A Local-Concentration-Based Feature Extraction Approach for Spam Filtering

Yuanchun Zhu and Ying Tan, *Senior Member, IEEE*

***Abstract*—Inspired from the biological immune system, we propose a local concentration (LC)-based feature extraction approach for anti-spam. The LC approach is considered to be able to effectively extract position-correlated information from messages by transforming each area of a message to a corresponding LC feature. Two implementation strategies of the LC approach are designed using a fixed-length sliding window and a variable-length sliding window. To incorporate the LC approach into the whole process of spam filtering, a generic LC model is designed. In the LC model, two types of detector sets are at first generated by using term selection methods and a well-defined tendency threshold. Then a sliding window is adopted to divide the message into individual areas. After segmentation of the message, the concentration of detectors is calculated and taken as the feature for each local area. Finally, all the features of local areas are combined as a feature vector of the message. To evaluate the proposed LC model, several experiments are conducted on five benchmark corpora using the cross-validation method. It is shown that the LC approach cooperates well with three term selection methods, which endows it with flexible applicability in the real world. Compared to the global-concentration-based approach and the prevalent bag-of-words approach, the LC approach has better performance in terms of both accuracy and F_1 measure. It is also demonstrated that the LC approach is robust against messages with variable message length.**

***Index Terms*—Artificial immune system (AIS), bag-of-words (BoW), feature extraction, global concentration (GC), local concentration (LC), spam filtering.**

I. Introduction

SPAM, also referred to as unsolicited commercial e-mail (UCE) or unsolicited bulk e-mail (UBE), has caused quite a few problems to our daily-communications life. Specifically, it occupies great resources (including network bandwidth, storage space, etc.), wastes users' time on removing spam from e-box, and costs much money for a loss of productivity. According to the statistics from Commtouch [1], spam made up 72% of the total e-mail traffic on average throughout the fourth quarter in 2008, and peaked at 94% in October. Ferris Research revealed its 2009 estimates for the cost of spam in [2]: spam would cost $130 billion worldwide, which was a 30% raise over the 2007 estimates. It also pointed out that three components constituted the total cost: user productivity cost, help desk cost, and spam control cost. Among them, user productivity cost took up the major portion, which contributed to 85% of the total cost.

To address the problem of spam filtering, many approaches have been proposed to filter spam from e-mail traffic. There are three main related research fields for anti-spam, namely term selection, feature extraction, and classifier design. In the field of classifier design, many machine learning (ML) methods are designed and applied to automatically filter spam. Some prevalent ML methods for spam filtering are naive bayes (NB) [3]–[5], support vector machine (SVM) [6]–[9], k-nearest neighbor (k-NN) [10], [11], artificial neural network (ANN) [12], [13], boosting [14], [15], and artificial immune system (AIS) [7], [16]–[20]. As the performance of an ML method depends on the extraction of discriminative feature vectors, feature extraction methods are crucial to the process of spam filtering. The researches of term selection and feature extraction have also attracted much attention from researchers all over the world. A description and analysis of current term selection methods and feature extraction approaches are given in Section II.

In this paper, we propose a local concentration (LC)-based feature extraction approach for anti-spam by taking inspiration from the biological immune system (BIS). The LC approach is considered to be able to effectively extract position-correlated information from messages by transforming each area of a message to a corresponding LC feature. Two implementation strategies of the LC approach are designed using a fixed-length sliding window and a variable-length sliding window. To incorporate the LC approach into the whole process of spam filtering, a generic LC model is designed and presented. The performance of the LC approach is investigated on five benchmark corpora PU1, PU2, PU3, PUA, and Enron-Spam. Meanwhile, accuracy and F_1 measure are utilized as evaluation criteria in analyzing and discussing the results.

The remainder of the paper is organized as follows. In Section II, we introduce the current term selection methods and feature extraction approaches. The LC-based model and the LC-based feature extraction approach are presented in Sections III and IV, respectively. In Section V, we give the descriptions of the copra and experimental setup, and analyze the results of the validation experiments in detail. Finally, the conclusions are given in Section VI.

Manuscript received January 06, 2010; revised August 14, 2010; accepted December 12, 2010. Date of publication December 30, 2010. This work was supported in part by the National High Technology Research and Development Program of China (863 Program) under Grant 2007AA01Z453, and was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 60875080 and Grant 60673020. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yong Guan.

The authors are with the Key Laboratory of Machine Perception (Ministry of Education) and Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: ytan@pku.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2010.2103060

II. Related Works

This section gives a brief overview of current term selection methods and prevalent feature extraction approaches, both of which have close relationship with our work.

A. Term Selection Methods

1) Information Gain (IG): In information theory, IG, also referred to as Kullback–Leibler distance [21], measures the distance between two probability distributions $P(x)$ and $Q(x)$. In the study of spam filtering, it is utilized to measure the goodness of a given term, i.e., the acquired information for e-mail classification by knowing the presence or absence of a given term t_i. The IG of a term t_i is defined as

$$I(t_i) = \sum_{C \in \{c_s, c_l\}} \left\{ \sum_{T \in \{t_i, \bar{t_i}\}} P(T, C) \log \frac{P(T, C)}{P(T)P(C)} \right\} \tag{1}$$

where C denotes an e-mail's class (c_s and c_l are the spam class and the legitimate e-mail class, respectively), and T denotes whether the term t_i appears in the e-mail (t_i and $\bar{t_i}$ means the presence and the absence of the term t_i, respectively). All the probabilities are estimated from the entire training set of messages.

2) Term Frequency Variance (TFV): Koprinska *et al.* [22] developed a TFV method to select the terms with high variance which were considered to be more informative. For terms occurring in training corpus, the ones occurring predominantly in one category (spam or legitimate e-mail) would be retained. In contrast, the ones occurring in both categories with comparable term frequencies would be removed. In the field of spam filtering, TFV can be defined as follows:

$$T(t_i) = \sum_{C \in \{c_s, c_l\}} \left[T_f(t_i, C) - T_f^{\mu}(t_i)\right]^2 \tag{2}$$

where $T_f(t_i, C)$ is the term frequency of t_i calculated with respect to category C, and $T_f^{\mu}(t_i)$ is the average term frequencies calculated with respect to both categories.

It was shown in [22] that TFV outperformed IG in most cases. However, the comparison between the top 100 terms with highest IG scores and the top 100 terms with highest TFV scores showed that those terms had the same characteristics as follows: 1) occurring frequently in legitimate, linguistic oriented e-mail, and 2) occurring frequently in spam e-mail but not in legitimate e-mail.

3) Document Frequency (DF): Calculated as the number of documents in which a certain term occurs, DF is a simple but effective way for term selection. According to the DF method, the terms whose DF is below a predefined threshold are removed from the set of terms. The DF of a term t_i is defined as follows:

$$D(t_i) = |\{m_j \mid m_j \in M, \text{and } t_i \in m_j\}| \tag{3}$$

where M denotes the entire training set of messages, and m_j is a message in M.

The essence of DF is to remove rare terms. According to its assumption, rare terms provide little information for classification, so the elimination of them does not affect overall performance. As shown in [23], the performance of DF was comparable to that of IG and χ^2 statistic (CHI) with up to 90% term elimination. One major advantage of DF is that its computational complexity increases linearly with the number of training messages.

TABLE I
Three Other Term Selection Methods

Method	Expression
χ^2 statistic	$\chi^2(t_i, c) = \frac{\lvert M\rvert(P(t_i,c)P(\bar{t}_i,\bar{c})-P(\bar{t}_i,c)P(t_i,\bar{c}))^2}{P(t_i)P(\bar{t}_i)P(c)P(\bar{c})}$
Odds ratio	$\tau(t_i, c) = \frac{P(t_i\|c)}{1-P(t_i\|c)} \frac{1-P(t_i\|\bar{c})}{P(t_i\|\bar{c})}$
Term strength	$S(t_i) = P(t_i \in y\|t_i \in x)$

4) Other Term Selection Methods: Term selection methods play quite important roles in spam filtering. Besides IG, TFV, and DF, there are many other methods to measure term importance. For better understanding and comparison, three other common ones [23]–[25] are also listed in Table I, where $c \in \{s, l\}$ is a given category, and $\bar{c} = \{s, l\} \backslash c$. In the term strength method, x and y are an arbitrary pair of distinct but related messages (falling in the same category) in the training corpus. All other variables have the same definitions as those in previous equations.

B. Feature Extraction Approaches

1) Bag-of-Words (BoW): BoW, also referred to as the vector space model, is one of the most popular feature extraction methods in spam filtering applications [24]. It transforms a message to a d-dimensional vector $\langle x_1, x_2, \ldots, x_d \rangle$ by considering the occurrence of preselected terms, which have been selected by utilizing a term selection method. In the vector, x_i can be viewed as a function of the term t_i's occurrence in the message. There are mainly two types of representation about x_i: Boolean type and frequency type [26]. In the case of Boolean type, x_i is assigned to 1 if t_i occurs in the message, otherwise it is assigned to 0. While in the case of frequency type, the value of x_i is calculated as the term frequency of t_i in the message. Schneider [27] showed that the two types of representation performed comparably in his experiments.

2) Sparse Binary Polynomial Hashing (SBPH): SBPH is a method to extract a large amount of different features in e-mail feature extraction [28], [29]. An N-term-length sliding window is shifted over the incoming message with a step of one term. At each movement, 2^{N-1} features are extracted from the window of terms in the following ways. The newest term of the window is always retained, and the other terms in the window are removed or retained so that the whole window is mapped to different features. SBPH performed quite promisingly in terms of classification accuracy as it could extract enough discriminative features. However, so many features would lead to a heavy computational burden and limit its usability.

3) Orthogonal Sparse Bigrams (OSB): To reduce the redundancy and complexity of SBPH, Siefkes *et al.* [29] proposed OSB for extracting a smaller set of features. The OSB method also utilizes a sliding window of N-term-length to extract features. However, different from the SBPH, only term-pairs with a common term in a window are considered. For each window of terms, the newest term is retained, then one of other terms is selected to be retained while the rest of terms are removed. After that, the remaining term-pair is mapped to a feature. Therefore, $N-1$ features would be extracted from a window of N-term-length, which greatly reduce the number of features compared to SBPH. The experiments in [29] showed that OSB slightly outperformed SBPH in terms of error rate.

4) AIS Approaches: Oda *et al.* [16] designed an immunity model for spam filtering. In their work, antibodies, which represented features in the model, were created by utilizing regular expressions. As a result, each antibody could match a mass of antigens (spam) which minimized the antibodies (features) set. To mimic the functions of BIS, different weights were assigned to antibodies. At the beginning of the approach, all the antibodies' weights were initialized to a default value. After a period of running, the weights of the antibodies that had matched more spam than legitimate e-mail, would increase. In contrast, other ones would decrease for the antibodies that had matched more legitimate e-mail. When weights fell below a predefined value, the corresponding antibodies would be culled from the model.

Tan and Ruan [18], [19] proposed a concentration-based feature construction (CFC) method, in which self-concentration and non-self-concentration were calculated by considering terms in self and non-self libraries. The terms in the two libraries were simply selected according to the tendencies of terms. If a term tended to occur in legitimate e-mail, it would be added to the self library. On the contrary, the terms tending to occur in spam would be added to the non-self library. After the construction of the two libraries, each message was transformed to a two-element feature by calculating the self and non-self concentrations of the message.

III. LC-Based Model

A. Background

BIS is an adaptive distributed system with the capability of discriminating "self cells" from "non-self cells." It protects our body from attacks of pathogens. Antibodies, produced by lymphocytes to detect pathogens, play core roles in the BIS. On the surfaces of them, there are specific receptors which can bind corresponding specific pathogens. Thus, antibodies can detect and destroy pathogens by binding them. All the time, antibodies circulate in our body and kill pathogens near them without any central controlling node. In the BIS, two types of immune response may happen: a primary response and a secondary response. The primary response happens when a pathogen appears for the first time. In this case, the antibodies with affinity to the pathogen are produced slowly. After that, a corresponding long-lived B memory cell (a type of lymphocyte) is created. Then when the same pathogen appears again, a secondary response is triggered, and a large amount of antibodies with high affinity to that pathogen are proliferated.

