Heliyon 5 (2019) e01802

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.heliyon.com

Machine learning for email spam filtering: review, approaches and open research problems

Emmanuel Gbenga Dada [a,*], Joseph Stephen Bassi [a], Haruna Chiroma [b], Shafi'i Muhammad Abdulhamid [c], Adebayo Olusola Adetunmbi [d], Opeyemi Emmanuel Ajibuwa [e]

[a] Department of Computer Engineering, University of Maiduguri, Maiduguri, Nigeria
[b] Department of Computer Science, Federal College of Education (Technical), Gombe, Nigeria
[c] Department of Cyber Security Science, Federal University of Technology Minna, Minna, Nigeria
[d] Department of Computer Science, Federal University of Technology Akure, Akure, Nigeria
[e] Department of Electrical Engineering, University of Ilorin, Ilorin, Nigeria

ARTICLE INFO

Keywords:
Computer science
Computer security
Computer privacy
Analysis of algorithms
Machine learning
Spam filtering
Deep learning
Neural networks
Support vector machines
Naïve Bayes

ABSTRACT

The upsurge in the volume of unwanted emails called spam has created an intense need for the development of more dependable and robust antispam filters. Machine learning methods of recent are being used to successfully detect and filter spam emails. We present a systematic review of some of the popular machine learning based email spam filtering approaches. Our review covers survey of the important concepts, attempts, efficiency, and the research trend in spam filtering. The preliminary discussion in the study background examines the applications of machine learning techniques to the email spam filtering process of the leading internet service providers (ISPs) like Gmail, Yahoo and Outlook emails spam filters. Discussion on general email spam filtering process, and the various efforts by different researchers in combating spam through the use machine learning techniques was done. Our review compares the strengths and drawbacks of existing machine learning approaches and the open research problems in spam filtering. We recommended deep leaning and deep adversarial learning as the future techniques that can effectively handle the menace of spam emails.

1. Introduction

In recent times, unwanted commercial bulk emails called spam has become a huge problem on the internet. The person sending the spam messages is referred to as the spammer. Such a person gathers email addresses from different websites, chatrooms, and viruses [1]. Spam prevents the user from making full and good use of time, storage capacity and network bandwidth. The huge volume of spam mails flowing through the computer networks have destructive effects on the memory space of email servers, communication bandwidth, CPU power and user time [2]. The menace of spam email is on the increase on yearly basis and is responsible for over 77% of the whole global email traffic [3]. Users who receive spam emails that they did not request find it very irritating. It is also resulted to untold financial loss to many users who have fallen victim of internet scams and other fraudulent practices of spammers who send emails pretending to be from reputable companies with the intention to persuade individuals to disclose sensitive personal information like passwords, Bank Verification Number (BVN) and credit card numbers.

According to report from Kaspersky lab, in 2015, the volume of spam emails being sent reduced to a 12-year low. Spam email volume fell below 50% for the first time since 2003. In June 2015, the volume of spam emails went down to 49.7% and in July 2015 the figures was further reduced to 46.4% according to anti-virus software developer Symantec. This decline was attributed to reduction in the number of major botnets responsible for sending spam emails in billions. Malicious spam email volume was reported to be constant in 2015. The figure of spam mails detected by Kaspersky Lab in 2015 was between 3 million and 6 million. Conversely, as the year was about to end, spam email volume escalated. Further report from Kaspersky Lab indicated that spam email messages having pernicious attachments such as malware, ransomware, malicious macros, and JavaScript started to increase in December 2015. That drift was sustained in 2016 and by March of that year spam email volume had quadrupled with respect to that witnessed in 2015. In March 2016, the volume of spam emails discovered by Kaspersky Lab is 22,890,956. By that time the volume of spam emails had

* Corresponding author.
E-mail address: gbengadada@unimaid.edu.ng (E.G. Dada).

https://doi.org/10.1016/j.heliyon.2019.e01802
Received 3 September 2018; Received in revised form 25 February 2019; Accepted 20 May 2019

skyrocketed to an average of 56.92% for the first quarter of 2016. Latest statistics shows that spam messages accounted for 56.87% of e-mail traffic worldwide and the most familiar types of spam emails were healthcare and dating spam. Spam results into unproductive use of resources on Simple Mail Transfer Protocol (SMTP) servers since they have to process a substantial volume of unsolicited emails [127]. The volume of spam emails containing malware and other malicious codes between the fourth quarter of 2016 and first quarter of 2018 is depicted in Fig. 1 below.

To effectively handle the threat posed by email spams, leading email providers such as Gmail, Yahoo mail and Outlook have employed the combination of different machine learning (ML) techniques such as Neural Networks in its spam filters. These ML techniques have the capacity to learn and identify spam mails and phishing messages by analyzing loads of such messages throughout a vast collection of computers. Since machine learning have the capacity to adapt to varying conditions, Gmail and Yahoo mail spam filters do more than just checking junk emails using pre-existing rules. They generate new rules themselves based on what they have learnt as they continue in their spam filtering operation. The machine learning model used by Google have now advanced to the point that it can detect and filter out spam and phishing emails with about 99.9 percent accuracy. The implication of this is that one out of a thousand messages succeed in evading their email spam filter. Statistics from Google revealed that between 50-70 percent of emails that Gmail receives are unsolicited mail. Google's detection models have also incorporated tools called Google Safe Browsing for identifying websites that have malicious URLs. The phishing-detection performance of Google have been enhanced by introduction of a system that delay the delivery of some Gmail messages for a while to carry out additional comprehensive scrutiny of the phishing messages since they are easier to detect when they are analyzed collectively. The purpose of delaying the delivery of some of these suspicious emails is to conduct a deeper examination while more messages arrives in due course of time and the algorithms are updated in real time. Only about 0.05 percent of emails are affected by this deliberate delay.

Though there are several email spam filtering methods in existence, the state-of-the-art approaches are discussed in this paper. We explained below the different categories of spam filtering techniques that have been widely applied to overcome the problem of email spam.

- **Content Based Filtering Technique:** Content based filtering is usually used to create automatic filtering rules and to classify emails using machine learning approaches, such as Naïve Bayesian classification, Support Vector Machine, K Nearest Neighbor, Neural Networks. This method normally analyses words, the occurrence, and distributions of words and phrases in the content of emails and used then use generated rules to filter the incoming email spams [28].
- **Case Base Spam Filtering Method:** Case base or sample base filtering is one of the popular spam filtering methods. Firstly, all emails both non-spam and spam emails are extracted from each user's email using collection model. Subsequently, pre-processing steps are carried out to transform the email using client interface, feature extraction, and selection, grouping of email data, and evaluating the process. The data is then classified into two vector sets. Lastly, the machine learning algorithm is used to train datasets and test them to decide whether the incoming mails are spam or non-spam [28].
- **Heuristic or Rule Based Spam Filtering Technique:** This approach uses already created rules or heuristics to assess a huge number of patterns which are usually regular expressions against a chosen message. Several similar patterns increase the score of a message. In contrast, it deducts from the score if any of the patterns did not correspond. Any message's score that surpasses a specific threshold is filtered as spam; else it is counted as valid. While some ranking rules do not change over time, other rules require constant updating to be able to cope effectively with the menace of spammers who continuously introduce new spam messages that can easily escape without been noticed from email filters [28]. A good example of a rule based spam filter is SpamAssassin [35].
- **Previous Likeness Based Spam Filtering Technique:** This approach uses memory-based, or instance-based, machine learning methods to classify incoming emails based to their resemblance to stored examples (e.g. training emails). The attributes of the email are used to create a multi-dimensional space vector, which is used to plot new instances as points. The new instances are afterward allocated to the most popular class of its K-closest training instances [33]. This approach uses the k-nearest neighbor (kNN) for filtering spam emails.
- **Adaptive Spam Filtering Technique:** The method detects and filters spam by grouping them into different classes. It divides an email corpus into various groups, each group has an emblematic text. A comparison is made between each incoming email and each group, and a percentage of similarity is produced to decide the probable group the email belongs to [137].

Many researchers and academicians have proposed different email spam classification techniques which have been successfully used to classify data into groups. These methods include probabilistic, decision

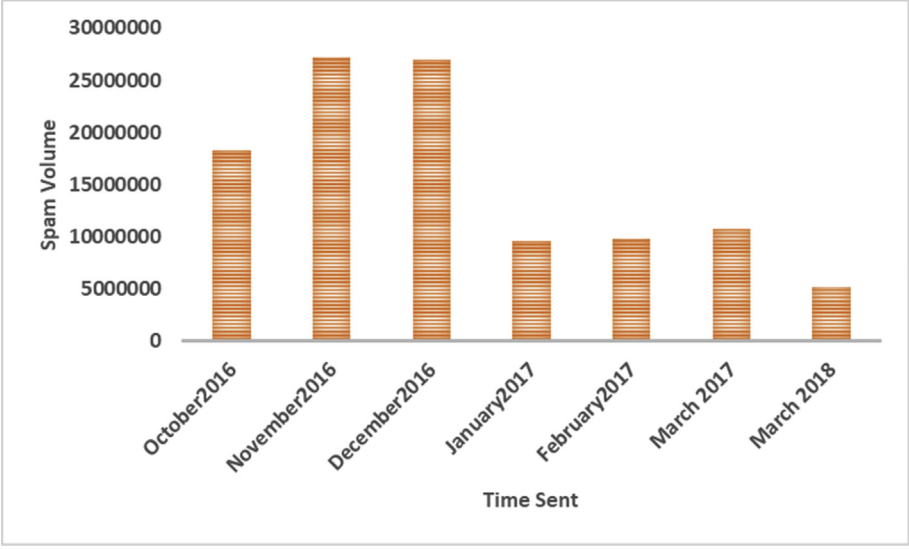

Fig. 1. The volume of spam emails 4th quarter 2016 to 1st quarter 2018.

tree, artificial immune system [4], support vector machine (SVM) [5], artificial neural networks (ANN) [6], and case-based technique [7]. It have been shown in literature that it is possible to use these classification methods for spam mail filtering by using content-based filtering technique that will identify certain features (normally keywords frequently utilised in spam emails). The rate at which these features appear in emails ascertain the probabilities for each characteristic in the email, after which it is measured against the threshold value. Email messages that exceed the threshold value are classified as spam [8]. ANN is a non-linear model that seeks to imitate the functions of biological neural networks. It is made up of simple processing components named neurons and carries out its computational operations by processing information [9,10]. Several research work have employed neural network to classify unwanted emails as spam by applying content-based filtering. These techniques decide the properties by either computing the rate of occurrence of keywords or patterns in the email messages. Literatures show that Neural Network algorithms that are utilised in email filtering attain moderate classification performance. Some of the most popular spam email classification algorithms are Multilayer Perceptron Neural Networks (MLPNNs) and Radial Base Function Neural Networks (RBFNN). Researchers used MLPNN as a classifier for spam filtering but not many of them used RBFNN for classification.

Support Vector Machines (SVM) has proved over the years to be one of the most powerful and efficient state-of-the-art classification techniques for solving the email spam problem [78]. They are supervised learning models that analyze data and identify patterns used for categorisation and exploring the relationship between variables of interest. SVM algorithms are very potent for the identification of patterns and classifying them into a specific class or group. They can be easily trained and according to some researchers, they outperform many of the popular email spam classification methods [130,131]. This is because during training, SVM use data from email corpus. However, for high dimension data, the strength and efficacy of SVM diminish over time due to computational complexities of the processed data [132,133]. According to [134], SVM is a good classifier due to its sparse data format and satisfactory recall and precision value. SVM has high classification accuracy. Moreover, SVM is considered a notable example of "kernel methods", which is one of the central areas of machine learning. Decision tree is another machine learning algorithm that has been successfully applied to email spam filtering. Decision trees (DT) need comparatively minute effort from users during training of datasets. DT completely perform variable analysis or feature selection of the email corpus data training. The performance of a tree does not depend on the relationships among parameters. A great benefit of decision tree is its capacity to assign unambiguous values to problems, decisions, and results of every decision [135]. This decreases vagueness in decision-making. Another huge advantage of the decision tree compared to other machine learning techniques is the fact that it makes open all the likely options and follows each option to its end in one view, giving room for straightforward evaluation among the different nodes of the tree. Despite the numerous advantages of Decision tree, it still has some drawbacks which are: unless there is appropriate pruning, controlling tree growth can be very difficult. Decision trees are a nonparametric machine learning algorithm that is incredibly adaptable and vulnerable to overfitting of training data [135]. This makes them to some extent poor classifiers and limit their classification accuracy. The different types of Decision trees that have been applied to email spam filtering are NBTree Classifier [80], C4.5/J48 Decision Tree Algorithm [81] and Logistic Model Tree Induction (LMT) [80]. Naïve Bayes is another wonderful machine learning algorithm that has been applied in email spam filtering. A Naive Bayes (NB) classifier simply apply Bayes' theorem on the context classification of each email, with a strong assumption that the words included in the email are independent of each other [38]. NB is desirable for email spam filtering because of its simplicity, ease of implementation and quick convergence compared to conditional models such as logistic regression [136]. It needs fewer training data. It is very scalable. No bottleneck is created by increase in the number of predictors and discrete unit of information [136]. NB can be used to solve both classification problems involving two or more classes. It can be used to make forecasting that is subject to or involving probability variation. They can effectively manage continuous and discrete data. NB algorithms are not susceptible to irrelevant features. Naıve Bayes algorithm is predominantly famous in business-related and open-source spam filters [51]. This is because apart from the advantages listed above, NB needs little training time or speedy assessment to detect and filter email spam. NB filters need training that can be offered by the earlier set of non-spam and spam messages [136]. It keeps the record of the changes that take place in each word that occurs in legitimate, illegitimate messages, and in both. NB can be applied to spam messages in diverse datasets having different features and attribute [136].

Stochastic optimization techniques such as evolutionary algorithms (EAs) have also been applied to spam filtering. This is because they do not have any sophisticated mathematical computation. Also, they can handle the solutions generated, they seek to recognise individuals that have the optimal solutions for the problem [11]. Several earlier works exist that integrated Genetic Algorithms with Neural Networks [12] to enhance the performance of neural network algorithms. A related approach of evolutionary computation methods such as Genetic Algorithms (GAs) is Particle Swarm Optimization (PSO), which is a technique that can be used for optimizing many continuous nonlinear functions and classification techniques. PSO is inspired by the social behaviour of animals such as flocks of bird and shoal of fishes. It has been applied in many areas of human endeavour such as neural network, swarm robotics, telecommunications, signal processing, data mining, and several other applications [129]. PSO algorithm operates on a population (swarm) of particles, with the characteristic of no crossover and mutation calculation as found in genetic algorithm. Every particle have a position and velocity. Each of the particle is a potential solution in the swarm. This makes it easy to implement [13]. What appears to be the most efficient spam filtering approach now is the automatic email filtering which have successfully been used for frustrating the malicious intentions of spammers. Some years back, the largest part of the spam email can be efficiently addressed by stopping emails originating from specified addresses or remove messages with specific subject lines. More deceitful and sophisticated techniques such as utilising arbitrary sender addresses and/or inserting haphazard characters to the beginning or the end of the message subject line are now been used by spammers to surmount the hurdle posed by the filtering methods [9]. Owing to the fact that a good number of real-world filters make use of the amalgamation of ML and application-specific knowledge in the form of hand-coded rules, comprehending the revolutionising attributes of spam is also germane, and many studies have been done on this subject [14,15]. However, in spite of the increasing research efforts on spam filtering, the growth of spam emails is still on alarming rate. This is evident with spammers devising more sophisticated methods for dodging detection, a very good example are emails with stego images (i.e. images with information hidden inside).

The two common approaches used for filtering spam mails are knowledge engineering and machine learning. Emails are classified as either spam or ham using a set of rules in knowledge engineering. The person using the filter, or the software company that stipulates a specific rule-based spam-filtering tool must create a set of rules. Using this method does not guarantee efficient result since there is need to continually update the rules. This can lead to time wastage and it is not suitable especially for naive users. Machine learning approach have proved to be more efficient than knowledge engineering approach. No rule is required to be specified, rather a set of training samples which are pre-classified email messages are provided. A particular machine learning algorithm is then used to learn the classification rules from these email messages [16]. Several studies have been carried out on machine learning techniques and many of these algorithms are being applied in the field of email spam filtering. Examples of such algorithms include

Deep Learning, Naïve Bayes, Support Vector Machines, Neural Networks, K-Nearest Neighbour, Rough sets, and Random Forests. The contributions of this work are given as follows:

a. We did a comprehensive evolutionary survey of the most important features of email spam, the evolution and developments. Through this, we highlighted some interesting research gaps and research directions.
b. We discussed the architectures of spam filters and the application of ML techniques to spam filtering process of Gmail, Yahoo mail and Outlook mail. The different components of the email spam filter were vividly discussed.
c. We presented an elaborate study of several techniques applied to email spam filtering and presented a phenomenal review of literatures on spam email filtering over the period (2004–2018).
d. We exposed researchers to some powerful machine learning algorithms that are not yet explored in spam filtering.
e. We stated in clear terms our findings on some open research problems in relation to spam filtering and recommended proactive steps for the development of machine learning techniques to curb future evolving of new variants of spam that might find it easy to evade filters.

The rest of this paper is organized as follows: Section 2 gives a succinct account of previous reviews, Section 3 is the background discussion, Section 4 describes the performance measures for evaluating the effectiveness of spam filters, Section 5 explains the machine learning algorithms that have found application in spam filtering, Section 6 is the comparative studies of existing machine learning techniques used in spam filtering, Section 7 unveils open research problems in machine learning for spam filtering and future direction before concluding in Section 8.

