Soft Comput (2007) 11:729–740
DOI 10.1007/s00500-006-0116-0

FOCUS

Artificial immune system inspired behavior-based anti-spam filter

Xun Yue · Ajith Abraham · Zhong-Xian Chi · Yan-You Hao · Hongwei Mo

Published online: 2 September 2006

Abstract This paper proposes a novel behavior-based anti-spam technology for email service based on an artificial immune-inspired clustering algorithm. The suggested method is capable of continuously delivering the most relevant spam emails from the collection of all spam emails that are reported by the members of the network. Mail servers could implement the anti-spam technology by using the "black lists" that have been already recognized. Two main concepts are introduced, which defines the behavior-based characteristics of spam and to continuously identify the similar groups of spam when processing the spam streams. Experiment results using real-world datasets reveal that the proposed technology is reliable, efficient and scalable. Since no single technology can achieve one hundred percent spam detection with zero false positives, the proposed method may be used in conjunction with other filtering systems to minimize errors.

Keywords Spam · Clustering algorithm · Artificial immune system · Artificial immune network

X. Yue (✉) · Z.-X. Chi · Y.-Y. Hao
Department of Computer Science and Engineering,
Dalian University of Technology, 116024 Dalian, China
e-mail: yuexun@sdau.edu.cn

X. Yue
College of Information Sciences and Engineering,
Shandong Agricultural University, 271018 Taian, China

A. Abraham
IITA Professorship Program, School of Computer Science and Engineering, Chung-Ang University, 221, Heukseok-dong, Dongjak-gu Seoul 156–756, Republic of Korea

H. Mo
Automation College, Harbin Engineering University,
150001 Harbin, China

1 Introduction

In recent years, personalized anti-spam filters of email client applications based on content filters have now become the standard for spam filters (Hinde 2003, Gyongyi and Garcia-Molina 2005). Spam filters may be implemented using rule-based filters (Mason 2004), nearest-neighbor classifiers (Sakkis et al. 2003), decision trees (Carreras and Marquez 2004) and Bayesian classifiers (Androutsopoulos et al. 2000) etc. The widespread adoption of personalized text classifiers has prompted the spam senders to develop novel techniques that involve addition of words and phrases to the body of an e-mail (Sophos Inc. 2004) with the intent to force the user's classifier to falsely accept the spam e-mail and to further degrade the classifier's performance.

E-mails are filtered inconsistently across different users irrespective of the user's interest. Since users have different interests or business needs, a good anti-spam filtering system should take into account of the different users' needs and interests into consideration and influence the overall decisions and behavior. The main theme of our research is centered on the fact that the collection of user activities can be automatically inferred from the variety of spam available on their users' spam folder. Using our proposed method the email server could then dynamically adapt and collaboratively deliver the same spam behavior-based patterns based on the feedback from the behavior-based patterns of the individual client users in the same e-mail server. Then spam filter server could work like a firewall by protecting the network from spam by blacklisting and other related techniques.

In this paper, we introduce some challenges to tackle the spam filtering problem. First we present how to define the behavior-based characteristics of spam. Second,

we describe how to dynamically and continuously identify the similar groups of spam while processing the data streams. Spam e-mails arrive in the form of continuous, high-volume, fast, and time-varying data streams. Clustering of such data streams is now facing a type of challenge to maintain the clusters in a dynamic environment with frequent updates without frequently performing complete and costly re-clustering.

Compared to other known approaches in the literature, we propose a new incremental immune-inspired clustering approach to mine the data streams. Artificial immune network model is selected because of their dynamic self organizing and highly distributed features, which has no centralized control and uses learning and memory to adapt according to the spam behavior characteristics. Artificial immune network models were developed quite independently by Timmis (2000) and De Castro and Von Zuben (2002) which have been used as alternative biologically motivated approaches to perform data clustering (De Castro and Timmis 2003). In an artificial immune network, repeated exposure to a given antigen considerably enhances the effectiveness of the immune response through the storage of high affinity memory cells from the early infections, where the system continuously improves its capability to perform its recognition of antigens. If e-mail data streams are represented as an antigen universe, a single item of data in the data streams represents an antigen that must be recognized. An antibody produced by the artificial immune network recognizes a set of antigens in the antigen universe by assuming all antigens in the universe can be recognized by the antibody set. Then the number of antibodies present determines the generality/specificity of the clusters and a small number of antibodies will result in few clusters and each cluster represents a very general description of the data. As the number of antibodies is increased, the specificity of the cluster and hence the concept it represents also increases and when a new data set is inserted, an incremental strategy is required to dynamically create the required clusters. Compared to other conventional clustering algorithms, the proposed clustering algorithm is able to dynamically track the ever-increasing large scale information without performing a complete re-clustering and is capable of continuously identifying similar groups of spam.

One commonly used approach for SMTP servers is blacklisting (Harris 2003), in which one simply declines all communications from known spammer IP's. The sender's IP address available in most email server software is an important tool for fighting spam. Since no single approach can achieve one hundred percent (spam detection with zero false positives), our proposed method may be used in conjunction with other filtering systems to minimize errors.

Rest of the paper is organized as follows. In Sect. 2, the related works are presented followed by description of the behavior-based characteristics of spam in Sect. 3. In Sect. 4, we present the behavior-based algorithm using artificial immune network and in Sect. 5, we present the experiment results of our research. Some conclusions are also provided towards the end.

