Soft Computing
https://doi.org/10.1007/s00500-019-04473-7

METHODOLOGIES AND APPLICATION

Machine intelligence-based algorithms for spam filtering on document labeling

Devottam Gaurav[1] · Sanju Mishra Tiwari[2] · Ayush Goyal[3] · Niketa Gandhi[4] · Ajith Abraham[5]

Abstract
The internet has provided numerous modes for secure data transmission from one end station to another, and email is one of those. The reason behind its popular usage is its cost-effectiveness and facility for fast communication. In the meantime, many undesirable emails are generated in a bulk format for a monetary benefit called spam. Despite the fact that people have the ability to promptly recognize an email as spam, performing such task may waste time. To simplify the classification task of a computer in an automated way, a machine learning method is used. Due to limited availability of datasets for email spam, constrained data and the text written in an informal way are the most feasible issues that forced the current algorithms to fail to meet the expectations during classification. This paper proposed a novel, spam mail detection method based on the document labeling concept which classifies the new ones into ham or spam. Moreover, algorithms like Naive Bayes, Decision Tree and Random Forest (RF) are used in the classification process. Three datasets are used to evaluate how the proposed algorithm works. Experimental results illustrate that RF has higher accuracy when compared with other methods.

Keywords Machine learning · Spam detection · Document labeling · Feature selection

1 Introduction

Email system serves as one of the most powerful communication systems for transmitting the user's information from one to another. It includes not only text but also images, files, etc. This platform helps the user in saving a huge amount of time and money in comparison with outdated techniques like telegrams, etc. Nowadays, approximately 281 billion emails are transmitted all over the world in our day-to-day life (https://cacm.acm.org/magazines/2018/7/229047-youve-got-mail/fulltext?mobile=false). One of the threats to such kind of platform is spam, where thousands of unwanted and unintended emails are transmitted daily. Spams are nothing but unwanted and unintended emails that are transmitted to other users having no prior permission. It may exist in various ways like unwanted and unwarranted advertising of products or services. In recent years, there has been tremendous growth in the volume of spam. In a similar way, authorized, perfect and flawless emails are called ham (Sharaff et al. 2016; Herrero et al. 2009).

Communicated by V. Loia.

✉ Sanju Mishra Tiwari
tiwarisanju18@ieee.org

Devottam Gaurav
gauravpurusho@gmail.com

Ayush Goyal
ayush.goyal@tamuk.edu

Niketa Gandhi
niketa@gmail.com

Ajith Abraham
ajith.abraham@ieee.org

[1] Department of Computer Science and Engineering, Chandigarh University, Punjab, India

[2] Ontology Engineering Group, Universidad Polytecnica de Madrid, Madrid, Spain

[3] Department of Electrical Engineering and Computer Science, Texas A&M University - Kingsville, Kingsville, TX, USA

[4] University of Mumbai, Mumbai, India

[5] Machine Intelligence Research Labs (MIR Labs), Auburn, WA 98071, USA

Published online: 02 November 2019

As per the study estimations, more than 70% of the emails are flawed (Nizamani et al. 2013). These are done for monetary benefit, advertisement, expanding the activity to malevolent sites, stacking contents, etc., by the spammers. This, in turn, leads the service provider to pay a huge amount of money. The cost may be assessed in terms of time spent by the user in going through these mails and removing it from the inboxes. These may also reduce the efficiency of the network with a way to harm the ham records, swamps the important data due to limited space, loss of network's bandwidth, etc. Due to these, users are forced to buy some extra space for a particular time period.

To overcome from this problem, a spam filtering method is used for classifying the emails into ham/spam. This technique is mainly divided into two types: The first one is knowledge engineering and machine learning method (Christina et al. 2010; Chebrolu et al. 2005; Gaurav et al. 2019). The first one contains few predefined protocols which are used for classifying the emails into ham/spam along with the address of a network, while the second one produces original data or resolves the issues by analyzing the data and to foresee future patterns. In other words, it produces models that can examine greater, progressively complex information and convey quicker, increasingly exact outcomes—even on an extremely vast scale. Thus, with the help of machine learning an unidentified threat in a superior way, after comparing the two methods, the latter one is much more efficient than the former one because there is no involvement of rules in it. Hence, in this paper, a machine learning technique is used.

The paper is organized with more details of the prior research work along with filtering techniques listed in Sect. 2. Section 3 highlights the proposed work. The mining of many textual features that are carried out along with the result is shown in Sect. 4. Section 5 illustrates the comparisons carried out between different datasets. At last, the conclusion is provided in Sect. 6.

2 Related works

The problem of spam emails has become a major concern nowadays. Many efforts have been made by eminent researchers to make a classification of emails into the spam/ham category along with their detection rate. They have carried out their tasks using machine learning. To solve these problems, the authors have also carried out the task of classification with the help of a new spam filtering technique with their rate of detection. Table 1 shows the relative work carried out by different eminent researchers along with the dataset used, methods carried out, extraction of features and results obtained. This literature part contains two parts: The first part is related to the detection of spam emails, and the second part is related to the classification of spam emails.

Sarwat et al. (2014) have presented their work for the detection of fraudulent emails with the help of cluster-based classification model (CCM). The size of the dataset used was 8000. This dataset was downloaded from the Nigeria Web site (Radev 2008). To carry out this classification task, various classification algorithms are used like

Table 1 Comparative study

Authors	Dataset	Methods	Feature selection	Results
Sarwat et al. (2014), Radev 2008)	8000 email	NB, SVM, DT, CCM	TF-IDF	Accuracy is 96%
Trivedi and Dey (2013)	Enron dataset	NB, Bayesian, bagging, boosting with re-sampling, AdaBoost	Genetic search method, term documentary matrix	Accuracy for Bayesian classifier is 92.99%
Bhat et al. (2014)	Dataset taken are from FB	NB, DT, k-NN, bagging and boosting.	Ensemble method	DT has performed better than others
Bassiouni et al. (2018)	Spambase UCI	RF, ANN, logistic regression, SVM, random tree, k-NN, decision table, Bayes Net, NB, RBF	ILFS	RF works better than others with accuracy = 95.45%
Youn and McLeod (2007)	Size varies from 5000 to 10,000	SVM, NB, DT	TF-IDF	DT performs better than others
Merugu et al. (2019)	Dataset taken from UCI which is 5574	SVM, RF, k-NN and NB are used with weighting method	TF-IDF	NB outbursts well than others with an accuracy of 97.6%

Naive Bayes (NB), Decision Tree (DT), Support Vector Machine (SVM) and CCM.

