A novel method in scam detection and prevention using data mining approaches

Mokhtari, M, Saraee, MH and Haghshenas, A

A Novel Method in Scam Detection and Prevention using Data Mining Approaches

Maryam Mokhtari[1], **Mohammad Saraee**[1], **Alireza Haghshenas**[2]

[1] Department of Electrical and Computer Engineering
Isfahan University of Technology, Isfahan, Iran
Mokhtari@ec.iut.ac.ir, Saraee@cc.iut.ac.ir

[2] Department of Computer Engineering
Iran University of Science & Technology, Tehran, Iran
Haghshenas@comp.iust.ac.ir

Abstract

‘Scam’ is a fraudulence message by criminal intent sent to internet user mailboxes. Many approaches have been proposed to filter out unsolicited messages known as ‘spam’ from legitimate messages known as ‘ham’. However up to this date no suitable approach has been proposed to detect Scams. Almost all spam filters which use Machine Learning approaches, classify scams as hams when scam messages are more similar to the average ham than spam. But such fraudulence messages can be very harmful to users as many people in the world lose their funds by relying on scam messages.

In this paper we use Data Mining techniques for scam detection. Bayesian Classifier, Naïve Bayes and K-Nearest Neighbor which are mostly used in spam detection are experimented and the results are reported. In addition, a new approach in scam detection is proposed. This approach uses K-Nearest Neighbor algorithm with modification to Document Similarity equation. Additionally, classification is not binary as ‘scam’ or ‘not scam’: a Fuzzy Decision is used instead of clear types of classes. Scam messages are successfully detected by applying this approach.

Keywords

Scam, Spam, Naïve Bayes, Nearest Neighbor, Fuzzy Decision, Clustering, Fraud

1. Introduction

Nowadays Fraudulent messages by criminal intent have become a complex problem in internet communications. Many people rely on these kinds of emails and lost their fund as a result.

We consider e-mail messages to be of three types: ham, spam, and scam. “Ham”, refers to legitimate e-mail messages. “Spam” messages are unsolicited pieces of email. The “Scam” messages are a subset of spam messages which are intelligent in design, such that they attempt to coax the individual to perform some action of illegal purpose beyond a simple “click me”. While most spams are harmless, scams are engineered by criminals; they earn a considerable amount of money from internet users who read scam messages stories and believe them. In 2003, the FTC reported the American public lost over $400 million to fraudulent activities. Scams communicated via e-mail and the Internet is on the rise as well. Brightmail reports that over three billion scam emails are now sent monthly over the Internet, noting a 50% increase from January to April 2004 alone. In March 2004, Zachary Hill was arrested by the FTC and the Department of Justice for identity theft and illegally attracting people via e-mail to fake websites masquerading as AOL and PayPal. During the tenure of his scam, Hill obtained at least 471 credit card numbers, 152 bank account and routing numbers, and 541 user names and passwords [2].

Many approaches have been proposed for the task of spam detection. The mostly used approaches are based on Bayesian Classifier and Naïve Bayes [9,16]. Naïve Bayes is simple and its accuracy is acceptable. Also K-Nearest Neighbor has been used to its simplicity and speed. Other approaches like Support Vector Machines, Decision Trees, Artificial Neural Networks, and many other machine learning methods or combinations of them have been used. A description of such approaches is discussed in section 3.

However, all these approaches perform well on static datasets, but scam messages changes dynamically and such approaches are not suitable for detecting them. While a simple spam filter can filter out spams very good, most scams can pass through most spam filters because scammers are more intelligent because they want to reach their criminal intent and earn internet users' frauds. So the opportunity for building better detection and investigative tools has again attracted the interest of many researchers in the world. Whereas many solutions have been proposed for the spam problem, no perfect attempt has been done on detection of Internet Frauds and hidden criminal aspects in emails.

In this paper, we try to design a filter which can:

- □ Filters out spams, more accurately than existing simple spam filters.
- □ Matches with dynamic changes and new spams created by spammers.
- □ Identifies the criminal intent hidden in e-mails.
- □ Has acceptable accuracy and speed.
- □ Is updated by users' feedback.
- □ Consider time as an inherent attribute of a message.
- □ Prune unusable training messages from dataset.