Inspired by the functions and principles of BIS, AIS was proposed in the 1990s as a novel computational intelligence model [20]. In recent years, numerous AIS models have been designed for spam filtering [7], [16]–[20]. One main purpose of both BIS and AIS is to discriminate "self" from "non-self." In the anti-spam field, detectors (antibodies) are designed and created to discriminate spam from legitimate e-mail, and the ways of creation and manipulation of detectors are quite essential in these AIS models.

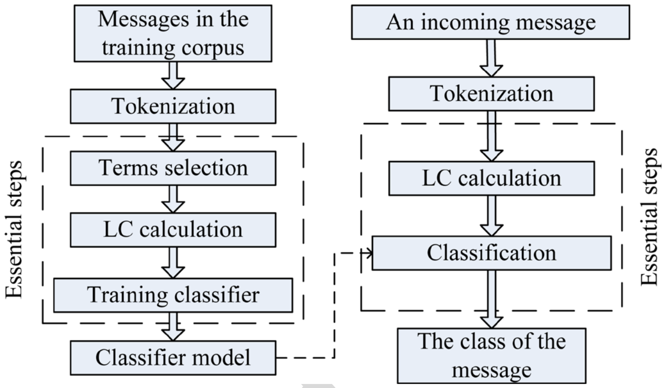


Fig. 1. Training and classification phases of the LC model. (a) Training phase of the model. (b) Classification phase of the model.

In this paper, taking some inspiration from BIS and CFC [18], [19], we propose an LC model, give a way of generating LC features of messages, and apply different term selection methods to the model. To mimic the functions of BIS, one key step is to define corresponding antibodies in an application. In the LC model for spam filtering, antibodies (spam genes and legitimate e-mail genes) are extracted from messages through term selection methods and tendency decisions. In addition, the difference between a primary response and a secondary response shows that the local concentrations of antibodies play important roles in immune behavior. Accordingly, we design two strategies of calculating local concentrations for messages. As local concentrations of antibodies help detect antigens in BIS, it is reasonable to believe that the proposed LC approach will provide discriminative features for spam filtering.

B. Structure of LC Model

To incorporate the LC feature extraction approach into the whole process of spam filtering, a generic structure of the LC model is designed, as is shown in Fig. 1. The tokenization is a simple step, where messages are tokenized into words (terms) by examining the existence of blank spaces and delimiters, while term selection, LC calculation and classification are quite essential to the model:

1) **Term selection:** In the tokenization step of the training phase, messages in the training corpus are transformed into a huge number of terms, which would cause high computational complexity. To reduce computational complexity, term selection methods should be utilized to remove less informative terms. Three term selection methods—IG, TFV, and DF were, respectively, applied to the LC model in our experiments. The experiments were conducted to compare their performances, aiming to show that the proposed model is compatible with various term selection methods. In addition, the experiments could reflect the effectiveness of the three methods. For a detailed introduction of term selection methods, please refer to Section II-A.
2) **LC calculation:** In BIS, antibodies distribute and circulate in bodies. Meanwhile, they detect and destroy specific

pathogens nearby. In a small area of a body, if the concentration of the antibodies with high affinity to a specific pathogen increases above some threshold, the pathogen would be destroyed. Thus, the local concentrations of antibodies determine whether the corresponding pathogens could be culled from the body. Inspired from this phenomenon, we propose an LC-based feature extraction approach. A detail description and analysis of it can be found in Section IV.

3) **Classification:** In the training phase, messages in the training corpus are at first transformed into feature vectors through the steps of tokenization, term selection, and LC calculation. Then the feature vectors are taken as the inputs of a certain classifier, after which a specific classifier model is acquired. Finally, the classifier model is applied to messages for classification. At this stage, our main focus is on the proposal of the LC-based feature extraction approach, so only SVM is adopted in the step of classification. We will investigate the effects of different classifiers on the performance of the model in the future.

IV. LC-Based Feature Extraction Approach

A. Motivation

Feature extraction approaches play quite important roles and attract much attention from researchers in spam filtering. However, there exist two main problems in most current feature extraction approaches, which are the high dimensionality of feature vectors and lack of consideration on the position related information. To address these two problems, we propose an LC approach, by taking inspirations from BIS, where local concentrations of antibodies determine whether the corresponding pathogens can be culled from the body. Mimicking BIS, messages are transformed into feature vectors of local concentrations with respect to “antibodies.” Our LC approach consists of two stages—detector sets (DS) generation and LC features construction, which are elaborated in Sections IV-B and IV-C.

B. Term Selection and DS Generation

Algorithm 1 shows the process of term selection and DS generation. The terms, generated by the step of tokenization, are at first selected by utilizing one certain term selection method, which can be any one of the approaches introduced in Section II-A. In term selection, importance of terms is measured by the criterion defined in the term selection method. Then unimportant (uninformative) terms are removed, and important terms are added to the preselected set which should be initialized as an empty set at the beginning. The purposes of term selection are to reduce the computational complexity of the LC calculation step and reduce possible noises brought by uninformative terms. The noises may occur when the uninformative terms are taken as discriminative features.

Algorithm 1 Term selection and DS generation

```
Initialize preselected set and DS as empty sets;

for each term in the terms set (generated by tokenization) do
    Calculate importance of the term according to a certain term selection methods;
end for

Sort the terms in descending order of the importance;
Add the front $m\%$ terms to the preselected set;

for each term $t_i$ in the preselected set do
    Calculate $Tendency(t_i)$ of the term $t_i$ according to (4);
    if $\| P(t_i \mid c_l) - P(t_i \mid c_s) \| > \theta, \theta \geqslant 0$ then
        if $P(t_i \mid c_l) - P(t_i \mid c_s) < 0$ then
            Add the term to $DS_s$;
        else
            Add the term to $DS_l$;
        end if
    else
        Discard the term;
    end if
end for
```

Taking the preselected set as a source, the DS are built based on tendencies of terms. The tendency of a term t_i is defined as follows:

$$\mathrm{Tendency}(t_i) = P(t_i \mid c_l) - P(t_i \mid c_s) \tag{4}$$

where $P(t_i \mid c_l)$ is the probability of t_i’s occurrence, given messages are legitimate e-mail, and $P(t_i \mid c_s)$ is defined similarly, i.e., the probability of t_i’s occurrence estimated in spam. $\mathrm{Tendency}(t_i)$ measures the difference between the term’s occurrence frequency in legitimate e-mail and that in spam. According to Algorithm 1, the terms, which occur more frequently in spam than in legitimate e-mail, are added to the spam detector set (DS_s), in which the terms represent spam genes. On the contrary, the terms, tending to occur in legitimate e-mail, are added to legitimate e-mail detector set (DS_l), in which the terms represent legitimate genes. The two DS (DS_s and DS_l) are then utilized to construct the LC-based feature vector of a message.

C. Construction of LC-Based Feature Vectors

To construct an LC-based feature vector for each message, a sliding window of w_n-term length is utilized to slide over the message with a step of w_n-term, which means that there is neither gap nor overlap between any two adjacent windows. At each movement of the window, a spam genes concentration SC_i and a legitimate genes concentration LC_i are calculated according to the two DS and the terms in the window as follows:

$$\mathrm{SC}_i = \frac{N_s}{N_t} \tag{5}$$

$$\mathrm{LC}_i = \frac{N_l}{N_t} \tag{6}$$

where N_t is the number of distinct terms in the window, N_s is the number of the distinct terms in the window which have been matched by detectors in DS_s, and N_l is the number of the distinct terms in the window which have been matched by detectors in DS_l.

Algorithm 2 shows the construction process of a feature vector. For the purpose of better understanding, we give an example as follows.

Algorithm 2 Construction of an LC-based feature vector

```
Move a sliding window of $w_n$-term length over a given
message with a step of $w_n$-term;

for each position $i$ of the sliding window do
    Calculate the spam genes concentration $\mathrm{SC}_i$ of the
    window according to (5);
    Calculate the legitimate genes concentration $\mathrm{LC}_i$ of the
    window according to (6);
end for

Construct the feature vector likes:
$\langle(\mathrm{SC}_1, \mathrm{LC}_1), (\mathrm{SC}_2, \mathrm{LC}_2), \ldots, (\mathrm{SC}_n, \mathrm{LC}_n)\rangle$.
```

Suppose one message is *"If you have any questions, please feel free to contact us ...,"* $\mathrm{DS}_s = \{\mathrm{free}, \mathrm{any}\}$, $\mathrm{DS}_l = \{\mathrm{If}, \mathrm{you}, \mathrm{questions}, \mathrm{feel}, \mathrm{to}, \mathrm{contact}\}$, and the parameter $w_n = 5$. Then according to Algorithm 2, the first window would be *"If you have any questions"*, $\mathrm{SC}_1 = 0.2$, and $\mathrm{LC}_2 = 0.6$. Similarly, the second window would be *"please feel free to contact"*, $\mathrm{SC}_2 = 0.2$, and $\mathrm{LC}_2 = 0.6$. The rest of SC_i and LC_i can be calculated in the same way by continuing to slide the window and do similar calculation. Finally, a feature vector $\langle(0.2, 0.6), (0.2, 0.6), \ldots\rangle$ can be acquired.

D. Two Strategies of Defining Local Areas

In this section, we present two strategies for defining local areas in messages—using a sliding window with fixed-length (FL) and using a sliding window with variable-length (VL).

1) Using a Sliding Window With Fixed-Length: When a fixed-length sliding window is utilized, messages may have different numbers of local areas (corresponding to different numbers of feature dimensionality), as messages vary in length. To handle this problem, we design two specific processes for dealing with the feature dimensionality of messages.

- **Implementation for short messages:** Suppose a short message has a dimensionality of six, and its dimensionality should be expanded to ten. Two methods could be utilized with linear time computational complexity. One is to insert zeros into the end of the feature vector. For example, a feature vector

$$\langle(\mathrm{SC}_1, \mathrm{LC}_1), (\mathrm{SC}_2, \mathrm{LC}_2), (\mathrm{SC}_3, \mathrm{LC}_3)\rangle$$

can be expanded as

$$\langle(\mathrm{SC}_1, \mathrm{LC}_1), (\mathrm{SC}_2, \mathrm{LC}_2), (\mathrm{SC}_3, \mathrm{LC}_3), (0, 0), (0, 0)\rangle .$$

The other is to reproduce the front features. For example, the feature vector would be expanded as

$$\langle(\mathrm{SC}_1, \mathrm{LC}_1), (\mathrm{SC}_2, \mathrm{LC}_2), (\mathrm{SC}_3, \mathrm{LC}_3), (\mathrm{SC}_1, \mathrm{LC}_1),\\ (\mathrm{SC}_2, \mathrm{LC}_2)\rangle.$$

In our preliminary experiments, the latter one performed slightly better. As we see, the reason is that the front features contain more information than simple zeros. Therefore, the second method is utilized in the LC model.

- **Implementation for long messages:** For long messages, we reduce their dimensionality by discarding terms at the end of the messages (truncation). The reason for choosing this method is that we will not do much reduction of features, but just do small adjustment so that all messages can have the same feature dimensionality. One advantage of it is that no further computational complexity would be added to the model. Preliminary experiments showed that the truncation performed well with no loss of accuracy for the remaining features could provide quite enough information for discrimination. It is shown in [30] and [31] that truncation of long messages can both reduce the computational complexity and improve the overall performance of algorithms.

In the model, we utilize both the two specific processes—implementation for short messages and implementation for long messages. We at first conduct parameter tuning experiments to determine a fixed value for the feature dimensionality. Then, if the dimensionality of a feature vector is greater than the value, the first kind of process will be utilized. Otherwise, the second one will be done. The parameter tuning experiments can ensure that the two specific processes do not affect the overall performance, and the LC model performs best with respect to the parameter of feature dimensionality.

