HTML tags, and unnecessary punctuation. Stop words were eliminated to improve the quality of the extracted features. The remaining text was transformed into numerical features using either CV or TF-IDF vectorization.

4.2 Model Training and Evaluation

SVM RBF: A machine learning model utilizing the SVM algorithm accompanied by a RBF kernel was implemented for learning from the designated training set.

Subsequently, an evaluation of the model's competency was conducted on the designated testing set, utilizing key performance metrics inclusive of precision, recall, F1-score, and overall accuracy.

SVM Linear: Analogously, another variant of the SVM model, this time incorporating a linear kernel, was fitted to the training data. The model's predictive power was rigorously assessed on a separate testing set, applying the identical three metrics to ensure consistency in evaluation methodology across models.

Naive Bayes: A Naive Bayes classifier, renowned for its basis in probabilistic principles, was subsequently employed in the same training set.

4.3 Evaluation Metrics

Precision: This metric, also known as the positive predictive value, quantifies the proportion of true positive predictions in the total positive predictions (both true and false positives) made by the model.

Recall: This metric measures the proportion of instances in which the model correctly identifies the true category as positive. It is an indicator of a model's ability to find and correctly classify all positive instances.

F1-score: This indicator is the harmonic average of the two indicators, recall rate, and precision, and provides a single indicator that encapsulates both aspects of these indicators. In cases where both false positives and false negatives need to be considered, and there is no clear preference between accuracy and recall, f1 scores can be used as a suitable single indicator of model performance.

4.4 Results

Figure 3 displays the results of the SVM RBF model. It achieved an accuracy of 0.98, with a high precision of 0.97 for ham emails and a slightly lower precision of 1.00 for spam emails. The result of the recall for ham emails was 1.00, while for spam emails, it was 0.83. These findings indicate that the SVM RBF model performed well in accurately identifying ham emails but faced some challenges in correctly classifying spam emails.

	precision	recall	f1-score	support
ham	0.99	0.99	0.99	887
spam	0.92	0.92	0.92	133
micro avg	0.98	0.98	0.98	1020
macro avg	0.96	0.96	0.96	1020
weighted avg	0.98	0.98	0.98	1020

Fig. 3. SVM RBF Results (Photo/Picture credit: Original)

Figure 4 presents the SVM Linear results. By optimizing the SVM with linear separation, the SVM Linear model achieved even better outcomes. It attained an accuracy of 0.99, with high precision and recall values for both ham and spam emails. The result of precision for ham emails was 0.99, and for spam, it was 0.98. The recall for ham emails was 1.00, and the other one was 0.94. These results suggest that the SVM Linear model excelled in accurately classifying both ham and spam emails.

	precision	recall	f1-score	support
ham	0.97	1.00	0.99	960
spam	0.99	0.84	0.91	155
accuracy			0.98	1115
macro avg	0.98	0.92	0.95	1115
weighted avg	0.98	0.98	0.98	1115

Fig. 4. SVM Linear Results (Photo/Picture credit: Original)

Figure 5 showcases the results of the Naive Bayes model. The Naive Bayes model achieved an overall accuracy of 0.98. It exhibited high precision (0.99) and recall (0.99) for ham emails, indicating its ability to accurately identify them. However, the precision and recall for spam emails were slightly lower at 0.92, suggesting some difficulty in correctly classifying spam emails compared to the other models.

	precision	recall	f1-score	support
ham	0.99	1.00	0.99	961
spam	0.98	0.94	0.96	154
accuracy			0.99	1115
macro avg	0.98	0.97	0.98	1115
weighted avg	0.99	0.99	0.99	1115

Fig. 5. Naive Bayes Results (Photo/Picture credit: Original)

Comparing the models, the SVM Linear model emerged as the top performer, achieving the highest accuracy and demonstrating balanced precision and recall for both ham and spam emails. In the context of accuracy and classification capabilities, it surpassed the performance of both the SVM with an RBF and the Naive Bayes models.

In conclusion, based on the tests conducted, the SVM Linear model performed the best out of the three models in the email spam classification task. However, further research and experimentation may be necessary to validate these findings and explore the models' robustness in different contexts.

5 Conclusion

In conclusion, spam filtering is an essential aspect of email management, aimed at reducing the impact of unwanted and malicious messages on users. Through the utilization of various techniques, including data collection, preprocessing, feature extraction, model selection, and evaluation, effective spam filtering systems can be developed. The deployment of machine learning techniques, including Naive Bayes and SVM, coupled with feature extraction methodologies like TF-IDF, facilitates the precise discernment and categorization of spam emails. Furthermore, the evaluation of the system using metrics like accuracy, precision, recall, and F1 scores provides insights into its performance. The incorporation of real-life spam examples and the removal of redundant data contribute to the enhancement of the spam filtering system's universality and accuracy.

To further improve spam filtering, future research should explore advanced machine learning models and techniques that can adapt to new spam variants and improve classification accuracy. Additionally, personalized filtering based on user preferences and needs can enhance the user experience and effectiveness of the filtering system. Overall, spam filtering systems continue to evolve and play a vital role in mitigating the negative effects of spam. With ongoing research and development efforts, these systems can become even more robust, efficient, and user-friendly, providing users with a safer and more enjoyable email communication experience.

References

1. T. Herath, et al. Security services as coping mechanisms: An investigation into user intention to adopt an email authentication service. Information systems journal 24.1, 61-84(2014).
2. A. Karim, et al. A comprehensive survey for intelligent spam email detection. IEEE Access 7, 168261-168295 (2019).
3. K. Akanksha, et al. Email Security. Journal of Image Processing and Intelligent Remote Sensing (JIPIRS) ISSN 2815-0953 2.06, 23-31(2022).
4. F. Jáñez-Martino, et al. A review of spam email detection: analysis of spammer strategies and the dataset shift problem. Artificial Intelligence Review 56.2, 1145-1173(2023).
5. T. Muralidharan, and N. Nissim. Improving malicious email detection through novel designated deep-learning architectures utilizing entire email. Neural Networks 157, 257-279(2023).
6. Dada, G. Emmanuel, et al. Machine learning for email spam filtering: review, approaches and open research problems. Heliyon 5.6, e01802(2019).
7. Mirza, M. Mohammad, and K. Umit. Enhancing IP Address Geocoding, Geolocating and Visualization for Digital Forensics. 2021 International Symposium on Networks, Computers and Communications (ISNCC), pp. 1-7. IEEE(2021).
8. J. Goodman. IP Addresses in Email Clients. CEAS(2004).
9. Weimiao Feng, et al. A support vector machine based naive Bayes algorithm for spam filtering. 2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC). pp. 1-8. IEEE(2016).

10. Zhefu Wu, et al. Passive indoor localization based on csi and naive bayes classification. IEEE Transactions on Systems, Man, and Cybernetics: Systems 48.9, 1566-1577(2017).
11. Chauhan, S. Alok, et al. Comparative Analysis of Supervised Machine and Deep Learning Algorithms for Kyphosis Disease Detection. Applied Sciences 13.8, 5012(2023).
12. Qin. Luo, et al. Design and implement a rule-based spam filtering system using neural network. 2011 International Conference on Computational and Information Sciences. IEEE(2011).
13. M. Rajalingam, V. Raman, and P. Sumari. Implementation of vocabulary-based classification for spam filtering. 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16). IEEE(2016).