In the stage of feature extraction, data were preprocessed; features were extracted by TF-IDF method and were shown in the vector form. These featured vectors were indicated with ham and spam. After the extraction of features, numerous tests were carried out with diverse sets of features and classification methods. The accuracy achieved by the authors was 96%. Trivedi and Dey (2013) have carried out the process of detection for spam emails with the help of two probabilistic classifiers such as NB and Bayesian. In respect of these two, three boosting algorithms were used here like bagging, boosting with resampling and AdaBoost. The experiments were carried out with three types of Enron datasets (i.e., Enron 4, Enron 5 and Enron 6). The lengths of each dataset were 6000. After preprocessing data, 375 useful features were extracted with genetic search method and term documentary matrix over 1359 attributes. The accuracies are lying between 88.1 and 92.9% when a genetic search method was used and Bayesian performs better than NB. In a similar way, Bhat et al. (2014) have carried out the task for spam detection with bagging and boosting technique. The size of the datasets includes 63,891 Facebook users. After preprocessing of data, the features were extracted from an online social network with the help of an ensemble technique. Numerous classification algorithms like NB, DT and k-NN (k-Nearest Neighbor) were used to observe their accuracies on WEKA tool. Finally, the conclusion is that DT outbursts better than others.

Bassiouni et al. (2018) have performed an experiment on classifying the spam emails from the ham mails. The Spambase UCI dataset was used behind this purpose. The size of the dataset is 4601.

After performing preprocessing of data, features were selected through Infinite Latent Feature Selection (ILFS) with the help of 10 classifiers. The names of these classifiers are RF, ANN, logistic regression, SVM, random tree, k-NN, decision table, Bayes Net, NB and RBF. Their respective accuracies are 95.4, 92.4, 92.4, 91.8, 91.5, 90.7, 90.3, 89.8, 89.8 and 82.6. RF outbursts better than the other classifiers in terms of accuracy, i.e., 95.45%. Similarly, Youn and McLeod (2007) have performed the experiment on the classification of spam emails with four different types of classifiers like neural network (NN), NB, SVM and J48. Various sizes of datasets were used for this experiment like 1000, 2000, 3000, 4000 and 4500. After performing preprocessing on data, features were selected on the basis of the TF-IDF method. Finally, the conclusion was that J48 outbursts among all. Merugu et al. (2019) have presented the classification task into spam and ham category on the basis of the weighting method. This method helps in easy classification of new emails into ham/spam based on the strength of words present in different categories.

The dataset used was collected from the UCI machine learning repository having a size of 5574. When data were cleaned, features were extracted through the TF-IDF method. This feature extraction method also uses the BoW model. NB outbursts well in the presence of others like RF, SVM and k-NN with an accuracy rate of 97.6%.

After observing from both the parts of related work, the conclusion is that spamming has become a serious issue in nowadays where efforts are being made in numerous ways using the different proposed methods of eminent researchers.

2.1 Methods for spam filtration

This section discusses the summary of different types of approaches used for filtering the spam, for example, classifying the methods based on techniques and classifying the methods based on theoretical approaches, etc. The spam filtering method tries to help in reducing or preventing the growth of a large number of unintended emails. These unintended emails are sometimes called as unsolicited commercial emails (UCE). These are needed to overcome the false effects of spam as:

(i) Spam causes inconvenience and squanders clients' an ideal opportunity to frequently check and erase this an extensive number of undesirable messages.
(ii) Spam may contain unequivocal content or pernicious code which includes infections, rootkits, worms, Trojans or another sort of harming programming and so forth.
(iii) Spam has moral issues like publicizing deceitful advertisements (for instance profit speedy), violent content, (for example, obscene pictures and grown-up material) that are inconvenient to the youthful ages.
(iv) When mailboxes are flooded with spam emails, then, it may lead to loss of important documents, thereby server becomes overloaded, gets delayed server response and chomps more wastage of space.

Various spam filtering methods are as follows:

(i) *Avoidance of distributed spammed email at the source* Botnets are the major source for spreading of false information in the cyber area. In other words, when a group of machines are connected to the Internet, it becomes part of botnets. Thus, it becomes a major issue for those whose computers are hijacked. When a computer acts like a zombie, it is being controlled by the hacker. Numerous activities are performed by them like diffusion of obscene material or different non-attractive activities, leaving no follows on your PC, etc.

All spammed emails are disseminated by zombie computers because 50% of the spam emails that are sent throughout the world are done by the zombie computers. The second way of distributing spam emails is done through the distributed denial of service attack (DDoS). It happens when a substantial number of Internet clients make concurrent solicitations to a site server, in order to counteract authentic clients to approach the site.

The zombies send a staggering measure of pointless data to the site's routers that they are not ready to process thus the network splits. To limit the distribution of spammed email in their source, email protocols are being developed continuously, email servers are jammed that creates spamming.

(ii) *Avoidance of Spammed email acceptance at the endpoint*

To stop false information coming at the endpoint, there are two ways:

- Theoretical methods are further divided into three parts: traditional, based on learning and hybrid techniques.
- Based on filtering, it is divided into two parts: user and server sides, respectively.

(a) *Traditional techniques* This technique uses the defined data given by the professionals to carry out the classification process. The stored data given by the professionals are called information-based data.

It helps in reducing the substantial quantity of negative procedures when it is a slice of email filtering method during classifying the data. This technique is further divided into different parts:

- *Technique used in analyzing the message* Analyzing the time-honored emails is done on the account of official signature, same signatures are used as per the updations are done in the database, statistical techniques are used for accessing the data based on Naive Bayes theorem. These are done just for checking the signatures of spammed emails.
- *Detecting the distributed mass* The purpose of using this technique is to notice the same kind of mail spreading to a large number of clients. The below ones are for this purpose:

- Votes given by the clients (Razor/Pyzor) (http://razor.sourceforge.net/; https://sourceforge.net/p/pyzor/mailman/pyzor-announce/).
- Analyzing the emails coming from some network (http://umanitoba.ca/computing/ist/email/exchange/securityspamindex.html).
- Generation of acknowledgment for trapping the spam (https://www.symantec.com/products/mail-security-exchange).

For mass identification of emails, the possibility of this technique is to use spam filtration for verifying the determined email signature. For the strategies, in view of recognition of reiterations, two fundamental problems are noticed. One is an unwanted exemplification and the other one is a discovery of genuine mass mailing. The former implies that every spam email has inconsequential contrasts. Due to this, it is difficult to gather unfaltering signs. To take care of this issue, the different relentless signs are utilized. For instance, in Yandex Mail System the strategy for shingles (http://company.yandex.ru/public/articles/antispam.xml) is figured it out.