2) Using a Sliding Window With Variable-Length: In this strategy, the length of a sliding window is designed to be proportional to the length of a message. Suppose we want to construct a $2N$-dimensional feature vector for each message. Then for an M-term length message, the length of the sliding window would be set to M/N-term for the message. In this way, all the messages can be transformed into $2N$-dimensional feature vectors without loss of information.

E. Analysis of the LC Model

For the purpose of better understanding, we now analyze the LC model from a statistical point of view. According to Algorithm 1, each term t_j in the DS_l satisfies

$$P(t_j \mid c_l) > P(t_j \mid c_s). \tag{7}$$

Thus, the terms in legitimate e-mail are more likely to fall into the DS_l, compared to those in spam, i.e.,

$$P(t_j \in \mathrm{DS}_l \mid c_l) > P(t_j \in \mathrm{DS}_l \mid c_s). \tag{8}$$

According to Algrithm 2, the LC_i of a sliding window depends on the number of terms (N_l) falling into the DS_l. From a statistical point of view, the probable number of terms (N_l) falling into the DS_l can be regarded as a good approximation of binomial distribution, i.e.,

$$P(N_l = r) \sim B_r(n, p), \tag{9}$$

$$B_r(n, p) = C_n^r p^r (1-p)^{n-r}, \quad r = 0, 1, 2, \ldots, n \tag{10}$$

where $p = P(t_j \in \mathrm{DS}_l)$ and n is the length of the sliding window.

Then we can obtain the expectation value of N_l for legitimate e-mail as follows:

$$\begin{aligned} E(N_l \mid c_l) &= \sum_{r=0}^{n} r C_n^r p_l^r (1-p_l)^{n-r} \\ &= np_l \\ &= nP(t_j \in \mathrm{DS}_l \mid c_l). \end{aligned} \tag{11}$$

Similarly, we can obtain the expectation value of N_l for spam as follows:

$$\begin{aligned} E(N_l \mid c_s) &= \sum_{r=0}^{n} r C_n^r p_s^r (1-p_s)^{n-r} \\ &= np_s \\ &= nP(t_j \in \mathrm{DS}_l \mid c_s). \end{aligned} \tag{12}$$

From (8), (11), and (12), we can obtain

$$E(N_l \mid c_l) > E(N_l \mid c_s) \tag{13}$$

which indicates that a sliding window in a legitimate e-mail tends to contain more legitimate genes than a sliding window in spam from a statistical point of view. Similarly, we can obtain $E(N_s \mid c_l) < E(N_s \mid c_s)$. Thus, an LC_i of a legitimate e-mail tends to be larger than that of spam, and an SC_i of a legitimate e-mail tends to be smaller than that of spam. In conclusion, the LC model can extract discriminative features for classification between spam and legitimate e-mail.

F. Evaluation Criteria

In spam filtering, many evaluation methods or criteria have been designed for comparing performance of different filters [24], [25]. We adopted four evaluation criteria, which were spam recall, spam precision, accuracy, and F_β measure, in all our experiments to evaluate the goodness of different parameter values and do a comparison between the LC approach and some prevalent approaches. Among the criteria, accuracy and F_β measure are more important, for accuracy measures the total number of messages correctly classified, and F_β is a combination of spam recall and spam precision.

1) **Spam recall:** It measures the percentage of spam that can be filtered by an algorithm or model. High spam recall ensures that the filter can protect the users from spam effectively. It is defined as follows:

$$R_s = \frac{n_{s\to s}}{n_{s\to s} + n_{s\to l}} \tag{14}$$

where $n_{s\to s}$ is the number of spam correctly classified, and $n_{s\to l}$ is the number of spam mistakenly classified as legitimate e-mail.

2) **Spam precision:** It measures how many messages, classified as spam, are truly spam. This also reflects the amount of legitimate e-mail mistakenly classified as spam. The higher the spam precision is, the fewer legitimate e-mail have been mistakenly filtered. It is defined as follows:

$$P_s = \frac{n_{s\to s}}{n_{s\to s} + n_{l\to s}} \tag{15}$$

where $n_{l\to s}$ is the number of legitimate e-mail mistakenly classified as spam, and $n_{s\to s}$ has the same definition as in (14).

3) **Accuracy:** To some extent, it can reflect the overall performance of filters. It measures the percentage of messages (including both spam and legitimate e-mail) correctly classified. It is defined as follows:

$$A = \frac{n_{l\to l} + n_{s\to s}}{n_l + n_s} \tag{16}$$

where $n_{l\to l}$ is the number of legitimate e-mail correctly classified, $n_{s\to s}$ has the same definition as in (14), and n_l and n_s are, respectively, the number of legitimate e-mail and the number of spam in the corpus.

4) $\boldsymbol{F_\beta}$ **measure:** It is a combination of R_s and P_s, assigning a weight β to P_s. It reflects the overall performance in another aspect. F_β measure is defined as follows:

$$F_\beta = (1+\beta^2)\frac{R_s P_s}{\beta^2 P_s + R_s}. \tag{17}$$

In our experiments, we adopted $\beta = 1$ as done in most approaches [24]. In this case, it is referred to as F_1 measure.

In the experiments, the values of the four measures were all calculated. However, only accuracy and F_1 measure are used for parameter selection and comparison of different approaches. Because they can reflect overall performance of different approaches, and F_1 combines both R_s and P_s. In addition, R_s and P_s, respectively, reflect different aspects of the performance, and they cannot reflect the overall performances of approaches, separately. That is also the reason why the F_β is proposed. We calculated them just to show the components of F_1 in detail.

V. Experiments

A. Experimental Corpora

We conducted experiments on five benchmark corpora PU1, PU2, PU3, PUA [26], and Enron-Spam[1] [32], using cross validation. The corpora have been preprocessed with removal of attachments, HTML tags, and header fields except for the subject. In the four PU corpora, the duplicates were removed from the corpora for duplicates may lead to over-optimistic conclusions in experiments. In PU1 and PU2, only the duplicate spam, which arrived on the same day, are deleted. While in PU3 and PUA, all duplicates (both spam and legitimate e-mail) are removed, even if they arrived on different days. In the Enron-Spam corpus, the legitimate messages sent by the owners of the mailbox and duplicate messages have been removed to avoid over-optimistic conclusions. Different from the former PU1 corpus (the one released in 2000) and Ling corpus, the corpora are not processed with removal of stop words, and no lemmatization method is adopted. The details of the corpora are given as follows.

1) **PU1:** The corpus includes 1099 messages, 481 messages of which are spam. The ratio of legitimate e-mail to spam is 1.28. The preprocessed legitimate messages and spam are all English messages, received by the first author of [26] over 36 months and 22 months, respectively.

[1] The five corpora are available from the web site: http://www.aueb.gr/users/ion/publications.html.

2) **PU2:** The corpus includes 721 messages, 142 messages of which are spam. The ratio of legitimate e-mail to spam is 4.01. Similar to PU1, the preprocessed legitimate messages and spam are all English messages, received by a colleague of the authors of [26] over 22 months.
3) **PU3:** The corpus includes 4139 messages, 1826 messages of which are spam. The ratio of legitimate e-mail to spam is 1.27. Unlike PU1 and PU2, the legitimate messages contain both English and non-English ones, received by the second author of [26]. While spam are derived from PU1, SpamAssassin corpus and other sources.
4) **PUA:** The corpus includes 1142 messages, 572 messages of which are spam. The ratio of legitimate e-mail to spam is 1. Similar to PU3, the legitimate e-mail contain both English and non-English messages, received by another colleague of the authors of [26], and spam is also derived from the same sources.
5) **Enron-Spam:** The corpus includes 33 716 messages, 17 171 messages of which are spam. The overall ratio of legitimate e-mail to spam is 0.96. It consists of six parts. In the first three parts, the ratio of legitimate e-mail to spam is about 3. While in the last three parts, the ratio of legitimate e-mail to spam is about 0.33. Experiments conducted on the whole Enron-Spam corpus using six-fold cross validation can help investigate the generalization performance of the model.

B. *Experimental Setup*

We conducted all the experiments on a PC with Intel E2140 CPU and 2G RAM. The SVM library LIBSVM is applied for the implementation of the SVM [33].

C. *Experiments of Parameter Selection*

Experiments have been conducted to tune the parameters of the LC model. In this section, we show and analyze the results of experiments on tuning important parameters of the LC model. All these experiments were conducted on PU1 corpus by utilizing ten-fold cross-validation, and IG was used as the term selection method of the models.

1) Selection of a Proper Tendency Threshold: Experiments were conducted with varied tendency threshold θ to investigate the effects of θ on the performance of the LC model. As shown in Fig. 2, the LC model performs well with small θ. However, with the increase of θ, the performance of the LC model degrades in terms of both accuracy and F_1 measure. As we see, the term selection methods have already filtered the uninformative terms, thus the threshold is not quite necessary. In addition, a great θ would result in loss of information. It is recommended that θ should be set to zero or a small value.

2) Selection of Proper Feature Dimensionality: For the LC model using a fixed-length sliding window (LC-FL), short messages and long messages need to be processed specifically so that all the messages can have the same feature dimensionality. Before that, the feature dimensionality needs to be determined. Therefore, we conducted experiments to determine the optimal number of utmost front sliding windows for discrimination. Fig. 3 depicts the results, from which we can see that the model performed best when ten utmost front sliding windows of each message were utilized for discrimination. In this case, all the messages would be transformed into 20-dimensional feature vectors through the specific process introduced in Section IV-D1.

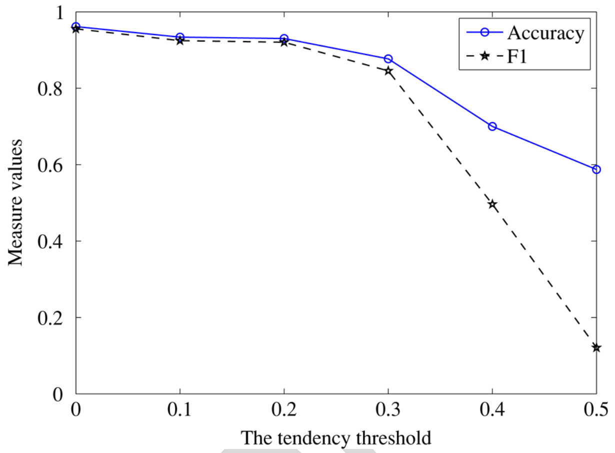


Fig. 2. Performance of the model with varied tendency threshold.

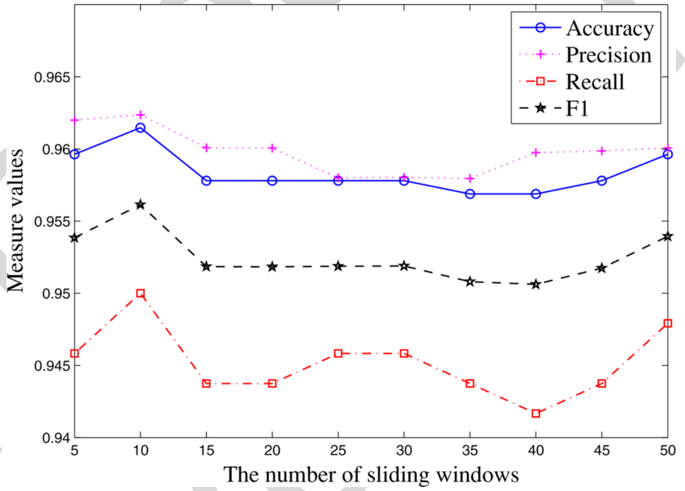


Fig. 3. Performance of the LC-FL model with different window numbers.

For the LC model using a variable-length sliding window (LC-VL), all the messages are directly transformed into feature vectors with the same dimensionality. However, there is still the necessity for determining the feature dimensionality, which corresponds to the number of local areas in a message. We conducted some preliminary experiments on PU1 and found that the LC-VL model performed optimally when the feature dimensionality was set to six or ten.

