Pacific Science Review A: Natural Science and Engineering 18 (2016) 145–149

HOSTED BY

Contents lists available at ScienceDirect

Pacific Science Review A: Natural Science and Engineering

journal homepage: www.journals.elsevier.com/pacific-science-review-a-natural-science-and-engineering/

Proposed efficient algorithm to filter spam using machine learning techniques

Ali Shafigh Aski [a, *], Navid Khalilzadeh Sourati [b]

[a] Department of Computer Engineering, Islamic Azad University, Sari Branch, Islamic Republic of Iran
[b] Islamic Azad University of Amol, Ayatollah Amoli Branch, Islamic Republic of Iran

ARTICLE INFO

Article history:
Received 29 June 2016
Accepted 26 September 2016
Available online 19 November 2016

Keywords:
Spam
Header
Machine learning
Classifier

ABSTRACT

Electronic spam is the most troublesome Internet phenomenon challenging large global companies, including AOL, Google, Yahoo and Microsoft. Spam causes various problems that may, in turn, cause economic losses. Spam causes traffic problems and bottlenecks that limit memory space, computing power and speed. Spam causes users to spend time removing it. Various methods have been developed to filter spam, including black list/white list, Bayesian classification algorithms, keyword matching, header information processing, investigation of spam-sending factors and investigation of received mails. This study describes three machine-learning algorithms to filter spam from valid emails with low error rates and high efficiency using a multilayer perceptron model. Several widely used techniques include C4.5 decision tree classifier, multilayer perceptron and Naïve Bayes classifier, all of which are used for training data whether in the form of spam or valid emails. Finally, the results are discussed, and outputs of considered techniques are examined in relation to the proposed model.

1. Introduction

The Internet is considered a very powerful tool. Email is an efficient way to exchange information. Considering the growth of the Internet and wide use of email, the rate of increase of spam is of great concern. Spam may originate from anywhere in the World Wide Web. Despite tools to prevent spam, it has been increasing daily. One way to assess the current situation is that organizations examine available means that can be used to even count the amount of spam. These means include corporate email systems, gateways, spam filtering and end user training. Internet users cannot disregard this important problem of the modern Internet world. Lack of mechanized systems to prevent spam will result in a spam-saturated World Wide Web, destruction of Internet products and severe loss of bandwidth.

1.1. Architecture of spam filtering rules and existing methods

There are various definitions for spam and its difference from valid mails. The shortest definition of spam is 'an unwanted electronic mail'. A major problem with introduction of spam filtering is that a valid email may be labelled spam or a valid email may be missed. To not filter spams causes problems; not only will inboxes be completely occupied by spam, but it will result in more serious problems including reduction of bandwidth and storage. There are techniques to identify emails received in the form of spam, as follows: black list/white list, Bayesian classifying algorithm [1], keyword matching and header information analysis [11].

A white list is a list of addresses from which users tend to receive emails. Users can also add email addresses, domain inputs or domains of functions. An advantage of white list is that it allows users or administrators to put email addresses of favourite people into the list in order to make sure that valid emails received from addresses in the white list are not labelled spam when receiving emails from different senders.

A black list is a list of addresses from which users do not tend to receive emails. The header reviewing process of an email involves a series of rules implemented as follows. An email will be labelled junk and transferred to a spam folder if its header is congruent to a

* Corresponding author.
E-mail addresses: alishafigh@gmail.com (A.S. Aski), Mr.khlilzadeh@gmail.com (N.K. Sourati).
Peer review under responsibility of Far Eastern Federal University, Kangnam University, Dalian University of Technology, Kokushikan University.

http://dx.doi.org/10.1016/j.psra.2016.09.017

header of training data in the black list. Otherwise, it will be transferred to the white list.

Bayesian classifications are the basis of many anti-spam methods; probability of a future event can be obtained by its occurrence in the past. Bayesian is an automatic classifier. Only text algorithms that have shown better efficiency are recently used for filtering [9]. Previously, various rule-based software packages were used for filtering operations [6]. Rule-based solutions have two substantial disadvantages. First, these systems required users to generate a series of rules; the users required broad knowledge of spam to formulate suitable rules. Second, these rules required reformulation by experts because features of spam change over time [14]. Basically, reformulation is time-consuming with a high level of error.