2 Related work

Spam filtering can be implemented at various stages of an email message transmission. These include email filtering by the message transfer agent (MTA), SMTP servers, and email client application using a variety of algorithms and rules.

The email client application based on content-based filters have now become the standard for spam filtering and a variety of algorithms and rules may be implemented using statistical techniques, unsupervised/supervised learning etc. Most of the techniques are based on the premise that it is possible to create a set of rules, exemplars or features that represent the '*spamminess*' of an email, and that if this is over some threshold, is considered to be spam. Such filters have been the focus of considerable interest, with some prior works based on rule-based filters (Mason 2004), nearest-neighbor classifiers (Sakkis et al. 2003), decision trees (Carreras and Marquez 2004) support vector machines (Drucker et al. 1999) and Bayesian classifiers (Androutsopoulos et al. 2000). The Bayesian spam filter and then the Markovian spam filters have dominated the market in the last few years. Unfortunately most Bayesian filters seem to reach a plateau of accuracy at 99.9%. The widespread adoption of personalized text classification has prompted the spam senders to develop a technique dubbed as '*Bayes poison*' which involves adding words and phrases to the body of an e-mail. This attack exploits the fact that most text classification algorithms treat a message as a "bag of words" where a human does not. The intent of this attack is to force the user's classier to falsely accept a single spam e-mail.

Blacklisting is not used very much today. The reason is that most spam headers are falsified containing false sender addresses, often the address of some random person. This leads to innocent people getting blocked because spammers have been sending spam using their mail-address as sender. Katirai (1999) used a genetic programming approach and the performance was compared with Bayes approach. It is also possible to stack

several spam filtering techniques in one filter resulting in a more reliable filter as illustrated by Sakkis et al. (2001).

Oda and White (2003a,b) used an artificial immune system model to protect email users effectively from spam. They tested the spam immune system with the publicly available *SpamAssassin* corpus[1] of spam and non-spam, and extended the original system by looking at several methods of classifying email messages with the detectors produced by the immune system. The resulting system classified the messages with similar accuracy to other spam filters, but used fewer detectors to do so, making it an attractive solution for circumstances where processing time is very important.

Roman et al. (2006) introduced a pre-challenge scheme, which is based on the challenge-response mechanism and takes advantage of some features of email systems. It assumes each user has a challenge that is defined by the user himself/herself and associated with his/her email address, in such a way that an email sender can simultaneously retrieve a new receiver's email address and challenge before sending an email in the first contact. Some new mechanisms are employed in their scheme to reach a good balance between security against spam and convenience to normal email users.

Zorkadis et al. (2005), proposed a novel approach for spam e-mail filtering based on efficient information theoretic techniques for integrating classifiers, for extracting improved features and for properly evaluating categorization accuracy in terms of false positives and false negatives. Random committee-based filters along with ADTree-based ones are efficiently combined through information theory. Empirical results revealed that the proposed information theoretic Boolean features exhibited a remarkably high spam categorization performance.

Özgür et al. (2004). proposed an anti-spam filtering method based on artificial neural networks and Bayesian networks for agglutinative languages in general and for Turkish in particular. Their algorithms have two main components. The first one deals with the morphology of the words and the second one classifies the e-mails by using the roots of the words extracted by the morphological analysis. In the experiments, a total of 750 e-mails (410 spam and 340 normal) were used and a success rate of about 90% was achieved.

Schryen (2006) presented a model of the Internet e-mail infrastructure as a directed graph and a deterministic finite automaton, and draws on automata theory to formally derive the modes of spam delivery possible. The author assessed the effectiveness of anti-spam approaches in terms of coverage of spamming modes.

Agrawal et al. proposed a method to identify spam at the router level and control it via rate limiting. Spam identification is done in two phases. In the first phase, the bulk stream of Email messages are identified and in second phase Bayesian classifiers are applied to identify whether it is a spam. If a bulk Email stream is classified as a spam then it is rated to limit it (e.g. no more than one copy per minute). The proposed method exploits the short time span delivery and bulkiness of spam Emails.

Sasaki and Shinnou (2005) proposed a spam detection technique using the text clustering based on vector space model. The method computes disjoint clusters automatically using a spherical k-means algorithm for all spam/non-spam mails and obtains centroid vectors of the clusters for extracting the cluster description. For each centroid vectors, the label ('spam' or 'non-spam') is assigned by calculating the number of spam email in the cluster. When new mail arrives, the cosine similarity between the new mail vector and centroid vector is calculated. Finally, the label of the most relevant cluster is assigned to the new mail. By using our method, we can extract many kinds of topics in spam/non-spam email and detect the spam email efficiently.

Zhao and Zhang (2005) presented a rough set based model to classify emails into three categories—spam, no-spam and suspicious, rather than two classes (spam and non-spam) in most currently used approaches. By comparing with popular classification methods like Naive Bayes classification, the error ratio that a non-spam is discriminated to spam can be reduced using the proposed model.

Lan and Zhou (2005) presented a filtering mechanism applying the idea of preference ranking. The preference ranking gives the similarity values for nominated emails and spam emails specified by users, so that the ISP/end users can deal with spam emails at filtering points.

Wu et al. (2005) proposed a novel anti-spam system which utilizes visual clues, in addition to text information in the email body, to determine whether a message is spam. Authors analyzed a large collection of spam emails containing images to identify a number of useful visual features for this application. Support Vector Machine was used as the underlying base classifier for anti-spam filtering.