3) Selection of a Proper Sliding Window Size: For the LC-FL model, the sliding window size is quite essential as it defines the size of local area in a message. Only when the size of local area is properly defined can we calculate discriminative LC vectors for messages. Fig. 4 shows the performance of the LC-FL model under different values of sliding window size. When the size was set to 150 terms per window, the model performed best in terms of both accuracy and F_1 measure. It also can be seen that

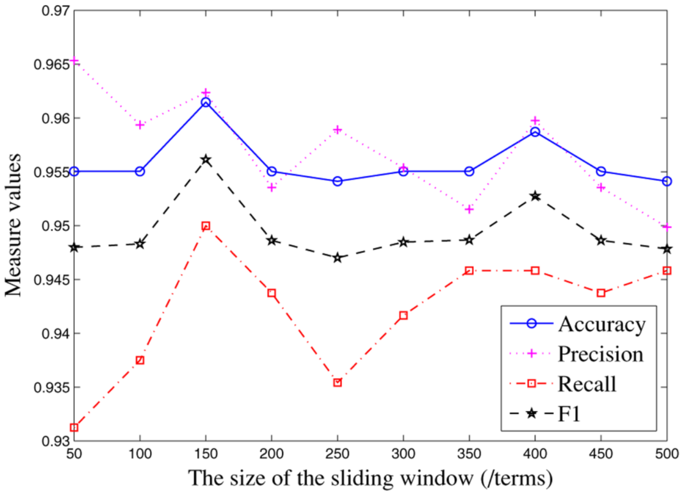


Fig. 4. Performance of the LC-FL model with different sliding window sizes.

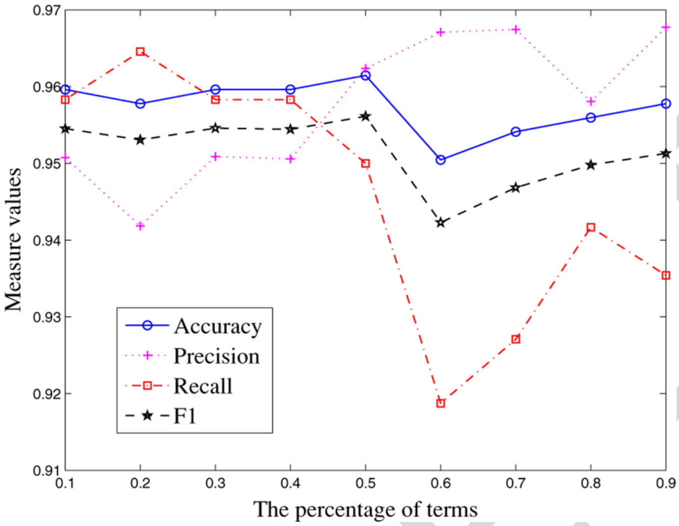


Fig. 5. Performance of the model with different percentage of terms.

the model performed acceptably when the parameter was set to other values.

4) Selection of Optimal Terms Percentage: The phase of term selection plays an important role in the LC model. The removal of less informative terms can reduce computational complexity and improve the overall performance of the model. We conducted experiments to determine the percentage of terms reserved after the phase of term selection. Therefore, the removal of uninformative terms can be maximized while avoiding removing informative ones.

Fig. 5 gives the results of the LC-FL model. When 50% terms were reserved after term selection, the model performed best in terms of both accuracy and F_1 measure. In the following experiments, we set the parameter to 50% for both the LC-FL model and the LC-VL model. We should pay attention to the fact that the model performed quite well when only 10% terms were reserved. This configuration can be applied to cost-sensitive situations.

D. Experiments of the Model With Three Term Selection Methods

To construct discriminative feature vectors for messages, both a term selection method and a feature extraction approach play quite essential roles. To some extent, a feature extraction approach depends on a proper term selection method. Therefore, it is necessary to verify whether the proposed LC approach can be incorporated with prevalent term selection methods.

We conducted comparison experiments of the model with three term selection methods IG, TFV, and DF. All these experiments were conducted on corpora PU1, PU2, PU3, and PUA using ten-fold cross-validation, and on corpus Enron-Spam using six-fold cross-validation. The performances of the LC-FL strategy and the LC-VL strategy are listed in Tables II and III, respectively. The two strategies performed quite well incorporated with any of these term selection methods. On one hand, the experiments showed that the proposed LC strategies could be incorporated with different term selection methods. On the other hand, the experiments had also reflected the effectiveness of the three term selection methods.

E. Comparison Between the LC Model and Current Approaches

In this section, we compared the two LC strategies with some prevalent approaches through the experiments on four PU corpora using ten-fold cross-validation and on corpus Enron-Spam using six-fold cross-validation. The approaches utilized in comparison are Naive Bayes-BoW, SVM-BoW [26], SVM-Global Concentration (SVM-GC), SVM-LC-FL, and SVM-LC-VL.

In Naive Bayes-BoW and SVM-BoW, Naive Bayes and SVM are utilized as their classifiers, respectively, BoW is utilized as the feature extraction approach, and IG is used as the term selection method [26]. In both SVM-LC-FL and SVM-LC-VL, SVM is utilized as their classifier. SVM-GC is a specific configuration of SVM-LC, in which the sliding window size is set to infinite. In such a case, each message is recognized as a whole window, and a two-dimensional feature (including a spam genes concentration and a legitimate genes concentration) is constructed for each message. In this way, it is similar to the CFC approach [18], [19]. The results of these experiments are shown in Table IV.

The comparison with Naive Bayes-BoW and SVM-BoW is mainly to compare the two LC strategies with the prevalent BoW approach. The results show that both of the two LC strategies outperformed the BoW approach in accuracy and F_1 measure. As mentioned before, we take accuracy and F_1 measure as comparison criteria without focusing on precision and recall. Because they are incorporated into the calculation of F_1 measure, and can be reflected by the value of F_1 measure.

The comparison between the two LC strategies and SVM-GC is to verify whether the two LC strategies can extract useful position-correlated information from messages. Both the two LC strategies correspond different parts of a message to different dimensions of the feature vector, while SVM-GC extracts position independent feature vectors from messages. As shown in Table IV, both the two LC strategies outperformed SVM-GC in accuracy and F_1 measure, which verified that the proposed

TABLE II
EXPERIMENTS OF THE LC-FL MODEL WITH THREE DIFFERENT TERM SELECTION METHODS ON CORPORA PU1, PU2, PU3, PUA, AND ENRON-SPAM, UTILIZING CROSS VALIDATION

Corpus	Feature sel.	Precision(%)	Recall(%)	Accuracy(%)	F_1(%)	Feature dim.
PU1	IG	96.04	95.42	96.24	95.73	20
	TFV	95.12	96.88	**96.42**	**95.99**	20
	DF	94.38	96.67	95.96	95.51	20
PU2	IG	95.74	75.71	94.37	84.55	20
	TFV	93.37	74.29	93.80	82.74	20
	DF	90.86	82.86	**94.79**	**86.67**	20
PU3	IG	95.99	95.33	**96.13**	**95.66**	20
	TFV	95.80	95.05	95.91	95.43	20
	DF	95.15	95.99	96.00	95.57	20
PUA	IG	96.01	94.74	**95.26**	**95.37**	20
	TFV	95.83	94.39	94.91	95.10	20
	DF	95.25	94.56	94.74	94.90	20
Enron-Spam	IG	94.07	98.00	96.79	**95.94**	20
	TFV	93.73	98.10	**96.80**	95.79	20
	DF	93.67	98.10	96.68	95.77	20

TABLE III
EXPERIMENTS OF THE LC-VL MODEL WITH THREE DIFFERENT TERM SELECTION METHODS ON CORPORA PU1, PU2, PU3, PUA, AND ENRON-SPAM, UTILIZING CROSS VALIDATION

Corpus	Feature sel.	Precision(%)	Recall(%)	Accuracy(%)	F_1(%)	Feature dim.
PU1	IG	94.85	95.63	95.78	95.21	6
	TFV	95.48	96.04	**96.24**	**95.72**	6
	DF	95.07	96.25	96.15	95.63	6
PU2	IG	95.74	77.86	94.79	85.16	6
	TFV	94.43	79.29	94.79	85.47	6
	DF	92.06	86.43	**95.63**	**88.65**	6
PU3	IG	96.68	94.34	96.03	95.45	6
	TFV	96.46	94.29	95.91	95.32	6
	DF	95.64	95.77	**96.15**	**95.67**	6
PUA	IG	95.60	94.56	**94.91**	**94.94**	6
	TFV	95.22	94.39	94.65	94.67	6
	DF	95.95	93.33	94.56	94.52	6
Enron-Spam	IG	92.44	97.81	**96.02**	**94.94**	6
	TFV	92.07	97.88	95.90	94.77	6
	DF	92.11	97.93	95.95	94.82	6

LC approach (including the LC-FL strategy and the LC-VL strategy) could effectively extract position-correlated information from messages.

Compared to BoW, the proposed LC strategies can greatly reduce feature vector dimensionality, and have advantages in processing speed. As shown in Table V, the two LC strategies outperformed the BoW approach significantly in terms of feature dimensionality and processing speed. However, BoW obtained poor performance when feature dimensionality was greatly reduced [26], while LC strategies performed quite promisingly with a feature dimensionality of 20.

F. Discussion

In Section V-E, it is shown that both the LC-FL strategy and the LC-VL strategy outperform the GC approach on all the corpora. The success of the LC strategies is considered to lie in two aspects. First, the LC strategies can extract position-correlated information from a message by transforming each area of a message to a corresponding feature dimension. Second, the LC strategies can extract more information from messages, compared to the GC approach. As the window size can be acquired when the parameter of the LC strategies are determined, the Global Concentration can be approximately expressed by the weighted sum of Local Concentration, and the weights are correlated with the window size. However, the Local Concentration cannot be deduced from Global Concentration. Thus, the Local Concentration contains more information than the Global concentration does.

The essence of the LC strategies is the definition of local areas for a message. As the local areas may vary with message length, we conducted experimental analysis to see whether drift of message length would affect the performance of the LC strategies. The average message length of corpora PU1, PU2, PU3, PUA, and Enron-Spam are 776 terms, 669 terms, 624 terms, 697 terms, and 311 terms, respectively. It can be seen that the average message length of Enron-Spam is quite shorter than the other four PU corpora. To further demonstrate the difference between Enron-Spam corpus and the PU corpora, the

TABLE IV
COMPARISON BETWEEN THE LC MODEL AND CURRENT APPROACHES

Corpus	Approach	**Precision(%)**	**Recall(%)**	**Accuracy(%)**	F_1(%)	**Feature dim.**
PU1	Naive Bayes-BoW	89.58	99.38	94.59	94.23	600
	SVM-BoW	93.96	95.63	95.32	94.79	600
	SVM-GC	94.97	95.00	95.60	94.99	2
	SVM-LC-FL	95.12	96.88	**96.42**	**95.99**	20
	SVM-LC-VL	95.48	96.04	96.24	95.72	6
PU2	Naive Bayes-BoW	80.77	90.00	93.66	85.14	600
	SVM-BoW	88.71	79.29	93.66	83.74	600
	SVM-GC	95.12	76.43	94.37	84.76	2
	SVM-LC-FL	90.86	82.86	94.79	86.67	20
	SVM-LC-VL	92.06	86.43	**95.63**	**88.65**	6
PU3	Naive Bayes-BoW	93.59	94.84	94.79	94.21	600
	SVM-BoW	96.48	94.67	96.08	95.57	600
	SVM-GC	96.24	94.95	96.05	95.59	2
	SVM-LC-FL	95.99	95.33	96.13	95.66	20
	SVM-LC-VL	95.64	95.77	**96.15**	**95.67**	6
PUA	Naive Bayes-BoW	95.11	94.04	94.47	94.57	600
	SVM-BoW	92.83	93.33	92.89	93.08	600
	SVM-GC	96.03	93.86	94.82	94.93	2
	SVM-LC-FL	96.01	94.74	**95.26**	**95.37**	20
	SVM-LC-VL	95.60	94.56	94.91	94.94	6
Enron-Spam	Naive Bayes-BoW	79.34	99.17	88.41	87.32	600
	SVM-BoW	90.88	98.87	95.13	94.62	600
	SVM-GC	91.48	97.81	95.62	94.39	2
	SVM-LC-FL	94.07	98.00	**96.79**	**95.94**	20
	SVM-LC-VL	92.44	97.81	96.02	94.94	6

TABLE V
PROCESSING SPEED OF THE APPROACHES

Approach	Naive Bayes-BoW	SVM-BoW	Naive Bayes-BoW	SVM-BoW	SVM-GC	SVM-LC-FL	SVM-LC-VL
Seconds/email	0.2	0.5	3	5	0.06	0.07	0.06
Feature dim.	120	120	600	600	2	20	6

Cumulative Distribution Function (CDF) of the message length in PU1 corpus and Enron-Spam corpus are depicted in Fig. 6.