2. Methodology

Most spam filtering methods use text techniques [12]; therefore, most of the problems are related to classification. The present study classifies rules to extract features from an email. Most developed models for minimizing spam have been machine learning algorithms [3,10]. Various systems have been introduced for automatic classification of emails [4]; some are as follows: decision-based systems [14], Bayesian classifiers [15], support vector machine [1,2], neural networks [15] and sample-based methods [7]. The present study discusses three important algorithms of machine learning techniques including C4.5 decision tree classifier, multilayer perceptron and naïve Bayes classifier provided in the proposed model. Various methods are presented in [13] to filter spam using machine learning algorithms.

2.1. Multilayer perceptron (MLP)

Fig. 1 shows a multilayer perceptron (MLP) neural network [5]. The model delivers information by activating input neurons containing values labelled on them. Activation of neurons is calculated in the middle or output layer, as follows:

$$a_i = \sigma\left(\sum_j W_{ij} O_j\right) \tag{1}$$

where a_i represent activation level of neuron i; j is neuron set of the previous layer; W_{ij} is the weight of the link between neuron i and j; O_j is the output of neuron j and $\sigma(x)$ represents a transfer function.

$$\sigma(x) = \frac{1}{1+e^x} \tag{2}$$

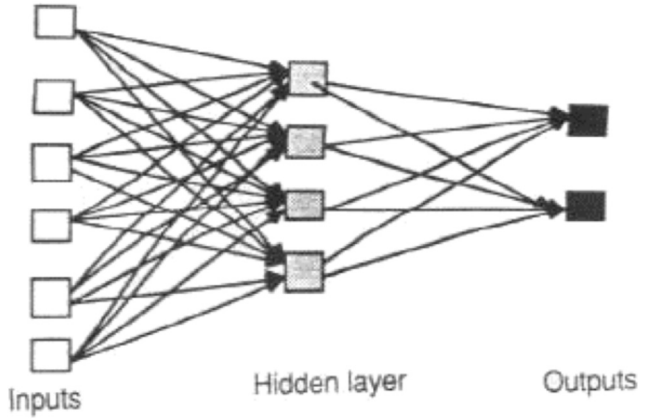

Fig. 1. Perceptron artificial neural network containing input, hidden and output layers.

A multilayer perceptron is trained using an error back-propagation strategy based on generalized delta rule.

An MLP indicates a nonlinear relationship between input and output vectors. This is achieved by connecting neurons of a node in the previous and next layers. Output neurons are multiplied by weighting coefficients. Then they are inserted in the nonlinear function of activation as input. In the training step, perceptron is given training information. Then network weights are adjusted to minimize the error between predicated and target output or to increase frequency of trainings to a predetermined maximum level. A series of unexperienced inputs is applied to the input to validate the training. These inputs need to be different from inputs used for network training. Training of neural networks is typically very complicated as an optimization problem containing a large number of variables. MLP is an in-depth optimizer to solve many problems; for example, when a fixed model or adequate knowledge is not available on values of inputs and their relationship with output. Fig. 2 shows a perceptron containing a bios input.

2.2. C4.5 decision tree classifier

Output of a C4.5 decision tree classifier is structural data in the form a binary tree. A C4.5 tree is modelled as follows. A training set is a set of base tuples to determine classes related to these tuples. A tuple X is represented by an adjective vector $X = (x_1, x_2, \ldots, x_n)$. Assume that a tuple belongs to a predefined class that is determined by an adjective called as class label. The training set is randomly selected from the base; this step is called the learning step. This technique is very efficient and extensively uses classification. The structure of the tree can be implemented with the following factors:

1. A node of the tree represents a test on an adjective;
2. A branch exiting from a node represents possible outputs of a test;
3. A leaf represents a class label.

A decision tree includes a rule set by which objective functions can be predicted. The J48 algorithm is an optimized version of C4.5. The algorithm used for this model uses greedy techniques. Fig. 3 shows classification of a sample based on a decision tree.

