https://doi.org/10.1007/s42979-023-02330-x

ORIGINAL RESEARCH

Machine-Learning-Based Spam Mail Detector

Panem Charanarur[1] · Harsh Jain[2] · G. Srinivasa Rao[3] · Debabrata Samanta[4] · Sandeep Singh Sengar[5] · Chaminda Thushara Hewage[5]

Received: 11 October 2022 / Accepted: 30 August 2023

Abstract
The proliferation of spam emails, a predominant form of online harassment, has elevated the significance of email in daily life. As a consequence, a substantial portion of individuals remain vulnerable to fraudulent activities. Despite Gmail's "spam mail filtration system," its effectiveness is not absolute. It occasionally misclassifies legitimate messages, leading to their confinement in the spam folder, or overlooks potentially harmful spam emails. This results in the occurrence of false positives. This research scrutinizes the historical data, cookies, caches, Session Restores, flash artifacts, and super cookies of Internet Explorer, Firefox, and Chrome on the Windows 10 platform. Data was collected through Google, Firefox, and Internet Explorer, operating within the Windows 10 environment. It has been observed that browsers store user behavior data on the host computer's hard drive. The implications of this study hold substantial value for computer forensics researchers, law enforcement professionals, and digital forensics experts. The study leverages Python, alongside pertinent libraries such as pandas, Numpy, Matplotlib, scikit-learn, and flask, to facilitate its investigation. The experiment result and analysis show KN and NB algorithms have the best accuracy and precision score compared to other Algorithms.

Keywords E-mail · Spam mail · Trap · Trash folder · Inbox · Cookies

Introduction

Detecting spam has become an urgent issue in recent years as a direct result of the meteoric rise in the volume of unwanted messages posted on social media. The more email addresses and social media accounts a person has, the more spam they will get. This holds true for sites like Facebook, Twitter, and YouTube as well. The usage of social media is growing at a startlingly fast pace, which is especially concerning given the current pandemic [18]. Users of social networking

Panem Charanarur, Harsh Jain, G. Srinivasa Rao, Debabrata Samanta, Sandeep Singh Sengar and Chaminda Thushara Hewage contributed equally to this work.

This article is part of the topical collection "Cyber Security and Privacy in Communication Networks" guest edited by Rajiv Misra, R. K. Shyamsunder, Alexiei Dingli, Natalie Denk, Omer Rana, Alexander Pfeiffer, Ashok Patel and Nishtha Kesswani.

✉ Sandeep Singh Sengar
SSSengar@cardiffmet.ac.uk

Panem Charanarur
panem.charanarur_tripura@nfsu.ac.in

Harsh Jain
haarshjain05@gmail.com

G. Srinivasa Rao
srinivasarao.gundu@gmail.com

Debabrata Samanta
debabrata.samanta369@gmail.com

Chaminda Thushara Hewage
chewage@cardiffmet.ac.uk

[1] Department of Cyber Security and Digital Forensics, National Forensic Sciences University Tripura campus, Agartala, Tripura, India

[2] School of Cyber Security and Digital Forensics, Ponda, Goa, India

[3] Department of Computer Science, Government Degree College Sitaphalmandi, Hyderabad, India

[4] Department of Computing and Information Technologies, Rochester Institute of Technology Kosovo, Germia Campus, Prishtina 10000, Kosovo

[5] Department of Computer Science, Cardiff Metropolitan University, Cardiff CF5 2YB, UK

Published online: 08 November 2023

platforms are subjected to an overwhelming number of text messages, the vast majority of which are spam. Viruses and other forms of malware are often disseminated via spam emails, which may take the shape of links, programs, accounts, news, reviews, rumors, and so on. To improve the safety of social media platforms, it is necessary to identify and control spam text.In this article, we give a thorough assessment of current research on approaches for recognizing and classifying spam texts that are posted on social networking platforms. The subject of spam identification and classification is investigated in this paper, along with the use of text-based approaches, as well as Machine Learning and Deep Learning [22, 19]. In addition to this, we go into the challenges associated with spam detection, including the methods and data sets that are presently being used by researchers working in this field. The term "spam" refers to any communication that is sent or received through digital media that was not sought by the recipient. This might be a social networking site, a microblogging service, an online video platform, an electronic mail service, or something else entirely. It is created by spammers to deceive users of social media platforms into visiting dangerous websites or clicking on links that lead to spam [7]. By encouraging recipients to visit harmful websites or download malicious software via the use of links included within the spam, the spammers who are responsible for sending these emails hope to achieve their aim. Because sending unsolicited emails may be rather lucrative for spammers [23], it is probable that they will continue to engage in this practice. Despite the many attempts that have been made, it seems that the amount of spam that is being received in people's inboxes is still increasing [14]. Both companies that rely on email for their day-to-day operations and average consumers who make use of email are losing money as a result of spam [17].

Aim and Objectives of the Study

The primary objective of this project revolves around enhancing the effectiveness of spam email detection and management. Within the proposed system, two distinct filtering models are employed to identify and categorize spam emails more efficiently.

The initial approach involves utilizing a concept known as "Opinion Rank," a mechanism designed to assess a user's credibility based on their email address. This assessment is derived from both high page rank and inverse page rank considerations. The Opinion Rank algorithm amalgamates these assessments, calculates their average, and assigns a unified ranking to establish an order. Subsequently, the system leverages Latent Dirichlet Allocation, a probabilistic topic modeling technique utilized to categorize content or documents based on specific topics. This optimization process aids in effectively filtering out spam emails, thereby contributing to a reduction in the influx of undesired messages.

In summary, this project is centered on elevating the efficiency of spam email identification and organization. The integration of Opinion Rank and Latent Dirichlet Allocation models collectively enhances the system's ability to accurately detect and manage spam, leading to a more streamlined and effective email communication experience.

Motivations

Due to the exponential rise in the volume of unwanted emails, commonly referred to as spam, there is an urgent demand for the development of more reliable and robust anti-spam filters. Recent advancements, particularly in the realm of machine learning, have exhibited notable efficiency in discerning and segregating spam emails. In this study, we present a comprehensive examination of some of the most widely adopted machine learning-based spam filtering approaches within the context of email communication.