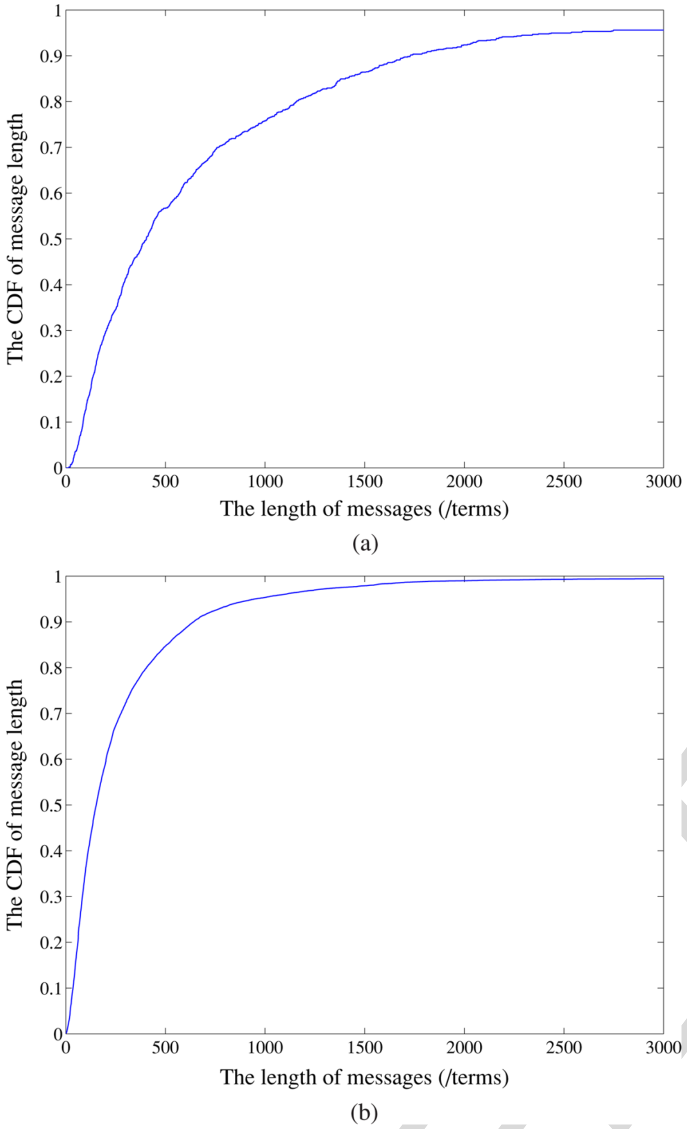
Even though the message length distribution in Enron-Spam corpus is quite different from that of PU corpora, it is shown in Section V-E that the LC strategies perform well on both Enron-Spam corpus and PU corpora. Thus, a preliminary conclusion can be drawn that the LC strategies are robust against variable message length, and the coexistence of short messages and long messages does not decrease the performance of the LC strategies. As long as the average message length is larger than the size of a window, the LC strategies can extract Local Concentration from messages. When almost all the messages become shorter than a window, the performance of the LC strategies would decay and become equivalent to that of the GC approach. However, the window size could be tuned accordingly when the message length changes too much. In that way, the LC strategies can still extract Local Concentration from messages with variable length. In future, we intend to focus on developing adaptive LC approaches, so that the definition of local area can be automatically adapted to the change of message length.

VI. CONCLUSION

We have proposed an LC approach for extracting local-concentration-features for messages. Two implementation strategies of the approach, namely the LC-FL strategy and the LC-VL strategy, have been designed. Extensive experiments have shown that the proposed LC strategies have quite promising performance and advantage in the following aspects.

1) Utilizing sliding windows, both the two LC strategies can effectively extract the position-correlated information for messages.
2) The LC strategies cooperate well with three term selection methods, which endows the LC strategies with flexible applicability in real world.
3) Compared to the prevalent BoW approach and the GC approach, the two LC strategies perform better in terms of both accuracy and F_1 measure.
4) The LC strategies can greatly reduce feature dimensionality and have much faster speed, compared to the BoW approach.
5) The LC strategies are robust against messages with variable message length.

In future work, we intend to incorporate other classifiers into the LC model and investigate their performance under these configurations. In addition, we hope the model can be developed as an adaptive anti-spam system, by taking into account the drift of spam content and the changing interests of users.



(a)

(b)

Fig. 6. CDF curves of message length in PU1 corpus and Enron-Spam corpus. (a) CDF curve of message length in PU1 corpus. (b) CDF curve of message length in Enron-Spam corpus.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and anonymous reviewers for providing insightful comments and constructive suggestions that greatly helped improving the quality of this paper.

REFERENCES

[1] Commtouch, Q4 2008 Internet Threats Trend Report Jan. 2009 [Online]. Available: http://www.pallas.com/fileadmin/img/content/publikationen/Commtouch-Pallas_2008_Q4_Internet_Threats_Trend_Report.pdf

[2] R. Jennings, Cost of Spam is Flattening—Our 2009 Predictions Ferris Research, Jan. 2009 [Online]. Available: http://www.ferris.com/2009/01/28/cost-of-spam-is-flattening-our-2009-predictions/

[3] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, A Baysian Approach to Filtering Junk e-mail AAAI Tech. Rep. WS-98-05, 1998, pp. 55–62.

[4] R. Segal, “Combining global and personal anti-spam filtering,” in *Proc. 4th Conf. Email and Anti-spam (CEAS’ 07)*, 2007***[Please provide page range or location of conference]***.

[5] A. Ciltik and T. Gungor, “Time-efficient spam e-mail filtering using n-gram models,” *Pattern Recognit. Lett.*, vol. 29, no. 1, pp. 19–33, 2008.

[6] H. Drucker, D. Wu, and V. Vapnik, “Support vector machines for spam categorization,” *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1048–1054, Sep. 1999.

[7] G. Ruan and Y. Tan, “Intelligent detection approaches for spam,” in *Proc. Third Int. Conf. Natural Computation (ICNC07)*, Haikou, China, 2007, pp. 1–7.

[8] S. Bickel and T. Scheffer, “Dirichlet-enhanced spam filtering based on biased samples,” *Adv. Neural Inf. Process. Syst.*, vol. 19, pp. 161–168, 2007.

[9] I. Kanaris, K. Kanaris, I. Houvardas, and E. Stamatatos, “Words versus character N-grams for anti-spam filtering,” *Int. J. Artif. Intell. T.*, vol. 16, no. 6, pp. 1047–1067, 2007.

[10] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. D. Spyropoulos, and P. Stamatopoulos, “Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach,” in *Proc. Workshop “Machine Learning and Textual Information Access,” 4th Eur. Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD’ 00)*, 2000, pp. 1–13.

[11] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. D. Spyropoulos, and P. Stamatopoulos, “A memory-based approach to anti-spam filtering for mailing lists,” *Inform. Retrieval*, vol. 6, no. 1, pp. 49–73, 2003.

[12] J. Clark, I. Koprinska, and J. Poon, “A neural network based approach to automated e-mail classification,” in *Proc. IEEE Int. Conf. Web Intelligence (WI’ 03)*, Halifax, Canada, 2003, pp. 702–705.

[13] C.-H. Wu, “Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks,” *Expert Syst. Appl.*, vol. 36, no. 3, pp. 4321–4330, Apr. 2009.

[14] X. Carreras and L. Márquez, “Boosting trees for anti-spam email filtering,” in *Proc. 4th Int. Conf. Recent Advances in Natural Language Processing (RANLP’ 01)*, 2001, pp. 58–64.

[15] J. R. He and B. Thiesson, “Asymmetric gradient boosting with application to spam filtering,” in *Proc. 4th Conf. Email and Anti-spam (CEAS’07)*, 2007***[Please provide page range or location of conference]***.

[16] T. Oda and T. White, “Developing an immunity to spam,” *Lecture Notes Comput. Sci. (LNCS)*, pp. 231–242, 2003.

[17] T. S. Guzella, T. A. Mota-Santos, J. Q. Uchôa, and W. M. Caminhas, “Identification of spam messages using an approach inspired on the immune system,” *Biosystems*, vol. 92, no. 3, pp. 215–225, Jun. 2008.

[18] Y. Tan, C. Deng, and G. Ruan, “Concentration based feature construction approach for spam detection,” in *Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN2009)*, Atlanta, GA, Jun. 14–19, 2009, pp. 3088–3093.

[19] G. Ruan and Y. Tan, “A three-layer back-propagation neural network for spam detection using artificial immune concentration,” *Soft Comput.*, vol. 14, pp. 139–150, 2010.

[20] D. Dasgupta, “Advances in artificial immune systems,” *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 40–49, Nov. 2006.

[21] Wikipedia [Online]. Available: http://en.wikipedia.org/wiki/Information_gain

[22] I. Koprinska, J. Poon, J. Clark, and J. Chan, “Learning to classify e-mail,” *Inform. Sci.*, vol. 177, pp. 2167–2187, 2007.

[23] Y. Yang and J. O. Pedersen, “A comparative study on feature selection in text categorization,” in *Proc. Int. Conf. Machine Learning (ICML’97)*, 1997, pp. 412–420.

[24] T. S. Guzella and M. Caminhas, “A review of machine learning approaches to spam filtering,” *Expert Syst. Appl.*, vol. 36, pp. 10206–10222, 2009.

[25] E. Blanzieri and A. Bryl, A Survey of Learning-Based Techniques of e-mail Spam Filtering University of Trento, Information Engineering and Computer Science Department, Trento, Italy, Tech. Rep. DIT-06-065, Jan. 2008.

[26] I. Androutsopoulos, G. Paliouras, and E. Michelakis, Learning to Filter Unsolicited Commercial E-mail NCSR “Demokritos” Tech. Rep. 2004/2, Oct. 2006, minor corrections.

[27] K.-M. Schneider, “A comparison of event models for naive bayes anti-spam e-mail filtering,” in *Proc. 10th Conf. Eur. Chapter of the Association for Computational Linguistics*, 2003, pp. 307–314.

[28] W. S. Yerazunis, “Sparse binary polynomial hashing and the CRM114 discriminator,” in *Proc. 2003 Spam Conf.*, Cambrige, MA, 2003.

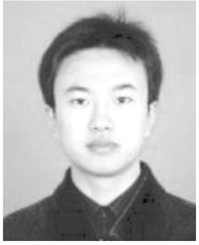
[29] C. Siefkes, F. Assis, S. Chhabra, and W. S. Yerazunis, “Combining winnow and orthogonal sparse bigrams for incremental spam filtering,” *Lecture Notes Comput. Sci.*, vol. 3202/2004, pp. 410–421, 2004.

[30] G. V. Cormack, “Content-based web spam detection,” in *Proc. 3rd Int. Workshop Adversarial Information Retrieval on the Web (AIRWeb’07)*, 2007***[Please provide page range or location of conference]***.

[31] D. Sculley, “Advances in Online Learning-Based Spam Filtering,” Ph.D. dissertation, Tufts Univ., Somerville, MA, 2008.