2.3. Naïve Bayesian classifier

A Naïve Bayesian classifier generally seems very simple; however, it is a pioneer in most information and computational applications for spam filtering [2,8]. A Bayesian network is an acyclic directed graph indicating probability distribution in a compressed way. A node in this graph shows a random variable, X_i. A directed edge between two nodes indicates potential interdependence between a variable shown by the parent node and another variable shown by a child node. The structure of this network assumes that a node X_i is conditionally independent from other vector and non-parent nodes. A node X_i is related to a potentially conditional table determining probability distribution on the node X_i by the amount allocated to parents of the node. A Bayesian classifier is simply a

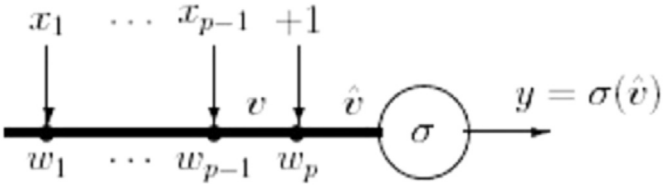

Fig. 2. Perceptron containing a bios input.

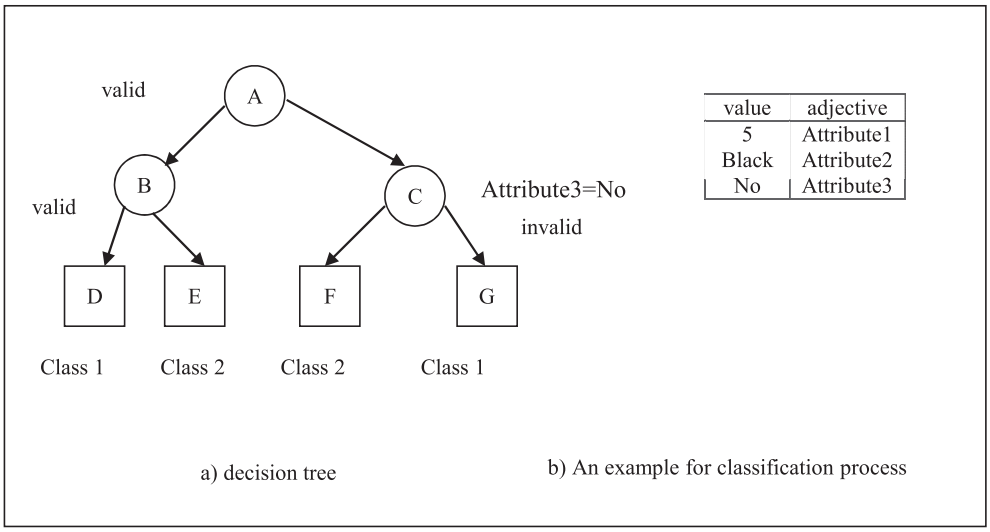

Fig. 3. Classification of a sample based on decision tree.

Bayesian network used for classification including group C which indicates variable of class label and variable X_i which indicates features. According to Bayes theory:

$$P(C = C_k|X = x) = \frac{P(X = x|C = c_k)P(C = c_k)}{P(X = x)} \quad (3)$$

An important problem with Bayes theory is independence of random variables, the lack of which makes it difficult to calculate $P(X = x|C = c_k)$. In the following, an old assumption on independence is provided for naïve Bayesian classifier; according to this assumption, a feature X_i will be a condition independent from other features if a class C variable is available.

The above assumption can be written as follows:

$$P(X = x|C = c_k) = \Pi_i P(X_i = x_i|C = c_k) \quad (4)$$

To clear the concept of the above formula as well as independence, consider the following example. Given that the word Coca Cola appears in 400 out of 3000 spams and given that the word appears in only 5 of 300 valid emails, then the probability of an email containing Coca Cola is spam is

$$P(cocacola|Spam) = \frac{\frac{400}{3000}}{\frac{400}{3000} + \frac{5}{300}} = 0/8889$$

3. Extraction of features and implementation of the considered algorithm

The work here was based on rules for proper scoring in terms of the efficiency of rules. The considered rules were provided in three forms: 1) email header information analysis, 2) keyword matching, and 3) main body of the message. A score was finally obtained for these rules.

The following introduces several efficient rules by which spam can be detected. The proposed algorithm to evaluate a spam works as follows:

The proposed model evaluated the email received in the system using 23 rules as shown in Table 1. Each rule was assigned a score and the sum of scores was calculated. Following evaluation of an email, a rule was applied to the email. As the first rule was applied to the received email followed by a positive result, the received email was scored. The process continued until the 23th rule was applied to the received email. The final score of the received email was compared to a threshold value. The received email was labelled Junk and sent to the Spam folder if its score was more than the threshold value (see Table 2).

4. Discussion

Results were obtained from studies of data of personal emails modelled by WEKA software which is a very powerful, open source and portable tool with a strong user interface to run machine

Table 1
Series of rules to assign a score to the received email.