To increase the readability of the manuscript and also enhance the understanding of the readers, the structure of this paper is depicted in Fig. 2 below:

2. Related work

There is a rapid increase in the interest being shown by the global research community on email spam filtering. In this section, we present similar reviews that have been presented in the literature in this domain. This method is followed so as to articulate the issues that are yet to be addressed and to highlight the differences with our current review. Lueg [17] presented a brief survey to explore the gaps in whether information filtering and information retrieval technology can be applied to postulate Email spam detection in a logical, theoretically grounded manner, in order to facilitate the introduction of spam filtering technique that could

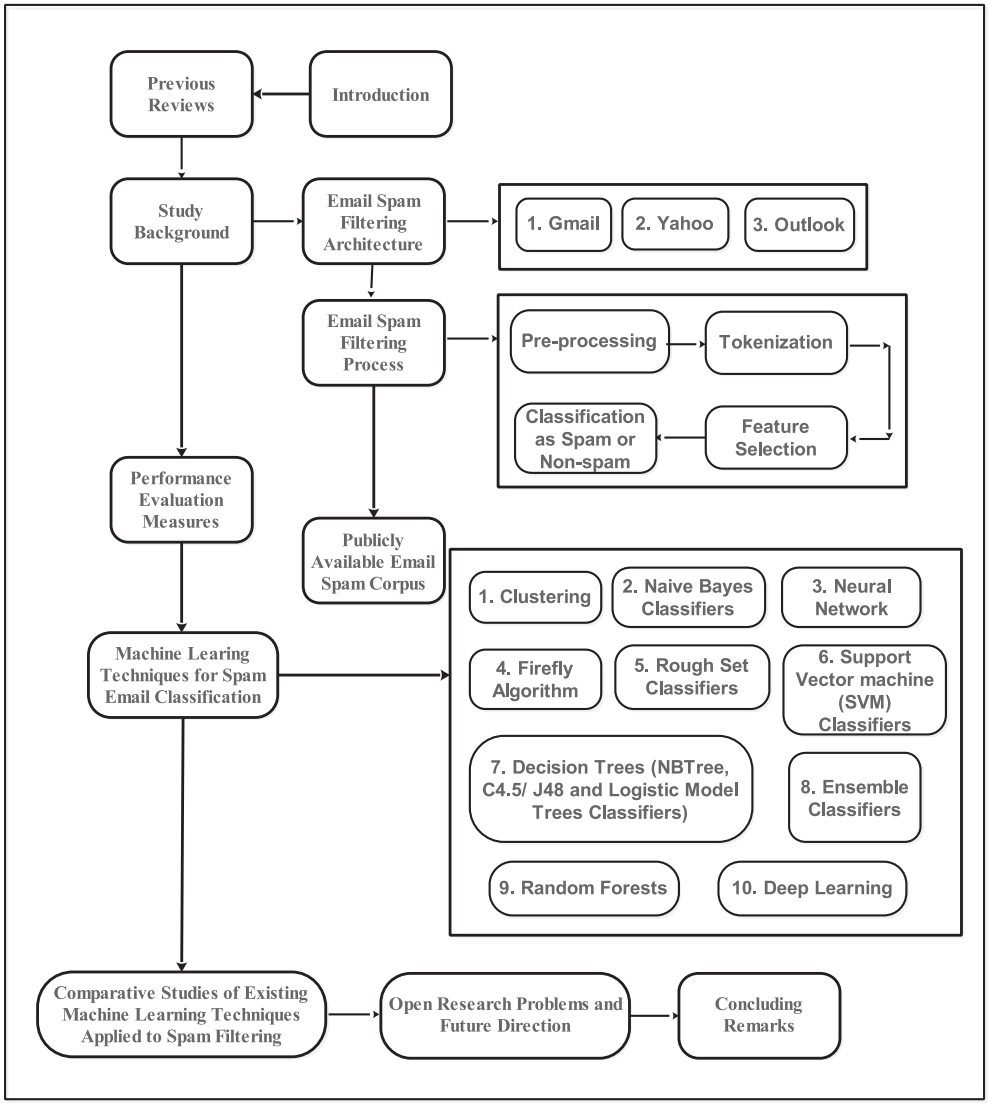

Fig. 2. Pictorial Representation of the Structure of this paper.

be operational in an efficient way. However, the survey did not present the details of the Machine learning algorithms, the simulation tools, the publically available datasets and the architecture of the email spam environment. It also fails short of presenting the parameters used by previous researches in evaluating other proposed techniques. Wang [18] reviewed the different techniques used to filter out unsolicited spam emails. The paper also to categorized email spams into different hierarchical folders, and automatically regulate the tasks needed to response to an email message. However, some of the limitations of the review article are that; machine learning techniques, email spam architecture, comparative analysis of previous algorithms and the simulation environment were all not covered.

The paper titled "Spam filtering and email-mediated applications" chronicles the details of email spam filtering system. It then presented a framework for a new technique for linking multiple filters with an innovative filtering model using ensemble learning algorithm. The article also explained the notion of operable email (OE) in an email-mediated application. Furthermore, a demonstration was made of OE in executing an email assistant and other intelligent applications on the world social email network [19]. However, the survey paper did not cover recent articles as it was published more than a decade ago. Cormack [20] reviewed previously proposed spam filtering algorithms up to 2008 with specific emphasis on efficiency of the proposed systems. The main focus of the review is to explore the relationships between email spam filtering with other spam filtering systems in communication and storage media. The paper also scrutinized the characterization of email spams, including the user's information requirements and the function of the spam sieve as a constituent of a huge and complex information system. However, certain important components of spam filters were not considered in the survey. These includes; the architecture of the system, the simulation environment and the comparative analysis of the performance of the reviewed filters.

Sanz, Hidalgo, and Pérez [21] detailed the research issues related to email spams, in what way it affects users, and by what means users and providers can reduce it effects. The paper also enumerates the legal, economic, and technical measures used to mediate the email spams. They pointed out that based on technical measures, content analysis filters have been extensively used and proved to have reasonable percentage of accuracy and precision as a result, the review focused more on them, detailing how they work. The research work explained the organization and the procedure of many machine learning approaches utilized for the purpose of filtering email spams. However, the review did not cover recent research articles in this area as it was published in 2008 and comparative analysis of the different content filters was also missing. A brief study on E-mail image spam filtering methods was presented by [22]. The study concentrated on email antispam filtering approaches used to transfer from text-based techniques to image-based methods. Spam and the spam filters premeditated to reducing it have spawned an upsurge in creativeness and inventions. However, the study did not cover machine learning techniques, simulation tools, dataset corpus and the architecture of email spam filtering techniques.

Bhowmick and Hazarika [23] presented a broad review of some of the popular content-based e-mail spam filtering methods. The paper focused mostly on machine learning algorithms for spam filtering. They surveyed the important concepts, efforts, effectiveness, and the trend in spam filtering. They discussed the fundamentals of e-mail spam filtering, the changing nature of spam, the tricks of spammers to evade spam filters of e-mail service providers (ESPs), and also examined the popular machine learning techniques used in combating the menace of spam. Laorden *et al.* [24] presented a detailed revision of the usefulness of anomaly discovery used for Email spam filtering that decreases the requirement of classifying email spam messages and only works with the representation of single class of emails. The review contains a demonstration of the first anomaly based spam sieving method, an improvement of the method, which used a data minimization technique to the characterized dataset corpus to decrease processing phase while retaining recognition rates and an investigation of the appropriateness of selecting non-spam emails or spam as a demonstration of normality.

This current review differed from the previous reviews presented in the preceding paragraph by focusing more on revisiting machine learning techniques used for email spam filtering. The review intends to cover the architecture of the email spam filtering systems, parameters used for comparative analysis, simulation tools and the dataset corpus. The period under review also include all recent research articles that are found to be useful for the advancement of the email spam filtering methods s shown in Table 1 below.

3. Background

Here we discussed the architecture of email server and the stages in processing email. We explained the different stages involved in preprocessing and feature selection.

3.1. Email spam filtering architecture

Spam filtering is aimed at reducing to the barest minimum the volume of unsolicited emails. Email filtering is the processing of emails to rearrange it in accordance to some definite standards. Mail filters are generally used to manage incoming mails, filter spam emails, detect and eliminate mails that contain any malicious codes such as virus, trojan or malware. The workings of email is influence by some basic protocols which include the SMTP. Some of the widely used Mail User Agents (MUAs) are Mutt, Elm, Eudora, Microsoft Outlook, Pine, Mozilla Thunderbird, IBM notes, Kmail, and Balsa. They are email clients that assists the user to read and compose emails. Spam filters can be deployed at strategic places in both clients and servers.

Spam filters are deployed by many Internet Service Providers (ISPs) at every layer of the network, in front of email server or at mail relay where there is the presence of firewall [25]. The firewall is a network security system that monitors and manages the incoming and outgoing network traffic based on predetermined security rules. The email server serves as an incorporated anti-spam and anti-virus solution providing a comprehensive safety measure for email at the network perimeter [26]. Filters can be implemented in clients, where they can be mounted as add-ons in computers to serve as intermediary between some endpoint devices [27]. Filters block unsolicited or suspicious emails that are a threat to the security of network from getting to the computer system. Also, at the email level, the user can have a customized spam filter that will block spam emails in accordance with some set conditions [28].

3.1.1. How Gmail, Yahoo and Outlook emails spam filters work

Different spam filtering formulas have been employed by Gmail, Outlook.com and Yahoo Mail to deliver only the valid emails to their users and filter out the illegitimate messages. Conversely, these filters also sometimes erroneously block authentic messages. It has been reported that about 20 percent of authorization based emails usually fail to get to the inbox of the expected recipient. The email providers have designed various mechanisms for use in email anti-spam filter to curtail the dangers posed by phishing, email-borne malware and ransomware to email users. The mechanisms are used to decide the risk level of each incoming email. Examples of such mechanisms include satisfactory spam limits, sender policy frameworks, whitelists and blacklists, and recipient verification tools. These mechanisms can be used by single or multiple users. When the satisfactory spam thresholds is too low it can lead to more spam evading the spam filter and entering the users' inboxes. Meanwhile having a very high threshold can lead to some important emails being isolated unless the administrator redirects them. This section discusses the operations of Gmail, Yahoo and Outlook emails antispam filters.

3.1.1.1. Gmail filter spam. Google's data centers makes use of hundreds

Table 1
Summary of previous reviews in email spam filtering.

Previous Reviews	Email Spam	Machine Learning	Comparative Analysis	Simulation Tool & Environment	Dataset Corpus	Architecture	Parameters	Period Covered
Lueg [17]	√							2000–2005
Wang [18]	√				√		√	1995–2005
Li et al. [19]	√	√	√	√	√		√	1997–2006
Cormack [20]	√	√			√		√	2000–2008
Sanz et al. [21]	√	√		√	√		√	2000–2008
Dhanaraj and Karthikeyani [22]	√						√	1994–2013
Bhowmick and Hazarika [23]	√	√	√		√	√	√	2004–2013
Laorden *et al.* [24]	√				√		√	2002–2014
Our Review	√	√	√	√	√	√	√	2000–2018

of rules to determine whether an email is valid or spam. Every one of these rules depicts specific features of a spam and certain statistical value is connected with it, depending on the likelihood that the feature is a spam. The weighted importance of each feature is then used to construct an equation. A test is conducted using the score against a sensitivity threshold decided by each user's spam filter. And consequently, it is classified as a lawful or spam email. Google is said to be using state of the art spam detection machine learning algorithms such as logistic regression and neural networks in its classification of emails. Gmail also use optical character recognition (OCR) to shield Gmail users from image spam. Also, machine-learning algorithms developed to combine and rank large sets of Google search results allow Gmail to link hundreds of factors to improve their spam classification. The evolving nature of spam over time revolves around factors such as domain reputation, links in message headers and others. These can make messages to unexpectedly end up in the spam folder. Spam filtering principally works on the foundation of "filters" settings that are continuously updated with the emergence of state of the art tools, algorithms, discovery of new spam and the feedback from Gmail users about likely spammers. Many spam filters employ text filters to eradicate hazards posed by spammers depending on the senders and their history.

3.1.1.2. Yahoo mail filter spam. Yahoo mail is the first free webmail providers in the world with over 320 million users. The email provider has its own spam algorithms that it uses to detect spam messages. The basic methods used by Yahoo to detect spam messages include: URL filtering, email content and spam complaints from users. Unlike Gmail, Yahoo filter emails messages by domains and not IP address. Yahoo mail uses combination of techniques to filter out spam messages. It also provide mechanisms that prevent a valid user from being mistaken for a spammer. Examples are ability of the users to troubleshoot SMTP Errors by referring to their SMTP logs. Another one is the complaint feedback loop service that helps a user to maintain a positive reputation with Yahoo. Yahoo whitelisting (internal whitelisting and Return Path Certification) is also provided. Unlike blacklisting, a whitelist blocks by letting the user specify the list of senders to receive mail from. The addresses of such senders are placed on a trusted-users list. Yahoo mail spam filters allows the user to use a combination of whitelist and other spam-fighting feature as a way to reduce the number of valid messages that are erroneously classified as spam. On the other hand, using whitelist alone will make the filter to be very strict and the implication is that any unapproved user would be blocked automatically. Many anti-spam systems use automatic whitelist. In this case, an anonymous sender's email address is checked against a database; if there is no history of spamming, their message is sent to the recipient's inbox and they are added to the whitelist.

3.1.1.3. Outlook email spam filter. After Gmail and Yahoo mail, we discussed Outlook from Microsoft in this section and how it handles spam filtering. In 2013, Microsoft changed the name of Hotmail and Windows Live Mail to Outlook.com. Outlook.com was patterned after Microsoft's Metro design language and directly imitates the interface of Microsoft Outlook. Outlook.com is a collection of applications from Microsoft, one of which is Outlook webmail service. Outlook webmail service allows the users to send and receive emails in their web browser. It allows the users to connect cloud storage services to their account so that when they want to send an email with file attachments, they can select files from not only their computer and OneDrive account but also from Google Drive, Box, and Dropbox account. Moreover, Outlook webmail service also allows users to encrypt their email messages and disallow the recipient from forwarding the email. Whenever a message is encrypted in Outlook.com, it is only the person with the password that will be able to decrypt the message and read it. This is a security measure that guarantees that only the intended recipient is permitted to read the message. The main difference between Outlook.com webmail service and the MS Outlook desktop application is that Outlook desktop application allows you to send and receive emails, via an email server, while Outlook.com is an email server. Outlook.com webmail service on-the-other-hand is for business and professionals who rely on email. Moreover, MS Outlook desktop application is a commercial software that comes along with the Microsoft Office package. It is a computer software program that provides services like email management, address book, notebook, a web browser and a calendar which allows users to plan their programmes and arrange upcoming meetings. About 400 million users are using Outlook.com. Statistics shows that their site receives about eight billion emails a day and out of which 30%–35% of those emails are delivered to the users' inboxes. Outlook.com have its own distinctive methods of filtering email spams.

3.2. Email spam filtering process

An email message is made up of two major components which are the header and the body. The header is the area that have broad information about the content of the email. It includes the subject, sender and receiver. The body is the heart of the email. It can include information that does not have a pre-defined data. Examples include web page, audio, video, analog data, images, files, and HTML markup. The email header is comprised of fields such as sender's address, the recipient's address, or timestamp which indicate when the message was sent by intermediary servers to the Message Transport Agents (MTAs) that function as an office for organising mails. The header line usually starts with a "From" and it goes through some modification whenever it moves from one server to another through an in-between server. Headers allow the user to view the route the email passes through, and the time taken by each server to treat the mail. The available information have to pass through some processing before the classifier can make use of it for filtering [29]. Fig. 3 below depicts a mail server architecture and how spam filtering is done.

The necessary stages that must be observed in the mining of data from an email message can be categorised into the following:

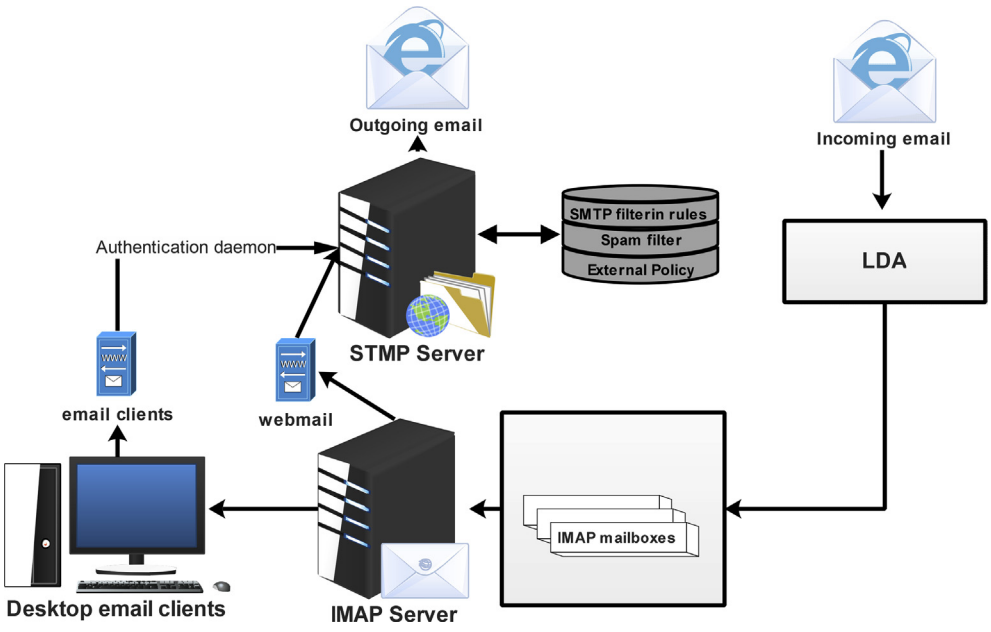

Fig. 3. Email server spam filtering architecture.

