Enhancing Spam Filtering: A Comparative Study of Modern Advanced Machine Learning Techniques

Chenwei Zhang

Qingdao No.2 Middle School, 266600 Qingdao, China

Abstract. Spam remains a persistent issue that not only consumes time and bandwidth but also poses significant cybersecurity threats. As a result, effective spam filtering has become essential. With an emphasis on Naïve Bayes (NB), Decision Trees (DT), and Support Vector Machines (SVM), this study offers a thorough analysis of the major machine learning techniques utilized in contemporary spam filtering. This paper investigates underlying principles of these methods, compares their performance through extensive experiments conducted on the Kaggle dataset, and discusses the current challenges and future directions for spam filtering technology. The study reveals that SVM is particularly effective for handling high-dimensional data, DT offers superior interpretability, and NB simplifies probabilistic classification. Experimental results demonstrate that while each method has its strengths and weaknesses, combining SVM with NB notably enhances classification accuracy. Despite these advances, spam filters still face challenges due to evolving spamming tactics. In order to address these persistent problems, the conclusion part highlights the need for more reliable and flexible spam filtering technologies and makes recommendations for future research directions.

1 Introduction

With the innovation of network technology, the use of email is gradually increasing. However, along with the working emails, spam emerges and becomes an annoying problem for users. It wastes users' time and bandwidth and threatens cybersecurity by publishing meaningless advertisements and suspicious links. Therefore, individuals need a spam filter to enhance the overall efficiency and accuracy of email communication. Spam filtering is a crucial technology in managing the email system by recognising unwanted mail and handling them automatically. It is able to use methods like machine learning and statistical analysis to distinguish between legitimate emails and spam.

Lueg carried out a brief investigation to see if information retrieval and filtering methods could be used to logically and theoretically base gaps in spam detection hypotheses. However, the survey lacked information on machine learning algorithms, simulation tools, publicly accessible datasets, and spam environment architecture [1,2]. Wang examined the various

Corresponding author: yuexin@ldy.edu.rs

methods for removing unsolicited spam emails, classifying spam into various folders with varying levels of hierarchy, and automatically modifying the actions needed to reply to emails [3]. Sanz, Hidalgo, and Pérez outlined the study questions related to spam, including how users are affected and how both providers and users can lessen its effects [2,4]. The study explains the organization and processes of various methods for filtering, but since it was published in 2008, it does not include recent research or compare different content filters [5]. A thorough analysis of several well-liked content-based email spam filtering techniques was given by Bhowmick and Hazarika, and Laorden et al. went into detail about the benefits of anomaly detection for spam filtering, emphasizing how this method only applies to the representation of a single class of emails and reduces the need for spam classification [6,7]. The field of spam filtering has advanced significantly in the past few years. Support Vector Machines (SVM) and decision trees (DT) have demonstrated as the one of the most powerful and efficient cutting-edged classification techniques for solving this problem [8]. Besides, naïve bayes (NB) and stochastic optimization techniques such as evolutionary algorithms (EA) have an important effect on machine learning [2].

This study's main goal is to investigate methods for spam filtering using machine learning. The paper begins by providing background information on spam emails. It then delves into the core technologies used in spam filtering, specifically SVM, DT, and NB, explaining and analysing their fundamental concepts and applications. Following this, the paper presents a comparative analysis of these algorithms based on relevant experimental results. This comparison highlights their relative effectiveness in spam classification. The study also addresses future developments and challenges in the field of spam classification, offering insights into potential advancements and persistent issues. The following is how the paper is organized: The first chapter introduces various spam filtering methods and technologies. An examination of the fundamental ideas and precepts that underpin these techniques is given in the second chapter. The third chapter discusses the experimental results and their implications. Finally, the fourth chapter summarizes the findings and outlines future directions for research in spam filtering.