- *Detecting the sender as spammers* These strategies depend on various blackhole arrangements of IP and addressed email. It is conceivable to use personal blackhole and white records or to utilize the service of RBL (real-time blackhole list) and DNSBL (DNS-based blackhole list) to verify the addresses. The merit in using this technique is that it helps in recognizing the spam at an early stage when the email is being received. The demerit is that the strategy used for adding and deleting the email address isn't translucent.
- *Verifying the source mail address and the name of the domain* This is one of the simple techniques of filtering the irrelevant data, where DNS's name is checked with the name of the sender's domain. This method becomes useless when genuine addresses are utilized by spam

ones. For this situation, it might be confirmed with a probability of sending the message from the current IP address. Initially, the sender's ID (http://www.microsoft.com/mscorp/safety/technologies/senderid/default.mspx) can be utilized where the sender's email address is shielded from misrepresentation by methods for distributing the arrangement of space name used in DNS. Moreover, in SPF (Sender Policy Framework) (http://www.openspf.org/Introduction), DNS convention is utilized for confirming the source's email address. The guideline conveys that if the proprietor of the domain needs SPF confirmation, then admittance of DNS to that domain may be added.

All the above strategy discussed depends on a few information for analyzing the data gathered by specialists of outsider providers who suffer from certain demerits:

- It is important to refresh the learning base routinely;
- It totally depends on updating information gathered by providers;
- The very low amount of security is guaranteed;
- Totally dependent on the corresponding natural language;
- Amount of detecting false data is very low.
- *Learning-based techniques* To overcome the problem of spreading false information, filtration is one of the solutions. It tends to work in an automated way of classifying the email into ham or spam. The existed proposed work shows an accuracy level of more than 91% (for instance, the assessment performed by Vidya Kumari and Kavitha (2019)). The algorithms used to filter the spam data from emails can be applied in various stages while transmitting the emails from one point to another, i.e., at routers, at the end server or at the end email box. When filtration is done at the endpoint server or mailboxes, then, an only partial part of the problem is solved, i.e., it helps in reducing the wastage of time done by the clients in screening the spam emails but it doesn't prevent from misusing the data. This happens because every one of the email is conveyed already to the destined server.

These are further divided into three main categories:

- *Detection of Spammed Images* Spamming in the image has been progressed toward becoming another sort of email spam. Spammers insert the message into the picture and then send a mail with that attachment.

In this situation, traditional techniques become insufficient. Filtration of an image proves to be an expensive and tedious work. Liu (Liu et al. 2010) carried out the process of filtering the image in a three-layered architecture. Layer 1 contains Mail Header Classifier; layer 2 contains the Image Header Classifier and layer 3 contains the Visual Feature Classifier. In layer 1, headers of emails are taken out first from the received emails. After making an analysis, features related to headers are taken out with the help of feature mining unit. Filtration is further carried out on the obtained features with the help of Bayesian classifier. If the result of Bayesian classifier is more than the threshold (let say T1), then the method gives the end result else the end result is further utilized in the next layer. If the result of layer 1 is lesser than the threshold obtained in layer 2 (let say T2), then the result of layer 1 is ignored else the end result of layer 2 is considered as final. If not, then the result of layer 2 is used in layer 3.

- *Detection of Bagged words* This model is utilized in NLP (natural language processing) and also in data mining. Here, textual data are denoted in the form of collected words, but not in order, ordered words and more in improper grammar. To carry out the filtration of spammed emails, two bagged words are taken. One pack is loaded up with spammed words that are there in unwanted emails, and the other pack is loaded up with authentic words that are there in ham mails. Considering email as a heap of words from one of these packs, Bayesian probabilities are utilized to decide to which pack this email has a place.
- *Detections of collaborative spams* This process mainly relies upon client produced marked data. Clients give input by categorizing mails as spam or ham. These categories are then used for training the spam filters. In spite of the fact that the information provided by the clients is extremely less, they are collected together for making the training size very huge. As a result, there is a considerable deviation in the clients' thoughts that which one belongs to spam or ham category. Therefore, spam filters when used on a worldwide classifier will be problematic. On the other hand, it becomes very tough to train the individual classifier with little information for all clients (Ahuja 2018)

Attenberg et al. (2009) carried out the task of collaborative spams successfully with the help of hashing trick on a customized worldwide classifier. The hashing trap maps every single individual classifier into a solitary low-dimensional space, where training is done with a weight vector which catches the individual facets of every client.

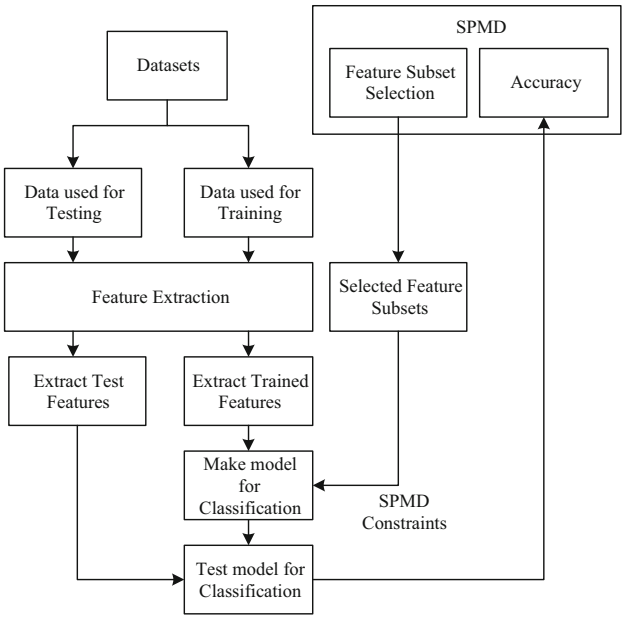

Fig. 1 Layout of proposed method

(c) *Hybrid-based techniques* Yoon et al. (2010) have proposed a hybrid filter for classifying the text messages into spam or ham data. This is done with the help of content-based filter and challenge-response protocols. To carry out such a process, three types of regions are considered: normal, uncertain and spam. To classify the messages in normal and spam, challenge-response protocols are used for uncertain messages. To check whether the source is a single user or a device, a thought-provoking question is sent by the message center. Upon receiving the response from the destination, matching takes place between correct ones and response. If there is a match, then, it is a normal text, else it is a spam one. For classifying the uncertain texts, a thought-provoking question is sent by the source to the destination in the form of CAPTCHA. If it is a legal one, then the correct response is given otherwise not.

3 Proposed work

3.1 Layout of proposed work

This section deals with the layout of the proposed work. To carry out the steps required in the algorithm further, five different stages are needed. In the first phase, the datum is collected from three different datasets like Enron dataset (http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html), Ling-Spam dataset (http://www.aueb.gr/users/ion/data/lingspam_public.tar.gz.2019) and PU dataset (http://www.aueb.gr/users/ion/data/PU123ACorpora.tar.gz.2019). In the next phase, the cleaning of data is performed using preprocessing. After cleaning of data, a spam filter is made to remove the unrelated features. When filtration is done, TF-IDF (Kim et al. 2019; Camastra et al. 2013) is used for extracting the features. After the extraction phase, features are selected with the help of different techniques like SelectKBest(), exhaustive feature search (EFS), etc. The purpose of feature selection technique is to select the most relevant one from the set of features on the basis of possibilities. In the end, three classifiers are used to classify the emails into ham/spam.