[32] V. Metsis, I. Androutsopoulos, and G. Paliouras, “Spam filtering with naive bayes—Which naive bayes?,” in *Proc. 3rd Conf. Email and Anti-Spam (CEAS’06)*, Mountain View, CA, 2006, pp. 125–134.

[33] C.-C. Chang and C.-J. Lin, LIBSVM: a Library for Support Vector Machines [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm



Yuanchun Zhu received the B.S. degree in computer science and the B.A. degree in English from Jilin University, Jilin, China, in 2007. He is currently majoring in computer science and working towards the Ph.D. degree at Key Laboratory of Machine Perception (Ministry of Education) and Department of Machine Intelligence, EECS, Peking University, Beijing.

His research interests include machine learning, swarm intelligence, AIS, bioinformatics, information processing, and pattern recognition.

Ying Tan (M’98–SM’02) received the B.S., M.S., and Ph.D. degrees in signal and information processing from Southeast University, Nanjing, China, in 1985, 1988, and 1997, respectively.

Since then, he became a Postdoctoral Fellow and then an Associate Professor at the University of Science and Technology of China. He was a Full Professor, advisor of Ph.D. candidates, and Director of the Institute of Intelligent Information Science of his university. He worked with the Chinese University of Hong Kong in 1999 and in 2004–2005. He was an electee of 100 talent program of the Chinese Academy of Science in 2005. Now, he is a Full Professor, advisor of Ph.D. candidates at the Key Laboratory of Machine Perception (Ministry of Education), Peking University, and Department of Machine Intelligence, EECS, Peking University, and he is also the head of Computational Intelligence Laboratory (CIL) of Peking University. He has authored or coauthored more than 200 academic papers in refereed journals and conferences and several books and book chapters. His current research interests include computational intelligence, artificial immune system, swarm intelligence and data mining, signal and information processing, pattern recognition, and their applications.

Dr. Tan is Associate Editor of the *International Journal of Swarm Intelligence Research* and the *IES Journal B, Intelligent Devices and Systems*, and Associate Editor-in-Chief of the *International Journal of Intelligent Information Processing*. He is a member of the Advisory Board of the *International Journal on Knowledge Based Intelligent Engineering System* and the Editorial Board of the *Journal of Computer Science and Systems Biology* and *Applied Mathematical and Computational Sciences*. He is also the Editor of *Springer Lecture Notes on Computer Science, LNCS 5263, 5264, 6145,* and *6146*, and Guest Editor of special issues of several journals including *Information Science*, *Soft Computing*, *International Journal of Artificial Intelligence*, etc. He was the general chair of the *International Journal on Swarm Intelligence* (ICSI 2010, ICSI 2011) and the program committee chair of ISNN2008. He was honored the second-class National Natural Science Award of China in 2009.

A Local-Concentration-Based Feature Extraction Approach for Spam Filtering

Yuanchun Zhu and Ying Tan, *Senior Member, IEEE*

***Abstract*—Inspired from the biological immune system, we propose a local concentration (LC)-based feature extraction approach for anti-spam. The LC approach is considered to be able to effectively extract position-correlated information from messages by transforming each area of a message to a corresponding LC feature. Two implementation strategies of the LC approach are designed using a fixed-length sliding window and a variable-length sliding window. To incorporate the LC approach into the whole process of spam filtering, a generic LC model is designed. In the LC model, two types of detector sets are at first generated by using term selection methods and a well-defined tendency threshold. Then a sliding window is adopted to divide the message into individual areas. After segmentation of the message, the concentration of detectors is calculated and taken as the feature for each local area. Finally, all the features of local areas are combined as a feature vector of the message. To evaluate the proposed LC model, several experiments are conducted on five benchmark corpora using the cross-validation method. It is shown that the LC approach cooperates well with three term selection methods, which endows it with flexible applicability in the real world. Compared to the global-concentration-based approach and the prevalent bag-of-words approach, the LC approach has better performance in terms of both accuracy and F_1 measure. It is also demonstrated that the LC approach is robust against messages with variable message length.**

***Index Terms*—Artificial immune system (AIS), bag-of-words (BoW), feature extraction, global concentration (GC), local concentration (LC), spam filtering.**

I. Introduction

SPAM, also referred to as unsolicited commercial e-mail (UCE) or unsolicited bulk e-mail (UBE), has caused quite a few problems to our daily-communications life. Specifically, it occupies great resources (including network bandwidth, storage space, etc.), wastes users' time on removing spam from e-box, and costs much money for a loss of productivity. According to the statistics from Commtouch [1], spam made up 72% of the total e-mail traffic on average throughout the fourth quarter in 2008, and peaked at 94% in October. Ferris Research revealed its 2009 estimates for the cost of spam in [2]: spam would cost $130 billion worldwide, which was a 30% raise over the 2007 estimates. It also pointed out that three components constituted the total cost: user productivity cost, help desk cost, and spam control cost. Among them, user productivity cost took up the major portion, which contributed to 85% of the total cost.

To address the problem of spam filtering, many approaches have been proposed to filter spam from e-mail traffic. There are three main related research fields for anti-spam, namely term selection, feature extraction, and classifier design. In the field of classifier design, many machine learning (ML) methods are designed and applied to automatically filter spam. Some prevalent ML methods for spam filtering are naive bayes (NB) [3]–[5], support vector machine (SVM) [6]–[9], k-nearest neighbor (k-NN) [10], [11], artificial neural network (ANN) [12], [13], boosting [14], [15], and artificial immune system (AIS) [7], [16]–[20]. As the performance of an ML method depends on the extraction of discriminative feature vectors, feature extraction methods are crucial to the process of spam filtering. The researches of term selection and feature extraction have also attracted much attention from researchers all over the world. A description and analysis of current term selection methods and feature extraction approaches are given in Section II.

In this paper, we propose a local concentration (LC)-based feature extraction approach for anti-spam by taking inspiration from the biological immune system (BIS). The LC approach is considered to be able to effectively extract position-correlated information from messages by transforming each area of a message to a corresponding LC feature. Two implementation strategies of the LC approach are designed using a fixed-length sliding window and a variable-length sliding window. To incorporate the LC approach into the whole process of spam filtering, a generic LC model is designed and presented. The performance of the LC approach is investigated on five benchmark corpora PU1, PU2, PU3, PUA, and Enron-Spam. Meanwhile, accuracy and F_1 measure are utilized as evaluation criteria in analyzing and discussing the results.

The remainder of the paper is organized as follows. In Section II, we introduce the current term selection methods and feature extraction approaches. The LC-based model and the LC-based feature extraction approach are presented in Sections III and IV, respectively. In Section V, we give the descriptions of the copra and experimental setup, and analyze the results of the validation experiments in detail. Finally, the conclusions are given in Section VI.

Manuscript received January 06, 2010; revised August 14, 2010; accepted December 12, 2010. Date of publication December 30, 2010. This work was supported in part by the National High Technology Research and Development Program of China (863 Program) under Grant 2007AA01Z453, and was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 60875080 and Grant 60673020. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yong Guan.

The authors are with the Key Laboratory of Machine Perception (Ministry of Education) and Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: ytan@pku.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2010.2103060

II. Related Works

This section gives a brief overview of current term selection methods and prevalent feature extraction approaches, both of which have close relationship with our work.

A. Term Selection Methods

1) Information Gain (IG): In information theory, IG, also referred to as Kullback–Leibler distance [21], measures the distance between two probability distributions $P(x)$ and $Q(x)$. In the study of spam filtering, it is utilized to measure the goodness of a given term, i.e., the acquired information for e-mail classification by knowing the presence or absence of a given term t_i. The IG of a term t_i is defined as

$$I(t_i) = \sum_{C\in\{c_s,c_l\}} \left\{ \sum_{T\in\{t_i,\bar{t_i}\}} P(T,C)\log\frac{P(T,C)}{P(T)P(C)} \right\} \tag{1}$$

where C denotes an e-mail's class (c_s and c_l are the spam class and the legitimate e-mail class, respectively), and T denotes whether the term t_i appears in the e-mail (t_i and $\bar{t_i}$ means the presence and the absence of the term t_i, respectively). All the probabilities are estimated from the entire training set of messages.

2) Term Frequency Variance (TFV): Koprinska *et al.* [22] developed a TFV method to select the terms with high variance which were considered to be more informative. For terms occurring in training corpus, the ones occurring predominantly in one category (spam or legitimate e-mail) would be retained. In contrast, the ones occurring in both categories with comparable term frequencies would be removed. In the field of spam filtering, TFV can be defined as follows:

$$T(t_i) = \sum_{C\in\{c_s,c_l\}} \left[T_f(t_i,C) - T_f^{\mu}(t_i)\right]^2 \tag{2}$$

where $T_f(t_i,C)$ is the term frequency of t_i calculated with respect to category C, and $T_f^{\mu}(t_i)$ is the average term frequencies calculated with respect to both categories.

It was shown in [22] that TFV outperformed IG in most cases. However, the comparison between the top 100 terms with highest IG scores and the top 100 terms with highest TFV scores showed that those terms had the same characteristics as follows: 1) occurring frequently in legitimate, linguistic oriented e-mail, and 2) occurring frequently in spam e-mail but not in legitimate e-mail.

3) Document Frequency (DF): Calculated as the number of documents in which a certain term occurs, DF is a simple but effective way for term selection. According to the DF method, the terms whose DF is below a predefined threshold are removed from the set of terms. The DF of a term t_i is defined as follows:

$$D(t_i) = \left|\{m_j \mid m_j \in M, \text{and } t_i \in m_j\}\right| \tag{3}$$

where M denotes the entire training set of messages, and m_j is a message in M.

The essence of DF is to remove rare terms. According to its assumption, rare terms provide little information for classification, so the elimination of them does not affect overall performance. As shown in [23], the performance of DF was comparable to that of IG and χ^2 statistic (CHI) with up to 90% term elimination. One major advantage of DF is that its computational complexity increases linearly with the number of training messages.

TABLE I
Three Other Term Selection Methods

Method	Expression
χ^2 statistic	$\chi^2(t_i,c) = \frac{\lvert M\rvert(P(t_i,c)P(\bar{t}_i,\bar{c})-P(\bar{t}_i,c)P(t_i,\bar{c}))^2}{P(t_i)P(\bar{t}_i)P(c)P(\bar{c})}$
Odds ratio	$\tau(t_i,c) = \frac{P(t_i\lvert c)}{1-P(t_i\lvert c)}\,\frac{1-P(t_i\lvert\bar{c})}{P(t_i\lvert\bar{c})}$
Term strength	$S(t_i) = P(t_i \in y \lvert t_i \in x)$

4) Other Term Selection Methods: Term selection methods play quite important roles in spam filtering. Besides IG, TFV, and DF, there are many other methods to measure term importance. For better understanding and comparison, three other common ones [23]–[25] are also listed in Table I, where $c \in \{s,l\}$ is a given category, and $\bar{c} = \{s,l\}\backslash c$. In the term strength method, x and y are an arbitrary pair of distinct but related messages (falling in the same category) in the training corpus. All other variables have the same definitions as those in previous equations.

B. Feature Extraction Approaches

1) Bag-of-Words (BoW): BoW, also referred to as the vector space model, is one of the most popular feature extraction methods in spam filtering applications [24]. It transforms a message to a d-dimensional vector $\langle x_1, x_2, \ldots, x_d\rangle$ by considering the occurrence of preselected terms, which have been selected by utilizing a term selection method. In the vector, x_i can be viewed as a function of the term t_i's occurrence in the message. There are mainly two types of representation about x_i: Boolean type and frequency type [26]. In the case of Boolean type, x_i is assigned to 1 if t_i occurs in the message, otherwise it is assigned to 0. While in the case of frequency type, the value of x_i is calculated as the term frequency of t_i in the message. Schneider [27] showed that the two types of representation performed comparably in his experiments.

2) Sparse Binary Polynomial Hashing (SBPH): SBPH is a method to extract a large amount of different features in e-mail feature extraction [28], [29]. An N-term-length sliding window is shifted over the incoming message with a step of one term. At each movement, 2^{N-1} features are extracted from the window of terms in the following ways. The newest term of the window is always retained, and the other terms in the window are removed or retained so that the whole window is mapped to different features. SBPH performed quite promisingly in terms of classification accuracy as it could extract enough discriminative features. However, so many features would lead to a heavy computational burden and limit its usability.