The organization of the rest of the paper is as follows: In section 2 a definition of scams & spams and differences between them is overviewed. In section 3, Spam Filtering approaches proposed till now is discussed. Section 4 discusses spam filtering as a text classification problem and describes pre-processing steps in this task. In section 5 a simple and useful approach in Spam Filtering, K-Nearest Neighbor is defined. In section 6 the mostly used approaches in spam detection on static datasets, Bayesian Classifier and Naïve Bayes is defined. Different Errors and Costs of Errors are discussed in section 7. Section 8 explains that approaches proposed to spam detection mostly work on static datasets. In section 9 sources of scams is introduced. In section 10 a new approach to match with dynamic changes and scam definition is proposed. In section 11 experimental results are shown. A Conclusion and suggestions for Future Works are given in Section 12.

2. Spams and Scams

In the following two sections we describe spam and scam and what are the problems caused by them.

2.1. What is Spam

Spam, also known as Unsolicited Commercial Email (UCE) and Unsolicited Bulk Email (UBE), is commonplace everywhere in email communication. Spam is a costly problem and many experts agree it is only getting worse [1]. It has become a serious problem, flooding users' inboxes and costing businesses billions of dollars in wasted bandwidth. The cost of spam to companies worldwide is estimated to be $20 billion a year and growing at the rate of almost 100% each year [7]. But for spammers, the costs of sending bulk mail are very low. With a low-bandwidth 56 kbps modem, hundreds of thousands of e-mails can be sent per hour. Internet providers and filtering companies have reported that spam now makes up 16e88% of all e-mails. The proportion varies considerably from one individual to another. The amount of spam received depends on the email address, the degree of exposure, the amount of time the address has been public and the upstream filtering. Because of the economics of spam and the difficulties inherent in stopping it, it is unlikely to go away completely [1,5]. Fig. 1 shows the volume of reports issued from www.SpamCop.net.

2.2. What is Scam

The scam has been conducted since at least 1989 in the form of physical mail, fax, and most recently through email. While much spam is innocuous, a portion is engineered by criminals to prey upon, or scam, unsuspecting people. The senders of scam attempt to mask their messages as non-spam and cheat through a range of tactics, including pyramid schemes, securities fraud, and identity theft via phishing mechanisms (e.g. redirection to faux PayPal or AOL websites). During 2003, the United States Federal Trade Commission (FTC) received more than a half-million consumer complaints, an increase of 25% on the previous year. Of these complaints, approximately 60% were concerned with various types of fraud. The Consumer Sentinel database, maintained by the FTC, now houses over 1.5 million complaints; one million of which correspond to consumer fraud. The total monetary loss for all fraud victims is in excess of $437 billion, with a median loss of $228. In 58% of the complaints, consumers report being contacted through the medium of the Internet [2]. In 2003, MessageLabs Inc. reported that the Nigerian scam grossed an estimated $2 billion dollars, ranking it one of the top grossing “industries” in Nigeria [8].

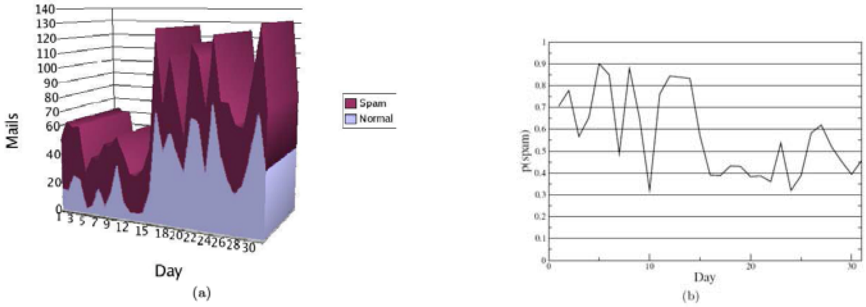


Figure 1. (a) Spam forwarded and reports sent. (b) resulting prior p(spam). (Source: [1])

A major challenge of Internet-fraud problems is the difficulty in discerning scam from spam and regular email. In fact, scam messages differ from other types of spam for several reasons. First and foremost, the scam’s major trait is its hidden criminal intent. In order to lure unsuspecting individuals, the text is engineered to read like regular e-mail, and thus pass successfully through spam filters. Second, messages from the same individual are not necessarily equivalent in text and story. Third, scam messages can be sent out over a longer time period than traditional bulk spam messages. Fourth, scam messages are not necessarily sent via the same physical routes as spam or via the same techniques, such as the commandeering of an open relay.