They are Naive Bayes (NB), Decision Tree (DT) and Random Forest (RF) (Staiano et al. 2013). These classifiers work on only labeled instances and use unlabeled instances for testing purpose. If the possibility of any feature is soaring, then it is mapped in the labeled group along with the predicted label.

These are continued till all the test instances are classified in one of the categories (spam or ham). The layout of the proposed methodology is shown in Fig. 1.

3.2 Algorithm 1 (SPMD)

Based on the gap identified in the literature, a new methodology is proposed based on the spam detection concept. The proposed method is named as spam mail detection (SPMD) method which is based on supervised learning (SL) approach. In the view of SL, it is expected that instances that belong to one label are created from one group and it also helps in providing an improved feature of labeled data from that of unlabeled instances. To carry out the process of spam filtration, text/document is labeled. This is done for mining the texts in a proper way. This process is called as document labeling. In the existing spam filtering methods, there is no automatic way available that can be utilized to allocate labels to an expansive number of texts and updating the model for classification concurrently. So, it requires a huge amount of human work which consumes a lot of time. To automate this process, a label for a particular class is consequently decided when a new dataset is made. This is done with the help of machine learning techniques. The proposed Algorithm 1 is given:

Algorithm 1 Spam Mail Detection (SPMD):

INPUT:

Labeled dataset corpus $Dt = \{dt_1, dt_2, \ldots, dt_m\}$, where m is the length of dataset, dt is the respective document present in Dt.

$i, j. k$: The positional parameters, where $\forall_{i,j,k} \in 1,2, \ldots, m$.

T: Set of tokens, where $T = \{t_1, t_2, \ldots, t_i\}$.

G: Set of messages, where $G = \{g_1, g_2, \ldots, g_j\}$.

L: Category set, where $L = \{l_1, l_2, \ldots, l_k\}$, $l_k = l_h + l_s$, l_s is a spam indicator and l_h is a ham indicator.

F: Set of features, where $F = \{f_{11}, f_{12}, \ldots, f_{ji}\}$, f_{ji} is the independent feature from the selected feature.

d_c: Combined data.

dt_c: The clean data.

D_{tr} : Training data$(70\% \; of \; d_c)$

D_{ts} : Test data $(30\% \; of \; d_c)$

N:Maximum times the term appears in Dt.

Y: Total amount of terms present in Dt.

Z: Total amount of occurrence of dt.

A : Set of algorithms, where $A = \{a_1, a_2, \ldots, a_m\}$.

K : Number of folds for cross validation.

Im_ACC : Improved Accuracy after cross validation is applied.

Fb:Final probable features calculated for each term tf.

$Pred$: The predicted labels.

Acc: The accuracy of algorithm.

OUTPUT: Classify the data as ham or spam.

$Dt \leftarrow Load(Dt)//load\ the\ datasets$
for $i, j, k = 1$ *to* m **do**
  for each *document* $dt_m \in Dt$ **do**
    for each *token* $t_i \in T$ **do**
      for each *message* $g_j \in G$ **do**
        for each *category* $l_k \in L$ **do**
          if $\left((t_i \in g_j) \&\& (g_j \in dt_m) \&\& (dt_m \in l_k)\right)$ **then**
            if $(l_k \in l_h)$ **then**
              $t_i \leftarrow Append(t_i, l_h)//l_h = 1$
            else
              $t_i \leftarrow Append(t_i, l_s)//l_s = 0$
          else
            continue
        end for of category
      end for of message
    end for of token
  end for of document
$d_c \leftarrow Combine\_Data(T, L)$
$dt_c \leftarrow Preprocess\_Text(d_c)$
$D_{tr}, D_{ts} \leftarrow Split(dt_c)$
for each *document* $dt_m \in Dt$ **do**
  for each *term* $tf_j \in D_{tr}$ **do**
    $tf_j = \sum_j^m \left(\frac{N}{Z}\right)$
    $idf_i = \log\left(\frac{N}{Y}\right)$
    $f_{ji} = \sum_{ji}^m \left(tf_j \times idf_i\right)$
    $F = \sum_{ji}^m f_{ji}$
  end for of term
$Fb \leftarrow Compute\_Prob(F)$
end for of document
for each *algorithm* A **do**
  $A.Fit(D_{tr}, Fb)$
  $Pred \leftarrow Predict(D_{ts})$
  $Acc \leftarrow Accuracy\_Score(Pred)$
  $Im\_ACC \leftarrow Cross\_Validation(D_{tr}, D_{ts}, K)$
end for of algorithm
end for

Algorithm 1 Description

Given dataset $Dt = \{dt_1, dt_2, \ldots, dt_m\}$, where dt_m is the mth document in Dt and L denotes the category set. The range of positional parameters $i, j.k$ is $\forall_{i,j,k} \in 1, 2, \ldots, m$ where m is the length of the dataset Dt. When a new document arrives, it is first broken down into tokens, where each token consists of the message in it and each message contains words in it. The tokens are represented by T, where $T = \{t_1, t_2, \ldots, t_i\}$. Each dt_m contains a set of jth messages, where $G = \{g_1, g_2, \ldots, g_j\}$ which are further combined with its own label to form a proper structure. The main work of ML is to make a Boolean categorization function with a technique used for filtration of spam in an automatic manner as: $\emptyset_G(dt_m) : Dt \rightarrow \{True, False\}$. Each of the messages ($G \in dt_m$) is assigned to anyone, instead of two. The messages G are referred to spam if $\emptyset_G(dt_m)$ has an indicator l_s else it is a ham message. Each ham message is denoted as l_h. With the help of prior representation, supervised ML algorithms take the following stages for filtration of spam as given below:

3.3 Data preprocessing

The representations of the email's contents are done with the help of featured vectors, i.e., message g_j is in the category l_k or not. When such vectors are united together for dataset collection, then these datasets are referred to as labeled datasets. Because of the huge number of email files, the subsequent datasets result in enormous and scanty structures. For these issues, some methods are used to reduce the dimensions of vectors before classifying it. To further reduce the dimensions in a better way, it can be also be done with the use of tokenization, stop word removal and lemmatization. After preprocessing of data, features are extracted by using the feature extraction method in the next stage.