The primary objective of this research is to provide a comprehensive overview of significant spam filtering paradigms, endeavors, effectiveness, and research trends. Our inquiry encompasses the analysis of methods employed to distinguish spam emails from legitimate ones, along with an exploration of diverse endeavors pursued by researchers to combat spam through the integration of machine learning techniques.

The analysis conducted in this study not only assesses the merits and demerits of prevalent machine learning algorithms employed in the domain of spam filtering, but it also underscores the persisting challenges that persist in this arena. Furthermore, our investigation underscores the potential prospects of deep learning, including deep adversarial learning, which we anticipate to emerge as pertinent solutions in addressing the intricate issue of spam emails.

In summation, this research endeavors to contribute a holistic understanding of the dynamic landscape of machine learning-based spam filtering. By evaluating the ongoing efforts, uncovering the strengths and weaknesses of current algorithms, and anticipating emerging trends, we aim to pave the way for more robust and effective anti-spam strategies in the realm of email communication.

Literature Review

Links to websites that include sexual material and films that have no discernible function are two common types of spam that may often be found on video-sharing networks like YouTube, for instance. Other common types of spam include advertisements. Some of these remarks are generated by computer programs that are executed in a completely hands-off manner. On online networks for the sharing of video games, the act of flooding a platform with messages, demanding membership in a specific group, violating copyright laws, and engaging in other behaviors of a similar nature is often referred to as spam. However, there are times when even seasoned internet users are unable to agree on what constitutes spam. The term "blog comment spam," which is sometimes referred to as "splog" in certain circles, describes remarks that go off-topic or are otherwise unrelated to the primary debate taking place on a blog. References to or direct linkages to the websites of unaffiliated commercial firms appear rather often in these remarks. Some blogs, which have earned the derogatory name of "splogs," are notorious for just copying and pasting text from other websites without offering any type of original thinking or analysis [4]. User evaluations of goods that have been posted on social networking sites are yet another potential source of spam [6]. According to the findings of study carried out by Liu and Pang, more than 35 percent of internet testimonials may be considered to be spam [2]. The purpose of writing fraudulent reviews such as these is to influence the judgments that buyers come to about a product and to artificially inflate its overall rating. As a consequence of this, it would appear that identifying fake reviews is a significant problem. If this significant problem is not resolved, it is possible that online review systems will become completely ineffective [20]. As a consequence of this, it would seem that recognizing fake reviews is difficult. As a direct consequence of this, it is probable that this will be the outcome. Users of social networking sites like Facebook and Twitter often express the concern that they will be inundated with an excessive amount of spam SMS messages sent from bogus accounts. This is one of the most popular anxieties among these users. Research such as the one that was carried out by [1] is just one example of the abundance of research that has used features of Facebook such as communities, URLs, videos, and photographs to analyze spam content. Other examples include research such as the one that was carried out. According to [8] research, statistical methods have the potential to be used to discriminate between spam accounts and real user profiles. By modeling their approach after that of spammers, came up with a one-of-a-kind technique for spotting spam in the material that users of social media platforms posted. They were able to do this using honey profiles [21]. Because graph models were able to uncover associations between social media users, they were also commonly used for the detection of spam based on the various attributes of the map. This was done by comparing the spam to the characteristics of the map. Analysis of the map's many features allowed for the successful completion of this task [12]. In recent years, there has been an increase in the adoption of algorithms that use machine learning in a variety of sectors, including the identification of spam [11].

The distinction between my work and prior research lies in several key aspects:

Approach and focus While prior research might have discussed general spam detection algorithms, my work within the ”Machine Learning based Spam Mail Detector” domain uniquely centers on developing and enhancing spam detection using machine learning techniques. Modeling addresses the challenge of accurately differentiating between legitimate emails and spam messages.

Data and features My study extensively utilizes labeled datasets of emails to train machine learning models for effective spam detection. This is a departure from some prior research that might have focused on rule-based or heuristic methods. By leveraging machine learning, my work adapts to evolving spam patterns and learns to identify subtle variations that could indicate spam.

Algorithm selection and evaluation Within the realm of "Machine Learning based Spam Mail Detector," my research involves the careful selection and tuning of machine learning algorithms such as KN and NB algorithms. The aim is to optimize detection accuracy and minimize false positives.

Problem Statement

The amount of a person's trust in the system has a tendency to shift throughout the course of their lifetime as they gather more life experience and become more active in their social networks. Few solutions distinguish between new and old tags, which is necessary to manage the associated trust dynamics. Future research could benefit from paying more attention to trust dynamics, which could make modeling more effective when used in practice. Since they depend on textual data, present approaches all too often operate on the assumption that the environment in question is exclusively monolingual. However, given that people from all over the globe use social networking sites, it stands to reason that tags and comments will be posted in a great number of different languages. Because of linguistic spam, it's possible that specific textual information may be ignored as being untrue. Therefore, a potential solution to this problem would be to include a large number of languages in the trust modelling process. Communication between users of various networks is a growing practice.For instance, users can sign in to various social networking sites with their Facebook login information. Consequently, one of the upcoming difficulties will be figuring out how to effectively distribute and integrate trust models across many domains. Although the audio and visual content aspects of multimedia materials have the potential to give important information about the relevance of the content, the majority of the currently available algorithms for noise and spam reduction rely primarily on the processing of textual tags and the analysis of user profiles.

Research Methodology

There are a number of techniques available for detecting and limiting spam emails. Our primary motivation for doing so is to investigate existing spam text detection and categorization methods. Here, we'll discuss the survey methodology we used to gather data for our in-depth analysis of spam filters.

Selection of Keywords and Data Sources

Initial search terms were selected with care, based on our study aims. After an initial search, many keywords were developed using fresh terms found in a number of connected publications. These terms were subsequently narrowed to better suit the aims of the study. Based on the objectives of the survey, we identified a set of keywords to use in the first search, and then we used those keywords to identify themes in the articles we read. We then narrow down the keyword pool to achieve our study objective.