Usually there is a scheme in which a stranger with an unfortunate story requests an individual for a certain amount of money, usually not a very large sum, to assist in the transfer of a large monetary sum. However, this message is a ruse to bilk the investors out of their money. Current Spam Filters have difficulty in filtering scam from ham as opposed to spam from ham. Based on these findings it is clear that a different type of system is necessary for filtering scam messages from the general population of e-mails [2].

Fig. 2. shows the relationship between three email types; Spams, Hams, Scams. Note that there exist certain messages which are viewed as spam by some individuals and ham by others (e.g. legitimate, but unsolicited advertisements); depicted in the intersection of Fig. 2. But scams are surely unsolicited by all types of users, having criminal intents. Even users may read and believe them. Fig. 3. shows a spam filter trying to classify emails in two classes: spam and ham. This spam filter is a classifier and emails are shown as vectors in a vector space. As figure shows, scams pass through spam filter successfully.

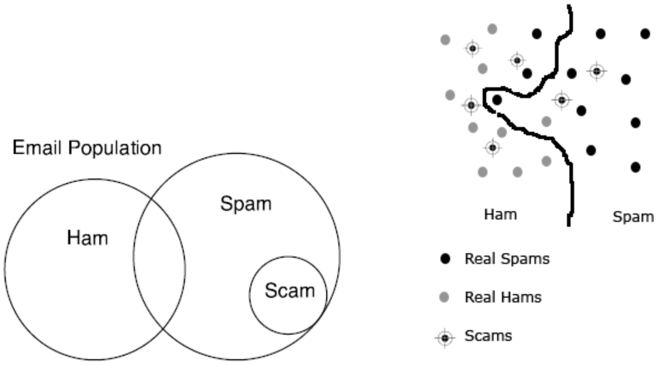


Figure 2. (a) Email types and their relationship; Ham: Legitimate Emails; Spam: Unsolicited Emails; Scam: Criminal Emails (b) A classifier (Spam Filter) has classified emails into two classes, spam and ham in a vector space; where each email is a vector. Scams were successful in passing through the spam filter.

3. Spam Filtering Approaches

Many data mining and machine learning researchers have worked on spam detection and filtering, commonly treating it as a basic text classification problem. The problem is popular enough that it has been the subject of a Data Mining Cup contest as well as numerous class projects [1]. Bayesian analysis has been very popular, but researchers have also used SVMs [10,11,12,3], decision trees, memory and case-based reasoning, rule learning, Artificial Neural Networks [3], K-Nearest Neighbor [3], Instance-based learning [9], Statistical Data Compression Models [6], Latent semantic indexing [13], memory based classifiers [9,19] and even genetic programming [1]. Other methods such as Markov Model have been used in character-level as opposed to common word-level methods [4]. Also combination of methods can be used, like combination of Naïve Bayes and memory based classifiers [14]. But all these evolutions have been on static datasets [14].

A rule-base model could be designed to define a structure for spams and hams. These rules should be updated by a human expert; this is a time-consuming and inaccurate method. Lazy learning techniques like case-based reasoning have been used for such dynamically changing contexts [14].

Bayesian Classifier, Naïve Bayes and K-Nearest Neighbor (KNN) were the mostly used approaches by spam filters as a static text classification. Naïve Bayes is most commonly used [9,16,17,1,3,18].

4. Spam Filtering as a text classification problem

4.1. Pre-Processing

Before classification, pre-processing steps should be applied on emails. These steps are:

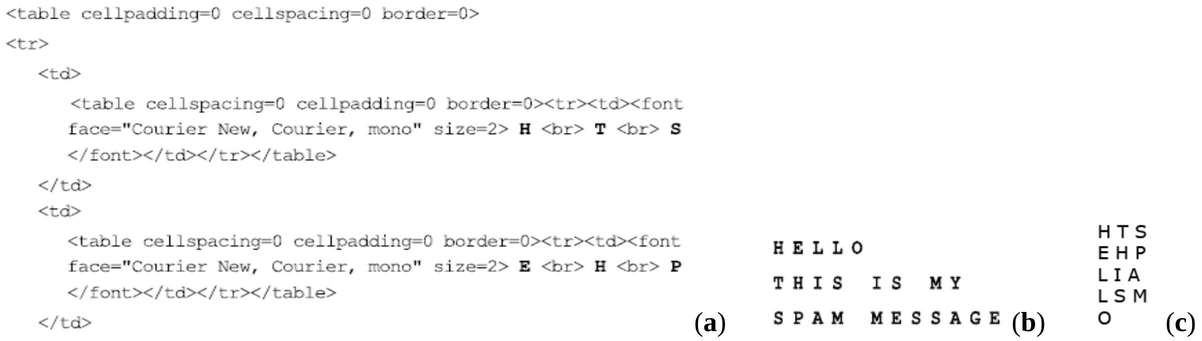
1. Separating the header from the body
2. Removing HTML tags
3. Replacing all sequences of whitespace characters (tabs, spaces and newline characters) by a single space.
4. Eliminating "stop words"
5. Eliminating words less than 3 characters.
6. Stemming
7. Frequency counting