Secker et al. (2003) proposed the artificial immune system for E-mail classification (AISEC) model for web information classification. The B-cells are represented by prototypes of uninteresting e-mail messages and the antigens are represented by the incoming email messages. If a B-cell is activated by an antigen, the message associated to it is labeled as uninteresting and

[1] SpamAssassin Corpus: http://www.spamassassin.org/ publiccorpus

Fig. 1 Example of a spam email

```
From mikeedo@emailisfun.com  Wed Jun 27 04:56:45 2001
Return-Path: <mikeedo@emailisfun.com>
Delivered-To: yyyy@netnoteinc.com
Received: from ns.mediline.co.in (unknown [203.197.32.212]) by
    mail.netnoteinc.com (Postfix) with ESMTP id 43F82130028 for
    <jm7@netnoteinc.com>; Wed, 27 Jun 2001 04:56:44 +0100 (IST)
Received: from gw02_[192.168.224.26] ([4.16.194.53]) by ns.mediline.co.in
    with Microsoft SMTPSVC(5.0.2195.1600); Wed, 27 Jun 2001 09:28:59 +0530
Received: from mail3.emailisfun.com by gw02 with ESMTP; Tue,
    26 Jun 2001 23:01:39 -0400
Message-Id: <00007409198a$00006e26$00006c63@mail3.emailisfun.com>
To: <mikeedo@emailisfun.com>
From: mikeedo@emailisfun.com
Subject: You Won The First Round! claim#        9462                27747
Date: Tue, 26 Jun 2001 23:01:34 -0400
MIME-Version: 1.0
Content-Transfer-Encoding: quoted-printable
X-Priority: 3
X-Msmail-Priority: Normal
Reply-To: wjilknmv@polbox.com
X-Originalarrivaltime: 27 Jun 2001 03:59:00.0584 (UTC) FILETIME=[7F62F680:
    01C0FEBD]
<html>
<body>
<p align=3D"center" style=3D"word-spacing: 0; margin-top: 0; margin-bottom=
: 0"><font size=3D"5" color=3D"#FF0000"><b>You Have Won The First Round!</=
b></font></p>
<p align=3D"center" style=3D"word-spacing: 0; margin-top: 0; margin-bottom=
: 0"><font size=3D"5" color=3D"#FF0000"><b><a href=3D"http://vdfe.weedwaac=
ker.com">Click Here To Collect!</a></b></font></p>
<p align=3D"center" style=3D"word-spacing: 0; margin-top: 0; margin-bottom=
: 0"> </p>
inadvertently received.<br>
Please <a href=3D"http://rmkid.weedwaacker.com">CLICK HERE</a> to be remov=
ed from
future mailings.</font></p>
</body>
</html>
```

sent to a special storage. The stimulation and suppression interactions were performed via incrementing and decrementing the stimulation counter. Clonal selection process is performed and a B-cell could be eliminated because of two processes. First, by elimination of those cells with stimulation counter equal to zero, and second when it makes a bad classification based on user feedback (when the B-cell classification and the user classification does not match).

3 Behavior-based characteristics of spam

We are interested in adaptively discovering the key ongoing activities pattern of the client user's spam. The user activities can be automatically inferred from the variety of data available on the users spam emails. We describe here our initial research on inferring such activities by examining only the user's email.

3.1 Behavior-based characteristics of spam

As illustrated in Fig. 1, each email is represented using both header and body features. Heeader features include the subject line, email addresses, domains of these email addresses, and words judged to be proper nouns. Body features include the bag of words found in the email body. The goal of our behavior-based anti-spam technology is also to give spam a "score" that can be used as input to the developed clusters and to use these numeric score to indicate how likely they are. After studying the typical spam message, we analyzed the behavior-based characteristics of spam and then used this information to further identify and block it.

3.1.1 Sender IP address and SMTP id number

The received lines in email headers contain the list of IP addresses that email has flowed through, as it is passed from one server to another. Many of these lines can be faked but the key line is the first one internal to the recipient's organization, which gives the IP address of the outside machine delivering the message across the internet to the recipient's organization and this line can be trusted. We will refer this line as the '*first internal line*'. Identifying this line in the clients email is the main goal of our research. We could keep looking through the list until we reach a machine listed in the MX record. If the machine listed in the MX record has received from an external sender, we can be very confident that we have found the sender and the 'score' of IP address and the SMTP ID number may be described as follows:

IP address:

$$X = \{x_4, x_3, x_2, x_1\} \quad \text{where } x_1, x_2, x_3, x_4 \in \text{integer}$$

$$\text{Score } \Psi = \sum_{i=1}^{4} x_i \times 10^{i-1}. \tag{1}$$

SMTP ID number:

$$Y = y_n y_{n-1} \cdots y_k \cdots y_3 y_2 y_1 \quad y_n y_{n-1} \cdots y_k \cdots \times y_3 y_2 y_1 \in \text{text}$$

$$\text{Score } \Psi = \text{size}\,(Y) \times 10^6 + \sum_{k=1}^{8} \text{ASCIIvalue}\,(y_k) \times 2^{k-1}. \quad (2)$$

3.1.2 URL link or reply email address in spam messages

Spam offenders use special e-mail addresses or web pages for '*mailto*' links. Messages inform the readers to reply to the mail with a subject of "*Remove*", and there is the major '*call-to-action*' for spammers. The 'score' for this case may be described as follows:

$N = \cdots\cdots @\alpha.\beta \cdots\cdots$ or

$N = \text{http://www.}\alpha.\beta \cdots\cdots$

$\alpha = z_n z_{n-1} \cdots z_k \cdots z_3 z_2 z_1$

$\beta = z_n z_{n-1} \cdots z_k \cdots z_3 z_2 z_1$

$z_n z_{n-1} \cdots z_k \cdots z_3 z_2 z_1 \in \text{text}$

$$\omega 1 = \text{size}(\alpha) \times 10^3 + \sum_{k=1}^{8} \text{ASCIIvalue}(z_k) \times 2^{k-1} \quad (3)$$

$$\omega 2 = \text{size}(\beta) \times 10^3 + \sum_{k=1}^{8} \text{ASCIIvalue}(z_k) \times 2^{k-1} \quad (4)$$

$$\text{Score } \Psi = \omega 1 \times 10^2 + \omega 2. \quad (5)$$

3.2 Algorithm to extract the behavior-based characteristics of spam from text file

A spam that users feed back to e-mail server is a text file and we should extract the characteristics of the spam and develop a '*spam score*' based on Eqs. (1), (2) and (5), that can be used as an input to cluster similar spam emails. The developed tool could automatically extract and calculate the spam score and use them as input to cluster similar spam messages. The algorithm is described below.

3.2.1 The read spam Eigen value algorithm

- *Input : strFileName / file name of spam*
- *Output: ES /a structure of Eigenvalue including server, SMTP ID and email*

Procedure ReadSpamEigenvalue (strFileName)

- *ES.nIPEigenvalue=0: ES.nSmtpidEigenvalue=0:ES.nEmailEigenvalue=0*
- *Read the file into a buffer from the address pbuf*
- *Find the location of 'Message-ID' sentence, let it be p message*
- *If pMessage is not null then*

Begin

- *find the location of the last "received" sentence before pMessage and let it be pReceived;*
- *From location pReceived to location pMessage, that is in this "Received" sentence, do*

Begin
Find the location of 'from', let it be pFrom
Find the location of 'by', let it be pBy
Find the location of 'ID', let it be pID
End

- *If pFrom is not null and pBy is not null then*

Begin
pFrom = pFrom + 5
Read the information between pFrom and pBy, let it be strServerA
Read the IP address in the strServerA, let it be A.B.C.D
Calculate the eigenvalues of A.B.C.D, let it be nIPEigenvalue
ES.nIPEigenvalue=nIPEigenvalue
End

- *If pId is not null then*

Begin
pId = pId+4
Read the SMTPID from pID,let it be strSMTPID
Calculate the eigenvalue of strSMTPID, let it be nSMTPIDEigenvalue
ES.nSMTPIDEigenvalue=nSMTPIDEigenvalue
End

- *Find the location of the last 'mailto:', let it be pEmail*
- *If pEmail is not null then*

  Begin

  PEmail = pEmail+7

  Read the Email from pEmail, let it be strEmail

  Calculate the eigenvalue of strEmail, let it be nEmailEigenvalue

  ES.nEmailEigenvalue = nEmailEigenvalue

  End

End

- *Return ES*

4 Identifying similar groups of spam based on immune-inspired clustering algorithm

In this section, we present the incremental artificial immune system inspired clustering approach which is capable of continuously identifying similar groups of spam.

4.1 Artificial immune network

Artificial immune system (AIS) is a biologically inspired paradigm for information processing. AIS is based on four immunological principles including the immune network theory, danger theory models, the mechanisms of negative selection and the clonal selection principles (Oda and White 2003a,b, Dasgupta 1999).

According to the immune network theory, the receptor molecules contained in the surface of the immune cells present markers, named idiotopes, can be recognized by receptors on other immune cells. These idiotopes are displayed in and/or around the same portions of the receptors that recognize non-self antigens. The recognition of idiotopes on a cell receptor by other cell receptors leads to ever increasing sets of connected cell receptors and molecules. The recognition of antigen by an antibody (cell receptor) leads to network activation, while the recognition of an idiotope by another antibody results in network suppression and the recognition of an antigen by a cell receptor results in network activation and cell proliferation. Clonal selection and expansion is the most accepted theory used to explain how the immune system copes with the antigens. In brief, the clonal selection and expansion theory states that when antigens invade an organism, a subset of the immune cells capable of recognizing these antigens proliferate and differentiate into active or memory cells. The active cells have the primary role of combating the invasion, while the memory cells have long life spans. An interesting phenomenon that occurs during the cellular proliferation is a mutational event with high rates. This mutation process, together with a strong selective force, ensures that the set of memory cells has improved capabilities of recognizing the antigens. As the total number of immune cells contained in an organism is limited and the number of possible invaders is almost limitless, the immune system has to be capable of generating enough cellular diversity. In addition, it has to be capable of extracting some general information contained in these invading antigens so as to prepare for more effective responses for future expositions. This information extraction process is a consequence of the mutation, selection and maintenance of the memory cells.