- **Pre-processing:** This is the first stage that is executed whenever an incoming mail is received. This step consists of tokenization.
- **Tokenization:** This is a process that removes the words in the body of an email. It also transforms a message to its meaningful parts. It takes the email and divides it into a sequence of representative symbols called tokens. Subramaniam, Jalab and Taqa [30] emphasised that these representative symbols are extracted from the body of the email, the header and subject. Guzella and Caminhas [16] asserted that the process of replacing information with distinctive identification symbols will extricate all the characteristics and words from the email exclusive of taking into account the meaning
- **Feature selection:** Sequel to the pre-processing stage is the feature selection phase. Feature selection a kind of reduction in the measure of spatial coverage that effectively exemplifies fascinating fragments of email message as a compressed feature vector. The technique is beneficial when the size of the message is large and a condensed feature representation is needed to make the task of text or image matching snappy [31]. Advance fee fraud, including inheritance, lottery, visa and customs-clearance scams, Romance scams, including marketing sex enhancement drugs to cure erection dysfunctional, online dating, military scams, Ads for porn sites, Ads for miscellaneous external sites, earning big money through "work-from-home" jobs, online shopping, pleading and gift requests, business proposals and others. Some of the most important features for spam filtering include: Message body and subject, Volume of the message, Occurrence count of words, Circadian patterns of the message (spam messages usually have many semantic discrepancies), Recipient age, Sex and country, Recipient replied (indicates whether the recipient replied to the message), Adult content and Bag of words from the message content. Sender Account Features used for spam filtering include: Sender Country (The distribution of countries as stated by users on their profile and as revealed by their IP address), Sender IP address, Sender Email, Sender & Recipient Age, Sender Reputation. The less important features are: Geographical distance between sender and receiver, Sender's date of birth, Username and password of the sender, Account lifespan, Sex of sender and Age of recipient. The recognition of spam e-mails with minimum number of features is important in view of computational complexity and time. Feature selection involves processes like stemming, noise removal and stop word removal steps.

3.3. Publicly available email spam corpus

The dataset contained in a corpus plays a crucial role in assessing the performance of any spam filter. Though there are many conventional datasets that are usually used for classifying text, it is just of recent that some researchers in the field of spam filtering are making the corpus used for evaluating the effectiveness of their proposed filter available to the public. A comprehensive list of the corpora made available to the public in the different techniques reviewed in this paper are in Table 2. Individual corpus possesses incredibly distinctive qualities which are indicated by the related information applied in the experiments conducted to evaluate the performance of the spam filter.

4. Analysis

4.1. Performance evaluation measures

Spam filters are usually evaluated on large databases containing ham and spam messages that are publicly available to users. An example of the performance measures that are used is classification accuracy (Acc). It is the comparative number of messages rightly classified, the percentage of messages rightly classified is used as an added measure for evaluating performance of the filter. It has however been highlighted that using Accuracy as the only performance indices is not sufficient. Other performance metrics such as recall, precision and derived measures used in the field of information retrieval must be considered, so also is false positives and false negatives used in decision theory. This is very important because of the costs attached to misclassification. When a spam message is wrongly classified as ham, it gives rise to a somewhat insignificant problem, because the only thing the user need to do is to delete such message. In contrast, when a non-spam message is wrongly labeled as Spam, this is obnoxious, because it indicates the possibility of losing valuable information as a result of the filter's classification error. This is very imperative especially in settings where Spam messages are automatically deleted. Therefore, it is inadequate to evaluate the performance of any Machine Learning algorithm used in spam filter using classification accuracy exclusively. Furthermore, in a setting that is lopsided or biased where the number of spam messages utilized for testing the performance of the filter is very much higher than that of ham messages, the classifier can record a very high accuracy by concentrating

Table 2
Publicly available email spam corpus.

Dataset name	Number of messages		Rate of spam	Year of creation	References
	Spam	Non-spam			
Spam archive	15090	0	100%	1998	Almeida and yamakami [32]
Spambase	1813	2788	39%	1999	Sakkis et al [33]
Lingspam	481	2412	17%	2000	Sakkis et al [33]
PU1	481	618	44%	2000	Attar et al [34]
Spamassassin	1897	4150	31%	2002	Apache spamassassin [35]
PU2	142	579	20%	2003	Zhang et al [36]
PU3	1826	2313	44%	2003	Zhang et al [36]
PUA	571	571	50%	2003	Zhang et al [36]
Zh1	1205	428	74%	2004	Zhang et al [36]
Gen spam	31,196	9212	78%	2005	Cormack and lynam [37]
Trec 2005	52,790	39,399	57%	2005	Androutsopoulos et al [38]
Biggio	8549	0	100	2005	Biggio et al [39]
Phishing corpus	415	0	100	2005	Abu-nimeh et al [40]
Enron-spam	20170	16545	55%	2006	Koprinska et al [41]
Trec 2006	24,912	12,910	66%	2006	Androutsopoulos et al [42]
Trec 2007	50,199	25,220	67%	2007	Debarr and wechsler [43]
Princeton spam image Benchmark	1071	0	100%	2007	Wang et al [44]
Dredze image spam Dataset	3297	2021	62%	2007	Dredze, gevaryahu and elias-bachrach [45]
Hunter	928	810	53%	2008	Gao et al [46]
Spamemail	1378	2949	32%	2010	Csminninggroup [47]

on the detection of spam messages solely. In a real world environment where there is nothing like zero probability of wrongly categorizing a ham message, it is required that a compromise be reached between the two kinds of errors, depending on the predisposition of user and the performance indicators used. The formulae for calculating the classification accuracy and classification error are depicted in Eqs. (1) and (2) below:

Assuming

NH = Number of non-spam messages to be classified
NS = Number of spam messages to be classified

$$\text{Classification Accuracy } (Acc) = \frac{|H \to H| + |S \to S|}{N_H + N_S} \tag{1}$$

$$\text{Classification Error } (Err) = 1 - Acc = \frac{|H \to S| + |S \to H|}{N_H + N_S} \tag{2}$$

According to [23], classification accuracy and error mutually take into account False Positive |H→S| and False Negative |S→H| occurrences to bear equal cost. It is necessary to point it out that disproportionate error costs is involved in spam filtering. Wrongly classifying a ham message as spam (also known as false positive event) is an expensive mistake compared to the spam message just evading the filter. Such incident is referred to as false negative event. When a legitimate e-mail is rightly classified as ham, it is called a true positive event |H→H|. However, when a spam e-mail is rightly classified as spam, then a true negative event |S→S| has occurred. Based on the above explanations, the false positive rate (FPR) can be defined as the ratio of ham or valid e-mails that are classified as spam. It is denoted using the formula in Eq. (3) below:

$$\text{FPR} = \frac{\text{No Of False Postives}}{\text{No Of False Positives} + \text{No Of True Negatives}} \tag{3}$$

Also, allowing spam emails that have been infected with malwares, spywares, adware, Trojan, botnet, viruses, worms, or phishing baits such as messages claiming to be from social web sites, dating sites, auction sites, banks, online payment processors or IT administrators are usually used to entrap victims. This can cause the users monumental losses. The ratio of spam messages that were wrongly classified as ham is called false negative rate (FNR). This is one more apt metric for evaluating the performance of a filter. The formula for computing the FNR is in Eq. (4) below:

$$\text{FNR} = \frac{\text{No Of False Negatives}}{\text{No Of True Positives} + \text{No Of False Negatives}} \tag{4}$$

Spam filters with a drastically reduced FPR and FNR are said to have a better performance. These standard characteristics (FNR and FPR) represents the efficiency of filters that directly aim at the classification decision borderline devoid of generating the probability estimate. On the other hand, the efficiency of filters that explicitly estimate the group conditional probabilities and then execute classification based on estimated probabilities can be represented by a curve called ROC (Receiver Operating Characteristics) curve. ROC curve, is a graphical plot that demonstrates the analytical capability of a spam filter as its bias level is modified [48]. The ROC curve is generated by plotting the true positive rate (TPR) against the false positive rate (FPR) at different threshold settings [49]. The true positive rate is referred to as sensitivity, recall or probability of detection [49] in machine learning. The false-positive rate is referred to as the squabble or likelihood of false alarm. This is computed by subtracting the value of the specificity from 1 (i.e. 1 - specificity). ROC testing are an outstanding standard of performance measure in spam filtering [48]. When the ROC curve of a spam filter closely sits on top of another, such filter can be classified a filter with superior performance in all implementation setups [20]. The two metrics imported from the field of information retrieval 'recall' and 'precision' are respectively utilised for obtaining the efficiency and characteristic of spam filters [50].

Given that

|S → H| = The number of spam messages classified as non-spam
|H → S| = The number of non-spam messages classified as spam respectively, and

where |H→ H| and |H→ S|

Eq. (5) below represents spam recall (Rs) and spam precision (Ps):

$$R_s = \frac{|S \to S|}{|S \to S| + |S \to H|} \text{ and } P_s = \frac{|S \to S|}{|S \to S| + |H \to S|} \tag{5}$$

Recall (Rs) also known as effectiveness can be defined as the comparative number of spam messages that the filter succeeded in preventing from entering email inbox. Precision (Ps) also described as the worth or reliability of the filter is calculated by dividing the number of messages categorised by the filter as spam but are truly ham by the total number of email messages [51,33]. Evaluating the performances of different spam filters using (Rs) and (Ps) is delicate considering the

different values that were involved in the computations that produced (Rs) and (Ps). According to [50,51], the cost of false positives is much more (λ times) than that of false negatives, where lambda (λ) is a numerical factor that stipulates how 'risky or 'harmful it is to wrongly classify a valid e-mail as spam. It also indicates how difficult it can be for the user to recuperate from such abysmal and unacceptable performance of the spam filter. Cost sensitivity should be considered as suggested [51]. This can be done by making every valid message as being equal to α messages. Clark [52] in his paper gave the formula for computing cost sensitive measures such as Weighted Accuracy (WAcc), Weighted Error Rate (WErr) and Total Cost Ratio (TCR).

Total Cost Ratio is used for measuring the accuracy of filters, it was proposed by [50]. Higher TCR implies better performance. When the value of TCR <1, it is better not to use the filter. In a situation where the cost is proportionate to time squandered, TCR measures the amount of time squandered by the user to delete all spam messages by himself despite the fact that spam filter is installed. It then compares it to the time spent to manually remove the spam emails that evade the filter in addition to the time required to recoup from valid messages that were erroneously blocked. The two main strengths of TCR is that it is a single-figure measurement, while majority of the other cost sensitive measures require a minimum of two figures. This nonetheless can give the wrong impression about the effectiveness of a filter as a better TCR might denote a greatly reduced false positive rate or a very high hit rate with a relatively high FPR. Likewise, TCR appears to be susceptible to the stabilising of the corpus. The stability of the corpus is a situation whereby the volume of spam and nonspam messages in the corpus are at variance. Portability of the values is one of the drawbacks of the TCR. Also, comparison can only be drawn among TCR values when all evaluated TCRs were calculated by making use of similar λ. The formulae for computing the Weighted Accuracy (WAcc), Weighted Error Rate (WErr) and Total Cost Ratio (TCR) are represented in Eqs. (6), (7), and (8) below:

$$W_{Acc} = \frac{\lambda|H \to H| + |S \to S|}{N_H + N_S} \text{ and } W_{Acc} = 1 - W_{Acc} \tag{6}$$

$$W_{Acc} = \frac{\lambda|H \to S| + |S \to H|}{N_H + N_S} \tag{7}$$

$$TCR = \frac{N_S}{\lambda|H \to S| + |S \to H|} \tag{8}$$

In computing cost sensitivity of filters, λ decides the strictness of penalty for wrongly classifying a non-spam email as spam. Integrated into the threshold is the cost sensitivity with the formula $\lambda/(1+\lambda)$. The model is reconstructed and assessed on diverse kinds of strictness level of λ. Table 3 below explains different strata of cost sensitivity of model that have been taken into consideration:

From the above Table 3, the efficiency of a filter for a given λ is compared with a baseline case by means of total cost ratio as explained in [16]. This a further prove of the enhancement derived from the use of filter. The λ is used for fine-tuning the weight of false positive and its performance is assessed by the cost sensitivity. The three values used for λ (999, 9, and 1) exemplify the conditions whereby a false positive is 999 times more expensive, or a false positive is 9 times an expensive error more than a false negative, or false positive and false negative are equal. Another metric for measuring the performance of a filter is the F-measure (F1-score or F-score). It is a measure of the accurateness of a test and is described as the weighted harmonic mean of the precision (Ps) and recall (Rs) of the test in a single equation. F-measure make use of a parameter that enables a compromise to be reached concerning recall and precision. F1 is the traditional F-measure that is commonly used and it presents uniform weight to recall and precision as shown in Eqs. (9) and (10).

Table 3
Levels of cost sensitivity of model.

λ	Maximum Tolerance Level $T = \lambda/(1+\lambda)$	Significance of having such cost sensitivity?
999	0.999	Filtered messages are thrown away and no additional processing is carried out.
99	0.9	Filtering a non-spam message is slightly penalized above allowing a spam message to go through. It is used to demonstrate that it is cumbersome re-sending a filtered spam message than deleting it manually.
1	0.5	If the receiver is not concerned as regards missing a non-spam message.

$$F_1 = \frac{2*\text{recall}*\text{precision}}{\text{Precision} + \text{Recall}} \tag{9}$$

$$F_{Beta} = \frac{(1 + Beta_2)*\text{recall}*\text{precision}}{\text{Recall} + Beta_2*\text{Precision} + \text{Recall}} \tag{10}$$

In a situation where we have $0 < Beta < 1$, it gives more importance to the precision while when we have $Beta > 1$, it gives more importance to the recall. It is glaring that F-measure (F1-score) is an exclusive case of weighted F-measure when $Beta = 1$.

5. Materials & methods

Of recent, spam mail classification is normally handled by machine learning (ML) algorithms intended to differentiate between spam and non-spam messages. Machine learning algorithms achieve this by using an automatic and adaptive technique. Rather than depending on hand-coded rules that are susceptible to the perpetually varying characteristics of spam messages, ML methods have the capacity to obtain information from a set of messages provided, and then use the acquired information to classify new messages that it just received. According to [53], ML algorithms have the capacity to perform better based on their experience. In this section we will review some of the most popular machine learning methods that have been applied to spam detection.

5.1. Clustering technique

Clustering deals with classifying a group of patterns into related classes. Clustering is a type of approach used in dividing objects or case examinations into comparatively similar collections known as clusters. Clustering techniques have drawn the attention of many researchers and academicians of recent and it has been applied in different fields of application. Clustering algorithms which are unsupervised learning tools are used on e-mail spam datasets which usually have true labels. Provided there are appropriate representations, a good number of clustering algorithms have the ability to classify e-mail spam datasets into either ham or spam clusters. Whissell and Clarke [54] proved this in their research work on e-mail spam clustering. The outputs were remarkably noteworthy as their technique performed better than those of existing modern semi-supervised techniques, thereby demonstrating that clustering can be a formidable technique for filtering spam e-mails. It classifies objects or opinions in such a manner that objects in the same group are more similar to each other than to those in other group. Two types of clustering methods that have been used for spam classification are density-based clustering and K-nearest neighbours (kNN). In [55], density based clustering is another method of document clustering which has been exploited to solve spam classification problem. As asserted by the authors, the method have the capacity to process encrypted messages, thereby upholding its confidentiality. The success of the technique is subject to its ability to locate sensitive comparators. The comparators are normally characterised with either speed or sensitivity. Locating the comparators that have speed and are sufficiently sensitive is the major barrier to the success of this technique.

kNN is a distribution free method, which does not rely on assumptions that the data is drawn from a given probability distribution [56].

This is rather important because in the real world, nearly all of the applied data disobey the standard hypothetical postulations made (such as Gaussian mixture, linearly separable, and others). Non-parametric algorithms like kNN can be used to salvage such situation. In kNN classifier, the classification model is not built from data, rather classification is carried out by matching the test instance with K training examples and decision is made as to which group it belong to depending on the resemblance to K closest neigbours [57]. The kNN is termed a lazy learner since the training data points is not used by it to perform generalisation. Simply put, there is no obvious training stage and if it exists it is extremely small. The implication is that the algorithm has a moderately speedy training phase. The absence of universality necessitates kNN to store all the training data. To be precise, the entire training data is required throughout the testing phase as decisions are made based on the complete training data set. The contradiction is quite clear here that there is no significant training stage, rather there is an extensive testing stage. There is an overhead cost of both time and memory. Additional time may possibly be required in the most awful case. Added memory is required to store all training data neigbours [57]. The authors in [58] opined that some of the strengths of kNN algorithm includes: there is no explicit training phase or it is very minimal. Once data is loaded into memory, it begins its classification process.