3) Orthogonal Sparse Bigrams (OSB): To reduce the redundancy and complexity of SBPH, Siefkes *et al.* [29] proposed OSB for extracting a smaller set of features. The OSB method also utilizes a sliding window of N-term-length to extract features. However, different from the SBPH, only term-pairs with a common term in a window are considered. For each window of terms, the newest term is retained, then one of other terms is selected to be retained while the rest of terms are removed. After that, the remaining term-pair is mapped to a feature. Therefore, $N - 1$ features would be extracted from a window of N-term-length, which greatly reduce the number of features compared to SBPH. The experiments in [29] showed that OSB slightly outperformed SBPH in terms of error rate.

4) AIS Approaches: Oda *et al.* [16] designed an immunity model for spam filtering. In their work, antibodies, which represented features in the model, were created by utilizing regular expressions. As a result, each antibody could match a mass of antigens (spam) which minimized the antibodies (features) set. To mimic the functions of BIS, different weights were assigned to antibodies. At the beginning of the approach, all the antibodies' weights were initialized to a default value. After a period of running, the weights of the antibodies that had matched more spam than legitimate e-mail, would increase. In contrast, other ones would decrease for the antibodies that had matched more legitimate e-mail. When weights fell below a predefined value, the corresponding antibodies would be culled from the model.

Tan and Ruan [18], [19] proposed a concentration-based feature construction (CFC) method, in which self-concentration and non-self-concentration were calculated by considering terms in self and non-self libraries. The terms in the two libraries were simply selected according to the tendencies of terms. If a term tended to occur in legitimate e-mail, it would be added to the self library. On the contrary, the terms tending to occur in spam would be added to the non-self library. After the construction of the two libraries, each message was transformed to a two-element feature by calculating the self and non-self concentrations of the message.

III. LC-Based Model

A. Background

BIS is an adaptive distributed system with the capability of discriminating "self cells" from "non-self cells." It protects our body from attacks of pathogens. Antibodies, produced by lymphocytes to detect pathogens, play core roles in the BIS. On the surfaces of them, there are specific receptors which can bind corresponding specific pathogens. Thus, antibodies can detect and destroy pathogens by binding them. All the time, antibodies circulate in our body and kill pathogens near them without any central controlling node. In the BIS, two types of immune response may happen: a primary response and a secondary response. The primary response happens when a pathogen appears for the first time. In this case, the antibodies with affinity to the pathogen are produced slowly. After that, a corresponding long-lived B memory cell (a type of lymphocyte) is created. Then when the same pathogen appears again, a secondary response is triggered, and a large amount of antibodies with high affinity to that pathogen are proliferated.

Inspired by the functions and principles of BIS, AIS was proposed in the 1990s as a novel computational intelligence model [20]. In recent years, numerous AIS models have been designed for spam filtering [7], [16]–[20]. One main purpose of both BIS and AIS is to discriminate "self" from "non-self." In the anti-spam field, detectors (antibodies) are designed and created to discriminate spam from legitimate e-mail, and the ways of creation and manipulation of detectors are quite essential in these AIS models.

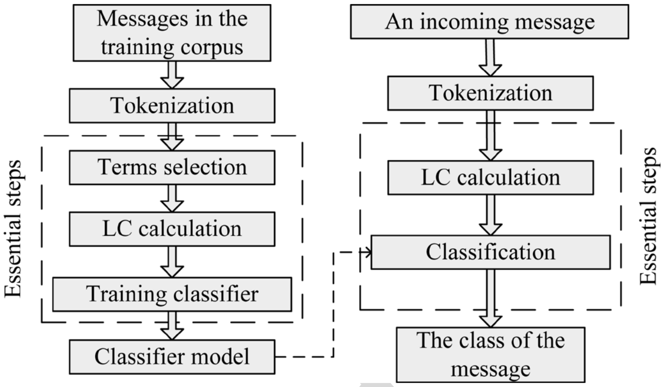


Fig. 1. Training and classification phases of the LC model. (a) Training phase of the model. (b) Classification phase of the model.

In this paper, taking some inspiration from BIS and CFC [18], [19], we propose an LC model, give a way of generating LC features of messages, and apply different term selection methods to the model. To mimic the functions of BIS, one key step is to define corresponding antibodies in an application. In the LC model for spam filtering, antibodies (spam genes and legitimate e-mail genes) are extracted from messages through term selection methods and tendency decisions. In addition, the difference between a primary response and a secondary response shows that the local concentrations of antibodies play important roles in immune behavior. Accordingly, we design two strategies of calculating local concentrations for messages. As local concentrations of antibodies help detect antigens in BIS, it is reasonable to believe that the proposed LC approach will provide discriminative features for spam filtering.

B. Structure of LC Model

To incorporate the LC feature extraction approach into the whole process of spam filtering, a generic structure of the LC model is designed, as is shown in Fig. 1. The tokenization is a simple step, where messages are tokenized into words (terms) by examining the existence of blank spaces and delimiters, while term selection, LC calculation and classification are quite essential to the model:

1) **Term selection:** In the tokenization step of the training phase, messages in the training corpus are transformed into a huge number of terms, which would cause high computational complexity. To reduce computational complexity, term selection methods should be utilized to remove less informative terms. Three term selection methods—IG, TFV, and DF were, respectively, applied to the LC model in our experiments. The experiments were conducted to compare their performances, aiming to show that the proposed model is compatible with various term selection methods. In addition, the experiments could reflect the effectiveness of the three methods. For a detailed introduction of term selection methods, please refer to Section II-A.
2) **LC calculation:** In BIS, antibodies distribute and circulate in bodies. Meanwhile, they detect and destroy specific

pathogens nearby. In a small area of a body, if the concentration of the antibodies with high affinity to a specific pathogen increases above some threshold, the pathogen would be destroyed. Thus, the local concentrations of antibodies determine whether the corresponding pathogens could be culled from the body. Inspired from this phenomenon, we propose an LC-based feature extraction approach. A detail description and analysis of it can be found in Section IV.

3) **Classification:** In the training phase, messages in the training corpus are at first transformed into feature vectors through the steps of tokenization, term selection, and LC calculation. Then the feature vectors are taken as the inputs of a certain classifier, after which a specific classifier model is acquired. Finally, the classifier model is applied to messages for classification. At this stage, our main focus is on the proposal of the LC-based feature extraction approach, so only SVM is adopted in the step of classification. We will investigate the effects of different classifiers on the performance of the model in the future.

IV. LC-Based Feature Extraction Approach

A. Motivation

Feature extraction approaches play quite important roles and attract much attention from researchers in spam filtering. However, there exist two main problems in most current feature extraction approaches, which are the high dimensionality of feature vectors and lack of consideration on the position related information. To address these two problems, we propose an LC approach, by taking inspirations from BIS, where local concentrations of antibodies determine whether the corresponding pathogens can be culled from the body. Mimicking BIS, messages are transformed into feature vectors of local concentrations with respect to "antibodies." Our LC approach consists of two stages—detector sets (DS) generation and LC features construction, which are elaborated in Sections IV-B and IV-C.

B. Term Selection and DS Generation

Algorithm 1 shows the process of term selection and DS generation. The terms, generated by the step of tokenization, are at first selected by utilizing one certain term selection method, which can be any one of the approaches introduced in Section II-A. In term selection, importance of terms is measured by the criterion defined in the term selection method. Then unimportant (uninformative) terms are removed, and important terms are added to the preselected set which should be initialized as an empty set at the beginning. The purposes of term selection are to reduce the computational complexity of the LC calculation step and reduce possible noises brought by uninformative terms. The noises may occur when the uninformative terms are taken as discriminative features.

Algorithm 1 Term selection and DS generation

```
Initialize preselected set and DS as empty sets;

for each term in the terms set (generated by tokenization) do
    Calculate importance of the term according to a
    certain term selection methods;
end for

Sort the terms in descending order of the importance;
Add the front $m\%$ terms to the preselected set;

for each term $t_i$ in the preselected set do
    Calculate Tendency($t_i$) of the term $t_i$ according to (4);

    if $\| P(t_i \mid c_l) - P(t_i \mid c_s) \| > \theta, \theta \geqslant 0$ then
        if $P(t_i \mid c_l) - P(t_i \mid c_s) < 0$ then
            Add the term to $DS_s$;
        else
            Add the term to $DS_l$;
        end if
    else
        Discard the term;
    end if

end for
```

Taking the preselected set as a source, the DS are built based on tendencies of terms. The tendency of a term t_i is defined as follows:

$$\mathrm{Tendency}(t_i) = P(t_i \mid c_l) - P(t_i \mid c_s) \tag{4}$$

where $P(t_i \mid c_l)$ is the probability of t_i's occurrence, given messages are legitimate e-mail, and $P(t_i \mid c_s)$ is defined similarly, i.e., the probability of t_i's occurrence estimated in spam. $\mathrm{Tendency}(t_i)$ measures the difference between the term's occurrence frequency in legitimate e-mail and that in spam. According to Algorithm 1, the terms, which occur more frequently in spam than in legitimate e-mail, are added to the spam detector set (DS_s), in which the terms represent spam genes. On the contrary, the terms, tending to occur in legitimate e-mail, are added to legitimate e-mail detector set (DS_l), in which the terms represent legitimate genes. The two DS (DS_s and DS_l) are then utilized to construct the LC-based feature vector of a message.

C. Construction of LC-Based Feature Vectors

To construct an LC-based feature vector for each message, a sliding window of w_n-term length is utilized to slide over the message with a step of w_n-term, which means that there is neither gap nor overlap between any two adjacent windows. At each movement of the window, a spam genes concentration SC_i and a legitimate genes concentration LC_i are calculated according to the two DS and the terms in the window as follows:

$$\mathrm{SC}_i = \frac{N_s}{N_t} \tag{5}$$

$$\mathrm{LC}_i = \frac{N_l}{N_t} \tag{6}$$

where N_t is the number of distinct terms in the window, N_s is the number of the distinct terms in the window which have been matched by detectors in DS_s, and N_l is the number of the distinct terms in the window which have been matched by detectors in DS_l.

Algorithm 2 shows the construction process of a feature vector. For the purpose of better understanding, we give an example as follows.

Algorithm 2 Construction of an LC-based feature vector

```
Move a sliding window of $w_n$-term length over a given
message with a step of $w_n$-term;
for each position $i$ of the sliding window do
    Calculate the spam genes concentration $\mathrm{SC}_i$ of the
    window according to (5);
    Calculate the legitimate genes concentration $\mathrm{LC}_i$ of the
    window according to (6);
end for
Construct the feature vector likes:
$\langle(\mathrm{SC}_1, \mathrm{LC}_1), (\mathrm{SC}_2, \mathrm{LC}_2), \ldots, (\mathrm{SC}_n, \mathrm{LC}_n)\rangle$.
```

Suppose one message is *"If you have any questions, please feel free to contact us ...,"* $\mathrm{DS}_s = \{\mathrm{free}, \mathrm{any}\}$, $\mathrm{DS}_l = \{\mathrm{If}, \mathrm{you}, \mathrm{questions}, \mathrm{feel}, \mathrm{to}, \mathrm{contact}\}$, and the parameter $w_n = 5$. Then according to Algorithm 2, the first window would be *"If you have any questions"*, $\mathrm{SC}_1 = 0.2$, and $\mathrm{LC}_2 = 0.6$. Similarly, the second window would be *"please feel free to contact"*, $\mathrm{SC}_2 = 0.2$, and $\mathrm{LC}_2 = 0.6$. The rest of SC_i and LC_i can be calculated in the same way by continuing to slide the window and do similar calculation. Finally, a feature vector $\langle(0.2, 0.6), (0.2, 0.6), \ldots\rangle$ can be acquired.