2 Methodology

2.1 Dataset description and preprocessing

A dataset of email text messages is displayed in Table 1, with each message classified as either spam or not by a binary number between 1 and 0. It can be applied to a number of Natural Language Processing (NLP) tasks, including spam detection and text classification [9]. This dataset can be used by scientists and researchers to train and assess machine learning models, which will enable them to produce efficient spam email filters [9].

Table 1. Spam email dataset [9].

Email content	Spam (1) or not spam (0)
Subject: naturally irresistible your corporate identity It is really hard to recollect a company: ...	1
Subject: your trusted source for prescription medication. best prescription generic meds 4 less ...	1
Subject: dear Ms. Feldman, please find enclosed a proposal for the d - g energy software license...	0
Subject: meeting to discuss presentation materials please respond to hello Vince and Kenneth, my ...	0

2.2 Proposed approach

The goal of this study is to showcase the machine learning methods used in spam filtering. The total pipeline is illustrated in Fig.1. First, this paper explores the application of four main algorithms in spam filtering, including SVM, DT, and NB. Each of these algorithms has its own features and application scenarios. SVM can be trained easily, however, because processing high-dimensional data requires processing it becomes more computationally complex over time, reducing the SVM's strength and effectiveness [10]. DT is another machine learning algorithm, and a major benefit of it is its ability to assign explicit values to each problem, decision, and outcome, and reduces ambiguity in decision making [11]. It also makes open all the likely options and provides space for direct evaluation between different nodes of the tree [2]. Furthermore, the application of NB's theorem to the contextual categorization of every email makes the strong assumption that the words within the email are unrelated to one another. Since EA does not require complex computations and identifies people who have the best answers to problems, it has also been used in spam filtering [2]. But this paper does not discuss EA.

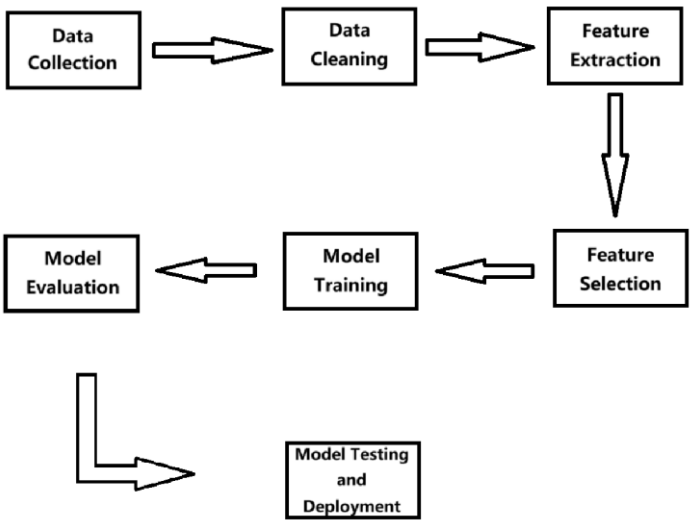

Fig. 1. The pipeline of spam filtering (Picture credit: Original).

2.2.1 Introduction to machine learning

Machine Learning (ML) is a dynamic subfield of artificial intelligence focused on enabling machines to efficiently process and interpret data. By leveraging algorithms trained on extensive datasets, ML creates models that empower machines to perform tasks traditionally reserved for human intelligence. These tasks include image categorization, data analysis, and price prediction. Machine learning (ML) comprises multiple techniques: supervised learning, in which models are trained on labelled data; unsupervised learning, which finds patterns in unlabelled data; semi-supervised learning, which combines the two approaches; and reinforcement learning, in which models learn by making mistakes and receiving feedback. The fundamental mechanism behind ML involves using algorithms—sets of rules refined and adjusted through historical data—to make predictions and classifications when encountering new information. Effective ML requires continuous refinement of these

algorithms to enhance their accuracy and reliability. By systematically updating these rules based on past experiences, ML systems gradually build robust models capable of handling complex data tasks with increasing proficiency. This iterative process is crucial for ensuring that the algorithms can adapt to new challenges and deliver precise outcomes in diverse applications.