In [58], the steps involved in a simple kNN algorithm for filtering spam mails is described in the algorithm below. Here Neighbours(d) return the k nearest neighbours of d, Closest (d, t) return the closest elements of t in d, and testClass(S) return the class label of S. A simple kNN algorithm for spam email classification is in the algorithm below:

Algorithm 1 kNN Algorithm for Spam Email Classification

1: Find Email Message class labels.
2: **Input** k, the number of nearest neighbors
3: **Input** D, the set of test Email Message;
4: **Input** T, the set of training Email Message.
5: L, the label set of test Email Message.
6: Read DataFile (TrainingData)
7: Read DataFile (TestingData)
8: **for** each d in D and each t in T **do**
9: Neighbors(d) = {}
10: **if** |Neighbors (d) | < k **then**
11: Neighbors(d) = Closest (d, t) ∪ Neighbors(d)
12: **end if**
13: **if** |Neighbors(d)| ≥ k **then**
14: *restrain(M, x_j, y_j)*
15: **end if**
16: **end for** 17: **return** Final Email Message Classification (Spam/Valid email)
18: **end**

5.2. Naïve Bayes classifier

The Bayesian classification exemplifies a supervised learning technique and at the same time a statistical technique for classification. It acts as a fundamental probabilistic model and let us seize ambiguity about the model in an ethical way by influencing the probabilities of the results. It is used to provide solution to analytical and predictive problems [123]. Bayesian classification is named after Thomas Bayes (1702–1761), who proposed the algorithm. The classification offers practical learning algorithms and previous knowledge and experimental data can be merged. Bayesian Classification offers a beneficial viewpoint for comprehending and appraising several learning algorithms. It computes exact likelihoods for postulation and it is robust to noise in input data. A Naive Bayes classifier is a straightforward probabilistic classifier that is founded on Bayes theorem with sound assumptions that are independent in nature. A better expression for the probability model should be autonomous characteristic model [59,60] as shown in Eq. (11):

$$\text{Bayes Theorem: Prob}\,(B \text{ given } A) \;=\; \text{Prob}\,(A \text{ and } B)/\text{Prob}\,(A) \tag{11}$$

The notion of class restrictive autonomy was created to make computation easier, and is the basis of tagging the algorithm 'naïve'. Nevertheless, the algorithm is effective and very robust. It performs just like other supervised learning algorithms. There have been an upsurge in the acceptance of NB as a simple and computationally efficient algorithm with satisfactory performances in solving real-world problems. As a result of its exceptional qualities, NB classifiers has found application as classification algorithm in text, spam email, sentiment analysis, recommender systems, spam reviews, and other online applications. Naive Bayes classifiers are particularly utilised in text classification (because it produces superior result in multi class problems and independence rule) and have greater success rate when compared to some other machine learning algorithms. Due to this obvious advantage, it is extensively applied in the field of spam filtering (detect spam e-mail) and sentiment analysis (in social media analysis, to recognise positive and negative customer opinions). Spam filtering is the most famous use of the NB classifier. It is a general method for differentiating unauthorised emails i.e. spam from the lawful ones, often referred to as ham. Most mail clients implement Bayesian spam filtering these days. Whereas users can install email-filtering software, server-side email filters utilising Bayesian spam filtering methods are entrenched inside software that makes e-mail facilities to perform effectively [61]. Virtually all the statistic-based spam filtering techniques are using Naïve Bayes' classifier to group the statistics of each token to a total score [62,126], and the score is used in making resolution on the filtering. According to [63], the token T which denote the spamminess (spam rating) is computed as illustrated in Eq. (12):

$$S[T] = \frac{C_{spam}(T)}{C_{spam}(T) + C_{Ham}(T)} \tag{12}$$

Where:

C_{Spam}(T) = The number of spam messages containing token T,
C_{Ham}(T) = The number of ham messages containing token T,

There will be need to merge the different token's spamminess to calculate the overall message spamminess in order to compute the probability for a message M with tokens {T_1,.....,T_N}. Computing the product of specific token's spamminess and comparing it with the product of specific token's hamminess is a straightforward way to make classifications. It is represented in Eq. (13) below:

$$\left(H[M] = \prod_{i=1}^{N}(1 - S\,[T_i])\right) \tag{13}$$

The message is classified as spam if the total spamminess product S [M] is greater than the hamminess product H [M]. The above description in [63] is used in the Naïve Bayes classification algorithm for email spam classification depicted below:

Algorithm 2 Naïve Bayes Classification Algorithm for Email Spam Classification

1: **Input** Email Message dataset
2: Parse each email into its component tokens
3: Compute probability for each token S [W] = C_{spam}(W)/(C_{ham}(W) + C_{spam}(W))
4: Store spamminess values to a database
5:**for** each message M **do**
6: **while** (M not end) **do**
7: scan message for the next token Ti
8: query the database for spamminess S(Ti)
9: compute probabilities of message collected S [M] and H [M]
10: compute the total message filtering signal by: I [M] = f (S [M], H [M])
11: $I[M] = \frac{I + S[M] - H[M]}{2}$
12: **if** I [M] > threshold **then**
13: msg is labeled as spam
14: **else**
15: msg is labeled as non-spam
16: **end if**
17: **end while**
18: **end for** 19: **return** Final Email Message Classification (Spam/Valid email)
20: **end**

5.3. Neural networks

Artificial Neural Networks are groups of simple processing units which are interconnected, and communicate with one another by means of a sizable number of weighted connections. Each of the units accepts input from the neighbouring units and external sources and calculates the output that is transmitted to other neighbours. The medium for fine-tuning the weights of the connections is also made available. Neural networks are potent algorithm for solving any machine-learning problem that requires classification [64]. Due to their resourcefulness, they are evolving as a major tool in the machine-learning researcher's set of tools. Nevertheless, neural networks are not commonly used in the detection of spam email as one may possibly envisage. As an alternative, nearly all state-of-the-art spam filters use naïve Bayes classifiers. This is due primarily to Paul Graham's well-known work titled "A Plan for Spam." Naïve Bayes is an excellent method for spam classification with high accuracy (99.99+%) and a low false-positive rate. What enhances its high accuracy is the huge number of well-interconnected processing components (neurons) that are working in harmony to provide solution to certain problems. For instance, Google recounted the increase in Gmail spam filters' accuracy from 99.5% to 99.9% after incorporating neural networks into it. This brings to mind that neural networks might be useful for improving the performance of spam filters, particularly when hybridised with Bayesian classification and other techniques. On the other hand, much research need to be done on the application of neural networks for spam detection, and nearly all of the current research takes the network configuration, momentum, and learning rate to be fixed. More research efforts needs to be focused on the efficacy of the network across datasets instead of the appropriateness of diverse network designs for the job [64]. According to [65], there are generally three kinds of units.

- **Input Unit:** This unit accepts signal from outside source.
- **Output Unit:** This unit transmits data outside the network.
- **Hidden Unit:** This unit accepts and transmits signals within the network.

The workings of the system is synchronised so that a great number of the units can function in parallel. ANN can be customized to accept a set of inputs and generate the needed set of outputs. This process is referred to as learning or training. There are two types of training in neural network.

- **Supervised:** Here, the network is given a set of inputs and matching output patterns, known as training dataset, to train the network.
- **Unsupervised:** In this instance, the network trains itself by producing groups of patterns. There is no earlier set of training data given to the system.

There are two conventional types of neural networks that are usually implied whenever ANN is used. They are the perceptron and the multilayer perceptron. This section will attempt to explain the perceptron algorithm and its application to email spam filtering. Below is a perceptron algorithm which is a standard Neural Network algorithm. The perceptron assists in locating a linear function of the attribute vector $f(x) = w^T x + b$ such that $f(x) > 0$ for vectors of one group [1], and $f(x) < 0$ for vectors of other group. Also, w = ($w_1, w_2, \ldots w_m$) are the weights of the function, and b is the supposed bias. The groups can be given the numbers +1 and -1, so search for a function $d(x) = \text{sign}\ (w^T x + b)$ is carried out. The perceptron learning begins by randomly selecting parameters (w_0,b_0) of the resolution and repeatedly bringing them up-to-date. A training sample (x,c) is selected at the nth iteration of the algorithm to the extent that the present decision function now group it as incorrect (i.e. sign $(w_n x + b_n) \neq c$). The rule depicted by Eq. (14) below is used in updating the parameters (w_n, b_n):

$$w_{n+1} = w_n + c_x \quad b_{n+1} = b_n + c \tag{14}$$

The criteria for terminating the algorithm is that a decision function must be located which accurately categorises all the training samples into different groups. The algorithm below is based on this explanation that was just given [66]. There are times when the training data cannot be separated linearly, in such cases the wisest action to take is to terminate the training algorithm once the number of data that are erroneously classified is sufficiently small [67]. The algorithm below represent the algorithm for a Perceptron Neural Network for email spam classification:

Algorithm 3 Perceptron Neural Network algorithm for Email Spam Classification

1: **Input** Sample email message dataset
2: Initialize w and b (to random values or to 0).
3: Find a training sample of messages (x,c) for which sign ($w^T x + b$).
4: **if** there is no such sample, **then**
5: Training is completed
6: Store the final w and stop.
7: **else**
8: update (w,b): $w = w + cx$,
9: $b = b + c$
10: **go to** step 8
11: **end if**
12: Determine email message class as sign ($w^T x + b$)
13: **return** Final Email Message Classification (Spam/Non-spam email)
14:**end**

The architecture of the Neural Network email spam classifier is depicted in Fig. 4 below.

5.4. Firefly algorithm

The firefly algorithm (FA) is a population based metaheuristic algorithm proposed by [68]. He got his inspiration from the sparkly behaviour of fireflies. The algorithm preserves and increase several candidate solutions by means of population physiognomies to direct the search [69]. The design of the algorithm was founded on the study of the concept of communication among fireflies at the time they are getting ready to copulate, and immediately they are exposed to danger. Fireflies share information among themselves by means of their sparkling attribute [70]. With about 2000 firefly species in the world, each one uses a dissimilar sparkling format. The fireflies normally generate a little spark with a particular format subject to what they are involved in. The light is generated by the biochemical production of light by living creatures. Depending on the form of the light, the right companion will communicate in return by either imitating the same form or answering back by using a precise form. Conversely, the intensity of light declines owing to distance. Therefore, a sparkling light exuding from a firefly gets a response from fireflies around it within a visual range of the flash. As [70, 71] noted that the properties of attraction and movement of fireflies could inspire an optimisation algorithm in which solutions follow better (brighter) solutions. The firefly algorithm for email spam classification is as shown below:

Algorithm 4 Firefly Algorithm for email spam classification

1: **Input** Email corpus with M number of features
2: Set k = 0
3: Get population of firefly N
4: Get the number of attributes M
5: Initialize the firefly population
6: **for** each firefly
7: Choose the firefly which has best fitness
8: Choose corresponding features from the testing part of the email spam corpus
9: Test the email message
10: k = k+1
11: Update each firefly
12: Classify the email message as either spam or Non-spam email
13: **end for**
14: **return** Final Email Message Classification (Spam/Non-spam email)
15:**end**

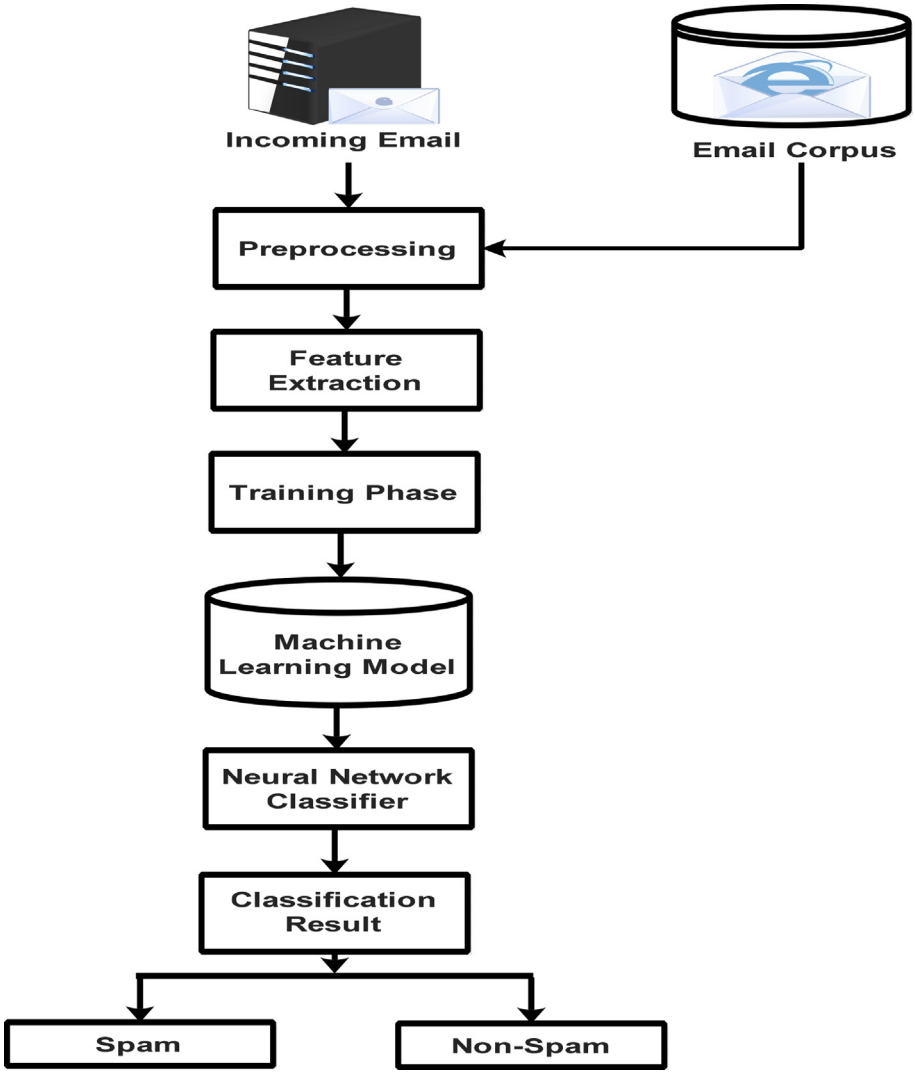

Fig. 4. Architecture of neural network (NN) Classifier.

5.5. Rough set classifiers

The rough set theory was proposed in 1982 by [72] in an effort to present a suitable framework for the automated conversion of data into knowledge. The technique is focused on the breakdown of categorisation of inexact, ambiguous or partial information stated in terms of the data gotten from experience. Rough set theory can be described as a recent mathematical method to fuzziness. The idea of Rough Set is built on the hypothesis that some knowledge is associated with every object of the universe. RS is a mathematical tool that concentrates on uncertainty [73]. It is in accordance with the notion that any inexact model can be estimated from underneath and from overhead by employing an association that is imperceptible in nature. One of the major feature of the RS philosophy is the need to discover redundancy and dependencies between features [74]. Rough Set theory has been applied to spam filtering because it provides efficient and less time consuming algorithms to extract hidden patterns in data. It also has the capacity to identify with ease the relationships that other conventional statistical techniques are finding difficult to find. Moreover, it accepts the use of both quantitative and qualitative data. It has the ability to estimate the minimum sets of data needed for grouping jobs.

Discovering the importance of data and creating a group of decision rules from the given data set are part of the strength of the RS classifiers. It is important to note that rough set theory expresses imprecision by using a borderline section of a set rather than by way of membership. Having the borderline section of a set empty implies that the set has been clearly defined (exact), if not the set is said to be rough (inexact). For a borderline section that contains at least one element in the set signifies that what we know about the set is not enough to exactly describe the set. According to [75], it can be observed that Rough Set techniques allow users to evaluate the significance of data. It allows the user to automatically generate the sets of decision rules from data. It is easy to understand. It offers straightforward interpretation of obtained results. It is suitable for concurrent (parallel/distributed) processing [76]. Fig. 5 below shows the email filtering process workflow of the Rough Set approach from the user mailbox.

The algorithm for spam email classification using rough set below is adopted from [100]:

Algorithm 5 Email spam classification algorithm using Rough Set

1: **Input** Email Testing Dataset (Dis_ testing dataset), Rule (RUL), b
2: **for** x ∈ Dis_T E **do**
3: **while** RUL (x) = 0 **do**
4: suspicious = suspicious ∪ {x};
5: **end while**
6: Let all r ∈ RUL (x) cast a number in favor of the non-spam class.

(*continued on next page*)

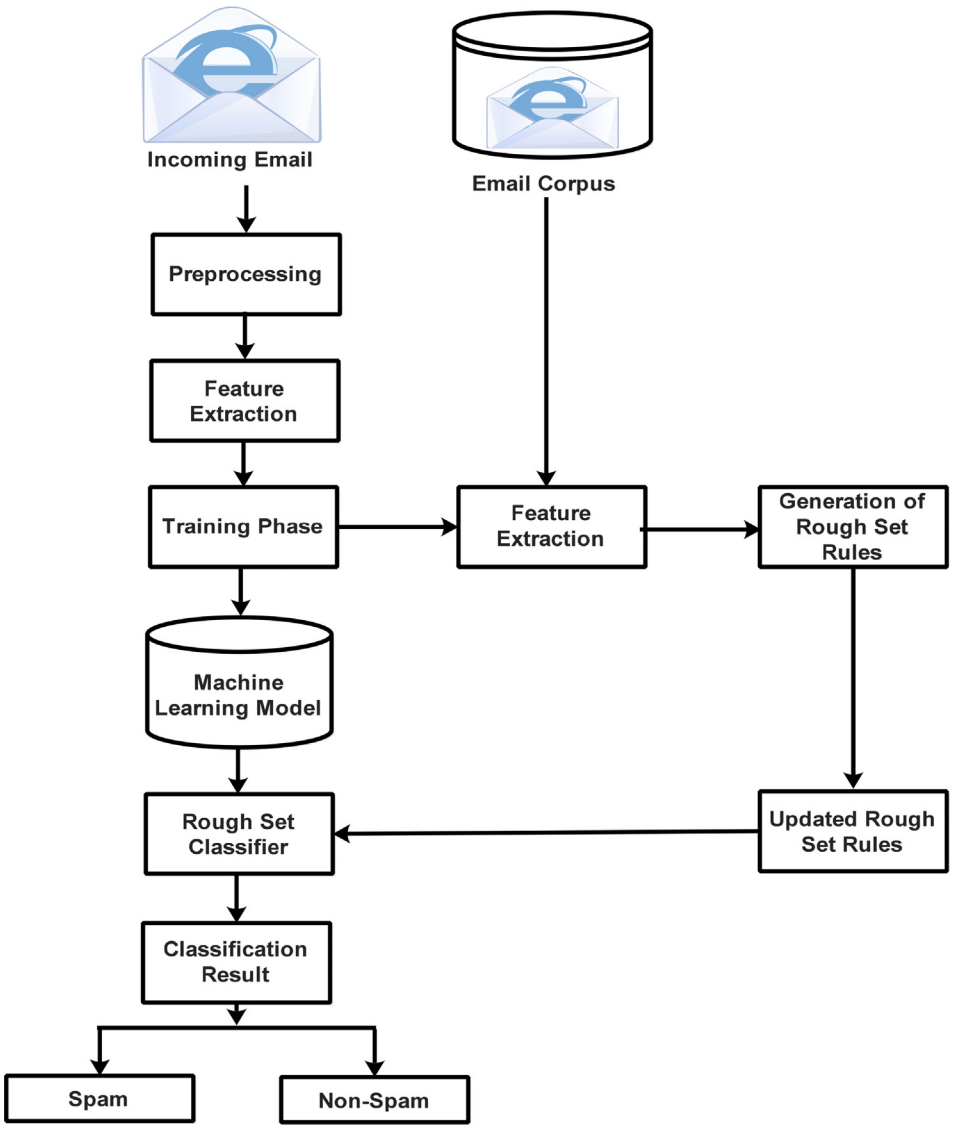

Fig. 5. Rough Set (RS) email filtering process workflow from user mailbox.