4.1.1. Filtering on Header

When IP Address of email sender is found in a black list, it is simply classified as a spam. A blacklist is available in http://spamlinks.net/filter-bl.htm. If you take part in a lottery, an email by subject of "You have won in lottery" makes you happy. A large amount of Nigerian scams had the subject of "Asalam Alaykom" and had sent to Muslim users (by identifying their country). In addition of subjects, messages sometimes are addressed to “Undisclosed recipients” or “nobody”. As basic header filtering became common in e-mail clients, these obvious text markers were simple to filter upon so spam could be discarded easily.

4.1.2. Removing HTML Tags

In the task of spam filtering, documents are email messages with too many HTML tags. These tags are not real text and should be removed. In removing HTML tags, spammers use frauds to confuse spam filters. As shown in Fig. 4, sometimes “vertical slicing” is performed [5]. In (a) HTML full text and its tags is shown. (b) is HTML file that a recipient sees. (c) is the text after removing all tags which is a nonmeaning text, not suitable for filtering model.



```
<table cellpadding=0 cellspacing=0 border=0>
<tr>
   <td>
      <table cellspacing=0 cellpadding=0 border=0><tr><td><font
      face="Courier New, Courier, mono" size=2> H <br> T <br> S
      </font></td></tr></table>
   </td>
   <td>
      <table cellspacing=0 cellpadding=0 border=0><tr><td><font
      face="Courier New, Courier, mono" size=2> E <br> H <br> P
      </font></td></tr></table>
   </td>
```

```
HELLO
THIS IS MY
SPAM MESSAGE
```

```
HTS
EHP
LIA
LSM
O
```

Figure 4. Removing HTML Tags and Vertical Slicing (a) HTML full text and its tags. (b) HTML file that a recipient sees. (c) Text after removing all tags which is a non-meaning text.

4.1.3. Eliminating Stop Words

First all sequences of whitespace characters (tabs, spaces and new line characters) is replaced by a single space. Then like any other text classification task, stop words should be eliminated which is 20-30% of total word counts and does not have any effect on characteristics of email message.

4.1.4. Stemming

In this step root (stem) of a word should be found (e.g. root of ‘users’ and ‘using’ is ‘use’). Some examples of the rules include:

□ If the word ends in 'ed', remove the 'ed' .
□ If the word ends in 'ing', remove the 'ing' .. etc.

Sometimes stemming rules is more difficult. (e.g. stem of ‘ran’ is ‘run’). Some useful algorithms for stemming are as follows [20]:

□ Brute Force Algorithms
□ Suffix Stripping Algorithms
□ Lemmatization Algorithms
□ Stochastic Algorithms
□ Hybrid Approaches
□ Affix Stemmers
□ Matching Algorithms

4.1.5. Frequency counting

Usually $tf - idf$ is used to count frequencies. To do this, the Term Frequency, Document Frequency and Document Length should be found. An email message is represented as a vector: ($w_1, w_2, \ldots, w_n$). w_i is a word in email message and a dimension in that vector space and is found by the following equation.

$$w_i = tf_i * idf_i \qquad (1)$$

$$= tf_i * \log(N / df_i) \qquad (2)$$

Where tf_i is the number of times the term occurred in the email message and idf_i is $\log(N/df_i)$; and df_i is the number of email messages contains term i and N is the total number of emails in the collection. $tf-idf$ shows the importance of a word in an email message. When lots of this word appears in an email message, and it is not appears too much in other email messages of the collection, the importance of it is higher.

$$D_i = (w_{i1}, w_{i2}, \dots, w_{in}) \tag{3}$$

$$= (tf-idf_{i1}, tf-idf_{i2}, \dots, tf-idf_{in}) \tag{4}$$

$$D_j = (tf-idf_{j1}, tf-idf_{j2}, \dots, tf-idf_{jn}) \tag{5}$$

$$similarity(D_i, D_j) = \frac{\sum_{k=1}^{n} w_{ik} * w_{jk}}{\sqrt{\sum_{k=1}^{n} w_{ik}^{2}} * \sqrt{\sum_{k=1}^{n} w_{jk}^{2}}} \tag{6}$$

When we have a new email message, similarity matching is used. A perfect classifier can classify messages in its vector space. This classifier could be a Neural Network, K-Nearest Neighbor, Bayesian Classifier or any other classifier. In classifiying there is two steps:

- Given training messages, compute a prototype vector for each class. (spam, ham)
- Given testing messages, assign to topic whose prototype (centroid) is nearest using similarity.