Database Selection

To conduct the literature review, we combed through a number of articles from various online scholarly journals. During the process of putting up the research papers for our study, we visited a librarian and made use of a broad variety of resources. We used a number of search phrases, such as "spam text," "spam tweets," "spam reviews," and "spam social media" are some instances [9, 3]. The process of recognizing spam messages Additionally, structuring it in accordance with the most pertinent classes necessitates a variety of unique operating processes. Compiling data from many digital sources, including Twitter, is the first thing that needs to be accomplished. Online discussion groups, Facebook, and email are all suitable possibilities. The next step is This stage is referred to as the pre-processing phase, and it is at this stage that a number of Natural Language Processing (NLP) strategies are now being used. Used to remove material that has been determined to be either irrelevant or duplicate. The third phase involves the extraction of characteristics from the data. Data derived from text using methods such as term frequency inversion [15]. N-grams, Document Frequency (also known as TF-IDF), and Word Embedding. These methods are put into practice. These characteristics. Words and text are converted by algorithms that extract and encode them. Into a numerical vector to make categorization easier. Figure 1 shows Spam and Non-Spam categorization.

The first and maybe most critical step in any cleaning operation is to vacuum the area to be cleaned. The textual data obtained from a dataset absolutely need preprocessing to be useful. Develop a strategy for getting rid of unnecessary items. Before going on to the next level, the dataset has to have any errors removed from it. Attempting to pull out characteristics from the text [13, 16]. There are effects from the outside. The text, as well as any punctuation, special characters, http connections, letters, and stop words, are all contained in the dataset. Before carrying out further processing on the text, the various methods of preprocessing shown in Fig. 3 may be used to clear it of any unnecessary information.

Tokenization

Dissecting words into their constituent parts (tokens) is a key step in this process. The content is cleaned up by getting rid of extraneous elements including HTML tags, punctuation, and more [10]. Whitespace tokenization is the gold standard of tokenization techniques. In this process, all of the whitespace in the text is eliminated, and the text is split down into individual words. Python's ”regular expressions” module is widely used for Natural Language Processing (NLP) activities and may be used to tokenize text. In Table 6 below, we see a representation of a sample sentence and associated tokens [5].

Stemming

The technique of etymological reduction examines how various words may be broken down into their simplest components. Using the Porter Stemmer program and the Natural Language Tool Kit library, you can generate meaningless words that are not found in any dictionary. When a larger portion of a word is removed during stemming than is necessary, over-stemming happens and the words are wrongly reduced to the same root word. It's possible that certain words will be incorrectly boiled down to more than one stem because to under stemming.

Lemmatization

To follow the development of a phrase, this approach employs lexical analysis and dictionaries. The words play, playing, and played are all derived from the same Lemma, or root word. As a result, the Lemma of these phrases is "play." The WordNet Lemmatizer module in the Python NLTK finds the WordNet corpus for lemmas. Correct lemmatization necessitates context explanations.

Normalization

During tokenization, phrases are broken down into their component parts to decrease the amount of tokens used. By doing so, it aids in decluttering texts. Sentiment

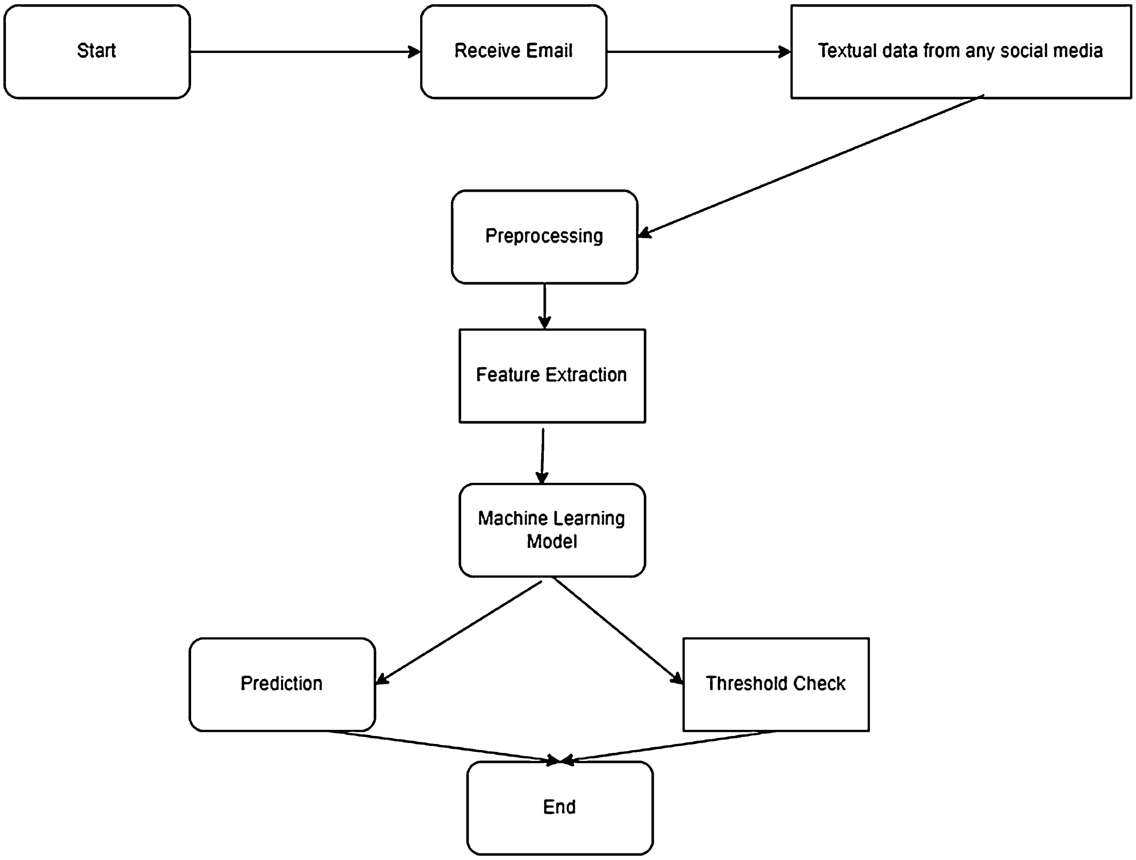


Fig. 1 Spam and Non-Spam categorization

categorization accuracy was improved by 4% when using a text normalization strategy for Tweets.