2.2.2 Introduction of SVM for detail

SVM is a supervised learning algorithm that includes a set of algorithms designed for solving classification and regression problems, and it has been demonstrated that SVM outperforms certain other related learning algorithms [12]. Since SVM can model complex, non-sequential, or multidimensional decision boundaries, its strength lies in its high precision, even though it might not be as quick as some other classification techniques [2]. The following is the SVM training and classification method for spam emails:

First, the email message x that needs to be classified is input, and parameters are set, including a training set S, a set of kernel functions, various parameter combinations $\{C_1, C_2, \dots, C_{num}\}$ and $\{\gamma_1, \gamma_2, \dots, \gamma_{num}\}$, as well as a parameter k, which may be used for k - fold cross-validation or could represent the number of nearest neighbours. Then, through an outer loop, each C_i is iterated over, with the current parameter $C = C_i$ being set. In the inner loop, for each C_i, every corresponding γ_j is iterated over, with the current $\gamma = \gamma_j$ being set. Subsequently, a trained SVM classifier $f(x)$ is generated using the current C and γ. If this is the first SVM classifier generated, it is saved as the current optimal classifier $f^*(x)$; otherwise, the current SVM classifier $f(x)$ is compared to the current best classifier $f^*(x)$ using k-fold cross-validation, and the classifier with higher accuracy is retained as the new optimal SVM classifier $f^*(x)$. This process is repeated multiple times to adjust different parameter combinations, maximising the accuracy of the classifier. Finally, the most ideal SVM classifier $f^*(x)$ is used to return the final classification result, determining whether the input email is spam. The objective of this process is to produce an optimal model that can most effectively classify emails as spam or non-spam.

2.2.3 Introduction of DT for detail

DT is an inbuilt and powerful graphical tool in ML and data analysis, which helps in making a decision or prediction from the given input data. The nodes denote any decision or test on the attribute; the branches represent the results of that test process, and the leaf node carries the final decision or prediction to be made. Starting from the root node, it proceeds to a recursive splitting of the subsets of data under some selected attributes, following criteria that may be either the Gini index or Information Gain. It continues until attainment of a certain depth or homogeneous subsets of data. They are preferred due to their simplicity, interpretability, and ability to handle numerical and categorical data. They do, however, tend to be prone to overfitting and may require techniques such as pruning for improving performance. Notwithstanding the challenges, DT forms the basis of many applications at large, including classification and regression tasks, and therefore are an important component of the bag of tools known as ML. Fig. 2 demonstrates an example of a decision tree.

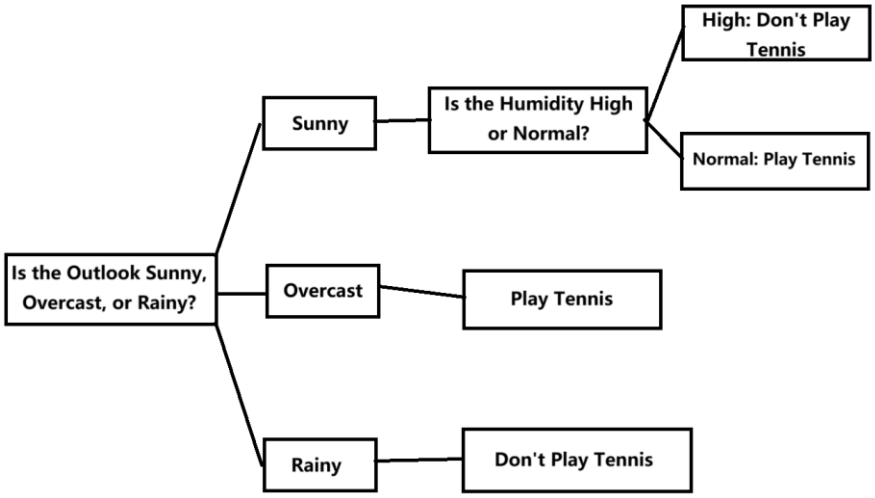

Fig. 2. An example of a DT (Picture credit: Original).