The computational aspects of immune network theory are relevant and it has proved itself to be a powerful model for computational systems. The immune network model discussed in this paper Artificial Immune NETwork (AINet) (De Castro and Von Zuben 2000, 2001) is an artificial immune network model originally developed to perform automatic data compression. The main role of the standard adaptive algorithm proposed in AINet was to reduce data redundancy, while at the same time extracting relevant information from the data set. The network cells within AINet are represented in a space of same dimensions as the input data, i.e. no dimensionality reduction is performed, but the network size is controlled based upon the immune network's dynamic and meta-dynamic processes. The network cells represent an 'internal image' of the input data set. In AINet, the input data are assumed unlabeled, thus resulting in a type of competitive learning algorithm. The recognition of an input pattern (antigen) results in cell proliferation, mutation and selection as suggested by the clonal selection theory. The recognition of components of the network itself results in network suppression; a process simulated by the elimination of all but one of the self-recognizing cells. By following these two immune principles, the AINet is capable of extracting relevant features contained in a set of input data and at the same time it eliminates data redundancy. The algorithm can be summarized as follows:

4.1.1 Classical AINet algorithm

Initialization: create an initial random population of network antibodies

- Repeat for each antigenic pattern, *do*:
  1. *Clonal selection and expansion*: For each network element, determine its affinity with the antigen presented. Select a number of high affinity elements and reproduce (clone) them proportionally to their affinity.

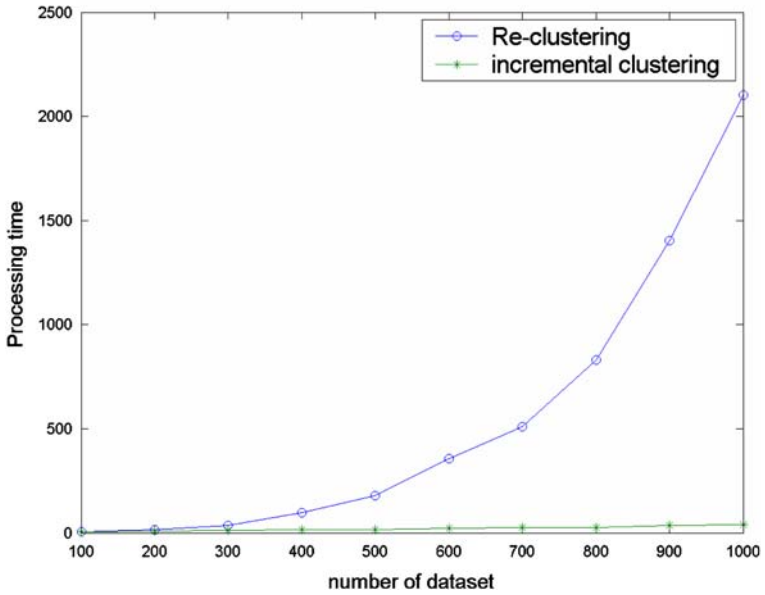

Fig. 2 Comparison of computational cost between ICAINet with incremental clustering and re-clustering

2. *Affinity maturation*: Mutate each clone inversely proportional to affinity. Reselect a number of highest affinity clones and place them into a clonal memory set.
3. *Clonal interactions*: Determine the network interactions (affinity) among all the elements of the clonal memory set.
4. *Clonal suppression*: Eliminate those memory clones whose affinity is less than a pre-specified threshold.
5. *Metadynamics*: Eliminate all memory clones whose affinity with the antigen is less than a pre-defined threshold.
6. *Network construction*: Incorporate the remaining clones of the clonal memory with all network antibodies, resulting in a matrix M of memory antibodies.

- *Network interactions*: Determine the distance between each pair of network antibodies and store these data in a matrix D.
- *Network suppression*: Eliminate all network antibodies whose affinity is less than a pre-specified threshold.
- *Diversity*: Introduce a number of randomly generated cells to the network;

Repeat until a pre-specified number of iterations is performed.

M represents an "internal image" of the input data set and we use the number of antibodies M to determine the generality/specificity of the clusters. It is necessary to use additional tools to automatically identify and separate clusters in the network of cells. The minimal spanning tree (MST) is a graph-theoretic technique which determines the dominant skeletal pattern of a point set by mapping the shortest path of linear, nearest-neighbor connections. The MST network represents a cumulative statistical summary of the spatial characteristics of such graphs and provides a visual, geometric summary. MST could be a useful technique to automatically detect and separate the network clusters.

4.2 Collaborative algorithm based on immune-inspired clustering algorithm

E-mail arrives in the form of continuous, high-volume, fast, and time-varying data streams, and the processing of such streams entails a near real-time constraint. From an artificial immune network perspective, repeated exposure to a given antigen considerably enhances the effectiveness of the immune response through the storage of high affinity memory cells from the early infections, where the system continuously improves its capability to perform its recognition of antigens.

We propose a new incremental immune-inspired clustering approach to mining spam data streams. The novel algorithm aims to enhance the incremental clustering capability of the classical AINet algorithm so that the clusters can be detected within a previously defined cluster. In order to maintain the clusters in a dynamic environment with a high volume of updates without costly re-clustering, we model the data stream as a sequence of time ordered 'windows' that contain a very limited number of objects at a given time. We further create the data clusters incrementally and dynamically based on the similarity between new data and the 'internal image' of the memory network .