Via meaning of the name
Via domain names
Via blocked IP number
By detecting apostrophe
Via evaluation of automatic white list (AWL)
Via addresses of in the black list
Via addresses of in the white list
Via type of content
Via bounded content
Via content of name
Via undeclared addresses of recipients
Via main header
Receiving from an address and sending to a similar address
Unclear subject field
If the subject field contains ambiguous letters
If the message is forward
If the message is replied
If there is no sign of the sender in the subject field
Whether there is a text
Whether the message contains emotional words
Whether the mark (") is constantly repeated in the body
Whether characters contain Latin alphabet compounds
There is large number of empty spaces In the body

Table 2
Comparison of results obtained from classifiers.

MLP	J48	Naïve Bayes	Evaluation rules
138.05	0.20	0.15	Data training time (sec)
1485	1449	1479	Valid classification of samples
99.2	96.6	98.6	Valid recognition rate (%)
2	4	5	False positive (%)

learning algorithms, techniques and pre-processing steps. The experimental dataset used for the study was drawn from a series of spam and valid emails entered into a personal system during a six-month period. Emails were well-reviewed and were subjected to the 23 default rules of the proposed model in order to classify emails. The primary dataset included 750 valid emails as well as 750 spams. To extract vector features of an email, the following methods were used: 1) email header review, 2) keyword review, 3) black list and white list.

Labels of a class in the proposed model included L and S; the former represented Legitimate, and the latter is an alternative for Spam.

Three classifiers including naïve Bayes, C4.5 decision tree and MLP neural networks were run on training data by WEKA software. The training data were tested in terms of message header information, black and white list and keyword review. Efficiency and accuracy of training data were evaluated by 10 classes of reliably valid data. An accuracy factor was calculated as follows.

4.1. Number of studied valid samples multiplied by training dataset to total data samples

Fig. 4 shows pseudo code of the considered algorithm compiled by C# language (see Figs. 5 and 6).

Two other basic concepts are used for spam filtering operations, false positive and false negative. The former refers to those spams classified as valid emails and the latter refers to valid emails wrongly classified as spam. False positive rate of a classifier is applied to its efficiency. Table 1 shows the efficiency of the above classifiers.

Efficiency of these three techniques depends on the following factors: 1) valid recognition rate, 2) data training time and 3) false positive rate. The efficiency of MLP neural network was better than the other models.

MLP requires more time to develop the model. J48 and naïve Bayes algorithm require more time on learning experimental data.

5. Conclusions

There are many ways to filter Internet spam. Considering the daily growth of spam and spammers, it is essential to provide

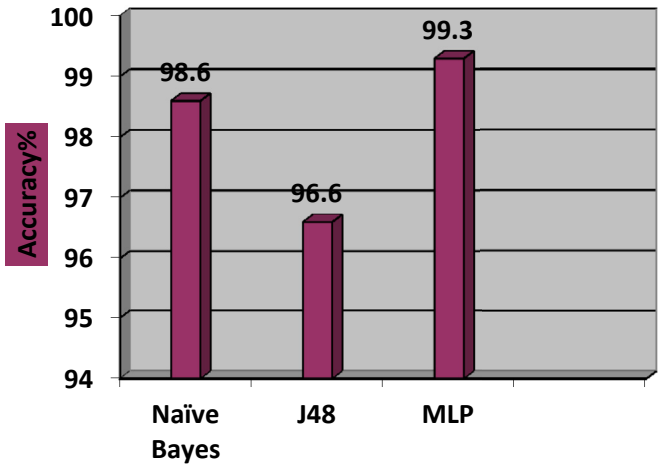

Fig. 5. Results from test of accuracy of classifiers.