3.4 Feature extraction

Feature extraction is an approach to choose a subset of unique features space. The quantity of feature that is in the space hampers the time used for computation as well as the accuracy of the classifier. The main theme for doing this is to do searching for a viable number of featured subsets by assessing it with proper features along with training data. To carry out such process, a spam filter is constructed. The general representation of spam filter is shown in Eq. (1):

$$\emptyset(G, \varphi) = \begin{Bmatrix} ham & \text{if } l_h = 1 \\ spam & \text{if } l_s = 0 \end{Bmatrix} \quad (1)$$

where φ is the vectored constraints. The function $\emptyset$ is used to specify whether a given message G is ham or spam. The purpose of spam filter is to remove the irrelevant features from the labeled datasets. These labeled datasets are grouped together with a frequent term tf_j and frequent document idf_i to form a TF-IDF (F) feature vector as in Eq. (4):

$$tf_j = \sum_{j}^{m} \left(\frac{N}{Z}\right) \quad (2)$$

$$idf_i = \sum_{i}^{m} \log\left(\frac{N}{Y}\right). \quad (3)$$

$$f_{ji} = \sum_{ij}^{m} (tf_j \times idf_i) \quad (4)$$

where N is the amount of term that has appeared in the dataset, z is the total amount of terms in the given dataset Dt, $F = \{f_{11}, f_{12}, .., f_{ji}\}$ and Y is the amount of document that has occurred. Equations (2) and (3) show the computation done for tf_j and idf_i, respectively. In the next stage, the selections of features are done from the extracted features.

3.5 Feature selection

It is the way toward choosing a subset of important features, which helps in constructing the model. It also helps in picking features that provide a great or better exactness while requiring less information. This technique can be utilized to distinguish and to expel unneeded and repetitive qualities from information that don't make any contribution to the model's exactness, otherwise it may certainly decline the exactness of the model. To make the model work in an easier manner, a labeled feature matrix (in Fig. 2) is created. This is simply a cross-product of $(F \times L)$ along with their own categories L as:

Each feature in the feature matrix consists of word and their labels for each and every data as given in Fig. 3.

From the feature matrix, appropriate features are selected using the selection technique given in Table 2. Let *ham_cnt* is the total number of ham labeled mails, *spam_cnt* is the total number of spam labeled mails, T be

Fig. 2 Labeled feature matrix

$$\begin{bmatrix} F_1 \cdots F_m & \\ f_{11} \quad \cdots \quad f_{1i} & l_1 \\ \vdots \quad \ddots \quad \vdots & \vdots \\ f_{j1} \quad \cdots \quad f_{ji} & l_k \end{bmatrix}$$

$$\{ID_1, ham, (word_1, l_1), (word_2, l_2), (word_3, l_3), \ldots\}$$
$$\{ID_1, spam, (word_1, l_1), (word_2, l_2), (word_3, l_3), \ldots\}$$

Fig. 3 Feature data

Table 2 Feature selection schemes

Feature selection scheme
Lack of selected features
Information gain
SelectKBest, Chi-square
Exhaustive feature search EFS)
Best first search (BFS)
GridSearchCV
Sequential backward Selection (SBS)
GreedyStepwise

the total number of mails, w is the word, $P(H)$ is the probability for ham feature and $P(S)$ is the probability for spam feature. This is clearly shown in Eq. (5) and Eq. (6), respectively.

$$P(H) = \left(\frac{ham\_cnt}{T}\right) \tag{5}$$

$$P(S) = \left(\frac{spam\_cnt}{T}\right) \tag{6}$$

After this, a conditional probability is applied. This is done to know out of total (ham or spam) feature, how much is the actual (ham or spam) word? Ham data(H) and word (w) are two independent events. These are shown in Eqs. (7), (8), (9) and (10):

$$P(H \cap w) = P(H) \times P(w) \tag{7}$$

$$P(S \cap w) = P(S) \times P(w) \tag{8}$$

$$P\left(w/H\right) = P(H \cap w)/P(H) \tag{9}$$

$$P\left(w/S\right) = P(S \cap w)/P(S) \tag{10}$$

Equations (11) and (12) are just used for summing the values finally.

$$P(H) = \sum \left(P\left(w/H\right) \times P(H)\right) \tag{11}$$

$$P(S) = \sum (\times P(S)) \tag{12}$$

When the probability is calculated for each ham and spam data, a comparison is done. To carry out the comparison process, a score is computed. This is done for ranking the features in a way to select the best feature among all. The model is represented as from Eq. (13):

$$\emptyset_F(P(H), P(S)) = \left\{ \begin{array}{ll} 1 & \text{if } (P(H) > P(S)) \\ 0 & \text{if } (P(S) > P(H)) \end{array} \right\} \tag{13}$$

If $(P(H) > P(S))$, then score is 1 otherwise score 0 is given. Higher the rank, higher will be the probability for selecting an optimal feature. Once a choice of selection is done, it greedily grabs the optimal feature from the trained features and removes the unrelated features. This feature selection strategy can be renamed as "**Greedy Selection Strategy**." In this way, all relevant features are selected from the extracted features. The features that are selected from the training part are called as trained features and for testing part are called as test features.

In this research, we are focused on novel feature selection methods. Principal component analysis (PCA) is a well-explored feature reduction method that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. Oliveira has illustrated the use of PCA on spam database (Oliveira 2019). Park and Klabjan (2018) also explored conventional and variants PCA on spam dataset. We have added a short discussion in Sect. 3.5. Our goal was to optimize the classification results using feature selection methods, and the proposed "**Greedy Selection Strategy**" can carry out the training process in a faster way and yields better result.

These selected/trained features are then converted into an understandable form for algorithms. Finally, an algorithm runs over the classifiers during classification stage.

3.6 During classification

To carry out the process of classification in an automated manner, a learning algorithm is built for the machine where it tries out to search for patterns from the data. This learning algorithm is called a model. For that a classifier is constructed. The purpose of creating a classifier is to classify the data to their respective labels or making the prediction for the data. These are done only when the set of algorithms $A = \{a_1, a_2, \ldots, a_m\}$ are applied on it. The classifier model is represented in Eq. (14) as:

$$\emptyset(A) = \left\{ \begin{array}{ll} ham & \text{if } (Pred_h = l_h) \\ spam & \text{if } (Pred_s = l_s) \end{array} \right\} \tag{14}$$

where Pred_h is the predicted label for ham data and Pred_s is the predicted label for spam data. After prediction, accuracy for the classifier is calculated based on the formula given in Eq. (15):

$$A_c = \left(\frac{Sp_c + Hp_c}{T_{em}}\right) \tag{15}$$

where A_c is the entire number of spammed mails that classified correctly; Hp_c is the entire number of hammed mails that classified correctly and T_em is the whole messages.