5. K Nearest Neighbor

To classify message d into class c: First k messages, nearest neighbors of d is found and defined as k-neighborhood N; then number of email messages n in N is counted that belong to c; then estimate $p(c\,|\,d)$ as n/k. Classification time is proportional to the training set size. Therefore this method is simple and fast in training and classification. Also we have only one calculation; $spam$ as c in $p(c\,|\,d)$, and $p(ham\,|\,d)$ is not needed.

6. Bayesian classifier

To classify a target message d, Bayesian classifiers select the class c that is most probable with respect to the email text using Bayes rule:

$$c(d) = \arg\max_{c \in C} [\, p(c\,|\,d) \,] \tag{7}$$

$$= \arg\max_{c \in C} [\, p(c) p(d\,|\,c) \,] \tag{8}$$

A statistical compression model M_c, built from the training data for class c, can be used to estimate the conditional probability of the message given the class $p(d\,|\,c)$:

$$c(d) = \arg\max_{M \in \{M_{ham}, M_{spam}\}} [\, p(c) p_M(d) \,] \tag{9}$$

In (9) the class priors may be ignored, equivalent to setting $p(c)$ to 0.5 for both spam and ham [1]. But to have an accurate spam filter it is better to obtain class priors from DataSet, previously labeled as ‘spam’ or ‘ham’. This labeling can be performed automatically or labeled by hand (users’ feedbacks). $p(spam)$ may be approximated by the ratio of the number of spams to the number of all messages in the training data.

Varying class priors can make different results. It may be problematic for researchers because it makes solution superiority more difficult to establish. A classifier that performs better than another on a dataset with 80% spam may perform worse on one with 40% spam [19]. $p(spam)$ and $p(ham)$ could be estimated. But it should be kept in mind that priors vary and static values are unrealistic.

In (8) probability estimates for $p(d \mid spam)$ and $p(d \mid ham)$ are computed as a product over all character occurrences in the email text. With increasing message length, these estimates converge towards zero. Therefore instead of $p_M(d)$ in (9) we can use $spam\_score(d)$.

$$spam\_score(d) = \frac{p(d \mid spam)^{1/|d|}}{p(d \mid spam)^{1/|d|} + p(d \mid ham)^{1/|d|}} \tag{10}$$

In (10) $|d|$ denotes the length of the message text. This helps produce scores that are relatively independent of message length [4]. $p(d \mid c)$ (c is $spam$ or ham), can be estimated as k/n where n is number of email messages in class c and k is number of email messages which have similarity more than a threshold. Threshold could be 0.5. Similarity is obtained from (6). Bayes classifier is useful in spam filtering.

6.1. Naïve Bayes

It could be assumed that the attributes of the vector D_i are independent in each class. In other words, the presence of one of the words in a message does not influence the probability of presence of other words. This is not a correct assumption, but it simplify the task.

$$p(d \mid c) = \prod_{i=1}^{n} p(w_i \mid c) \tag{11}$$

7. Error Costs

Viewing the filter as a spam detector, a spam message is a positive instance and a legitimate message is a negative instance. Judging a legitimate email to be spam (a false positive error) is usually far worse than judging a spam email to be legitimate (a false negative error). A false negative simply causes slight irritation, i.e., the user sees an undesirable message. On the other hand, a false positive can be critical: if spam is deleted permanently from a mail server, it can mean a legitimate message has been discarded without a trace. If spam is moved to a low-priority mail folder for later human scanning, false positives may be much more tolerable. Filtering even a small amount of legitimate email defeats the purpose of filtering because it forces the user to start reviewing the spam folder for missed messages and cause a user to reconsider the value of spam filtering. This argues for assigning a very high cost to false positive errors [1]. Therefore instead of $p(spam)$ we can use PCF_{spam}:

$$PCF_{spam} = \frac{p(spam).\cos t(FN)}{p(spam).\cos t(FN) + p(ham).\cos t(FP)} \tag{12}$$

True costs of filtering errors may simply be unknown to the data mining researcher, or may be known only approximately [1]. If we assume that the cost of a false positive (that is, of classifying a legitimate message as spam) is about ten times that of a false negative, this reduces to:

$$PCF_{spam} = \frac{p(spam)}{p(spam) + p(ham)*10} \tag{13}$$

When most spams are harmless, False Positive error cost is much more than False Negative error. By a false negative error, it means that user sees an email that just don’t like. But false negative error in scams has more cost than spams, which means that user sees an email by criminal intent that may harm him.