4.2.1 Incremental clustering artificial immune network (ICAINet) algorithm

Step1: initialize the sequence of time ordered 'windows' and the parameters of algorithm

Step2. if it is the beginning of the time ordered "windows" then

Step2-1: For each $x_i \in X$ *: do: // Repeat for each antigen,*

Step2-2: if x_i *is first one of* X *then*

Initialization: $Ab = \{y_1, y_2, \ldots y_l\}$

: // create an initial random population of network antibodies;

Step2-3: immune principle 1- proliferation: clonal selection and affinity maturation algorithms are incorporated in algorithm 1

Step2-4: immune principle 2- network suppression: the network interactions allows the network to control its number of cells

Step2-5: if stopping criteria is not met then go to Step2-1

Step3: $Ab \leftarrow M$ *// the set of memory cells M computed previously will be used as network antibodies for the next time ordered 'windows'*

Step4: For each $x_i \in \Delta X$ *: // repeat for each antigenic pattern, do:*

Step 4-1: $D = \{\| d(x_i, y_j)\}$ *where* $x_i \in \Delta X \quad y_j \in M$

Step4-2: If $D > d\min$- *threshold then*

Step4-3: immune principle 1 -proliferation: clonal selection and affinity maturation algorithms are incorporated in algorithm 1,

Step4-4: immune principles 2- network suppression: the network interactions allow the network to control its number of cells.

Else

Step4-5: $M \leftarrow M - x_i$

Endif

Step5 : if the stopping criteria is not met then goto Step 3

Step6: use of the minimal spanning tree (MST) to calculate clusters

Step7: repeat

The main idea of the proposed incremental approach is to use an 'internal image' memory network to represent the input data set in order to reduce data redundancy, whilst at the same time also extract relevant information from the data set. The number of antibodies present would determine the generality/specificity of the clusters and when new data set arrives, the similarity between new data and the previous 'internal image' memory network is computed and is used to create clusters incrementally and dynamically (instead of attempting to directly restructure the clusters).

The key role of the algorithm is to reproduce (cloning) those cells capable of appropriately recognizing the specific pathogens. During the proliferative phase of the immune cells, they are subjected to a controlled mutation event with high rates (somatic hyper-mutation). Those mutated offspring cells that have increased their capability of recognizing a specific pathogen are then selected for survival and further reproduction. This whole mutational process followed by selective events is called affinity maturation of the immune response, because it allows the immune system to increase its capability in recognizing (affinity with) pathogens. A population of immune cells are reproduced using mutation and then natural selection process helps to develop an adaptive immune response. A small number of antibodies will

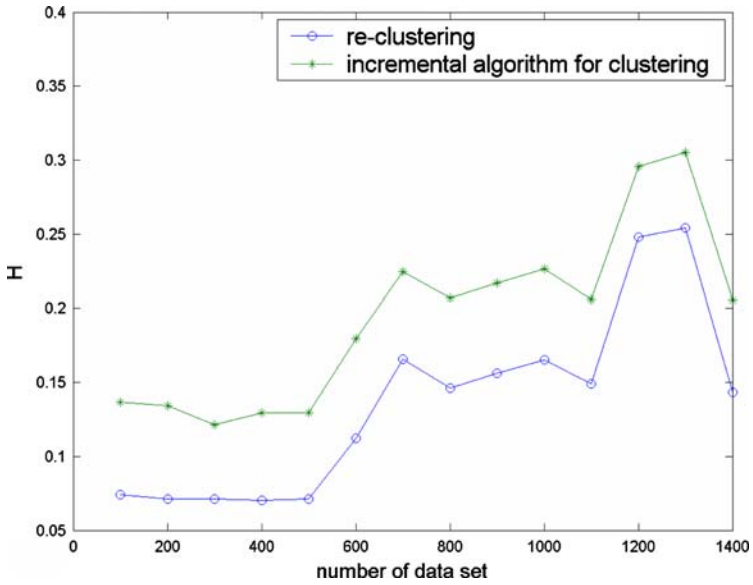

Fig. 3 Performance comparison of homogeneity (H) between incremental algorithm and re-clustering approaches

result in few clusters, and therefore each cluster represents a very general description of the data.

5 Simulated experiments and results

In this section, we present the evaluation of our algorithm and related experiments. The simulations were carried out on a 1.6 GHz Intel Pentium 4 CPU, PC, with 256 M host memory.

5.1 Data set and setup of parameters

We used the April 2003 standard testing corpus (Mason 2004) as the incoming data streams for our experiments. The set of 1,400 messages was shuffled into fourteen time ordered windows, with each shuffle providing an effectively random sequencing of the messages. After some preliminary tests with the AINet algorithm, we used a combination of parameters as depicted in Table 1.

Table 1 AINet parameters used for experiments

Parameter	Value	Meaning
n	4	Best-matching cells taken for each Ag
N	10	Clone number multiplier
qi	0.2	Percentile amount of clones to be Re-selected
gen	40	Maximum number of generations
tp	1	Pruning threshold
ts	0.26	Suppression threshold
mi	4	Learning (hypermutation) rate
Sc	0.01	Pre-specified number of iterations
d-threshold	0.2	Min-threshold

We have used two types of performance measures to study the effectiveness of the proposed algorithm (ICAINet).

(a) Running time according to the changes with the number of spam.
(b) Homogeneity (H) and separation (S) to evaluate the quality of clustering

$$H = \frac{1}{n}\sum_{i=1}^{n} \text{dist}\,(d_i,\ \text{center}_k). \tag{6}$$

The metric H is calculated as the average distance between each data point and the center of the cluster it belongs to:

$$S = \frac{1}{\sum i \neq j |C_i| \cdot |C_j|} \sum_{ii \neq j} |C_i| \cdot |C_j| \cdot \text{dist}\,(\text{center}_i, \text{center}_j). \tag{7}$$

The metric S is calculated as the weighted average distance between cluster centers. The metric H reflects the compactness of the clusters while S reflects the overall distance between clusters.