(*continued*)

Algorithm 5 Email spam classification algorithm using Rough Set

7: Predict membership degree based on the decision rules;
8: R = r ∈ RUL (x)|r predicts non-spam;
9: Estimate Rel (Dis_T E | x ∈ non-spam);
10: Rel (Dis_T E | x ∈ non-spam) = ∑r ∈ R Predicts (non-spam);
11: Certainty$_x$ = 1/cer × Rel (Dis_T E | x ∈ non-spam);
12: **while** Certaintyx≥1 – b **do**
13: suspicious = suspicious ∪ {x};
14: **end**
15: spam = spam ∪ {x};
16: **return** Final Email Message Classification (Spam/Non-spam/Suspicious email)
17:**end**

5.6. Support vector machine classifiers

Support Vector Machines (SVM) are supervised learning algorithms that have been proven to perform better than some other attendant learning algorithms [128]. SVM is a group of algorithms proposed by [77] for solving classification and regression problems. SVM has find application in providing solution to quadratic programming problems that have inequality constraints and linear equality by differentiating different groups by means of a hyperplane. It takes full advantage of the boundary [78]. Though the SVM might not be as fast as other classification methods, the algorithm draws its strength from its high accuracy because of its capacity to model multidimensional borderlines that are not sequential or straightforward. SVM is not easily susceptible to a situation where a model is disproportionately complex such as having numerous parameters comparative to the number of observations. These qualities make SVM the ideal algorithm for application in the areas of digital handwriting recognition, text categorization, speaker recognition, and so on. We briefly describe the binary *C*-SVM classifier which was explained in [79]. Here *C* denote the cost parameter to regulate modeling error which arises when a function is too closely fit to a limited set of data points by penalising the error ξ. During training, assuming we have a set of data to be trained, hypothetically there is only a merger of parameter (C, γ) which have the ability to produce the most superior SVM classifier. Grid-search on parameter C and γ is the only viable technique usually applied in SVM training to obtain this merger of parameter. The k-fold rotation estimation is employed in grid search to choose the SVM classifier with the most ideal rotation estimation prediction of accuracy [79]. The SVM training and classification algorithm for spam emails is

presented in the algorithm below:

Algorithm 6 Support Vector Machine (SVM) algorithm

1: **Input** Sample Email Message x to classify
2: A training set *S*, a kernel function, $\{c_1, c_2, \ldots c_{num}\}$ and $\{\gamma_1, \gamma_2, \ldots \gamma_{num}\}$.
3: Number of nearest neighbours k.
4: **for** $i = 1$ to *num*
5: set $C=C_i$;
6: **for** $j = 1$ to q
7: set $\gamma = \gamma$;
8: produce a trained SVM classifier $f(x)$ through the current merger parameter (C, γ);
9: **if** ($f(x)$ is the first produced discriminant function) **then**
10: keep $f(x)$ as the most ideal SVM classifier $f^*(x)$;
11: **else**
12: compare classifier $f(x)$ and the current best SVM classifier $f^*(x)$ using *k*-fold cross-validation
13: keep classifier with a better accuracy.
14: **end if**
15: **end for**
16: **end for**
17: **return** Final Email Message Classification (Spam/Non-spam email)
18: **end**

5.7. Decision tree

A Decision Tree (DT) is a type of classifier whose pattern looks like that of a tree structure. According to [28,98], decision tree induction is a distinctive technique that leads to gaining knowledge on classification. Each node of a DT is either a leaf node that specifies the value of the intended feature (class). It can also be a decision node that indicates certain test to be conducted on the value of a feature, with one branch and a sub-tree (which is a subset of the larger tree) representing every likely result of the test. A decision tree can be employed to provide solution to classification problem by beginning at the root of the tree and going through it until it gets to a leaf node that gives the classification result. Decision tree learning is an approach that have been applied to spam filtering. The aim is to produce a DT model and train the model in order for it to forecast the value of a goal variable centered on a number of input variables. The respective inner node communicates with some of the input variables [124]. Individual leaf denotes a value of the goal variable provided that the values of the input variables are from the path that leads from the root to the leaf. It is possible to learn a tree by breaking the fundamental set into different subsets depending on the value of the feature that was given before. This procedure is iterated for each resultant subset repeatedly which suggest the reason it is known as recursive partitioning. The recursion stops once all the subsets at a particular node all have goal variables that are similar. Another criteria that can lead to the termination of the recursion is when dividing the set is no more enhancing the predictions. There are different types of decision tree as explained below:

5.7.1. NBTree classifier

This is a type of decision tree that hybridised Naïve Bayes classifier with decision tree thereby combining the strengths of both algorithms. This approach works by applying Naïve Bayes classifier at the nodes while decision tree is developed with one variable that is divided at each node. For a database that is big in size, the NBTree classifier is very helpful, if the size of the database is non uniform and the features are not unavoidably autonomous, the strength of the NBTree becomes prominent. The database of the spam emails follows the above described pattern. The task of interpreting the classifier is straightforward just as we have in Naïve Bayes. The decision tree partitions the data into different part and Naïve Bayes explains the different subdivision (also known as leaf) [80].

5.7.2. C4.5/J48 decision tree algorithm

J48 is a modified, redistributed and freely available version of C4.5 decision tree algorithm. J48 is developed by studying data at the nodes which are used to examine the meaning of prevailing attributes. The authors claimed in [81], that the most commonly used and the most effective decision tree algorithm method is the C4.5 algorithm. A tree model is produced by the decision tree through the use of only one feature at a time. The algorithm uses the value of the feature to rearrange the dataset. And proceed to search for the areas of the dataset that obviously have one class and indicate those areas as leaves. For the remaining areas that contain classes that are more than one, the algorithm selects alternative features. It also maintains the dividing process with just the number of occurrences in such areas pending the time that the leaves are completely created, or there is absence of feature that can be utilised to create at least one leave varied in the conflicted areas. The decision tree produced by C4.5 can be applied for solving different classification problems. The algorithm selects the features that it can further divide into subclasses at each node. The output of the categorisation or result obtained is denoted by a leaf node [28].

5.7.3. Logistic model tree induction (LMT)

LMT is a type of decision tree that uses logistic regression models on leaves. This classifier has proved to have a higher degree of accuracy and robustness in diverse research fields. The main drawback of this approach is the high computational complexity incurred when there is an inducement of the logistic regression models into the tree. A prediction of a model is created by organising the tree down to the leaf and applying the logistic prediction model related to such leaf. The strength of the logistic model is that it is simple to decode and translate compared to C4.5 trees. Moreover, it has been proven that trees produced by LMT have a reduced size compared to those created by C4.5 induction. The authors in [80] revealed in their paper that there is a reduction in training time needed to create the logistic model tree compared to Naïve Bayes classifier and also gives superior result compared to Naïve Bayes classifier when they were applied to solve email spam filtering problem.

In this review work, we discussed the popular iterative Dichotomiser 3 (ID3) algorithm proposed by Ross Quinlan to build the decision tree using entropy and information gain. The entropy evaluates the adulteration of a random corpus of email samples while the information gain is used to compute entropy by dividing the email sample by some features. Assuming we have an email dataset E with classifications c_j, entropy is computed using Eq. (15) below:

$$\text{entropy } (E) \sum_{j=1}^{|c|} \Pr(c_j) \log_2 \Pr(C_j) \tag{15}$$

The relationship between the entropy and information gain represented in Eq. (16) below, where *entropy Ai(E)* is the estimated entropy of Feature *Fi* which is exploited in dividing the email messages as either spam or legitimate mail.

$$\text{gain } (E,\ F_i) = \text{entropy}(D) - \text{entropy}_{Fi}(E) \tag{16}$$

The decision tree algorithm for classifying email messages using entropy algorithm is presented below:

Algorithm 7 Decision Tree algorithm for Spam Filtering

1: **Input** Email Message dataset
2: Compute entropy for dataset
3: **while** condition **do**
4: **for** every attribute/feature
5: calculate entropy for all categorical values
6: take average information entropy for the current attribute.
7: calculate gain for the current attribute
8: pick the highest gain attribute
9: **end for**
10: **end while**
11: **return** Final Email Message Classification (Spam/Non-spam email)
12:**end**

By partitioning the email dataset in relation to least entropy, the resultant email dataset has the highest information gain and so impurity

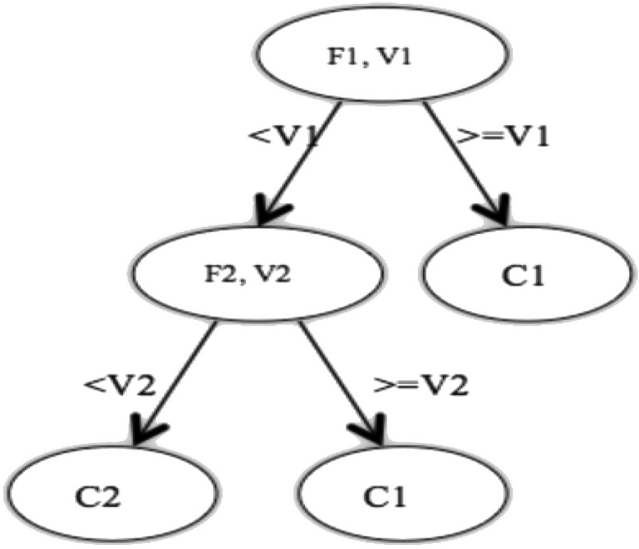

Fig. 6. Decision Tree Algorithm for email spam filtering.

(emails contain both spam and ham) of the dataset is reduced. The dataset can be tested using the decision tree algorithm after the tree is created from the training email dataset. The email dataset being tested undergo some processing in the tree using some predefined rules pending the time it will get to a leaf node. The label in the leaf node is then assigned to the tested data. Below in Fig. 6 is a theoretical tree that illustrate how the decision tree algorithm carries out its spam filtering operation. F represents the features or words in the email message. V depicts the values or word frequencies of some words contained in the email message. C depicts the labels which are either spam/ham.

5.8. Ensemble classifiers

Ensemble learning is a new approach in which a group of different classifiers are trained and assembled to further improve the classification accuracy of the complete system on identical problem, in this case it is spam filtering. They are a class of machine learning algorithm that work in harmony and are applied to enhance the classification performance of the whole system. In [82], the authors advocated the assembling of different filters as a very fascinating approach to effectively handle spams which now comes in different forms. The most widely accepted ensemble classifiers are bagging and boosting [83]. These algorithms train classifier instances on various subsets of the complete data set. Bagging combines the outputs of trained classifiers on sample drawn from a larger sample of the data set.

Dietterich [84] presented an overview of several ways of building ensembles and also presented an account to justify the reason for using ensembles and why their performance is better than their single members. Adeva, Beresi and Calvo [85] presented the significance of variety for the productive merging of different classifiers. A remarkable implementation of ensemble approach is found in Random Forests where a number of decision trees are created for an identical problem and their outputs are combined to obtain the most superior classification decision on the whole [86]. Bagging (also known as bootstrap aggregating) is an ensemble meta-learning algorithm that is normally used for decision tree methods. For instance, the random forest algorithm is an ensemble method for decision trees that is recognised for attaining great classification accuracy. Bagging ensembles was used by [39] to prevent poisoning attacks on spam filters. Random forests have been successfully used to create a model for spam detection as explained by [43,125].

Boosting is a very efficient technique that combines a series of "weak" learners to create a single learner that is stronger than the individual learner [39]. Boosting is classified as a learning algorithm which is centered on the theory of hybridisation of several weak hypotheses, a very good example is the AdaBoost system. The objective of boosting is to obtain a very accurate classification rule by amalgamating several weak rules or weak hypotheses each of which may be only relatively accurate. A learner is trained in every phase of the classification process, and the result of each phase is used to add credence to data for the upcoming phases [87]. AdaBoost is the most popular boosting algorithm. It was proposed by [88]. AdaBoost can produce a good output even when the performance of the weak learners are unsatisfactory. At present Boosting is now been applied in the field of classification, regression, face recognition and so on. Boosting algorithms that utilised confidence rated projections are being applied to solve spam filtering problem. Literature have also shown that they can produce classification results that are better than that of Bayesian and decision tree approaches [87]. AdaBoost has become a widely accepted machine learning algorithm because of its astounding performance in solving classification problems. It is believed among some statisticians that AdaBoost has some relationship with logistic regression probability maximisation [89]. The widespread use of AdaBoost according to Rob Schapire is not unconnected with the advantages that the approach have over some other learning algorithm. AdaBoost is fast, the algorithm is straightforward and easy to program, absence of parameter tuning (except T) makes is less cumbersome. It is adaptable and can combine well with any learning algorithm. Also, there no need of any previous knowledge about weak learner. It is verifiably efficient, provided it can always locate rough rules of thumb. The algorithm is very adaptable, and can be used with data that is textual, numeric or discrete in nature. It has been expanded further to learning problems that are outside binary classification. The AdaBoost algorithm for detecting spam email is show in algorithm 8 below:

Algorithm 8 AdaBoost Algorithm for Email Spam Classification (Adapted from [127])

1: **Input** set of email messages corpus M
2: **while** condition **do**
3: use the labeled message corpus M (labeled) to trains the classifier.
4: use the classifier to test the M (unlabeled) messages and produce scores using a scoring function.
5: relate each message with the matching score computed above.
6: label the messages with the least scores.
7: add the recently labeled messages into M (labeled) corpus.
8: eliminate the recently labeled message from the M (unlabeled) corpus.
9: **end while**
10: train the message corpus
11: given $(x_1, y_1)\ldots (x_n, y_n)\ \varepsilon\ S_t$ where $y_1 = 0.1$
12: weights $w_1 \ldots w_f = 1/f$, where f = number of features in an email message
13: **for** t = 1 to T **do**
14: $\sum_i w_i = 1$
15: error $e_j = \sum_i w_i \left|h_j(x_i) - (y_i)\right|$
16: Select classifier h_j with the least error
17: Update weights $w_{t+1,i} = w_{t,i}\beta_t^{1-ei}$ where $e_i = \begin{cases} 0 & \text{if classified correctly} \\ 1 & \text{otherwise} \end{cases}$
18: $\beta_t = \dfrac{e_t}{1 - e_t}$
19: $\alpha_t = \log(1/\beta_t)$
20: **end for**

21: **return** Final Email Message Classification (Spam/Non-spam email) $h(x) =$

$$\begin{cases} 1 & if \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

22:**end**

5.9. Random forests (RF)

Random forest (RF) is an example of ensemble learning approach and regression technique appropriate for solving problems that pertains to classifying data into groups [90]. Random forest was first proposed by [91]. The algorithm carry out prediction through the use of decision

trees. At the training stage, some of decision trees are created by the program writer. These decision trees are subsequently utilised for the task of predicting the group; this is accomplished by taking into account the selected groups of every distinct trees and the group that have the highest number of vote is taken as the result. RF technique is gaining more popularity these days and it has find application in different fields and in literature it has been used to provide solution to analogous problem, as found in [41,92,93]. Some of the strength of Random forests is that it usually have lesser classification error and superior f-scores compared to decision trees. Also, its performance is usually up to or even better than that of SVMs, even though they are considerably easier to comprehend for humans. Its performance is really good with disproportionate data sets that is characterised by some missing variables. It provides an efficient mechanism for computing the approximate value of missing data and preserving precision in situations where a considerable percentage of the data are lost. RF allows the user to grow as many trees as possible. The speed of execution is high. It was demonstrated in [91] that using RF to process a set of data with 50,000 cases and 100 variables, will generate 100 trees in 660 seconds on a computer with processor speed of 800Mhz. In situations where the size of the data sets is enormous, much of the needed memory is for the storage of the data and three arrays of whole numbers having equal magnitudes as the data. Computing the closeness shows that increase in the storage space needed is directly proportional to the amount of instances multiplied by the quantity of trees. RFs produces several trees used for classification. The task of classifying a new data from an input vector begins by placing the input vector along each of the trees in the forest. Every tree will perform its classification which is often referred to as the tree "votes" for that group. The forest decides which of the groups have the overall highest votes in the forest. The algorithm 8 below succinctly outline the steps involved in the construction of forest trees.

Algorithm 9 Random Forests Algorithm for Email Classification

1: **Input** *X*: number of nodes
2: **Input** *N*: number of features in the Email Message
3: **Input** *Y:* number of trees to be grown
4: **while** termination conditions is not true **do**
5: Select a self-staring Email Message S indiscriminately from the training corpus *Y*
6: Create tree $R_{??}$ from the selected self-starting Email Message S
7: Choose n features arbitrarily from N; where n≪N
8: Compute the optimal dividing point for node d among the n features
9: Divide the parent node to two offspring nodes through the optimal divide
10: Execute steps 1–3 till the maximum number of nodes (x) is created
11: Create your forest by iterating steps 1–4 for Y number of times
12: **end while**
13: generate result of every created trees $\{R_t\}_1^Y$
14: use a new Email Message for every created trees beginning at the root node
15: designate the Email Message to the group compatible with the leaf node.
16: merge the votes or results of every tree
17: **return** Final Email Message Classification (Spam/Non-spam email) group having the highest vote (*G*).
18: **end**

5.10. Deep learning algorithms

Deep learning is a new emerging area which exploits artificial intelligence and machine learning to learn features directly from the data, using multiple nonlinear processing layers. Deep learning models can achieve very high accuracy in email spam classification. Deng and Yu [94] discussed various deep learning techniques, their classification into supervised, unsupervised and hybrid deep networks depending on their architectures and applications like computer vision, language modeling, text processing, multimodal learning, and information retrieval.