Table 3 Experimental results*

Methods	E			LS			PU		
	NB	DT	RF	NB	DT	RF	NB	DT	RF
Info gain	67.03	84.27	91.50	77.41	83.21	91.30	56.61	77.32	88.24
SBS, Greedy	56.36	83.88	91.77	82.28	82.75	91.40	56.99	76.57	87.73
SFS, GridSearch	56.00	84.49	91.67	82.79	83.41	91.93	57.10	76.47	87.02
EFS, BFS	55.13	85.54	92.78	*83.29*	*84.90*	*92.97*	56.90	76.24	89.20
SelectKBest	65.90	83.84	91.81	78.57	83.21	92.97	57.93	76.85	87.72
WFS	67.13	81.98	91.72	79.03	82.25	92.56	57.13	76.29	87.71

E Enron dataset, *LS* Ling-Spam dataset, *Info Gain* information gain, *SBS* sequential backward selection, *EFS* exhaustive feature selection, *SFS* sequential forward selection, *BFS* best first search, *WFS* wrapper-based feature selection

4 Experimental results

The experiments are carried out with the help of steps given in Algorithm 1. For carrying out the analyses as specified before, Pycharm software has been used. The data have been collected from three datasets, namely Enron dataset (http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html), Ling-Spam dataset (http://www.aueb.gr/users/ion/data/lingspam_public.tar.gz.2019) and PU dataset (http://www.aueb.gr/users/ion/data/PU123ACorpora.tar.gz.2019). For evaluating the experimental results, the datasets are divided in the ratio of 70:30 which means 70% of the data are contained in the training set and the remaining is contained in the test set.

In this, the classifiers are trained on the given training set, making predictions on the test set, and at last the computation of accuracy is done. In the proposed method, it is shown that the accuracy for detection of spammed emails depends on the classifier as well as on the strategies used for the selection of features. The selections of features are shown in Table 2.

At first, experiments are carried out without any strategies used for selection of features. This brought about generally poor results in terms of accuracy for three algorithms. The outcomes can be seen in Table 3. Furthermore, in the next stage, the experiments are carried out with strategies used for selection of features on every one of the classified algorithms. At last, the comparisons are done between with and without strategies used for selection of features. The outcome of accuracy outlined in Table 3 illustrates about the highest values.

The accuracy of the second highest data is depicted in Table 3 with italic fonts. There has been an increase in the accuracy of DT and NB from 76.24 to 85.54% and 55.13 to 67.13%, respectively. Thus, on making the final conclusion, RF shows the increased accuracy from 87.02 to 92.97%. The impact of every one of the methods used for selection of features on every algorithm, i.e., DT, NB and RF, is portrayed in Fig. 4.

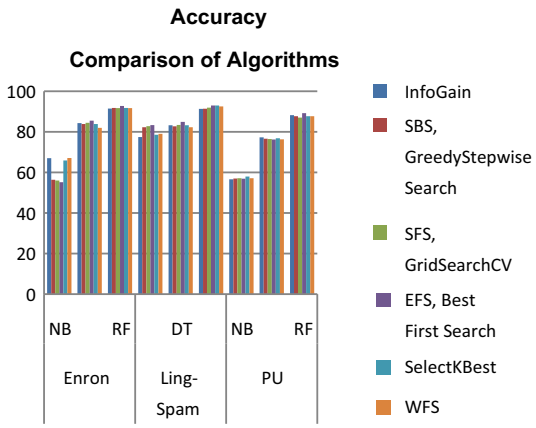

Fig. 4 Comparative study of different machine learning algorithms. *E* Enron dataset, *LS* Ling-Spam dataset

It can likewise be seen from the outcomes that, when the selection of features are done in a proper way, it hampers the NBs' performance.

RF is the best among the rest classifiers in the presence of with and without the strategies used for selection of features. From the outcomes, it tends to be seen that with the strategies used for selection of features such as sequential feature selector, GridSearchCV and exhaustive feature selector, best first search has performed in a good manner in comparison with others—aside from the NB in the Enron dataset

5 Comparative analysis

In accessing the dataset required for carrying out the research works for a specific purpose, several challenges were faced. Among these, one of the challenges being faced is to carry out the classification task of spam emails. This is likewise applied in the assessment of recently proposed spam filtering techniques. It gives a simple way to access the datasets that are readily available. In this research, spam detection techniques have been proposed

Table 4 Classification results of three ML algorithms

	Recall			Precision			F-Measure		
	E	LS	PU	E	LS	PU	E	LS	PU
NB	64.34	85.14	56.12	74.02	90.97	99.91	68.84	86.90	72.68
DT	82.00	82.00	76.00	82.00	82.00	77.00	82.00	82.00	76.00
RF	85.00	87.00	78.00	85.00	87.00	78.00	85.00	87.00	78.00

E Enron dataset, *LS* Ling-Spam dataset

with the help of three datasets, namely Enron dataset (http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html), Ling-Spam dataset (http://www.aueb.gr/users/ion/data/lingspam_public.tar.gz.2019) and PU dataset (http://www.aueb.gr/users/ion/data/PU123ACorpora.tar.gz.2019).

The research work is further carried out with three different types of algorithms like NB, DT and RF with no feature selection strategy. These datasets are further split in the ratio of 70:30. For Ling-Spam dataset, RF got more than 90% of TPR (True Positive Rate), while DT got TPR nearly 90% except NB achieves the lowest TPR as nearly 85%. And for FPR (false positive rate), nearly about every algorithms have given a decent result, but RF achieves the lowest one, i.e., 37.06%.

For Enron dataset, the three algorithms have given nearly about the similar outcome as RF got lower FPR as compared to the Enron dataset which is 13.10%. The accuracy achieved by RF got more than 92% as compared to Enron dataset.

While comparing the three datasets, i.e., Enron dataset (http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html), Ling-Spam dataset (http://www.aueb.gr/users/ion/data/lingspam_public.tar.gz) and PU dataset (http://www.aueb.gr/users/ion/data/PU123ACorpora.tar.gz), RF achieves the highest accuracy among all which is 92.56% in the Ling-Spam dataset except NB achieves the lowest one, i.e., 57.13% in PU dataset.

The same result can be observed in the case of TPR where RF achieves the highest TPR among all which is 91.17% in Ling-Spam dataset, but NB got the lowest one which is 57.12% in PU dataset. Table 4 depicts the outcome used for comparison among three ML algorithms having all featured data along with the performance measures (precision, recall and F-measure) (Kumar et al. 2012; Gupta et al. 2013; Mishra et al. 2018) which are used for calculation between the three datasets.

As from Table 4, the outcome used for classification purpose among three ML algorithms with three datasets can be noted. After making an observation on the data listed in Table 4, the conclusion is that RF has the highest F-Measure than others. While comparing the datasets, we observed that there is a slight difference in F-Measure in Ling-Spam dataset which is still a satisfying performance but NB achieves a lower F-Measure of 72.68% in comparison with others which itself shows that NB does not know how to improve itself after the addition of the extracted word feature. Finally, from this, RF depicts the highest F-Measure of about 87% in Ling-Spam dataset.

With having new detection features, the result of RF depicts the lowest FPR of around 13.95% in Enron dataset among others, but has a high F-Measure of around 86% both in Enron and in Ling-Spam datasets for the proposed detection features. These detection features have been carried out on all the datasets. The details are shown in Table 5. The detection rate (DR) seems to be higher than all other algorithms which are more than 92% in Ling-Spam dataset.