Stopwords Removal

Because stop words such as "a," "the," "an," and "so" may greatly reduce the size of a dataset, we need to give serious consideration to using them in a more frequent fashion. They are examples of linguistic terms that are often used yet do not have a clear definition. Using the NLTK Python Library, it is possible to successfully get rid of them.

Feature-Extraction Techniques

Because text is written as a string of characters, but machine learning algorithms perform better with numerical data, the text input must be converted into numerical vectors. This is due to the fact that text is written as a series of letters. This technique is used to derive essential information from a text with the purpose of improving readability across a variety of platforms, including computers.

Bag of Words (BoW)

It constructs a collection of word presence features from all the words in an instance. Consider the papers to be containers for the words. Each bag contains a word collection, and each document serves as a storage space for the documents included inside that bag. The technique's method is referred to as feature extraction since it builds a word presence feature set from every word that is present in an instance. A document's term frequency and instances of the same word can be retrieved and displayed in vector format. By combining n-grams with skip-gram word embedding, created a model for detecting spam in peer reviews. To identify fake reviews, they used deep learning models on 400 good and negative testimonials for hotels posted on TripAdvisor. It is possible to see how each document is related to its words.

N-grams

N-grams are sequences of n-words or n-tokens that may be found in written communication. They have a significant bearing on the performance of a variety of natural language processing applications. It was found by ltk and Güngor that increasing the value of n for the first n-words heuristic to 50 produced better results. In the process of screening spam from e-mails, this was one of the ideas that was explored.

Term Frequency-Inverse Document Frequency (TF-IDF)

To determine the frequency with which certain words occur in a given dataset, one method that may be used is called the bag of words technique. It is possible to use the inverse-document frequency (IDF) on its own to give a unique perspective on the issue. You can draw the reader's attention to the most important issues that are discussed in a document using star ratings to highlight certain keywords in the document.

One Hot Encoding

Each word or phrase is represented by either a single 1 or a 0 depending on whether it is a 1 or a 0. Each and every one of the words in the language is connected to its very own one-of-a-kind hot vector. The word list may be thought of as a matrix, and it could be put into action by making use of the NLTK Python Module.

Word Embedding

For situations when we just have a little quantity of information, one-hot encoding is the way to go. This approach allows us to encode a large vocabulary as the complexity grows significantly. Vector representations of words are used in word embedding, a method of word representation.

Word2Vec

The word2vec neural network has two layers and analyses text by converting individual words into vectors. A continuous skip-gram or bag of words, depending on the task at hand, may be implemented. The CBOW model can be trained more quickly and has a somewhat higher level of accuracy for phrases that are more often used.

Glove Word Embedding

It is a paradigm for automatically producing, without the assistance of humans, a vector that may represent words or sentences. The degree to which two statements are semantically related to one another is used as a factor in the calculation used to determine how far apart the two phrases are. Pennington, Socher, and Manning were the pioneering researchers who were the first to use it in their study. This is accomplished by the use of matrix factorization techniques, and it shows the frequency with which words appear in a corpus through the utilization of a co-occurrence matrix. The results of computing each word embedding's co-occurrence probability are shown in Eq. (1).

$$F(t_a, t_b, t_c) = P_{ac} * P_{bc} \tag{1}$$

If texts t_a and t_c are discovered together, there is a possibility that they will appear together as part of a P_{ac} co-occurrence in the future. The number P_{bc} represents the likelihood that both t_b and t_c will be found in the same piece of writing at the same time. Research using techniques such as tf-idf, bag of words, and n-grams is summarized.

Mathematical Equation

Calculating the likelihood that a message that contains a certain term is unsolicited commercial email Let's say the message that we suspect includes the phrase "REPLICA". The vast majority of individuals who regularly check their email are aware that this letter is almost certainly an example of spam and, more specifically, a solicitation to sell knockoff versions of timepieces manufactured by well-known manufacturers. The program used to identify spam, on the other hand, is unable to "know" such things; all it can do is calculate probabilities. The program makes use of a formula that is derived from Bayes' theorem to arrive at that conclusion.

$$\Pr(S) = 0.8; \Pr(H) = 0.2 \tag{2}$$

where:

- $\Pr(S)$ is the probability that a message is spam, given that it contains the word "replica;"
- $\Pr(H)$ is the probability that the word "replica" appears in spam messages;
- S is the probability that any given message is spam;
- H is the probability that any given message is not spam (is "ham");
- W is the probability that the word "replica" appears in ham messages. the frequency with which a phrase appears in spam. Recent data shows that the probability that every given communication would contain spam has climbed to at least 80%:

$$\Pr(S) = 0.5; \Pr(H) = 0.5 \tag{3}$$

On the other hand, the vast majority of Bayesian spam detection software operates on the presumption that there is no a priori reason for any incoming message to be spam rather than ham and therefore assigns an identical probability of 50% to each scenario.

$$\Pr(S \mid W) = \frac{\Pr(W \mid S)}{\Pr(W \mid S) + \Pr(W \mid H)} \tag{4}$$

It is believed that the filters that employ this theory are "not prejudiced," which means that they do not have any preconceived notions about the emails that are being received. This assumption makes it possible to reduce the complexity of the general formula to:

$$p = \frac{p_1 p_2 \cdots p_N}{p_1 p_2 \dots p_N + \left(1 - p_1\right)\left(1 - p_2\right) \dots \left(1 - p_N\right)} \tag{5}$$