8. Dynamically Changing

Class priors of hams and spams change over time and assuming them static is not realistic. Over time, the writers of scams can change any number of features, such as the motive for money transfer of the name and title subject of who is in need of help. Moreover, sections of the story or plead may change as well, such as when a paragraph of the message is removed or added. It is not uncommon to find that over time, there is a continual tweaking of the scam,where a part of the scam is changed while keeping most parts in common [2].

Something of an arms race has emerged between spammers and the spam filters used to combat spam. As filters are adapted to contend with today's types of spam emails, the spammers alter, obfuscate and confuse filters by disguising their emails to look more like legitimate email. This dynamic nature of spam & scam raises a requirement for update in any filter that is to be successful over time in identifying spam [14].

9. A Novel Approach

In this section a new approach in scam detection is proposed. This approach uses K-Nearest Neighbor algorithm (KNN), but with modification to Document Similarity equation. In addition classification is not binary as 'scam' or 'not scam': a fuzzy decision is used instead of clear types of classes. Scam Filtering is not a simple text classification task, as scammers are more intelligent than spammers. Scammers change their stories part by part gradually and after a while they may not use the first story at all. Thus time of sending the message is important and scam filtering may be categorized as a temporal data mining task. 'Time' is an inherent attribute in an email message.

The proposed approach affects the classification by the time of data that has been sent: the more aged messages lose their importance as time elapses until messages from very past is removed completely. This is similar to what human brain does, although it forgets events gradually, after a while may not remember them at all. This approach removes the following disadvantages:

- □ A large amount of data is stored and processing this dataset takes lots of time.
- □ Enough memory does not exist to store all the data.
- □ By scams dynamically changing, past data have less confidence than current data and very far past data have no confidence and using them in classification decreases task's accuracy.

In the following, we completely describe implementation of this approach:

At first we have a dataset of messages which has been labeled previously by hand or by users' feedback.

Message label is an integer number between 0 and 100, where 100 means this message is surely a scam and 0 means this message is surely not a scam. Any number between 0 and 100 is acceptable which shows level of being scam in a message; indeed in labeling messages a fuzzy decision has been used. We named this label "Scam-Score Label" and when a message is a record in dataset, Scam-Score Label is added to it as a column of concerned record. Filter is updated by users' feedbacks; any time a user feedback introduces a message as a 'scam', Scam-Score Label of a cluster of messages similar to that message increases by 1 and any time user introduces a message 'not scam', Scam-Score Label decreases by 1. A cluster of similar messages is a group of messages with Document Similarity more than a threshold. Document Similarity here is obtained by (16).

Addition of Scam-Score Label, another label is added which is 'Time Label'. It is added to columns of message data record. This is a number between 0 and 48. 48 shows message is sent in current month and 1 shows 48 months later (four years ago). Data before 4 years ago is thrown away. Each month passes, Time Label of messages decreases by 1. Anytime a message gets zero Time Label it is removed from dataset. Whatever a message has higher time label is has more confidence.

When a new message arrives, a row is added to dataset to record this new data. Its Time Label is 48. We show the new message as a vector X, when attributes of this new message are like D_i in (3) and uses $tf-idf$.

$$X = (x_1, x_2, ..., x_n) \tag{14}$$

$$= (tf-idf_1, tf-idf_2, \ldots, tf-idf_n) \tag{15}$$

To find Scam-Score Label two approaches is proposed in the following; second approach is more accurate.

1. The following equation is used to find the most similar message to this new message (instead of (6)).

$$similarity(D_i, X) = \frac{\sum_{k=1}^{n} w_{ik} * x_k}{\sqrt{\sum_{k=1}^{n} w_{ik}^{2}} * \sqrt{\sum_{k=1}^{n} x_k^{2}}} * t(X) \qquad (16)$$

$t(X)$ is Time-Label of X. Indeed a 1-Nearest Neighbor, by a little change is used. Scam-Score Label of this new message is equal to Scam-Score of the most similar message.