2.2.4 Introduction of NB for detail

NB is a pretty simple, general probabilistic ML algorithm applied for classification tasks. Bayes' Theorem is at the core of this algorithm: computation of the probability of a class given a set of features, under the assumption of these features as conditionally independent from each other. Based on Bayes' Theorem, an NB classifier is a simple classification algorithm that functions on the premise that features, given the class label, are independent of one another. Spam filtering is the best-known use of NB classifiers, and the classification algorithm for it is depicted below:

Initially, an email message dataset is inputted, and each email is parsed into its constituent tokens. The algorithm then calculates the spamminess of each token using the formula

$$S[T] = \frac{C_{spam}(T)}{C_{spam}(T) + C_{ham}(T)} \tag{1}$$

where the counts of spam messages with token T are represented by $C_{spam}(T)$ and non-spam messages with the same token are represented by $C_{ham}(T)$. These metrics for spamminess are kept in a database. For each message, the algorithm scans through the tokens, queries the database for their spamminess $S(Ti)$, and computes the probabilities for spam $S[M]$ and non-spam $H[M]$. The overall filtering signal is then calculated using

$$I[M] = \frac{I + S[M] - H[M]}{2} \tag{2}$$

The message is categorized as spam if I[M] surpass a preset threshold; if not, it is categorized as non-spam. The final result is a classification of each email as either spam or valid.

3 Result and Discussion

3.1 Results display for NB and SVM

Thesis tallied the number of spam messages $N_{S\to L}$ that were mistakenly identified as legitimate mail (false negatives) and the number of legitimate messages $N_{L\to S}$ that were wrongly identified as spam (false positives) for each algorithm in Table 2 [13]. P is the precision of the classifier and G is calculated as the ratio of the classifier precision and the trivial classifier precision ($\frac{N_L}{N}$). To represent the overall number of messages, $N = 1099$, the number of spam messages, $N_S = 481$, and the number of legitimate messages, $N_L = 618$ should be used.

Table 2. Basic algorithm performance [13].

Algorithms	$N_{L\to S}$	$N_{S\to L}$	P	F_L	F_S	G
NB	0	138	87.4%	0.0%	28.7%	1.56
SVM	10	11	98.1%	1.6%	2.3%	1.74

It is clear that NB produces no false positives in this experiment at all (show in Table 3). But for SVM, it requires an enhancement of lower probability of false positives. Therefore, there is a way to combine these two methods together to get a filter with higher accuracy. To begin, let f and g represent two spam filters with extremely low false positive rates. If either f or g flags message x as spam, then mark it as such. Otherwise (if $f(x) = g(x) = L$) classify it as legitimate mail. This paper denotes this combination as $f \cup g$.

If message x has the property $f(x) = g(x) = c$, then can categorize x as a member of c. Suppose $f(x)$ is not equal to $g(x)$, for example, $f(x) = L$ and $g(x) = s$. The fact that f picked the safe even though it was the incorrect choice is probably the reason the algorithm returns different results since g is unlikely to mistakenly categorize valid communications as spam [14]. Therefore, it makes sense to believe that S, not L, is the real class of x.

Table 3. Basic algorithm performance characteristics of their union [13].

Algorithms	$N_{L\to S}$	$N_{S\to L}$	P	F_L	F_S	G
NB ∪ SVM	0	61	94.4%	0.0%	12.7%	1.68