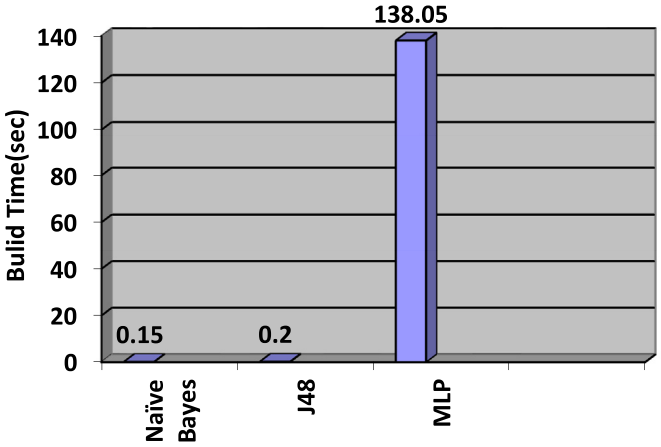

Fig. 6. training time of classifiers.

```
Int  Number=0 , SumNumber=0 , MaxNumber= 23;
For(int   i=1;i<=2;i++)
{
Switch(i)
  {
   Case 1: SumNumber+= RuleAccept();break;
   Case 2: If(Email=="ok Rule Two Accept "){ SumNumber+= RuleAccept();break;}
  }
{
If(SumNumber>= MaxNumber){NewMail="Spam" }
Else{ NewMail=sendtoInbox }

Public  int RuleAccept()
{
If(Email=="ok Rule One Accept"){Number++;}
Else{ Number =0}
Return   Number;
}
```

Fig. 4. Pseudo code of the considered algorithm written by C#.

effective mechanisms and to develop efficient software packages to manage spam. Using valid emails and spam the present study extracted data from emails using machine learning algorithms to develop a new model. Measuring the rate of 10 classes of valid emails and running MLP algorithm on test data, the model demonstrated higher efficiency than naïve Bayes classifier algorithms and J48 with a low rate of false positive. The proposed algorithm can be modelled to be implemented on a Mail Server and Mail Client in order to eliminate problems, such as bandwidth reduction and very low efficiency, from which users usually suffer.

References

[1] I. Anderouysopoulos, J. Koutsias, K.V. Chandrianos, G. Paliouras, C. Spyrpolous, An evaluation of naïve Bayesian anti-spam filtering, in: Proceeding of 11th Euro Conference on Match Learn, 2000.

[2] I. Anderouysopoulos, J. Koutsias, K.V. Chandrianos, G. Paliouras, C. Spyrpolous, An experimental comparison of naïve Bayesian and keyword-based anti-spam filtering with personal e-mail messages, in: Proceeding of the an International ACM SIGIR Conference on Res and Devel in Inform Retrieval, 2000.

[3] A. Asuncion, D. Newman, UCI Machine Learning Repository, 2007. http://www.ics.uci.edu/mlearn/MLRRepository.html.

[4] E. Blanzieri, A. Bryl, A Survey of Learning-based Techniques of Email Spam Filtering Tech. Rep, DIT-06-056, University of Trento, Information Engineering and Computer Science Department, 2008.

[5] J. Clark, I. Koprinska, J. Poon, A neural network-based approach to automated email classification, in: Proceeding of the IEEE/WIC International Conference on Web Intelligence, 2003.

[6] S.J. Delany, D. Bridge, Textual case-based reasoning for spam filtering: a comparison of feature-based and feature-free approaches, Artif. Intell. Rev. 26 (1–2) (2006) 75–87.

[7] F. Fdez Riverola, E. Iglesias, F. Diaz, J.R. Mendez, J.M. Chorchodo, Spam hunting: an instance-based reasoning system for spam labeling anf filtering, Decis. Support Syst. 43 (3) (2007) 722–736.

[8] A. Seewald, An evaluation of Naïve Bayes variants in content-based learning for spam filtering, Intell. Data Anal. (2009).

[9] Ahmed Khorsi, An overview of content-based spam filtering techniques, Informatica 31 (3) (October 2007) 269–277.

[10] Ian H. Witten, Eibe Frank, Data Mining – Practical Machine Learning Tools and Techniques, second ed., Elsevier, 2005.

[11] Chih – Chien Wang, Sheng – Yi Chen, Using header session messages to anti-spamming, Comput. Secur. 26 (5) (2007) 381–390.

[12] M. Chang, C.K. Poon, Using phrase as features in email classification, J. Syst. Softw. 82 (2009) 1036–1045.

[13] T.S. Guzella, T.M. Caminhas, A review of machine learning approaches to spam filtering, Expert Syst. Appl. 36 (2009) 10206–10222.

[14] C. Wu, Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks, Expert Syst. (2009).

[15] Anti-Spam filtering using neural networks and Bayesian classifiers. Yue Yang and Sherif Elfayoumy. Proceeding of the 2007 IEEE international symposium on computational intelligence in robotics and automation.