Combining Individual Probabilities

In natural languages such as English, for example, the chance of discovering an adjective is influenced by the likelihood that the language already has a noun. However, when it comes to computer programming, the statistical correlations between individual words are almost never known, and as a result, the connection between them cannot be entirely accurate.

$$\frac{1}{p} - 1 = \frac{\left(1 - p_1\right)\left(1 - p_2\right) \dots \left(1 - p_N\right)}{p_1 p_2 \cdots p_N} \tag{6}$$

P is the chance that the message in question includes either the first word (for example, “replica”) or two words (for example, “watches”) of a specific category. It is possible to rewrite the formula that is used to aggregate the probabilities of individual occurrences such that it takes into account floating-point underflow.

$$\begin{gathered} \ln\left(\frac{1}{p} - 1\right) = \sum_{i=1}^{N}\left[\ln\left(1 - p_i\right) - \ln p_i\right] \\ \text{Let } \eta = \sum_{i=1}^{N}\left[\ln\left(1 - p_i\right) - \ln p_i\right] \end{gathered} \tag{7}$$

Taking log on both sides, let, therefore.

$$\frac{1}{p} - 1 = e^{\eta} \tag{8}$$

The alternative method for calculating the combined probability:

$$p = \frac{1}{1 + e^{\eta}} \tag{9}$$

The mathematical equation for precision and recall are as respective

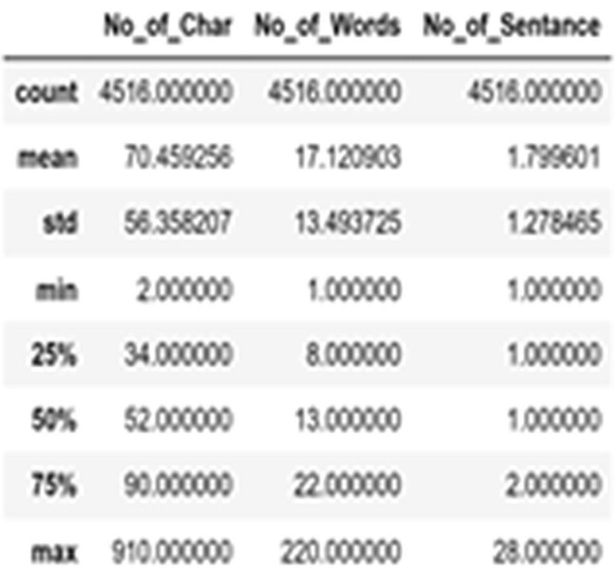
$$\text{Precison} = \text{tp}/\text{tp} + \text{fp} \tag{10}$$

$$\text{Recall} = \text{tp}/\text{tp} + \text{fn} \tag{11}$$

tp: True positive
fp: False positive
tn: True negative
fn: False negative

Table 1 Compare between Logistic and Random Forest

Sl. no.	Model	Accuracy	Precision
1	Logistic (Model 3)	99%	40%
2	Random Forest	78%	44%



	No_of_Char	No_of_Words	No_of_Sentance
count	4516.000000	4516.000000	4516.000000
mean	70.459256	17.120903	1.799601
std	56.358207	13.493725	1.278465
min	2.000000	1.000000	1.000000
25%	34.000000	8.000000	1.000000
50%	52.000000	13.000000	1.000000
75%	90.000000	22.000000	2.000000
max	910.000000	220.000000	28.000000

Fig. 2 Quick overview of our dataset (both ham and spam mail)

Findings

Random forest makes use of the same ”significant variables” that are utilized in linear models. The following table summarizes the main characteristics of the available models from which the company may make its final decision shown in Table 1.

The approach has a 95% chance of correctly predicting who will pay back the loan, but cannot predict who would default. Nearly 100% accuracy in projecting 0% default rate is a true plus. Since our primary goal is to anticipate defaulters, immediate action is required. A quick overview of our dataset (both ham and spam mail) is shown in Fig. 2. The relation between words in spam and ham mail in red and yellow respectively shown in Fig. 3.

Figure 4 shows most occurred word cloud of spam mail.

Tried some more mail shown in Fig. 5.

Results and Discussion

Here’s a list of software and hardware components that used for this research:

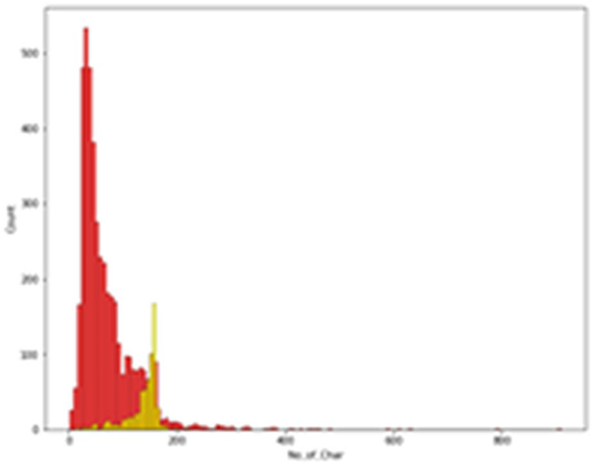


Fig. 3 Relation between word in spam and ham mail in red and yellow respectively

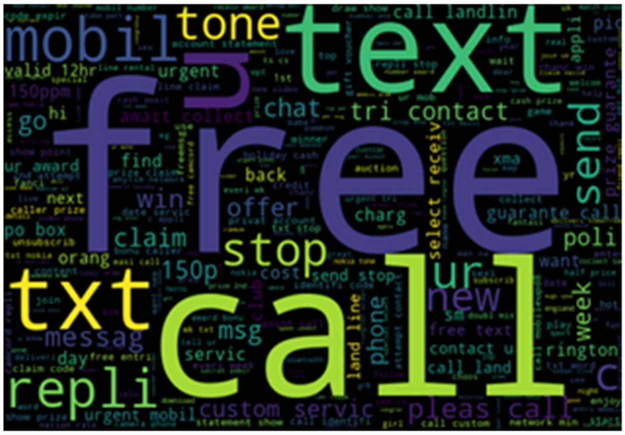


Fig. 4 Most occurred word cloud of spam mail

Software:

Programming languages:

Python: Widely used for machine learning and natural language processing tasks.