5.2 Performance evaluation

We first compare the effectiveness of ICAINet for clustering. Figures 2, 3 and 4 illustrate the clustering accuracies and effectiveness of the proposed algorithm for different sizes of data sets. As evident from Fig. 2, ICAINet obtains faster (incremental clustering) clustering

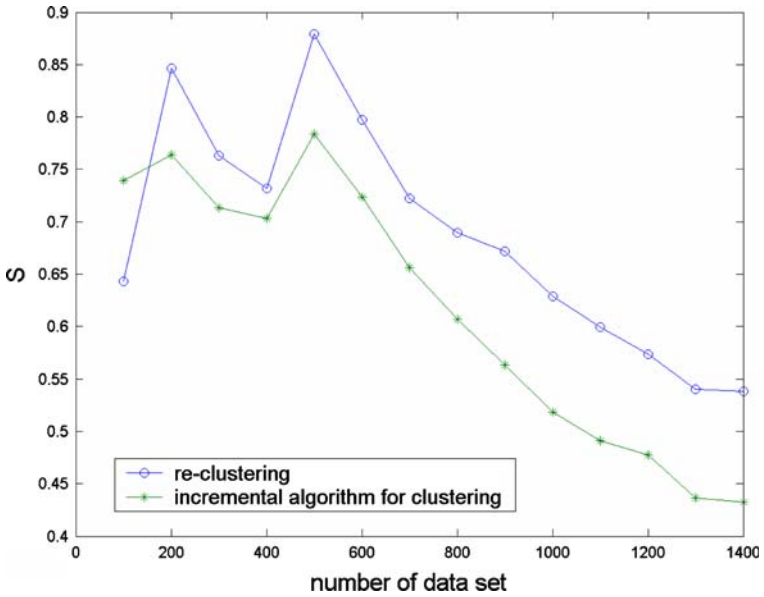

Fig. 4 Performance comparison of separation (S) between incremental algorithm and re-clustering approaches

Table 2 The performance of ICAINet

Number of spam	Running time (s)			Number of immune networks			Number of clusters		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
100	8.9	9.6	9.9	33	43	48	4	7.2	9
200	21.9	22.5	23.1	47	55	60	8	10.1	12
300	30.8	33.4	35.6	55	60	72	8	12.4	16
400	44.2	45.9	48.7	60	66	76	12	18.5	22
500	53.2	58.6	60.3	66	71	80	17	21.2	26
600	69.6	71.1	75.7	71	76	88	12	15.9	20
700	79.1	84.5	88.2	68	75	85	22	29.3	32
800	89.3	97.2	101.2	72	85	92	24	30.6	36
900	104.2	111.3	123.7	80	88	97	8	15.8	21
1,000	118.7	124.8	137.7	76	78	92	20	26.6	31
1,100	126.5	137.5	148.3	75	80	88	19	22.2	34
1,200	133.5	149.7	155.8	83	93	101	14	16.7	28
1,300	156.8	162.3	170.2	88	92	103	24	30.2	38
1,400	163.9	174.8	182.7	86	99	112	29	36.0	42

results than conventional re-clustering especially for bigger data sets.

The computational cost of ICAINet algorithm for clustering crucially depends on the size increment of the incremental data set X and size of the initial starting dataset. We denote the computational costs as follow:

$$\mathrm{Time}(t) = \begin{cases} \mathrm{time}(X_0) & t = 1 \\ (\mathrm{Time}(t-1) + \mathrm{Time}(\Delta x) - \mathrm{Time}(\Delta y)) & \text{other} \end{cases} \tag{8}$$

where X_0 is the size of initial starting dataset, Δx is the size of incremental data set, Δy refers to the similarity distance between new data Δx and the internal image memory network computed last time is less than d min _threshold.

A decrease in H or an increase in S results in a better quality of clusters. Figures. 3 and 4 depicts that the re-clustering algorithm results (mean of 50 times) in a lower value of H and a higher value of S when compared to the incremental algorithm. The ICAINet incremental algorithm obtained better clustering results than the conventional re-clustering algorithm.

Table 2 depicts the effectiveness of ICAINet algorithm, including running time, number of immune networks and number of clusters. It is obvious that for higher number of spam emails, the quality of clustering is much better. In the other words, the more individual client user feedbacks about the spam, the better spam

filters could be further developed and less likely the individual client users will be caught further.

6 Conclusion and discussions

Spam is a big and complex problem today! Some recent studies by the federal trades commission (FTC) reveal that the public are getting less spam because of better anti-spam technologies. There is a significant financial incentive for spammers to acquire the knowledge to defeat any spam filters. Because of this reason, any robust, long-term anti-spam solution must use multiple techniques and also must involve cooperation among all parties who are interested in finding solutions. Being motivated to protect our networks from the escalating costs of spam, we proposed a behavior-based collaborative algorithm for e-mail servers using an artificial immune system. From the experiment results, the new approach has shown its characteristics of reliability, efficiency and scalability, and could be easily used in conjunction with other filtering systems.