2. Similar to K-Nearest Neighbor, Scam-Score of the new message is average of Scam-Score of k most similar messages.

9.1. Appropriate Alerts for users

Users, who receive this new message, also receive an alert at start of message or end of it. This alert depends on value of Scam-Score:

If Scam-Score is more than 80 it is transferred to a folder named "Scams".

If Scam-Score is between 50 and 80 an alert is sent to receiver like this: "Be Careful! This email appears to be a fraudulent message."

If Scam-Score is between 30 and 50 a lower level message is sent, like this: "Do you know the sender? Be Careful!"

No message is deleted, because at list a little False Positive error is remained.

Users can help us by sending a feedback: by clicking a button "This is scam" and "This is not scam". We use this feedback as described in above to remove misclassifications.

10. Experiments

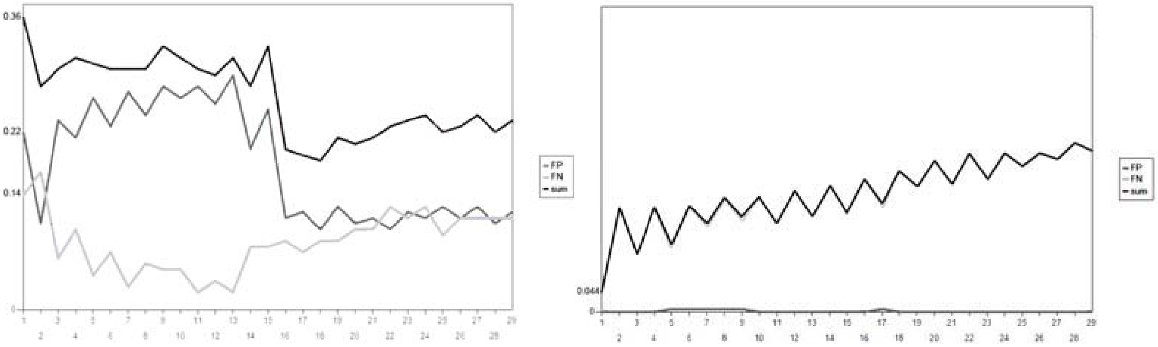
As an experiment we compare two approaches:

- K-Nearest Neighbor
- The new proposed approach (Temporal KNN)

Implementation has been done with C# .NET 2005. Experiments have been tested on a dataset with nearly 500 messages, hams and scams. One half used for training (hams and scams are classified and known in advance). And the other half used for testing (A mixture of scams and hams without knowing which one is scam or ham) Results of experiments are shown in Fig. 5. We named the two approaches KNN (pure K-Nearest Neighbor) and KNN2 (The new proposed approach: Temporal KNN). K in both methods is a variable and changes from 1 t0 30 which is the X axis of following charts.

FP is False Positive Error and FN is False Negative Error. Sum is the whole error which is FN+FP. FN in (b) is nearly overlapped by Sum, because of nearly zero FP Error.

It can be understood from the following charts that in pure K-Nearest Neighbor, higher value for K has better effect, when in the new proposed approach it is not much suitable. In average the new proposed approach has a better result in comparison with pure KNN; Mean Error in pure KNN is around 0.1 when in new approach it is nearly 0.05.



(a) (b)

Fig. 5. Experimental Results on a dataset with 500 messages, Hams, Spams and Scams.
X axis: varying K, FP: False Positive, FN: False Negative, sum: FP+FN
(a) Pure K Nearest Neighbor (KNN) (b) The new modified (KNN2)

11. Conclusions and Feature Works

In this paper mostly used spam detection approaches on static datasets were described: Bayesian Classifier, Naïve Bayes, and K-Nearest Neighbor (KNN). Also a new approach was proposed to match dynamically changes but also simple, accurate and fast. Experiments show this new temporal KNN have better results than pure KNN. The result is also useful for other types of datasets which may have fraudulent intents, like securities and bank frauds.

For Future Works, combination of proposed methods with an expert system could be experimented which may give better results, when allows human interfering in any step like pre-processing, frequency counting, building clusters, define similarity and also other steps.

In addition, types of scams which make use of images rather than text could be experimented. To do this, different image processing approaches combined to Data Mining Methods are needed.

Scam detection is a new field to research and many Data Mining approaches are not still experimented in this task. This is still an open Data mining problem.