Data analysis and machine learning libraries:

Scikit-learn: A comprehensive machine learning library for Python.

NLTK (Natural Language Toolkit): For text processing and analysis.

Text preprocessing tools:

BeautifulSoup: For HTML parsing.

Web Scraping Tools: Such as Scrapy or Beautiful Soup for collecting data.

Integrated Development Environments (IDEs):

Jupyter Notebook: Interactive environment for data analysis and model development.

Hardware:

Computer system:

A modern computer with sufficient processing power, memory, and storage for running data analysis and training machine learning models.

Storage:

Adequate storage space to store datasets, model checkpoints, and research materials.

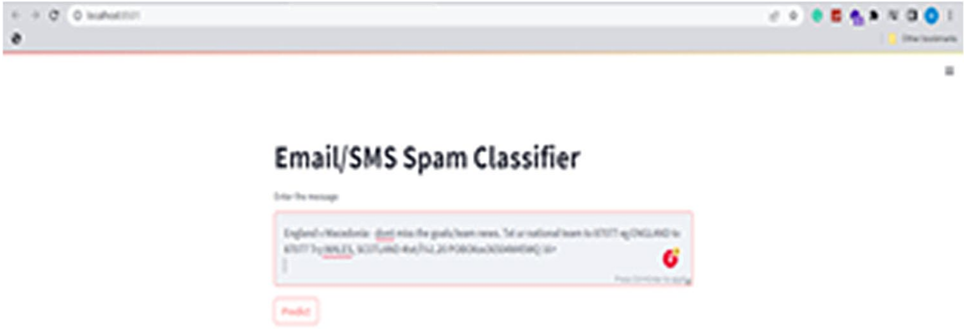
Internet connectivity:

High-speed internet for data collection, research, and collaboration.

Table 2 shows the results of road marking detection. Table 3 shows the accuracy results of 11 different types of algorithms. Results based on accuracy and precision are shown in Table 4. The description of the combined data is shown in Table 5. The discussion of spam data is shown in Table 6. Discussion of ham data is shown in Table 7. Tables 8 show the raw table taken from Kaggle.com. After counting a number of characters and words the modified data are shown in Table 9. Some of the statistics of data with the relationship between char word and sentence in spam data are shown in Table 10. The relationship between char word and sentence in ham data is shown in Table 11. After applying the different model in our ML model, accuracy and precision data is shown in Table 12.

Although this is just the beginning of the research, we have tried to present this paper in a fresh and original way. Everything we have covered is completely new. This process is always being improved, and we strive to have 100% algorithm correctness. The accuracy and Precision of Nave

Fig. 5 Tried some more mail



Email/SMS Spam Classifier

Enter the message

England v Macedonia - dont miss the goals/team news. Txt ur national team to 87077 eg ENGLAND to 87077 Try:WALES, SCOTLAND 4txt/ú1.20 POBOXox36504W45WQ 16+

Table 2 Accuracy of the algorithm

Test image	Results	
	Number of correct detections	Number of missing lines
1	3	2
2	5	0
3	6	2
4	5	1
5	6	0
6	5	2
7	4	0
8	5	1
9	7	5
10	4	2
11	4	2

Bayes are 0.959381 and 1 respectively, which is better than all other algorithms, as shown in the table below, which compares each algorithm we tested with our refined dataset. After making numerous changes to our Nave based algorithm, we were able to achieve the accuracy of 97.1954%. And we're still aiming to get 100% accuracy. Table 13 shows Comparative study with other algorithms.

Conclusions and Future Scope

Since it highlights some of the most significant initiatives in the field, this survey is useful for scholars interested in social media spam detection. In the not-too-distant future, one of our objectives is to expand the number of ways to detect spam and to provide an explanation of the benefits and drawbacks associated with each approach. In conclusion, the ubiquity of spam emails, a prevailing form of online harassment, has underscored the indispensability of email in daily life. Consequently, a significant portion of individuals remain exposed to potential fraudulent activities. Despite Gmail's deployment of a "spam mail filtration system," its efficacy is not absolute, occasionally resulting in the misclassification of legitimate messages and the inadvertent exclusion of potentially harmful spam emails. This phenomenon manifests as the occurrence of false positives, undermining the system's reliability.

The crux of this research involved a meticulous examination of historical data, cookies, caches, Session Restores, flash artifacts, and super cookies emanating from Internet Explorer, Firefox, and Chrome platforms within the Windows 10 environment. The study entailed data collection through Google, Firefox, and Internet Explorer browsers, providing insights into user behavior data stored on the host computer's hard drive. The implications of this study resonate profoundly within the realms of computer forensics, law enforcement, and digital forensics. The meticulous employment of Python, fortified by libraries such as pandas, Numpy, Matplotlib, scikit-learn, and flask, facilitated the investigation process, underscoring the significance of versatile tools in advancing the field.

The culmination of experimentation and analysis reveals a noteworthy revelation: the KN and NB algorithms emerged with the most impressive Accuracy and Precision scores, surpassing their algorithmic counterparts. This outcome underscores the potential of these algorithms to excel in spam detection, reaffirming their efficacy in addressing the challenges of email security and privacy.