Our main interest in developing the ICAINet was to cluster large data streams, such as those which arise in network monitoring, intrusion detection and related web applications. The future research direction focuses on a multi-layered immune network model which is capable of continuously identifying similar groups of spam and also to minimize the time for processing each stream element and the amount of memory available to the query processor.

Acknowledgements This work is partially supported by National Nature Science Foundation of China, No.60305007. Authors would like to thanks the three anonymous referees for the constructive comments that helped to enhance the quality and presentation of this paper.

References

Androutsopoulos I, Koutsias J, Chandrinos KV, Paliouras G, Spyropoulos CD (2000) An evaluation of naive Bayesian anti-spam filtering. In: Proceedings of the workshop on machine learning in the new information age

Carreras X, Marquez L (2004) Boosting trees for anti-spam email filtering. In: Proceedings of RANLP-01, 4th international conference on recent advances in natural language processing

Dasgupta D (1999) Artificial immune systems and their applications. Springer, Berlin Heidelberg New York

Drucker H, Wu D, Vapnik VN (1999) Support vector machines for spam categorization. IEEE Trans Neural Netw 10(5):1048–1054

De Castro LN, Timmis JI (2003) Artificial immune systems as a novel soft computing paradigm. Softcomputing 7:526–544

De Castro LN, Von Zuben FJ (2000) An evolutionary immune network for data clustering. In: Proceedings of the IEEE Brazilian symposium on neural networks, pp 84–89

De Castro LN, Von Zuben FJ (2001) aiNet: an artificial immune network for data analysis. In: Chapter XII Abbass HA, Saker RA, Newton CS (eds) Data mining: a heuristic approach, Idea Group Publishing, USA, pp 231-259

De Castro LN, Von Zuben FJ (2002) Artificial immune system: a new computational intelligence approach. Springer, Berlin Heidelberg New York

Gyongyi Z. Garcia-Molina H (2005) Spam: it's not just for inboxes anymore. IEEE Comput 38(10):28–34

Harris E (2003) The next step in the spam control war. Greylisting. White paper, August 2003

Hinde S (2003) Spam: the evolution of a nuisance. Comput Secur 22(6): 474–478

Katirai H (1999) Filtering junk E-mail: a performance comparison between genetic programming and Naïve Bayes. MSc Thesis, University of Waterloo

Lan M, Zhou W (2005) Spam filtering based on preference ranking. In: The fifth international conference on computer and information technology, pp 223–227

Mason J (2004) The SpamAssassin homepage. http://www.Spamassassin.org/index.html

Oda T, White T (2003) In: Proceedings of the congress on evolutionary computation (CEC 2003), Canberra, 1, pp 390–396

Oda T, White T (2003) Developing an immunity to spam. Genetic and evolutionary computation — GECCO 2003. In: Genetic and evolutionary computation conference, Chicago, July 12–16, 2003, Proceedings, Part I Series: Lecture notes in computer science, vol 2723. Springer, Berlin Heidelberg New York, pp 231-242

Özgür L, Güngör T, Gürgen F (2004) Adaptive anti-spam filtering for agglutinative languages: a special case for Turkish. Pattern Recognit Lett 25(16): 1819–1831

Roman R, Zhou J, Lopez J (2006) An anti-spam scheme using pre-challenges. Comput Commun (in press) (http://dx.doi.org/10.1016/j.comcom.2005.10.037)

Sakkis G, Androutsopoulos I, Olaiouras G, Karkaletsis V, Spyropoulos CD, and Stamatopoulos P (2001) Stacking classifiers for anti-spam filtering of e-mail. In: Proceedings of conference on empirical methods in natural language processing, Carnegie Mellon University, Pittsburgh

Sakkis G, Androutsopoulos I, Paliouras G, Karkaletsis V, Spyropoulos C, Stamatopoulos P (2003) A memory-based approach to anti-Spam filtering for mailing lists. Inf Retrieval 6:49–73

Sasaki M, Shinnou H (2005) Spam detection using text clustering. In: 2005 international conference on cyberworlds, pp 316–319

Schryen G (2006) A formal approach towards assessing the effectiveness of anti-spam procedures. In: Proceedings of the 39th annual Hawaii international conference on systems sciences, vol 6, pp 129–138

Secker A, Freitas AA, Timmis J (2003) AISEC: an artificial immune system for E-mail classification. In: Sarker R, Reynolds R, Abbass H, Kay-Chen T, McKay R, Essam D, Gedeon T (eds) Proceedings of the congress on evolutionary computation, Canberra, pp 131–139

Sophos Inc. (2004) Field guide to spam http://www.sophos.com/Spaminfo/explained/fieldguide.html. Continuously updated. Accessed March 2, 2004

Spam Whacking Working in US (2006) Comput Fraud Secur, 2006(1):2–3

Timmis J (2000) Artificial immune systems: a novel data analysis technique inspired by the immune network theory. Ph.D. Dissertation, Department of Computer Science, University of Wales

Wu C-T, Cheng K-T, Zhu Q, Wu Y-L (2005) Using visual features for anti-spam filtering. IEEE Int Conf Image Process 3: 509–512

Zhao W, Zhang Z (2005) An email classification model based on rough set theory. In: Proceedings of the 2005 international conference on active media technology (AMT 2005), pp 403–408

Zorkadis V, Karras DA, Panayotou M (2005) Efficient information theoretic strategies for classifier combination, feature extraction and performance evaluation in improving false positives and false negatives for spam e-mail filtering. Neural Netw 18(5–6):799–807