3.2 Discussion

ML algorithms have become integral to spam filtering, employing various classification techniques to identify and manage unwanted emails. However, many of these algorithms are designed to handle static and well-defined categories, making them susceptible to vulnerabilities, particularly when confronted with manipulated or adversarial data. This susceptibility poses a significant challenge to data reliability and accessibility, necessitating the development of robust strategies to enhance spam filter security. An ongoing issue in spam filtering is the difficulty in improving accuracy. While researchers strive to refine the predictive capabilities of spam filters, spammers continually adapt their tactics to circumvent these defences. This cat-and-mouse dynamic highlight the urgent need for more sophisticated techniques and algorithms that can outsmart evolving spam strategies. Additionally, current spam filters primarily address text-based emails, leaving a gap in handling more complex forms of spam. For instance, some spammers embed text within images, such as staged images, to evade detection. This method effectively bypasses traditional text-based filters. To address this, researchers must develop advanced image-based spam filters capable of recognizing and processing spam content embedded in various visual formats. This advancement is crucial for staying ahead of sophisticated spamming techniques and ensuring comprehensive spam detection.

4 Conclusion

This paper explored three prominent ML approaches: SVM, NB, and DT, and their applications in spam filtering. It also reviewed several publicly available datasets used to assess the efficiency of spam filters. The study aimed to analyse the performance and usability of these spam filtering techniques and to identify potential issues. Extensive experiments were conducted to evaluate these methods. The results indicate that while current spam filters continue to face challenges related to insecurity and inaccuracy, combining SVM with NB significantly mitigates these issues. Despite this progress, there is still room for improvement. The paper focused on only two specific methods for spam filtering. Future research will expand to include additional approaches such as EA, Neural Networks (NN), and Random Forests (RF). This next phase of research will examine the effectiveness and complexity of these methods, aiming to provide a more comprehensive understanding of their potential and limitations in practical applications.

References

1. C.P. Lueg, From spam filtering to information retrieval and back: seeking conceptual foundations for spam filtering. Proceedings of the American Society for Information Science and Technology, 42(1) (2005)
2. E.G. Dada, J.S. Bassi, H. Chiroma, A.O. Adetunmbi & O.E. Ajibuwa, Machine learning for email spam filtering: review, approaches and open research problems. Heliyon, 5(6) (2019)
3. X.L. Wang, Learning to classify email: a survey. In 2005 International conference on machine learning and cybernetics, 9, 5716-5719 (2005)
4. E.P. Sanz, J.M.G. Hidalgo & J.C.C Pérez, Email spam filtering. Advances in computers, 74, 45-114 (2008)
5. S. Dhanaraj & V. Karthikeyani, A study on e-mail image spam filtering techniques. International conference on pattern recognition, informatics and mobile engineering, 49-55 (2013)
6. A. Bhowmick & S.M. Hazarika, Machine learning for e-mail spam filtering: review, techniques and trends. arXiv preprint:1606.01042 (2016)
7. G. Laorden, X. Ugarte-Pedrero, I. Santos, B. Sanz, J. Nieves & P.G. Bringas, Study on the effectiveness of anomaly detection for spam filtering. Information Sciences, 277, 421-444 (2014)
8. Z.S. Torabi, M.H. Nadimi-Shahraki & A. Nabiollahi, Efficient support vector machines for spam detection: a survey. International Journal of Computer Science and Information Security, 13(1), 11 (2015)
9. A. Kumar, “Spam email Dataset”, 2023 Retrieved on 2024, Retrieved from: https://www.kaggle.com/datasets/jackksoncsie/spam-email-dataset
10. B. Yu & Z.B. Xu, A comparative study for content-based dynamic spam classification using four machine learning algorithms. Knowledge-Based Systems, 21(4), 355-362 (2008)
11. K.K. Hiran, R.K. Jain, K. Lakhwani & R. Doshi, Machine Learning: Master Supervised and Unsupervised Learning Algorithms with Real Examples (2021)
12. D. Sculley & G.M. Wachman, Relaxed online SVMs for spam filtering. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, 415-422 (2007)

13. K. Tretyakov, Machine learning techniques in spam filtering. In Data mining problem-oriented seminar, 3(177), 60-79 (2004)
14. M. Barreno, B. Nelson, R. Sears, A.D. Joseph & J.D. Tygar, Can machine learning be secure?. In Proceedings of ACM Symposium on Information, computer and communications security, 16-25 (2006)