In totality, this research serves as a testament to the necessity of continuous vigilance in the realm of spam detection and computer forensics. By dissecting the intricacies of

Table 3 Accuracy results of 11 different types of algorithms

Sl. no.	Algorithm	Accuracy	Precision	Accuracy_max_ft_3000	Precision_max_ft_3000	Accuracy_scaling	Precision_scaling	Accuracy_num_chars	Precision_num_chars
1	KN	0.900387	1	0.905222	1	0.905222	0.97619	0.928433	0.771186
2	NB	0.959381	1	0.971954	1	0.978723	0.946154	0.940039	1
3	ETC	0.977756	0.991453	0.979691	0.97561	0.979691	0.97561	0.976789	0.975
4	RF	0.970019	0.990826	0.975822	0.982906	0.975822	0.982906	0.974855	0.982759
5	SVC	0.972921	0.974138	0.974855	0.974576	0.971954	0.943089	0.866538	0
6	AdaBoost	0.962282	0.954128	0.961315	0.945455	0.961315	0.945455	0.971954	0.950413
7	xgb	0.971954	0.950413	0.968085	0.933884	0.968085	0.933884	0.970019	0.942149
8	LR	0.951644	0.94	0.95648	0.969697	0.967118	0.964286	0.961315	0.971154
9	GBDT	0.951644	0.931373	0.946809	0.927835	0.946809	0.927835	0.948743	0.929293
10	BgC	0.957447	0.861538	0.959381	0.869231	0.959381	0.869231	0.968085	0.913386
11	DT	0.935203	0.838095	0.931335	0.831683	0.932302	0.84	0.943907	0.877358

Table 4 Results based on accuracy and precision

Sl. no.	Algorithm	Variable	Value
1	ETC	Accuracy	0.977756
2	SVC	Accuracy	0.972921
3	xgb	Accuracy	0.971954
4	RF	Accuracy	0.970019
5	AdaBoost	Accuracy	0.962282
6	NB	Accuracy	0.959381
7	BgC	Accuracy	0.957447
8	LR	Accuracy	0.951644
9	GBDT	Accuracy	0.951644
10	DT	Accuracy	0.935203
11	KN	Accuracy	0.900387
12	ETC	Precision	0.991453
13	SVC	Precision	0.974138
14	xgb	Precision	0.950413
15	RF	Precision	0.990826
16	AdaBoost	Precision	0.954128
17	NB	Precision	1
18	BgC	Precision	0.861538
19	LR	Precision	0.94
20	GBDT	Precision	0.931373
21	DT	Precision	0.838095
22	KN	Precision	1

Table 5 Description of the combined data

Parameter	Num_characters	Num_words	Num_sentences
Count	653	653	653
Mean	137.479326	27.675345	2.977029
Std	30.014336	7.011513	1.493676
Min	13	2	1
25%	131	25	2
50%	148	29	3
75%	157	32	4
Max	223	46	9

Table 6 Description of spam data

Parameter	Num_characters	Num_words	Num_sentences
Count	4516	4516	4516
Mean	70.45682	17.123339	1.815545
Std	56.356802	13.491315	1.364098
Min	2	1	1
25%	34	8	1
50%	52	13	1
75%	90	22	2
Max	910	220	38

Table 7 Description of ham data

Parameter	Num_characters	Num_words	Num_sentences
Count	5169	5169	5169
Mean	78.923776	18.456375	1.962275
Std	58.174846	13.323322	1.433892
Min	2	1	1
25%	36	9	1
50%	60	15	1
75%	117	26	2
Max	910	220	38

browser data storage and evaluating the performance of diverse algorithms, this study contributes to the evolving landscape of digital security, enriching the toolkit of professionals tasked with safeguarding digital communications and thwarting cyber threats.

Table 8 Raw data (from kaggle.com)

Text size	V1	V2	Un_manned:2	Un_manned:3	Un_manned:4
2464	Ham	They will pick and drop in car so no problem	NaN	NaN	NaN
1248	Ham	HI HUN! IM NOT COMIN2NITE-TELL EVERY1 IM SORR...	NaN	NaN	NaN
1413	Spam	Dear ‘ve been invited to XCHAT. This is our f..	NaN	NaN	NaN
2995	Ham	They released vday shirts and when U put it on..	NaN	NaN	NaN
4458	spam	Welcome to UK-mobile-date this msg is FREE giv..	NaN	NaN	NaN

Table 9 The modified data

Sl. no.	Mode	Text	Num_characters	Num_words
1	0	Go until jurong point,crazy... available only	111	24
2	0	Ok lar...Joking wif u oni..	29	8
3	1	Free entry in 2.a wkly comp to win FA Cup fina	155	37
4	0	U dun say so early hor...U c already then say..	49	13
5	0	Nah I don’t he goes to usf, he lives aro	61	15

Table 10 Relationship between char word and sentence in spam data

Parameter	Num_characters	Num_words	Num_sentences
Count	4516.000000	4516.000000	4516.000000
Mean	70.456820	17.123339	1.815545
Std	56.356802	13.491315	1.364098
Min	2.000000	1.000000	1.000000
25%	34.000000	8.000000	1.000000
50%	52.000000	13.000000	1.000000
75%	90.000000	22.000000	2.000000
Max	910.000000	220.000000	38.000000

Table 11 Relationship between char word and sentence in ham data

Parameter	Num_characters	Num_words	Num_sentences
Count	653.000000	653.000000	653.000000
Mean	137.479326	27.675345	2.977029
Std	30.014336	7.011513	1.493676
Min	13.000000	2.000000	1.000000
25%	131.000000	25.000000	2.000000
50%	148.000000	29.000000	3.000000
75%	157.000000	32.000000	4.000000
Max	223.000000	46.000000	9.000000

Table 12 Accuracy and precision data

Sl. no.	Algorithm	Variable	Value
1	Accuracy	ETC	0.977756
2	Accuracy	SVC	0.972921
3	Accuracy	xgb	0.971954
4	Accuracy	RF	0.971954
5	Accuracy	AdaBoost	0.970019
6	Accuracy	NB	0.962282
7	Accuracy	BgC	0.959381
8	Accuracy	LR	0.957447
9	Accuracy	GBDT	0.951644
10	Accuracy	DT	0.951644
11	Accuracy	KN	0.935203
12	Precision	ETC	0.900387
13	Precision	SVC	0.991453
14	Precision	xgb	0.974138
15	Precision	RF	0.950413
16	Precision	AdaBoost	0.990826
17	Precision	NB	0.954128
18	Precision	BgC	1.000000
19	Precision	LR	0.861538
20	Precision	GBDT	0.861538
21	Precision	DT	0.940000
22	Precision	KN	0.931373

Table 13 Comparative study with other algorithms

Sl. no.	Algorithm	Accuracy	Precision
1	KN	0.905222	1
2	NB	0.971954	1
3	ETC	0.979691	0.975610
4	RF	0.975822	0.982906
5	SVC	0.974855	0.974576

Funding This study was funded by the Cybersecurity, Information Networks Centre (CINC), Cardiff Metropolitan University, United Kingdom.