The fundamental constituent of deep learning is the multilayered hierarchical data representation typically in the form of a neural network with more than two layers. These type of techniques allow spontaneous integrating of data features of a upper level to the lower ones. A neural network (NN) comprises of various unified neurons. The type of application being used will determined the number of neurons and the connections among them. According to [94], the deep learning methods can be categorised into the following:

1. Unsupervised Deep networks (also known as generative learning). Examples of deep networks include Autoencoder, Sparse Autoencoder (SAE), Stacked Sparse Autoencoder (SSAE), Restricted Boltzmann Machines (RBMs), Deep Belief Networks (DBNs), Deep Boltzmann Machines (DBMs), and generalised denoising autoencoders. They can be used to expressively produce samples by sampling from the networks, and so they are referred to as generative models. In unsupervised deep learning, there is no provision for labels during training and the overall purpose is to denote a function which is used to represent unseen structure from an unlabeled data. Unsupervised deep learning models performs different functions such as density estimation, clustering, feature learning (or representation learning), dimension reduction and others. This makes them a better algorithm for email spam filtering. The main benefit of the unsupervised learning lies in situations where there is a large volume of unlabelled data. Using unsupervised deep learning methods for such huge data enables easy learning of features that are better likened to hand-crafted features.
2. Supervised Deep networks. They are networks that are are meant to provide classifying power for the aim of pattern classification. They are usually characterised by the subsequent allocations of classes conditioned on the visible data. The sum–product network (SPN) and Convolutional Neural Network (CNN) are examples of supervised deep network.
3. Hybrid deep networks (Fusion of Unsupervised and Supervised). Examples of hybrid deep networks include the use of generative models of Deep Belief Networks (DBNs) to pre-train deep convolutional neural networks (deep CNNs). Pre-training DNNs or CNNs using a set of normalized deep autoencoders, including denoising autoencoders, contractive autoencoders, and sparse autoencoders

According to [95] the Convolutional Neural Network (CNN) has been a hot research area of late. CNN is advantageous because of its reliable fault tolerance, parallel processing and self-learning ability. It have been successfully applied in the field of email spam filtering. Albelwi and Mahmood [96] described the Deep Convolutional Neural Networks (CNNs) as a kind of feed-networks that draws its inspiration from biology. The network has light local connections and weight distribution among its neurons. A CNN comprises of several trainable phases piled up on top of each other, after which there is a supervised classifier and series of arrays termed feature maps which denote both input and output of each stage. CNNs are characteristically made up of several categories of layers, comprising of convolutional, pooling, and fully-connected layers. Automatic learning of feature description that is very discriminative without the need for hand-crafted features is possible in CNNs through the piling up of several layers. The workings of a CNN is different from the traditional backpropagation neural network (BPN) because a BPN operates on isolated hand-crafted image features whereas, a CNN operates precisely on an email message to mine valuable, essential features for classification. Below is a CNN algorithm for spam email classification.

Algorithm 10 Convolutional Neural Networks for Email Classification

1: **Input** Pretreatment of Email Message
2: **Input** parameters N
3: file = getfile ()//Find the Message Corpus
4: label = getlabel (file)//Find the labelled Messages
5: test = gettest (file)//Find the Email Message
6: vec = getword2vec ()//Load the word vector
7: random = random (label)//Randomized
8: **while** condition **do**
9: Nf = CV(len (xshuffle),nf)//Cross-validation

(*continued on next page*)

(*continued*)

Algorithm 10 Convolutional Neural Networks for Email Classification

```
10: for trindex, teindex in kf do
11: xtotal, ytotal = xshuffle [trindex],yshuffle [trindex]
12: xtrain, xdev,ytrain, ydev = split (xtotal, ytotal)
13://Divide the data set
14: for i < N do
15: conv = getconv ()//Convolution layer
16: h = sigmoid (conv)
17: N = getk ()//Get the value of N
18: tensorr = gettensor ()
19: for x,y in xtrain, ytrain do
20: value, indice = topk (tensorr)
21://Get the Email Message feature and location information
22: tensors = get (value, indice)
23://Get the corresponding tensor
24: tensora = append (tensors)
25: end for
26: end for
27: con = con (tensorp)
28: conn = sigmoid (con)//Sigmoid
29: getsoftmax (conn)//softmax
30: end for
31: if getdev () then
32: tr = false
33: end if
34: end while
35: return Final Email Message Classification (Spam/Non-spam email)
36: end
```

6. Study area

6.1. Comparative studies of existing machine learning techniques applied to spam filtering

Several Machine Learning techniques used in email spam filtering have been published in literatures. It is obvious from our study that in the bid to apply machine learning algorithm to solve the email spam problem, different learning algorithms are proposed each time thereby adding to the ever-expanding pool of machine learning algorithms for filtering spam mails. Evaluation of the performance of filters using only some figures is becoming more of an uphill task. Some studies have been devoted to evaluating the effectiveness of filters under similar situations using certain benchmarks. Such studies help us to identify the techniques that have superior performance under some circumstances. They also made clear the significance of taking into consideration other traits in the email in addition the headers and the body.

Karthika and Visalakshi [97] proposed an approach that combined and implemented Support Vector Machine (SVM) and Ant Colony Optimization (ACO) algorithms for spam classification. The proposed technique is a hybrid model which relies on selecting the features of emails for their classification. The SVM basically works as the classifier i.e. the classification algorithm while the feature selection of emails is implemented by the ACO algorithm for efficiency and accuracy. The proposed method permits more than one mail to be classified at a time with an improved speed of execution. The proposed method performs better than some state-of-the-art (i.e. kNN, NB and SVM) classification methods in terms of accuracy, precision and recall. Adopting ACO algorithm for feature selection offers an improved efficiency in spam mails classification. The drawback of their approach is low performance. This shows that there is still room for further improvement of the email spam classification.

Awad and Foqaha [1] used the hybrid of Radial Basis Function Neural Networks (RBFNN) with PSO algorithm (HC-RBFPSO) algorithms for spam email filtering. One of the strength of their proposed method is that it has good classification accuracy. It can effectively act as a reliable alternative to other existing spam mail classification techniques. Classification accuracy was the only parameter upon which the performance of the proposed method was evaluated. Other factors like precision and recall were not considered in the evaluation of the system. Their proposed method is not an improvement over already existing methods.

Sharma and Suryawanshi [99] applied the hybrid of kNN and Spearmen Correlation for detecting spam email. It is very difficult to assess the effectiveness of their proposed system as the performance was not evaluated against proven classification techniques. Thus, it is difficult to ascertain if it is an improvement of existing methods.

Awad and ELseuofi [100] reviewed six state of the art machine learning methods (Bayesian classification, k-NN, ANNs, SVMs, Artificial Immune System and Rough sets) and their applicability to the problem of spam email classification. Their performances in terms of precision, accuracy, and recall were compared using SpamAssassin dataset. The techniques compared all have very poor performance. The performances of kNN and AIS was very poor and also proves to be poor classifier for spam email classification. Rough sets algorithm has the worst performance. The performance result of the six machine learning methods was measured in terms of spam recall, precision and accuracy. The neural networks was the most simple and fastest algorithm among the six, while the rough sets method is the most complicated.

Rajamohana, Umamaheswari and Abirami [101] proposed adaptive binary flower pollination algorithm (ABFPA) which is a global optimisation technique which was used to extract features for review spam classification. The dataset used for measuring the performance of their approach was built by Ott et al [118]. The performance of their proposed technique was very low compared to some existing techniques.

Alkaht and Al-Khatib [102] used the multi stage Neural Network to filter spam emails. Their proposed technique outperformed Multi-Layer Perceptron (MLP) and perceptron classifiers. Also, applying this method in spam filtering produced better result than applying it for scenes classification in multispectral images, where originally it was adopted. The algorithm's performance for scenes classification and spam filtering was poor. The time taken to train the data set by their proposed method is too long. The test on Arabic and bilingual emails classification was not enough to judge the capability of the proposed system; the result was not good. This is basically due to the low quality of the dataset used for training, which was randomly collected emails. The small size of training dataset was not enough to cover the diversity of the whole dataset. Arabic language morphological properties were not taken into consideration to let the classifier be as general as possible. The proposed method did not use email features that can be extracted and used for training datasets to detect spam mails.

Sharma, Prajapat and Aslam [103] applied multilayer perceptron neural network (MLP) and naive Bayesian models using keywords selection method. They also quantify their results using statistical measures on emails as either spam or ham on TREC07 dataset. A major drawback of MLP model is that the training is slow as it takes a longer time to build compared to NB. Mousavi and Ayremlou [104] proposed the Naïve Bayesian algorithm for spam classification. The implementation of an algorithm called Porter Stemming in MATLAB is used for suffix stripping in email spam classification. The training of the algorithm with a larger training set yielded a better performance in terms of precision, with lower rate of recall. Their work did not consider classification accuracy in evaluating the training set. The performance of the training set was not evaluated against any tested algorithm. Hence, the performance of the proposed algorithm cannot be ascertained. The scope of the research work is too narrow; almost proffering nothing. The research work, together with its implemented algorithm does not make any significant contribution.

Dhanaraj and Palaniswami [105] applied the combination of Firefly algorithm and Naïve Bayes for email classification in in a distributed environment using the CSDMC2010 spam corpus dataset. Firefly algorithm was used to optimise and select the feature space with the best fitness. The spam classification was done with the Naïve Bayes classifier. The proposed method was not an improvement over existing methods in terms of specificity. The performance of the proposed method was not

evaluated using other performance metrics that can prove the effectiveness of their approach. Choudhary and Dhaka [106] applied the Genetic Algorithm for automatic classification of emails. The algorithm was able to successfully differentiate between spam and ham emails. The proposed method was not evaluated in terms of common email classification metrics like accuracy, precision, recall, and computation time. Hence, it performance in comparison to other methods cannot be rated. Other limitations peculiar to GA algorithm and its variants also applies to the proposed method. The over-all efficiency of the genetic algorithm based email identification depends on the large number of parameters like:

Table 4
Summary of published papers that attempted spam filtering using Machine Learning techniques.

Reference	Dataset Description	Proposed Technique	Compared Algorithm(s)	Performance Metrics	Limitation(s)
Karthika and Visalakshi [97]	Spambase dataset	Hybridised ACO and SVM	Hybridised ACO and SVM with KNN, NB and SVM.	Accuracy, precision and recall	Very low performance.
Awad and Foqaha [1]	Spambase dataset	Combined Radial Basis Function Neural Networks (RBFNN) with PSO algorithm (HC-RBFPSO)	PSO, RBFNN, MLP and ANN	Accuracy	Time taken to build MLP is very high. No improvement on existing methods.
Sharma and Suryawanshi [99]	Spambase dataset	kNN Classification with Spearmen Correlation	kNN with spearman and kNN with Euclidean	Accuracy, precision, recall, and F-measure.	Low performance
Awad and ELseuofi [100]	SpamAssassin	Bayesian classification, k-NN, ANNs, SVMs, Artificial Immune System and Rough sets	Bayesian classification, k-NN, ANNs, SVMs, Artificial Immune System and Rough sets.	Recall, precision and accuracy	Many of the state-of-the-art spam classification techniques were not examined.
Rajamohana, Umamaheswari and Abirami [101]	Dataset built by built by Ott et al. (2011) [118].	Adaptive binary flower pollination algorithm (ABFPA)	ABFPA, BPSO, SFLA for feature selection while ABFPA, NB and kNN	Global best positions.	Standard evaluation metrics were not used to evaluate the performance of the proposed method.
Alkaht and Al Khatib [102]	Randomly collected emails	Multi-stage Neural Networks for filtering spam	NN, MLP and Perceptron	Accuracy.	The method was not evaluated using standard email corpus. The training was time consuming
Sharma, Prajapat and Aslam [103]	TREC07 dataset	MLP	MLP and NB	Accuracy, precision, and recall	Low performance
Mousavi and Ayremlou [104]	Selected emails.	NB algorithm for spam classification	Not compared	Precision and Recall	No meaningful contribution to knowledge. Also, the performance of the method was not compared with other existing methods
Dhanaraj and Palaniswami [105]	CSDMC2010 dataset	Firefly and Bayes classifiers	Firefly, NB, NN and PSO algorithm.	Sensitivity, specificity and accuracy	Low performance.
Choudhary and Dhaka [106]	Words in data dictionary	GA	Not compared	Not stated	Performance not compared with other technique.
Palanisamy, Kumaresan and Varalakshmi [107]	Ling dataset	Negative selection and PSO	NSA, PSO, SVM, NB and DFS-SVM	Accuracy	Only accuracy of the method was used in assessing it performance.
Shrivastava and Bindu [108]	2248 emails	GA with Heuristic Fitness Function	Not compared	Classification accuracy.	Accuracy of the method not compared with that of other techniques.
Zavvar, Rezaei and Garavand [109]	Spambase dataset	PSO, ANN and SVM	PSO, SOM, kNN and SVM	AUC	The method only have AUC value as the only performance metrics.
Idris and Mohammad [110]	Datasets from Machine Learning and intelligent system	AIS	Not compared	False positive rate.	No standard metric to evaluate its performance neither was the effectiveness compared with any other standard spam filtering method.
Sosa [111]	2200 e-mails from several senders to various receivers	Forward feature selection using a single-layer ANN as classificator with double cross-validation with 5-Fold	Not compared	Classification accuracy.	The effectiveness of the method was not measured with other known technique.
Karthika, Visalakshi and Sankar [112]	Spambase	GA-NB and ACO-NB	GA-NB and ACO-NB	Accuracy, recall, precision and F-measure.	The is no improvement gain in the proposed algorithm compared to the existing approaches.
Bhagyashri and Pratap [113]	SpamAssasin	Bayes Algorithm	Not compared	Precision, recall and accuracy.	The performance of the method was not compared with other standard algorithms.
Zhao and Zhang [115]	Spambase	Rough Set	RS and NB	Classification Accuracy, Precision and Recall.	Low performance
Kumar and Arumugan [116]	Collected emails	Probabilistic neural network for classification of spam mails while Particle Swarm Optimization is used for feature selection	PNN, BLAST and NB	Specificity and sensitivity.	Low performance
Akinyelu and Adewumi [90]	2000 phishing and ham mails	Random Forest	Compared with Fette et al [92]	False Positive and False Negative	Adequate performance metrics not used to evaluate the effectiveness of the method.
Akshita [117]	PU1, PU2, PU3, PUA and Enron Spam	Deep Learning for Java (DL4J) Deep Networks	Dense MLP, SDAE and DBN	Accuracy, Recall, Precision and F1	Time consuming training

email data set, number of words in the data dictionary, chromosome size, size of each group in the data dictionary and so on. On the other hand, the type of mail also affects the performance of GA based filtering techniques like url, image type, text type and so on.

Palanisamy, Kumaresan and Varalakshmi [107] applied the hybrid of combined negative selection algorithm (NSA) and PSO using a local outlier factor (LOF) as the fitness function for the detector generation for email classification on Ling spam dataset. The performance of their system cannot be really evaluated as factors like precision, recall, computation time, and false positives were not used in evaluating the performance of the system. Only the classification accuracy of the method was used in assessing it performance. The proposed work can be further enhanced to improve its efficiency by using better optimization technique as the classification accurate is still very low. Shrivastava and Bindu [108] applied Genetic Algorithm with Heuristic Fitness Function for email spam classification. The method was tested on 2248 mails. Genetic Algorithms do not work well when the population size is small and the rate of change is too high. Performance measures like precision, recall, computation time, and false positives were not used in evaluating the performance of the system. Zavvar, Rezaei and Garavand [109] implemented email spam detection by using the fusion of Particle Swarm Optimization, Artificial Neural Network and Support Vector Machine on spambase datasets retrieved from UCI repository. The proposed method was compared with other methods such as data classification Self Organizing Map (SOM) and K-Means based on Area Under Curve (AUC). The proposed method have low performance. Factors like precision, recall, computation time, and false positives were not used in evaluating the performance of the system.

Idris and Mohammad [110] presented an AIS based email classification technique for spam detection. The drawback of their technique is that they did not use any standard metric to evaluate its performance neither was the effectiveness compared with any other standard spam filtering method. Factors like accuracy, precision and computation time were not used in the evaluation of the proposed system. Sosa [111] applied Sinespam, a spam classification technique using Machine Learning to classify a corpus of 2200 e-mails from several senders to various receivers gathered by the ISP. While the NN method have a relatively high accuracy and is advantageous, its spam precision performance is not sufficient for it to be used without supervision. To enhance the efficiency of this technique, there is need for added members or adjustments of the feature set. A combination of keywords and descriptive characteristics may provide more accurate classification, as well as the combination of spam classification techniques. Renuka, Visalakshi and Sankar [112] used the hybrid of GA-NB and ACO-NB for email classification on spambase dataset. The spam classification is implemented using Naïve Bayes algorithm while feature selection is executed using ant colony optimization algorithm. The proposed technique performed very poorly.

Bhagyashri and Pratap [113] applied the Bayesian filter for automatic emails classification. The classification using Bayesian filter is done according to the method defined by Graham [114]. The performance of the proposed method was tested using SpamAssassin dataset. A major drawback of the Bayesian spam filtering technique is that it may be vulnerable to Bayesian poisoning, a technique used by spammers in an bid to reduce the efficacy of spam filters that depend on Bayesian filtering. The performance of the method was not compared with any other standard algorithm. Zhao and Zhang [115] used the Rough Set and Naïve Bayes for email classification on spambase dataset taken from UCI repository. They used the rough set theory to develop email classification model. Their method have a moderately low performance in terms of precision and recall. The proposed system was not tested with real time data. Kumar and Arumugam [116] applied the combination of probabilistic neural network for classification of spam mails while Particle Swarm Optimization is used for feature selection. The proposed system have low performance in terms of accuracy, specificity and sensitivity.