5.1 Model optimization

To judge the performance of the algorithm, the two important questions arises here are: (1) Are the features defined in a well manner? (2) Do the amount of data required during the training phase are sufficient? To answer such questions, the possible solution is to check the size of the dataset during training time. With the increase in the size of training data, it becomes very much complex for a

Table 5 Outcome of new detection features

	FPR			Detection rate			F-Measure		
	E	LS	PU	E	LS	PU	E	LS	PU
NB	51.43	60.94	60.69	55.13	83.29	56.90	55.00	76.00	56.00
DT	14.54	45.89	29.54	84.54	84.90	76.24	85.00	85.00	76.00
RF	13.95	44.36	24.72	92.78	92.97	89.20	86.00	86.00	78.00

E Enron dataset, *LS* Ling-Spam dataset

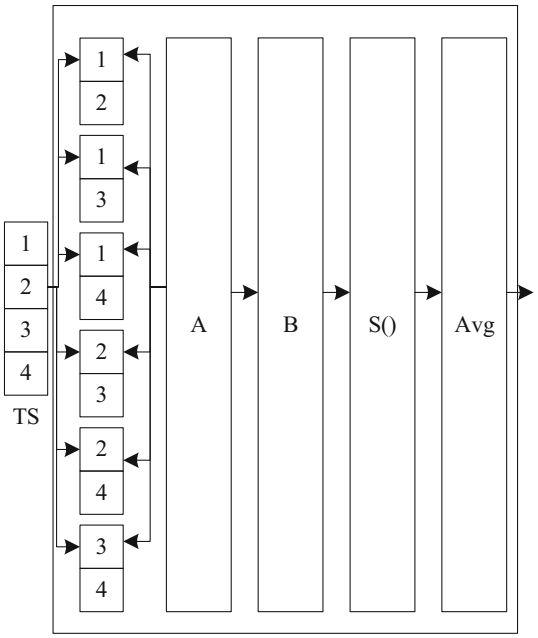

Fig. 5 Four cross-fold validation techniques. A—trained features, B—test features, S()—selection() and TS—training set

model to learn in a correct way and to represent/fit each part of the training data. Moreover, it may be possible that some part of the training data may or may not fit. This is due to the presence of some noise in the training data. As a result of this, the cross-validation score is reduced more effectively. As a result of this, the cross-validation score for the test part increases. This occurs due to an increase in the capacity of model for generalizing the data. But in the case of smaller training datasets, the model may overfit and performs ineffectively on the test set.

This is because the ultimate aim of an algorithm is to make the model to perform better. However, there are some drawbacks of the proposed algorithm as: (1) It suffers from generalizing the result on the test part in a more accurate way. These occur because some amounts of noisiness are present in the test dataset. Due to this, there is a little error in the training part, but with a high error in the test part. (2) The performance of the proposed algorithm decreases slightly in order to fit the whole dataset. These happen only when there is a lesser number of data that are used in training time. To solve such drawbacks of the proposed algorithm, a k-fold ($k = 4$) cross-validation technique is used. The validation technique (in Fig. 5) is used to assess the performance of the classification algorithm among the datasets.

The selection function which is used for selection of feature is based on the four cross-fold validation technique which is repeated as per accuracy of the classifier. The heuristic function is used along with the selection function to improve the accuracy of classifier while searching for relevant features from datasets. The number of validation turns that are required for smaller datasets is more than that of larger datasets because the smaller dataset requires a smaller amount of time during the learning phase.

Despite having some drawbacks of the proposed algorithm, there are some advantages as: (1) It provides clear thought regarding the viable dimension of every classifier for detecting the mail as ham or spam. (2) The accuracy level that can be accomplished by the proposed algorithm is higher than other algorithms. These can be clearly justified when the result of the detection rate (DR) is more than 92%.

To optimize the model, a penalty is provided as supplementary to the selection function. The goal of this penalty is to simply help in breaking the ties of smaller subsets of features. The term "breaking the ties" means only picking smallest one from the two subsets of features. To achieve this goal, the penalty is made less than or equal to 0.1%. While optimizing the data, a lot of issues are faced but some of them are mentioned below:

- To optimize the NP-hard problems, many algorithms are required to find the hypothesis by doing approximations because they don't have access to fundamental distributions. This kind of issue is very similar to bias variance tradeoff where accuracies are estimated by trading off the datum.
- It is very difficult to find the relevant features from the subsets for NP-hard problems because classifier's accuracy may reduce leisurely when irrelevant features are taken during the learning process.

Due to these issues, the process of selecting the relevant features is summarized by finding the optimal features. However, optimal features may or may not be unique because similar accuracies can be achieved by using two dissimilar feature sets.

5.2 Calculating the complexity

Time complexity is one of the most significant criteria for calculating the effectiveness of an algorithm. To compute the complexity of the proposed algorithm, it is needed to find out the number of calculation steps taken by the algorithm. The number of calculation steps may vary from the size of the dataset. Let the size of the dataset be n, and m be the number of relevant features. If the size of the dataset is very large, then a large amount of time is taken for computation and vice versa. To carry out such process, initially, a labeled feature matrix of $(n \times m)$ is created. To compute the time complexity of this labeled feature matrix, n may be approximately equal to m, i.e.,$n \cong m$. This tends to $O(n^2)$ because the size of the dataset is very large initially. Some parts may contain noisy features and others

may not. If training is done with this labeled feature matrix, then computation time will be very high, and hence, the performance of the proposed algorithm will decrease. To reduce the computation time or increase the performance of the proposed algorithm, relevant features are selected. In other words, noisy features are removed which may take constant time, i.e.,$O(1)$. After removal of the noisy features, the dataset contains only $(n-1)$ features. These feature sets are further divided into smaller subsets. This is done to make the training process easier. Hence, the time complexity remains at $O((n-1)logn) \cong O(nlogn)$. From $(n-1)$ digit, (1) is neglected because digit (1) is very small. Finally, time complexity can be evaluated in three cases: (a) The best case will occur when there are only relevant features in the dataset, then, it will take $O(nlogn)$. (b) The worst case will occur when both relevant and noisy features are present in the dataset, then it will take $O(n^2)$. (c) For average case, the time complexity will be $O(nlogn)$.

Space complexity $S(n)$ measures the quantity of memory required for storage purpose by the algorithm at any point. Space complexity is summation of total number of elements and stack space. Stack space is basically the amount of extra space taken by the algorithm, i.e.,

$$S(n) = Totalelements + StackSpace$$

Since there are only n number of elements, so it will take only $O(n)$. For stack space computation, it will take $O(logn)$.This is because division is being taken for making the training process to train the dataset easily. Hence, total $S(n)$ is:

$$S(n) = O(n) + O(logn) \cong O(n)$$

6 Conclusions

In the last decade, spam mail has become an increasing threat to mail communication. Thousands of undesirable email messages are generated in a bulk format by spammers. These, in turn, may often lead to a waste of time spent by users in searching and deleting the spam emails, loss of the user's network bandwidth and an unnecessary increase in the traffic volume, etc. Most spam email generally contains advertisements of products or services, which may be useless or unnecessary for the user. In this paper, a spam mail detection (SPMD) method is proposed to detect spam emails. Three algorithms are used for classification purpose like Naive Bayes, Decision Tree and Random Forest. Among these classifiers, Random Forest was shown to achieve the highest accuracy of 92.97%.