Data availability No data are available for this study.

Declarations

Conflict of interest The authors have no conflicts of interest to declare.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Abernethy J, et al. Graph regularization methods for web spam detection. Mach Learn. 2010;81(2):207–25. https://doi.org/10.1007/s10994-010-5171-1.
2. Abu-Nimeh S, Chen TM. Proliferation and detection of blog spam. IEEE Secur Privacy. 2010;8(5). https://doi.org/10.1109/MSP.2010.113.
3. Blanzieri E, Bryl A. A survey of learning-based techniques of email spam filtering. Artif Intell Rev. 2008;29(1):63–92. https://doi.org/10.1007/s10462-009-9109-6.
4. Chu Z, et al. Detecting automation of Twitter Accounts: are you a human, bot, or cyborg? IEEE Trans Depend Secure Comput. 2012;9(6):811–24. https://doi.org/10.1109/TDSC.2012.75.
5. Deshpande VP, et al. An evaluation of Naïve Bayesian anti-spam filtering techniques. 2007 IEEE SMC Information Assurance and Security Workshop, p.333–40. IEEE Xplore. 2007. https://doi.org/10.1109/IAW.2007.381951.
6. Drucker H, Wu D, Vapnik VN. Support vector machines for spam categorization. IEEE Trans Neural Netw. 1999;10(5):1048–54. https://doi.org/10.1109/72.788645.
7. Fattahi J, Mejri M. SpaML: a bimodal ensemble learning spam detector based on NLP techniques. 2020. https://arxiv.org/abs/2010.07444v2.
8. Fisher D, et al. Revisiting Whittaker & Sidner's email overload ten years later. In: Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work. ACM; 2006. p. 309–312. https://doi.org/10.1145/1180875.1180922.
9. Gurunath R, Samanta D. A novel approach for semantic web application in online education based on steganography. Int J Web-Based Learn Teach Technol (IJWLTT). 2022;17(4):1–13. https://doi.org/10.4018/IJWLTT.285569.
10. Gurunath R, et al. Insights into deep steganography: a study of steganography automation and trends. Cyber Secur Netw Secur. 2022:129–55. https://doi.org/10.1002/9781119812555.ch6.
11. Gyongyi Z, Garcia-Molina H. Web spam taxonomy. 2005. Semantic Scholar. https://www.semanticscholar.org/paper/Web-Spam-Taxonomy-Gy%C3%B6ngyi-Garcia-Molina/a9bee91c071d8e8d2c040af9e16f457b51a147fa.
12. Heymann P, Koutrika G, Garcia-Molina H. Fighting spam on social web sites: a survey of approaches and future challenges. IEEE Internet Comput. 2007;11(6):36–45. https://doi.org/10.1109/MIC.2007.125.
13. Hovold J. Naive Bayes spam filtering using word-position-based attributes and Length-Sensitive Classification Thresholds. 2005. https://www.semanticscholar.org/paper/Naive-Bayes-spam-filtering-using-attributes-and-Hovold/76b6697e667653b1cb574009f60c17355b9e7dac.
14. Imam NH, Vassilakis VG, Kolovos D. An empirical analysis of health-related campaigns on Twitter Arabic hashtags. 7th International Conference on Data Science and Machine Learning Applications (CDMA), 2022. p. 29–41. https://doi.org/10.1109/CDMA54072.2022.00011.
15. Khalid I, Khan MS. Email classification analysis using machine learning techniques. Appl Comput Inf. https://doi.org/10.1108/ACI-01-2022-0012.
16. Metsis V, Androutsopoulos I, Paliouras G. Spam filtering with Naive Bayes-which Naive Bayes? In: Semantic Scholar. https://www.semanticscholar.org/paper/Spam-Filtering-with-Naive-Bayes-Which-Naive-Bayes-Metsis-Androutsopoulos/7f5ce28afc0c2eafd4a6ef711e399bee4056c3b8.
17. Piskorski J, et al. Exploring linguistic features for web spam detection: a preliminary study. In: Proceedings of the 4th international workshop on Adversarial information retrieval on the web. Association for Computing Machinery, ACM Digital Library. 2008. p. 25–28. https://doi.org/10.1145/1451983.1451990.
18. Raja PV, et al. Email spam classification using machine learning algorithms. Second International Conference on Artificial Intelligence and Smart Energy (ICAIS). 2022. p. 343–8. https://doi.org/10.1109/ICAIS53314.2022.9743033.
19. Saad O, et al. A survey of machine learning techniques for spam filtering. Semantic Scholar. https://www.semanticscholar.org/paper/A-Survey-of-Machine-Learning-Techniques-for-Spam-Saad-Hassanien/7ed185947e8b29e6187c14a8b59d3aa421302779.
20. Social Networking Service. Wikipedia. https://en.wikipedia.org/w/index.php?title=Social_networking_service&oldid=1178107951. Accessed 1 Oct 2023.
21. Spirin N, Han J. Survey on web spam detection: principles and algorithms. ACM SIGKDD Explor Newsl. 2012; 13(2):50–64. https://doi.org/10.1145/2207243.2207252.
22. Sumithra A, et al. Probability-based Naïve Bayes algorithm for email spam classification. 2022. p. 1–5. https://doi.org/10.1109/ICCCI54379.2022.9740792.
23. Yerima SY, Bashar A. Semi-supervised novelty detection with one class SVM for SMS spam detection. In: 29th International Conference on Systems Signals and Image Processing (IWSSIP), CFP2255E-ART. 2022. p. 1–4. https://doi.org/10.1109/IWSSIP55020.2022.9854496.