Akinyelu and Adewumi [90] proposed Random Forest for classifying phishing attacks with the method used by Fette et al [92] on a dataset comprising 2000 phishing and ham emails. The proposed technique only used two performance metrics to measure the performance of the proposed technique. This is not sufficient to determine the effectiveness of a method.

Akshita [117] applied the Deep Learning technique to content based spam classification. The author used DL4J deep network on PU1, PU2, PU3, PUA and Enron spam datasets. All the methods used have good performance. The performance of the proposed system was compared with Dense MLP, SDAE and DBN. The major shortcoming of SDAE and DBN is the huge time required in training the spam filters.

A summary of published papers that attempted spam filtering using machine learning techniques is in Table 4 below.

The previous literatures surveyed revealed some important suggestions that could help in fighting the threat posed by spam emails. However, there are some drawbacks which could possibly be caused by the area of focus of the researchers. Presented below are the summary of research gaps found in many of the surveyed literatures.

- Some papers focused on feature-free methods for email spam filtering since it have proven to have higher accuracy than the feature-based technique. It should however be noted that feature-free techniques have a high computational cost since it usually take much longer time in its e-mail classification task. It also suffers from implementation complexity.
- Some studies considered using subject line, header, and message body as the most important feature in classifying messages as either spam or ham. However, it is worth mentioning that suspicious subject line, header and body alone can lead to error in spam mail classification. Users might also need to select features manually.
- Other researcher discovered that bag of words model are relatively effective features for filtering spam and phishing emails, and email headers are features which are as critical as message body in detecting spam mails.
- Most of the researchers did not put the computational cost into consideration in the choice of which machine learning technique to use for spam mail filtering. Their main focus is performance in terms of classification accuracy.
- Some researchers used the behavioural patterns of spammers as an important aspect of spam detection while machine learning algorithms were used for extracting the important features from the message body. Comprehensive feature engineering might be required for better accuracy.
- Except for Deep learning, the other machine learning techniques applied to email spam filtering have the limitation of average fault tolerance, lack of parallel processing and low self-learning capability.

7. Discussion

Machine learning algorithms have been extensively applied in the field of spam filtering. Substantial work have been done to improve the effectiveness of spam filters for classifying emails as either ham (valid messages) or spam (unwanted messages) by means of ML classifiers. They have the ability to recognise distinctive characteristics of the contents of emails. Many significant work have been done in the field of spam filtering using techniques that does not possess the ability to adapt to different conditions; and on problems that are exclusive to some fields e.g. identifying messages that are hidden inside a stego image. Most of the machine learning algorithms used for classification of tasks were designed to learn about inactive objective groups. The authors in [119] posited that when these algorithms are trained on data that has some data that have been poisoned by an enemy, it makes the algorithms susceptible to a number of different attacks on the reliability and accessibility of the data. As a matter of fact, manipulating as minute as 1% of the training data is enough in certain instances [120]. Though it might be strange to hear that the data supplied by an enemy is used to train a system, it does

happen in some real world systems. Examples include spam detection systems, spam connection, financial fraud, credit card fraud, and other unwelcome deeds where the earlier deeds of the enemy are a major origin of training data. The unfortunate thing is that a good number systems are re-trained regularly using the new instances of undesirable activities. This serves as a launching pad for attacker to launch more attacks on such system.

One of the open problem that needs to be addressed is handling of threat to the security of the spam filters. Though some attempt have been made to address this problem. For example, the threat model for adaptive spam filters proposed by [121] categorises attacks based to whether they are causative or exploratory, targeted or indiscriminate, and if they are meant to interrupt reliability or accessibility. The purpose of causative attack is to trigger error in categorisation of messages, whereas an exploratory attack aims to determine the classification of a message or set of messages. An attacks on integrity is meant to have a negative influence on the classification of spam, on the other hand, attacks on accessibility is meant to have a negative influence on the c classification of ham. The fundamental purpose of a spammer is to send spam which cannot be seized by the filter (or user) and labeled as spam. There are other potential capabilities of attack which all depend entirely on the ability to send random messages grouped as spam. A larger percentage of spam filters are nevertheless susceptible to different kinds of attack. For example, Bayes filter is susceptible to mimicry attack [120]. Naïve Bayes and AdaBoost also demonstrated endless deterioration to adversary control attack.

Further research work need to be conducted to tackle the fact that email spam filtering is a concept drift problem. As such, while the spam filter researchers are trying to increase the prognostic accuracy of the filter, the spammers are also evolving and trying to surpass the efficiency of the spam filters. It becomes very important to develop more efficient techniques that will adequately handle the trend or progression in spam features that makes them to evade many spam filters undetected. The most successful technique applied in filtering spam is the content based spam filtering approach which classify emails as either spam or ham depending on the data that made up the content of the message. Examples of this technique include Bayesian Filtering, SVM, kNN classifier, Neural Network, AdaBoost classifier, and others. Systems based on machine learning approach facilitates learning and adjustment to recent dangers posed to the security of spam filters. They also have the capacity to counter curative channels that spammers are using.

We hereby suggest that the future of email spam filters lies in deep learning for content-based classification and deep adversarial learning techniques. Deep learning is a kind of machine learning technique that allows computers to learn from experience and knowledge devoid of explicit programming and mine valuable patterns from primitive data [122]. The traditional machine learning algorithms finds it very hard to mine adequately-represented features because to the limitations that characterised such algorithms. The shortcomings of the usual machine learning algorithms include: need for knowledge from expert in a particular field, curse of dimensionality, and high computational cost. Deep learning have been applied to solve representation problem by creating several naive features to represent a complicated concept. Deep learning will be far more effective in solving the problem of spam email because as number of available training data is increasing, the effectiveness and efficiency of deep learning becomes more pronounced. Deep learning models have the capacity to solve sophisticated problems by using intricate and huge models. Thus, they exploit the computational power of modern CPUs and GPUs. Deep learning is generally considered to be a black box since we have imperfect knowledge of the explanations behind its high performance. Despite the huge success of deep learning in solving many problems, it has been discovered lately that deep neural networks are susceptible to adversarial examples. Adversarial examples are unnoticeable to human but can effortlessly fool deep neural networks during the testing/deploying phase. The helplessness to adversarial examples becomes one of the foremost dangers for using deep neural networks in situations where safety is very crucial. Therefore, the adversarial deep learning technique is a great method that is yet to be exploited in email spam filtering.

Summarily, the open research problems in email spam filtering are itemized below:

- Lack of effective strategy to handle the threats to the security of the spam filters. Such an attack can be causative or exploratory, targeted or indiscriminate attack.
- The inability of the current spam filtering techniques to effectively deal with the concept drift phenomenon.
- Majority of the existing email spam filters does not possess the capacity to incrementally learn in real-time. Conventional spam email classification techniques are no longer viable to cope in real time environment that is characterised by evolving data streams and concept drift.
- Failure of many spam filters to reduce their false positive rate.
- Development of more efficient image spam filters. Most spam filters can only classify spam messages that are text. However, many savvy spammers send spam email as text embedded in an image (stego image) thereby making the spam email to evade detection from filters.
- The need to develop adapted, scalable, and integrated filters by applying ontology and semantic web to spam email filtering.
- Lack of filters that have the capacity to dynamically update the feature space. Majority of the existing spam filters are unable to incrementally add or delete features without re-creating the model totally to keep abreast of current trends in email spam filtering.
- The need to apply deep learning to spam filtering in order to exploit its numerous processing layers and many levels of abstraction to learn representations of data.
- The inevitable need to design spam filters with lower processing and classification time using Graphics Processing Unit (GPU) and Field-Programmable Gate Array (FPGA) with their advantage of low power consumption, reconfigurability, and real-time processing capability for real-time processing and classification.

8. Conclusion

In this study, we reviewed machine learning approaches and their application to the field of spam filtering. A review of the state of the art algorithms been applied for classification of messages as either spam or ham is provided. The attempts made by different researchers to solving the problem of spam through the use of machine learning classifiers was discussed. The evolution of spam messages over the years to evade filters was examined. The basic architecture of email spam filter and the processes involved in filtering spam emails were looked into. The paper surveyed some of the publicly available datasets and performance metrics that can be used to measure the effectiveness of any spam filter. The challenges of the machine learning algorithms in efficiently handling the menace of spam was pointed out and comparative studies of the machine learning technics available in literature was done. We also revealed some open research problems associated with spam filters. In general, the figure and volume of literature we reviewed shows that significant progress have been made and will still be made in this field. Having discussed the open problems in spam filtering, further research to enhance the effectiveness of spam filters need to be done. This will make the development of spam filters to continue to be an active research field for academician and industry practitioners researching machine learning techniques for effective spam filtering. Our hope is that research students will use this paper as a spring board for doing qualitative research in spam filtering using machine learning, deep leaning and deep adversarial learning algorithms.

Declarations

Author contribution statement

Emmanuel Gbenga Dada, Joseph Stephen Bassi, Haruna Chiroma, Shafi'i Muhammad Abdulhamid, Adebayo Olusola Adetunmbi, Opeyemi Emmanuel Ajibuwa: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Competing interest statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] M. Awad, M. Foqaha, Email spam classification using hybrid approach of RBF neural network and particle swarm optimization, Int. J. Netw. Secur. Appl. 8 (4) (2016).
[2] D.M. Fonseca, O.H. Fazzion, E. Cunha, I. Las-Casas, P.D. Guedes, W. Meira, M. Chaves, Measuring characterizing, and avoiding spam traffic costs, IEEE Int. Comp. 99 (2016).
[3] Visited on May 15, 2017, Kaspersky Lab Spam Report, 2017, 2012, https://www.securelist.com/en/analysis/204792230/Spam_Report_April_2012.
[4] E.M. Bahgat, S. Rady, W. Gad, An e-mail filtering approach using classification techniques, in: The 1st International Conference on Advanced Intelligent System and Informatics (AISI2015), November 28-30, 2015, Springer International Publishing, BeniSuef, Egypt, 2016, pp. 321–331.
[5] N. Bouguila, O. Amayri, A discrete mixture-based kernel for SVMs: application to spam and image categorization, Inf. Process. Manag. 45 (6) (2009) 631–642.
[6] Y. Cao, X. Liao, Y. Li, An e-mail filtering approach using neural network, in: International Symposium on Neural Networks, Springer Berlin Heidelberg, 2004, pp. 688–694.
[7] F. Fdez-Riverola, E.L. Iglesias, F. Diaz, J.R. Méndez, J.M. Corchado, SpamHunting: an instance-based reasoning system for spam labelling and filtering, Decis. Support Syst. 43 (3) (2007) 722–736.
[8] S. Mason, New Law Designed to Limit Amount of Spam in E-Mail, 2003. http://www.wral.com/technolog.
[9] I. Stuart, S.H. Cha, C. Tappert, A neural network classifier for junk e-mail, in: Document Analysis Systems VI, Springer Berlin Heidelberg, 2004, pp. 442–450.
[10] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, Elsevier, 2011.
[11] S.N. Qasem, S.M. Shamsuddin, A.M. Zain, Multi-objective hybrid algorithms for radial basis function neural network design, Knowl. Based Syst. 27 (2012) 475–497.
[12] J.D. Schaffer, D. Whitley, L. Eshelman, Combinations of genetic algorithms and neural networks: a survey of the state of the art, Combinations of Genetic Algorithms and Neural Networks, 1992, pp. 1–37.
[13] E. Elbeltagi, T. Hegazy, D. Grierson, Comparison among five evolutionary-based optimization algorithms, Adv. Eng. Inf. 19 (2005) 43–53.
[14] L.H. Gomes, C. Cazita, J.M. Almeida, V. Almeida, W.J. Meira, Workload models of spam and legitimate e-mails, Perform. Eval 64 (7–8) (2007) 690–714.
[15] C.C. Wang, S.Y. Chen, Using header session messages to anti-spamming, Comput. Secur. 26 (5) (2007) 381–390.
[16] T.S. Guzella, W.M. Caminhas, A review of machine learning approaches to spam filtering, Expert Syst. Appl. 36 (7) (2009) 10206–10222.
[17] C.P. Lueg, From spam filtering to information retrieval and back: seeking conceptual foundations for spam filtering, Proc. Assoc. Inf. Sci. Technol. 42 (1) (2005).
[18] X.L. Wang, Learning to classify email: a survey, in: 2005 International Conference on Machine Learning and Cybernetics (Vol. 9, pp. 5716-5719), IEEE, Aug 2005.
[19] W. Li, N. Zhong, Y. Yao, J. Liu, C. Liu, Spam filtering and email-mediated applications, in: Paper presented at the International Workshop on Web Intelligence Meets Brain Informatics, 2006.
[20] G.V. Cormack, Email spam filtering: a systematic review, Found. Trends Inf. Retr. 1 (4) (2008) 335–455.
[21] E.P. Sanz, J.M.G. Hidalgo, J.C.C. Pérez, Email spam filtering, Adv. Comput. 74 (2008) 45–114.
[22] S. Dhanaraj, V. Karthikeyani, A study on e-mail image spam filtering techniques, in: Paper presented at the International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013.
[23] A. Bhowmick, S.M. Hazarika, Machine Learning for E-Mail Spam Filtering: Review, Techniques and Trends, arXiv:1606.01042v1 [cs.LG] 3 Jun 2016, 2016, pp. 1–27.
[24] C. Laorden, X. Ugarte-Pedrero, I. Santos, B. Sanz, J. Nieves, P.G. Bringas, Study on the effectiveness of anomaly detection for spam filtering, Inf. Sci. 277 (2014) 421–444.
[25] I. Katakis, G. Tsoumakas, I. Vlahavas, Email mining : emerging techniques for email management, in: A. Vakali, G. Pallis (Eds.), Web Data Management Practices: Emerging Techniques and Technologies, Idea Group Publishing, USA, 2007 chap 10.
[26] T. Savita, B. Santoshkumar, Effective spam detection method for email, international conference on advances in engineering & technology - 2014 (ICAET-2014), OSR J. Comp. Sci. (IOSR – JCE) (2014) 68–72.
[27] B. Irwin, B. Friedman, Spam Construction Trends, in: Information Security for South Africa (ISSA), 2008, pp. 1–12.
[28] V. Christina, S. Karpagavalli, G. Suganya, Email spam filtering using supervised machine learning techniques, Int. J. Comput. Sci. Eng. 02 (09) (2010) 3126–3129.
[29] Available at, Mail Server Solution, 2017, http://telco-soft.in/mailserver.php.
[30] T. Subramaniam, H.A. Jalab, A.Y. Taqa, Overview of textual anti-spam filtering techniques, Int. J. Phys. Sci. 5 (12) (2010) 1869–1882.
[31] M.F. Porter, An algorithm for suffix stripping, Program: Electron. Libr. Inf. Syst. 14 (3) (1980) 130–137.
[32] T.A. Almeida, A. Yamakami, Content-based spam filtering, in: The 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, 2010, pp. 1–7.
[33] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, Stacking classifiers for anti-spam filtering of E-mail, in: Empirical Methods in Natural Language Processing, 2001, pp. 44–50.
[34] A. Attar, R.M. Rad, R.E. Atani, A survey of image spamming and filtering techniques, Artif. Intell. Rev. 40 (1) (2011) 71–105.
[35] J.R. Mendez, F. Díaz, E.L. Iglesias, J.M. Corchado, A comparative performance study of feature selection methods for the anti-spam filtering domain, in: Advances in Data Mining. Applications in Medicine, Web Mining, Marketing, Image and Signal Mining, Springer Berlin Heidelberg, 2006, pp. 106–120.
[36] L. Zhang, J. Zhu, T. Yao, An evaluation of statistical spam filtering techniques spam filtering as text categorization, ACM Trans. Asian Lang. Inf. Process 3 (4) (2004) 243–269.
[37] G.V. Cormack, T.R. Lynam, On-line supervised spam filter evaluation, ACM Trans. Inf. Syst. 25 (3) (2007).
[38] I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, G. Paliouras, C.D. Spyropoulos, An evaluation of naive bayesian anti-spam filtering, in: Proceedings of 11th European Conference on Machine Learning (ECML 2000), Barcelona, 2000, pp. 9–17.
[39] B. Biggio, I. Corona, G. Fumera, G. Giacinto, F. Roli, Bagging classifiers for fighting poisoning attacks in adversarial classification tasks, in: Multiple Classifier Systems, Springer Berlin Heidelberg, 2011, pp. 350–359.
[40] S. Abu-nimeh, D. Nappa, X. Wang, S. Nair, A comparison of machine learning techniques for phishing detection, in: eCrime 07: Proceedings of the Antiphishing Working Groups 2nd Annual eCrime Researchers Summit, New York,USA, 2007, pp. 60–69.
[41] I. Koprinska, J. Poon, J. Clark, J. Chan, Learning to classify e-mail, Inf. Sci. 177 (10) (2007) 2167–2187.
[42] I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, G. Paliouras, C.D. Spyropoulos, Learning to filter spam E-mail : a comparison of a naïve bayesian and a memory based approach, in: Proceedings of 4th European Conference on Principles and Practice of Knowledge Discovery in Databases, Lyon, France, September 2000, 2000, pp. 1–12.
[43] D. Debarr, H. Wechsler, Spam detection using clustering , random forests and active learning, in: CEAS 2009 Sixth Conference on Email and Anti-spam, 2009.
[44] Z. Wang, W. Josephson, Q. Lv, M. Charikar, K. Li, Filtering image spam with near-duplicate detection, in: Proc of the Fourth Conf on Email and Anti-spam, 2007.
[45] M. Dredze, R. Gevaryahu, A. Elias-Bachrach, Learning fast classifiers for image spam, in: Proc of the Fourth Conf on Email and Anti-spam, 2007.
[46] Y. Gao, M. Yang, X. Zhao, B. Pardo, Y. Wu, T. Pappas, A. Choudhary, Image spam hunter, in: Proc. Of the 33th IEEE International Conference on Acoustics, Speech, and Signal Processing, Las Vegas, NV, USA, 2008.
[47] Available at, CS Mining Group, 2010, http://www.csmining.org/index.php/malicious-software-datasets-.html.
[48] T. Fawcett, An introduction to ROC analysis, Pattern Recogn. Lett. 27 (8) (2006) 861–874.
[49] Mathswork, Detector Performance Analysis Using ROC Curves – MATLAB & Simulink Example, Retrieved August 11, 2017 from, 2016, http://www.mathworks.com/help/phased/examples/detector-performance-analysis-using-roc-curves.html.
[50] I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, C.D. Spyropoulos, An experimental comparison of naïve Bayesian and keyword-based anti-spam filtering with personal e-mail messages, in: Proc of the Ann Int ACM SIGIR Conf on Res and Devel in Inform Retrieval, 2000.
[51] I. Androutsopoulos, G. Paliouras, E. Michelakis, Learning to Filter Unsolicited Commercial E-Mail. Tech. Rep., National Centre for Scientific Research Demokritos, Athens, Greece, 2011.
[52] K.P. Clark, A Survey of Content-Based Spam Classifiers, 2008, pp. 1–19.
[53] T.M. Mitchell, Machine Learning, first ed., McGraw-Hill, 1997.