Acknowledgment

The authors wish to acknowledge DarleyConsulting.com, Annexia.org, xtdNetand.nl and Untroubled.org to make their datasets free accessible, that we used in our implementation. The authors also wish to acknowledge CodProject.com for their free useful components.

References

[1] Tom Fawcett, " 'In vivo' spam filtering: a challenge problem for data mining", KDD Explorations vol.5 no.2, December 2003.

[2] Airoldi E, Malin B. "Data mining challenges for electronic safety: the case of fraudulent intent detection in e-mails", In Proceedings of the Privacy and Security Aspects of Data Mining Workshop, in conjunction with the 4th IEEE Internation Conference on Data Mining. Brighton, England, November 2004, pp. 57–66.

[3] K. Tretyakov, "Machine learning techniques in spam filtering", Institute of Computer Science, University of Tartu Data Mining Problem-oriented Seminar, MTAT, vol. 3, pp. 60-79, 2004.

[4] Bratko, A. and Filipic, B. "Spam filtering using character-level markov models: Experiments for the TREC 2005 spam track," Text Retrieval Conference, 2005.

[5] Cournane, A., and Hunt, R. “An analysis of the tools used for the generation and prevention of spam”. Computers & Security, 23, 2 (2004), 154-166.

[6] Bratko A., Cormack G. V., Filipic B., Lynam T. R. and Zupan B., “Spam filtering using statistical data compression models”, Journal of Machine Learning Research 7 (Dec 2006), 2699-2720.

[7] I. Androutsopoulos, J. Koutsias, V. Konstantinos, V. Chandrinos, G. Paliouras, C. Spyropoulos, “An evaluation of naive bayesian antispam filtering” in: G. Potamias, V. Moustakis, M. van Someren (Eds.), Proceedings of the ECML 2000 Workshop on Machine Learning in the New Information Age (2000), pp. 9–17.

[8] D. Leonard. “E-mail threats increase sharply”. IDG News Service, December 12, 2002.

[9] I. Androutsopoulos, J. Koutsias, G. Paliouras, V. Karkaletsis, G. Sakkis, C. Spyropoulos, P. Stamatopoulos, “Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach”, in: 4th PKDD workshop on machine learning and textual information access, 2000.

[10] I. Androutsopoulos, G. Paliouras, E Michelakis, “Learning to filter unsolicited commercial email”. Tech rpt 2004/2, NCSR Demokritos, 2004.

[11] H.D. Drucker, D. Wu, V. Vapnik, “Support vector machines for spam categorization”, IEEE Transactions On Neural Networks 10 (5) (1999) 1048–1054.

[12] A. Kolcz, J. Alspector, “SVM-based filtering of e-mail spam with content-specific misclassification costs”, in: Proceedings of TextDM’2001, IEEE ICDM-2001 Workshop on Text Mining, San Jose CA, 2001.

[13] K.R. Gee, “Using latent semantic indexing to filter spam”, in: Proceedings of the 2003 ACM Symposium on Applied Computing (SAC), (ACM, 2003), pp. 460–464.

[14] Delany, S.J., Cunningham, P., Tsymbal, A. & Coyle, L.: “A case-based technique for tracking concept drift in spam filtering”, Knowledge-Based Systems, vol.18(4-5), pp.187-195, 2005

[15] P. Pantel, D. Lin, SpamCop: “A spam classification and organization program”, in: Learning for Text Categorization—Papers from the AAAI Workshop, Madison Wisconsin, 1998 pp. 95–98, (AAAI Technical Report WS-98-05).

[16] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz, “A bayesian approach to filtering junk email”, in: AAAI-98 Workshop on Learning for Text Categorization. Madison, Wisconsin, 1998, pp. 55–62, (AAAI Technical Report WS-98-05).

[17] Carpinter, J. & Hunt, R. “Tightening the net: A review of current and next generation spam filtering tools”. Computers & Security 25(8): 566-578 (2006)

[18] F. Provost, T. Fawcett, and R. Kohavi. “The case against accuracy estimation for comparing induction algorithms. In J. Shavlik, editor, Proceedings of ICML- 98, pages 445-453, San Francisco, CA, 1998. Morgan Kaufmann.

[19] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. Spyropoulos, P. Stamatopoulos, “A memory-based approach to anti-spam filtering for mailing lists”, Information Retrieval 6 (1) (2004) 49–73.

[20] Wikipedia, The free encyclopedia.