Funding This study was not funded by any grant.

Compliance with ethical standards

Conflict of Interest The authors have declare that they have no conflict of interest.

Human animal rights No animals were involved. This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

Ahuja L (2018) Handling web spamming using logic approach. In: International conference on advances in computing and data sciences. Springer, Singapore, pp 380–387

Attenberg J, Weinberger K, Dasgupta A, Smola A, Zinkevich M (2009) Collaborative email-spam filtering with the hashing trick. In: Proceedings of the sixth conference on email and anti-spam

Bassiouni M, Ali M, El-Dahshan EA (2018) Ham and spam e-mails classification using machine learning techniques. J Appl Secur Res 13(3):315–331

Bhat SY, Abulaish M, Mirza AA (2014) Spammer classification using ensemble methods over structural social network features. In: Proceedings of the 2014 IEEE/WIC/ACM international joint conferences on web intelligence (WI) and intelligent agent technologies (IAT), vol 02. IEEE Computer Society

Camastra F, Ciaramella A, Staiano A (2013) Machine learning and soft computing for ICT security: an overview of current trends. J Ambient Intell Humaniz Comput 4:235–247

Chebrolu S, Abraham A, Thomas JP (2005) Feature deduction and ensemble design of intrusion detection systems. Comput Secur 24(4):295–307

Christina V, Karpagavalli S, Suganya G (2010) A study on email spam filtering techniques. Int J Comput Appl 12(1):0975–8887

DCC Spam Control Delayed Your E-Mail. http://umanitoba.ca/computing/ist/email/exchange/securityspamindex.html. Accessed 20 Dec 2018

Gaurav D, Yadav JKPS, Kaliyar RK, Goyal A (2019) Detection of false positive situation in review mining. Soft Computing and signal processing. Springer, Singapore, pp 83–90

Gupta S, Kumar P, Abraham A (2013) A profile based network intrusion detection and prevention system for securing cloud environment. Int J Distrib Sensor Netw 9(3):364575

Herrero A, Corchado E, Pellicer MA, Abraham A (2009) MOVIH-IDS: a mobile-visualization hybrid intrusion detection system. Neurocomputing 72(13–15):2775–2784

http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html. Accessed 31 Jan 2019

http://www.aueb.gr/users/ion/data/lingspam_public.tar.gz. Accessed 05 Feb 2019

http://www.aueb.gr/users/ion/data/PU123ACorpora.tar.gz. Accessed 10 Feb 2019

https://cacm.acm.org/magazines/2018/7/229047-youve-got-mail/fulltext?mobile=false. Accessed 20 Feb 2019

Staiano A, Di Taranto MD, Bloise E, Agostino MND, D'Angelo A, Marotta G, Gentile M, Jossa F, Iannuzzi A, Rubba P, Fortunato G (2013) Investigation of single nucleotide polymorphisms associated to familial combined hyperlipidemia with random forests. In: Neural nets and surroundings. Springer, Berlin, Heidelberg, pp 169–178

Kim D, Deokseong S, Suhyoun C, Pilsung K (2019) Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec. Inf Sci 477:15–19

Kumar RK, Poonkuzhali G, Sudhakar P (2012) Comparative study on email spam classifier using data mining techniques. In: Proceedings of the international multi-conference of engineers and computer scientists, vol 1, Hong Kong, pp 14–16

Liu TJ, Tsao WL, Lee CL (2010) A high performance image-spam filtering system. In: 2010 ninth international symposium on distributed computing and applications to business engineering and science (DCABES). IEEE, pp 445-449

Merugu S, Reddy MCS, Goyal E, Piplani L (2019) Text message classification using supervised machine learning algorithms. In: Kumar A, Mozar S (eds) ICCCE 2018. ICCCE 2018. Lecture Notes in Electrical Engineering, vol 500. Springer, Singapore, p 2019

Microsoft Sender ID Framework. http://www.microsoft.com/mscorp/safety/technologies/senderid/default.mspx. Accessed 14 Jan 2019

Mishra S, Sagban R, Yakoob A, Gandhi N (2018) Swarm intelligence in anomaly detection systems: an overview. Int J Comput Appl 1–10. (2018)

Nizamani S, Memon N, Wiil UK, Karampelas P (2013) Modeling suspicious email detection using enhanced feature selection. arXiv:1312.1971

Oliveira JP (2019) Spam dataset analysis. https://rstudio-pubs-static.s3.amazonaws.com/65173_80cf15e9415c48d5a60bc54b042fccfe.html. Accessed 08 Aug 2019

Park YW, Klabjan D (2018) Three iteratively reweighted least squares algorithms for L1-norm principal component analysis. Knowl Inf Syst 54(3):541–565

Pyzor's homepage. https://sourceforge.net/p/pyzor/mailman/pyzor-announce/. Accessed 14 Dec 2018

Radev D (2008) CLAIR collection of fraud email, ACL data and code repository. ADCR2008T001

Razor's homepage. http://razor.sourceforge.net/. Accessed on 05 Dec 2018

Sarwat N, Menon N, Glasdam M, Nguyen DD (2014) Detection of fraudulent emails by employing advanced feature abundance. Egypt Inform J 15:169–174

Sender Policy Framework. http://www.openspf.org/Introduction. Accessed 24 Jan 2019

Sharaff A, Nagwani NK, Dhadse A (2016) Comparative study of classification algorithms for spam email detection. In: Shetty N, Prasad N, Nalini N (eds) Emerging research in computing, information, communication and applications. Springer, New Delhi

Symantec Brightmail Anti-Spam. https://www.symantec.com/products/mail-security-exchange. Accessed 23 Dec 2018

Trivedi SK, Dey S (2013) Interplay between probabilistic classifiers and boosting algorithms for detecting complex unsolicited emails. J Adv Comput Netw 1(2):132–136

Vidya Kumari KR, Kavitha CR (2019) Spam detection using machine learning in R. In: Smys S, Bestak R, Chen JZ, Kotuliak I (eds) International conference on computer networks and communication technologies. Lecture Notes on Data Engineering and Communications Technologies, vol 15. Springer, Singapore

Yandex: Some Automatic Spam Detection Methods. http://company.yandex.ru/public/articles/antispam.xml. Accessed 03 Jan 2019

Yoon JW, Hyoungshick K, Huh JH (2010) Hybrid spam filtering for mobile communication. Comput Secur 29(4):446–459

Youn S, McLeod D (2007) A comparative study for email classification. In: Elleithy K (ed) Advances and Innovations in systems, computing sciences and software engineering. Springer, Dordrecht