[54] J.S. Whissell, C.L.A. Clarke, Clustering for semi-supervised spam filtering, in: Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-abuse and Spam Conference (CEAS '11), 2011, pp. 125–134.

[55] S. Dipika, D. Kanchan, Spam e-mails filtering techniques, Int. J. Tech. Res. Appl. 4 (6) (2016) 7–11.

[56] T. Saravanan, A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm, Retrieved on August 8, 2017 from, 2010, https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/.

[57] P.N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Pearson Addison Wesley, Boston, 2006 c2006. 0321321367.

[58] S. Zhu, W. Dong, W. Liu, Hierarchical reinforcement learning based on KNN classification algorithms, Int. J. Hosp. Inf. Technol. 8 (8) (2015) 175–184.

[59] I. Biju, J.J. Wendy, Implementing spam detection using Bayesian and porter stemmer keyword stripping approaches, in: TENCON 2009-2009 IEEE Region 10 Conference, 2009, pp. 1–5.

[60] J. Wu, T. Deng, Research in anti-spam method based on bayesian filtering, in: Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on, 2, 2008, pp. 887–891.

[61] S. Ray, 6 Easy Steps to Learn Naive Bayes Algorithm (With Code in Python), Retrieved on August 9, 2017 from, 2015, https://www.analyticsvidhya.com/blog/2015/09/naive-bayes-explained/.

[62] M.N. Marsono, M.W. El-Kharashi, F. Gebali, Binary LNS-Based Naïve Bayes Inference Engine for Spam Control: Noise Analysis and FPGA Synthesis, IET Computers & Digital Techniques, 2008.

[63] K. Li, Z. Zhong, Fast statistical spam filter by approximate classifications, in: Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems, 2006. Saint Malo, France.

[64] A. Edstrom, Detecting Spam with Artificial Neural Networks, Retrieved on August 10, 2017 from, 2016, http://homepages.cae.wisc.edu/~ece539/project/s16/Edstrom_rpt.pdf.

[65] A. Chandra, S. Mohammad, B. Rizwan, Web spam classification using supervised artificial neural network algorithms, Adv. Comput. Intell.: Int. J. (ACII) 2 (1) (2015) 21–30.

[66] O.A.S. Carpinteiro, I. Lima, J.M.C. Assis, A.C.Z. de Souza, E.M. Moreira, C.A.M. Pinheiro, A Neural Model in Anti-spam Systems, Lecture Notes in Computer Science, Springer, Berlin, 2006.

[67] D. Ndumiyana, M. Magomelo, L. Sakala, Spam detection using a neural network classifier, Online J. Phys. Environ. Sci. Res. 2 (2) (2013) 28–37. ISSN 2315-5027.

[68] X.S. Yang, Firefly algorithms for multimodal optimisation, Proc. 5th symposium on stochastic algorithms, foundations and applications, in: O. Watanabe, T. Zeugmann (Eds.), Lecture Notes in Computer Science 5792, 2009, pp. 169–178.

[69] J. Dugonik, I. Fister, Multi-population firefly algorithm, in: Proc. Of the 1st Student Computer Science Research Conference, Ljubljana, Slovenia, 2014, pp. 19–23.

[70] W.A. Khan, N.N. Hamadneh, S.L. Tilahun, J.M. Ngnotchouye, A Review and Comparative Study of Firefly Algorithm and its Modified Versions, Intech Publishing House, 2016, pp. 281–313. Chapter 13.

[71] A. Kundur, Evaluation of Firefly Algorithm Using Benchmark Functions, Department of Computer Science. North Dakota State University of Agriculture and Applied Science, 2013. Master thesis.

[72] Z.I. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic, New York, NY, USA, 1991.

[73] S.S. Roy, V.M. Viswanatham, P.V. Krishna, N. Saraf, A. Gupta, R. Mishra, Applicability of rough set technique for data investigation and optimization of intrusion detection system, in: Quality, Reliability, Security and Robustness in Heterogeneous Networks, Springer Berlin Heidelberg, 2013, pp. 479–484.

[74] N. Pérez-Díaz, D. Ruano-Ordás, F. Fdez-Riverola, J.R. Méndez, Rough sets for spam filtering: selecting appropriate decision rules for boundary classification, Appl. Soft Comput. 13 (8) (2012) 1–8.

[75] N.B. Agnieszka, Mining rule-based knowledge bases inspired by rough set theory, Fundam. Inf. 148 (1-2) (2016) 35–50.

[76] N. Pérez-Díaz, D. Ruano-Ordás, F. Fdez-Riverola, J.R. Méndez, Boosting Accuracy of Classical Machine Learning Antispam Classifiers in Real Scenarios by Applying Rough Set Theory, Hindawi Publishing Corporation, Scientific Programming, 2016, 2016, Article ID 5945192, 10 pages.

[77] V. Vapnik, The Nature of Statistical Learning Theory, Springer Verlag, 1995.

[78] Z.S. Torabi, M.H. Nadimi-Shahraki, A. Nabiollahi, Efficient support vector machines for spam detection: a survey. (IJCSIS), Int. J. Comput. Sci. Inf. Secur. 13 (1) (2015) 11–28.

[79] S.K. Chen, Y.H. Chang, SVM classifier algorithm, in: Proc. Of 2014 International Conference on Artificial Intelligence and Software Engineering(AISE2014), 6, DEStech Publications, Inc, 2014, p. 655.

[80] S. Chakraborty, B. Mondal, Spam mail filtering technique using different decision tree classifiers through data mining approach - a comparative performance analysis, Int. J. Comput. Appl. 47 (16) (2012) 26–31, 0975 – 888.

[81] K. Masud, M.R. Rashedur, Decision tree and naïve Bayes algorithm for classification and generation of actionable knowledge for direct marketing, J. Softw. Eng. Appl. 6 (2013) 196–206.

[82] P.H.C. Guerra, D. Guedes, J.W. Meira, C. Hoepers, M.H.P.C. Chaves, K. Steding-Jessen, Exploring the spam arms race to characterize spam evolution, in: Proceedings of the 7th Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS), Redmond, WA, 2010, July.

[83] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.

[84] T.G. Dietterich, Ensemble methods in machine learning, Lect. Notes Comput. Sci. 1857 (2000) 1–15.

[85] J.J.G. Adeva, U.C. Beresi, R.A. Calvo, Accuracy and Diversity in ECOC Ensembles of Text Categorizers, available: Retrieved on August 09, 2017 from, 2000, http://citeseer.ist.psu.edu/732806.html.

[86] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

[87] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, Ann. Stat. 38 (2) (2000).

[88] Y. Freund, R.E. Schapire, A Decision - theoretic generalization of on - line learning and an application to boosting, JCSS 55 (1997) 119–139.

[89] Singer Schapire, Improved boosting algorithms using confidence-rated predictions, Mach. Learn.: Mach. Learn. 37 (1999).

[90] A.A. Akinyelu, A.O. Adewumi, Classification of phishing email using random forest machine learning technique, J. Appl. Math. 6 (2016). Article ID 425731, Retrieved on July 12, 2017 from.

[91] L. Breiman, A. Cutler, Random Forests-Classification Description, Department of Statistics Homepage, 2007. http://www.stat.berkeley.edu/~breiman/RandomForests/cchome.htm.

[92] I. Fette, N. Sadeh, A. Tomasic, Learning to detect phishing emails, in: Proceedings of the 16th International World Wide Web Conference (WWW '07), 649–656, Alberta, Canada, May 2007, 2007.

[93] C. Whittaker, B. Ryner, M. Nazif, Large-scale automatic classification of phishing pages, in: Proceedings of the 17th Annual Network & Distributed System Security Symposium (NDSS '10), The Internet Society, San Diego, Calif., USA, 2010.

[94] L. Deng, D. Yu, Deep Learning: Methods and Applications, Now publishers, Boston, 2014.

[95] S. Zhao, Z. Xu, L. Liu, M. Guo, Towards Accurate Deceptive Opinion Spam Detection Based on Word Order-Preserving CNN, arXiv:1711.09181v1 [cs.CL] 25 Nov 2017, 2017, pp. 1–8. Available at, https://pdfs.semanticscholar.org/1687/0bed28831f6bd49a0228177351d1870fafd1.pdf.

[96] S. Albelwi, A. Mahmood, A framework for designing the architectures of deep convolutional neural networks, Entropy 19 (6) (2017) 242.

[97] R. Karthika, P. Visalakshi, A hybrid ACO based feature selection method for email spam classification, WSEAS Trans. Comput. 14 (2015) 171–177.

[98] C. Balakumar, D. Ganeshkumar, A data mining approach on various classifiers in email spam filtering, Int. J. Res. Appl. Sci. Eng. Technol. 3 (1) (2015) 8–14.

[99] A. Sharma, A. Suryawansi, A novel method for detecting spam email using KNN classification with spearman correlation as distance measure, Int. J. Comput. Appl. 136 (6) (2016) 28–34.

[100] W.A. Awad, S.M. Elseuofi, Machine learning methods for spam E-mail classification, Int. J. Comput. Sci. Inf. Technol. 3 (1) (2011) 173–184.

[101] S.P. Rajamohana, K. Umamaheswari, B. Abirami, Adaptive binary flower pollination algorithm for feature selection in review spam detection, in: IEEE International Conference on Innovations in Green Energy and Healthcare Technologies, 2017, pp. 1–4.

[102] I.J. Alkaht, B. Al-Khatib, Filtering SPAM Using Several Stages Neural Networks, Int. Rev. Comp. Softw. 11 (2016) 2.

[103] A.K. Sharma, S.K. Prajapat, M. Aslam, A comparative study between naïve Bayes and neural network (MLP) classifier for spam email detection, Int. J. Comput. Appl. (2014).

[104] A. Mousavi, A. Ayremlou, Bayesian Spam Classifier, Available online at, 2011, http://cs229.stanford.edu.

[105] K.R. Dhanaraj, V. Palaniswami, Firefly and Bayes classifier for email spam classification in a distributed environment, Aust. J. Basic Appl. Sci. 8 (17) (2014) 118–130.

[106] M. Choudhary, V.S. Dhaka, Automatic E-mails classification using genetic algorithm, in: Special Conference Issue: National Conference on Cloud Computing and Big Data, 2013, pp. 42–49.

[107] C. Palanisamy, T. Kumaresan, S.E. Varalakshmi, Combined techniques for detecting email spam using negative selection and particle swarm optimization, Int. J. Adv. Res. Trends Eng. Technol. 3 (2016). ISSN: 2394-3777.

[108] J.N. Shrivastava, M.H. Bindu, E-mail classification using genetic algorithm with heuristic fitness function, Int. J. Comput. Trends Technol. 4 (8) (2013) 2956–2961.

[109] M. Zavvar, M. Rezaei, S. Garavand, Email spam detection using combination of particle swarm optimization and artificial neural network and support vector machine, Int. J. Mod. Educ. Comput. Sci. (2016) 68–74.

[110] I. Idris, A.S. Muhammad, An improved AIS based E-mail classification technique for spam detection, in: Proceedings of the Eight International Conference on eLearning for Knowledge-Based Society, Thailand, 2012.

[111] J.N. Sosa, Spam Classification Using Machine Learning Techniques – Sinespam, Master of Science Thesis, 2010. Master in Artificial Intelligence (UPC-URV-UB).

[112] D.K. Renuka, P. Visalakshi, T. Sankar, Improving E-mail spam classification using ant colony optimization algorithm, Int. J. Comput. Appl. (2015) 22–26.

[113] G. Bhagyashri, H. Pratap, Auto E-mails classification using bayesian filter, Int. J. Adv. Technol. Eng. Res. 3 (4) (2013).

[114] P. Graham, A Plan for Spam, Retrieved on August 07, 2017 from, 2002, http://www.paulgraham.com/spam.html.

[115] W. Zhao, Z. Zhang, An email classification model based on rough set theory, in: Proceedings of the 2005 International Conference on Active Media Technology, 2005, 2005 (AMT 2005).

[116] S. Kumar, S. Arumugam, A probabilistic neural network based classification of spam mails using particle swarm optimization feature selection, Middle East J. Sci. Res. 23 (5) (2015) 874–879.

[117] Akshita Tyagi, Content Based Spam Classification- A Deep Learning Approach, A Thesis Submitted To The Faculty Of Graduate Studies, University Of Calgary, Alberta, Canada, 2016.

[118] M. Ott, Y. Choi, C. Cardie, J.T. Hancock, Finding deceptive opinion spam by any stretch of imagination, in: ACM Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 1, 2011, pp. 309–319.
[119] M. Barreno, B. Nelson, R. Sears, A.D. Joseph, J.D. Tygar, Can machine learning be secure?, in: Proceedings of the 2006 ACM Symposium on Information Computer and Communications Security, Taipei, Taiwan, 2006, pp. 16–25.
[120] B. Nelson, M. Barreno, F.J. Chi, A.D. Joseph, B.I.P. Rubinstein, U. Saini, C. Sutton, J.D. Tygar, K. Xia, Exploiting Machine Learning to subvert your spam filter, in: Proceedings of the 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats, San Francisco, California, 2008, 2008, pp. 1–9.
[121] M.A. Barreno, Evaluating the Security of Machine Learning Algorithms, EECS Department, University of California, Berkeley, 2008.
[122] X. Yuan, P. He, Q. Zhu, R. RanaBhat, X. Li, Adversarial Examples: Attacks and Defenses for Deep Learning, rXiv:1712.07107v2 [cs.LG] 5 Jan 2018, 2018. Retrieved from, https://arxiv.org/pdf/1712.07107.pdf.
[123] G. Bandana, Design and Development of Naïve Bayes Classifier, North Dakota State University of Agriculture and Applied Science, Graduate Faculty of Computer Science, 2013. Master thesis.
[124] G. Holmes, G. Pfahringer, B. Kirkby, R. Frank, E.M. Hall, Multiclass Alternating Decision Trees, ECML, 2002, pp. 161–172.
[125] S.M. Lee, D.S. Kim, J.H. Kim, J.S. Park, Spam detection using feature selection and parameters optimization, in: 2010 International Conference on Complex, Intelligent and Software Intensive Systems, 1, 2010, pp. 883–888.
[126] M.N. Marsono, M.W. El-Kharashi, F. Gebali, Targeting spam control on middleboxes: Spam detection based on layer-3 e-mail content classification, Elsevier Computer Networks, 2009.
[127] P. Sahil, G. Dishant, A. Mehak, K. Ishita, J. Nishtha, Comparison and analysis of spam detection algorithms, Int. J. Appl. Innov. Eng. Manag. (IJAIEM) 2 (4) (2013) 1–7.
[128] D. Sculley, G. Wachman, in: W. Kraaij, A.P. de Vries, C.L.A. Clarke, N. Fuhr, N. Kando (Eds.), Relaxed Online SVMs for Spam Filtering, SIGIR, ACM, 2007, pp. 415–422.
[129] E.G. Dada, E.I. Ramlan, pdPSO: the Fusion of Primal-Dual interior point method and particle swarm optimisation algorithm, Malays. J. Comp. Sci., Malaysia 31 (1) (2018) 17–34.
[130] B. Schölkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and beyond, MIT press, 2002.
[131] B. Schölkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and beyond (Adaptive Computation and Machine Learning), 2001, 2001.
[132] B. Yu, Z. Xu, A comparative study for content-based dynamic spam classification using four Machine Learning algorithms, Knowl. Based Syst. 21 (4) (2008) 355–362.
[133] Y. Feng, H. Zhou, An effective and efficient two-stage dimensionality reduction algorithm for content-based spam filtering, J. Comput. Inf. Syst. 9 (4) (2013) 1407–1420.
[134] Priyanka Chhabra, Rajesh Wadhvani, Sanyam Shukla, Spam filtering using support vector machine, Special Issue of IJCCT 1 (2) (2010) 3–4, for International Conference [ACCTA-2010], 3-5 August 2010. Pp. 166-171.
[135] Jason Brownlee, Master Machine Learning Algorithms, Discover How They Work and Implement Them From Scrath, 2019. Available at: https://machinelearningmastery.com/master-machine-learning-algorithms/.
[136] Nurul Fitriah Rusland, Norfaradilla Wahid, Shahreen Kasim, Hanayanti Hafit, Analysis of Näıve Bayes algorithm for email spam filtering across multiple datasets, IOP Conf. Ser. Mater. Sci. Eng. 226 (2017), 012091.
[137] L. Pelletier, J. Almhana, V. Choulakian, Adaptive filtering of spam, in: Second Annual Conference on Communication Networks and Services Research (CNSR'04), 2004.