D. Two Strategies of Defining Local Areas

In this section, we present two strategies for defining local areas in messages—using a sliding window with fixed-length (FL) and using a sliding window with variable-length (VL).

1) Using a Sliding Window With Fixed-Length: When a fixed-length sliding window is utilized, messages may have different numbers of local areas (corresponding to different numbers of feature dimensionality), as messages vary in length. To handle this problem, we design two specific processes for dealing with the feature dimensionality of messages.

- **Implementation for short messages:** Suppose a short message has a dimensionality of six, and its dimensionality should be expanded to ten. Two methods could be utilized with linear time computational complexity. One is to insert zeros into the end of the feature vector. For example, a feature vector

  $$\langle(\mathrm{SC}_1, \mathrm{LC}_1), (\mathrm{SC}_2, \mathrm{LC}_2), (\mathrm{SC}_3, \mathrm{LC}_3)\rangle$$

  can be expanded as

  $$\langle(\mathrm{SC}_1, \mathrm{LC}_1), (\mathrm{SC}_2, \mathrm{LC}_2), (\mathrm{SC}_3, \mathrm{LC}_3), (0, 0), (0, 0)\rangle\ .$$

  The other is to reproduce the front features. For example, the feature vector would be expanded as

  $$\langle(\mathrm{SC}_1, \mathrm{LC}_1), (\mathrm{SC}_2, \mathrm{LC}_2), (\mathrm{SC}_3, \mathrm{LC}_3), (\mathrm{SC}_1, \mathrm{LC}_1), (\mathrm{SC}_2, \mathrm{LC}_2)\rangle.$$

  In our preliminary experiments, the latter one performed slightly better. As we see, the reason is that the front features contain more information than simple zeros. Therefore, the second method is utilized in the LC model.
- **Implementation for long messages:** For long messages, we reduce their dimensionality by discarding terms at the end of the messages (truncation). The reason for choosing this method is that we will not do much reduction of features, but just do small adjustment so that all messages can have the same feature dimensionality. One advantage of it is that no further computational complexity would be added to the model. Preliminary experiments showed that the truncation performed well with no loss of accuracy for the remaining features could provide quite enough information for discrimination. It is shown in [30] and [31] that truncation of long messages can both reduce the computational complexity and improve the overall performance of algorithms.

In the model, we utilize both the two specific processes—implementation for short messages and implementation for long messages. We at first conduct parameter tuning experiments to determine a fixed value for the feature dimensionality. Then, if the dimensionality of a feature vector is greater than the value, the first kind of process will be utilized. Otherwise, the second one will be done. The parameter tuning experiments can ensure that the two specific processes do not affect the overall performance, and the LC model performs best with respect to the parameter of feature dimensionality.

2) Using a Sliding Window With Variable-Length: In this strategy, the length of a sliding window is designed to be proportional to the length of a message. Suppose we want to construct a $2N$-dimensional feature vector for each message. Then for an M-term length message, the length of the sliding window would be set to M/N-term for the message. In this way, all the messages can be transformed into $2N$-dimensional feature vectors without loss of information.

E. Analysis of the LC Model

For the purpose of better understanding, we now analyze the LC model from a statistical point of view. According to Algorithm 1, each term t_j in the DS_l satisfies

$$P(t_j \mid c_l) > P(t_j \mid c_s). \tag{7}$$

Thus, the terms in legitimate e-mail are more likely to fall into the DS_l, compared to those in spam, i.e.,

$$P(t_j \in \mathrm{DS}_l \mid c_l) > P(t_j \in \mathrm{DS}_l \mid c_s). \tag{8}$$

According to Algrithm 2, the LC_i of a sliding window depends on the number of terms (N_l) falling into the DS_l. From a statistical point of view, the probable number of terms (N_l) falling into the DS_l can be regarded as a good approximation of binomial distribution, i.e.,

$$P(N_l = r) \sim B_r(n, p), \tag{9}$$

$$B_r(n, p) = C_n^r p^r (1 - p)^{n-r}, \quad r = 0, 1, 2, \ldots, n \tag{10}$$

where $p = P(t_j \in \mathrm{DS}_l)$ and n is the length of the sliding window.

Then we can obtain the expectation value of N_l for legitimate e-mail as follows:

$$\begin{aligned} E(N_l \mid c_l) &= \sum_{r=0}^{n} r C_n^r p_l^r (1-p_l)^{n-r} \\ &= np_l \\ &= nP(t_j \in \mathrm{DS}_l \mid c_l). \end{aligned} \tag{11}$$

Similarly, we can obtain the expectation value of N_l for spam as follows:

$$\begin{aligned} E(N_l \mid c_s) &= \sum_{r=0}^{n} r C_n^r p_s^r (1-p_s)^{n-r} \\ &= np_s \\ &= nP(t_j \in \mathrm{DS}_l \mid c_s). \end{aligned} \tag{12}$$

From (8), (11), and (12), we can obtain

$$E(N_l \mid c_l) > E(N_l \mid c_s) \tag{13}$$

which indicates that a sliding window in a legitimate e-mail tends to contain more legitimate genes than a sliding window in spam from a statistical point of view. Similarly, we can obtain $E(N_s \mid c_l) < E(N_s \mid c_s)$. Thus, an LC_i of a legitimate e-mail tends to be larger than that of spam, and an SC_i of a legitimate e-mail tends to be smaller than that of spam. In conclusion, the LC model can extract discriminative features for classification between spam and legitimate e-mail.

F. Evaluation Criteria

In spam filtering, many evaluation methods or criteria have been designed for comparing performance of different filters [24], [25]. We adopted four evaluation criteria, which were spam recall, spam precision, accuracy, and F_β measure, in all our experiments to evaluate the goodness of different parameter values and do a comparison between the LC approach and some prevalent approaches. Among the criteria, accuracy and F_β measure are more important, for accuracy measures the total number of messages correctly classified, and F_β is a combination of spam recall and spam precision.

1) **Spam recall:** It measures the percentage of spam that can be filtered by an algorithm or model. High spam recall ensures that the filter can protect the users from spam effectively. It is defined as follows:

$$R_s = \frac{n_{s \to s}}{n_{s \to s} + n_{s \to l}} \tag{14}$$

where $n_{s \to s}$ is the number of spam correctly classified, and $n_{s \to l}$ is the number of spam mistakenly classified as legitimate e-mail.

2) **Spam precision:** It measures how many messages, classified as spam, are truly spam. This also reflects the amount of legitimate e-mail mistakenly classified as spam. The higher the spam precision is, the fewer legitimate e-mail have been mistakenly filtered. It is defined as follows:

$$P_s = \frac{n_{s \to s}}{n_{s \to s} + n_{l \to s}} \tag{15}$$

where $n_{l \to s}$ is the number of legitimate e-mail mistakenly classified as spam, and $n_{s \to s}$ has the same definition as in (14).

3) **Accuracy:** To some extent, it can reflect the overall performance of filters. It measures the percentage of messages (including both spam and legitimate e-mail) correctly classified. It is defined as follows:

$$A = \frac{n_{l \to l} + n_{s \to s}}{n_l + n_s} \tag{16}$$

where $n_{l \to l}$ is the number of legitimate e-mail correctly classified, $n_{s \to s}$ has the same definition as in (14), and n_l and n_s are, respectively, the number of legitimate e-mail and the number of spam in the corpus.

4) **F_β measure:** It is a combination of R_s and P_s, assigning a weight β to P_s. It reflects the overall performance in another aspect. F_β measure is defined as follows:

$$F_\beta = (1+\beta^2)\frac{R_s P_s}{\beta^2 P_s + R_s}. \tag{17}$$

In our experiments, we adopted $\beta = 1$ as done in most approaches [24]. In this case, it is referred to as F_1 measure.

In the experiments, the values of the four measures were all calculated. However, only accuracy and F_1 measure are used for parameter selection and comparison of different approaches. Because they can reflect overall performance of different approaches, and F_1 combines both R_s and P_s. In addition, R_s and P_s, respectively, reflect different aspects of the performance, and they cannot reflect the overall performances of approaches, separately. That is also the reason why the F_β is proposed. We calculated them just to show the components of F_1 in detail.

V. Experiments

A. Experimental Corpora

We conducted experiments on five benchmark corpora PU1, PU2, PU3, PUA [26], and Enron-Spam[1] [32], using cross validation. The corpora have been preprocessed with removal of attachments, HTML tags, and header fields except for the subject. In the four PU corpora, the duplicates were removed from the corpora for duplicates may lead to over-optimistic conclusions in experiments. In PU1 and PU2, only the duplicate spam, which arrived on the same day, are deleted. While in PU3 and PUA, all duplicates (both spam and legitimate e-mail) are removed, even if they arrived on different days. In the Enron-Spam corpus, the legitimate messages sent by the owners of the mailbox and duplicate messages have been removed to avoid over-optimistic conclusions. Different from the former PU1 corpus (the one released in 2000) and Ling corpus, the corpora are not processed with removal of stop words, and no lemmatization method is adopted. The details of the corpora are given as follows.

1) **PU1:** The corpus includes 1099 messages, 481 messages of which are spam. The ratio of legitimate e-mail to spam is 1.28. The preprocessed legitimate messages and spam are all English messages, received by the first author of [26] over 36 months and 22 months, respectively.

[1] The five corpora are available from the web site: http://www.aueb.gr/users/ion/publications.html.

2) **PU2:** The corpus includes 721 messages, 142 messages of which are spam. The ratio of legitimate e-mail to spam is 4.01. Similar to PU1, the preprocessed legitimate messages and spam are all English messages, received by a colleague of the authors of [26] over 22 months.
3) **PU3:** The corpus includes 4139 messages, 1826 messages of which are spam. The ratio of legitimate e-mail to spam is 1.27. Unlike PU1 and PU2, the legitimate messages contain both English and non-English ones, received by the second author of [26]. While spam are derived from PU1, SpamAssassin corpus and other sources.
4) **PUA:** The corpus includes 1142 messages, 572 messages of which are spam. The ratio of legitimate e-mail to spam is 1. Similar to PU3, the legitimate e-mail contain both English and non-English messages, received by another colleague of the authors of [26], and spam is also derived from the same sources.
5) **Enron-Spam:** The corpus includes 33 716 messages, 17 171 messages of which are spam. The overall ratio of legitimate e-mail to spam is 0.96. It consists of six parts. In the first three parts, the ratio of legitimate e-mail to spam is about 3. While in the last three parts, the ratio of legitimate e-mail to spam is about 0.33. Experiments conducted on the whole Enron-Spam corpus using six-fold cross validation can help investigate the generalization performance of the model.

B. Experimental Setup

We conducted all the experiments on a PC with Intel E2140 CPU and 2G RAM. The SVM library LIBSVM is applied for the implementation of the SVM [33].

C. Experiments of Parameter Selection

Experiments have been conducted to tune the parameters of the LC model. In this section, we show and analyze the results of experiments on tuning important parameters of the LC model. All these experiments were conducted on PU1 corpus by utilizing ten-fold cross-validation, and IG was used as the term selection method of the models.

1) Selection of a Proper Tendency Threshold: Experiments were conducted with varied tendency threshold θ to investigate the effects of θ on the performance of the LC model. As shown in Fig. 2, the LC model performs well with small θ. However, with the increase of θ, the performance of the LC model degrades in terms of both accuracy and F_1 measure. As we see, the term selection methods have already filtered the uninformative terms, thus the threshold is not quite necessary. In addition, a great θ would result in loss of information. It is recommended that θ should be set to zero or a small value.

2) Selection of Proper Feature Dimensionality: For the LC model using a fixed-length sliding window (LC-FL), short messages and long messages need to be processed specifically so that all the messages can have the same feature dimensionality. Before that, the feature dimensionality needs to be determined. Therefore, we conducted experiments to determine the optimal number of utmost front sliding windows for discrimination. Fig. 3 depicts the results, from which we can see that the model performed best when ten utmost front sliding windows of each message were utilized for discrimination. In this case, all the messages would be transformed into 20-dimensional feature vectors through the specific process introduced in Section IV-D1.

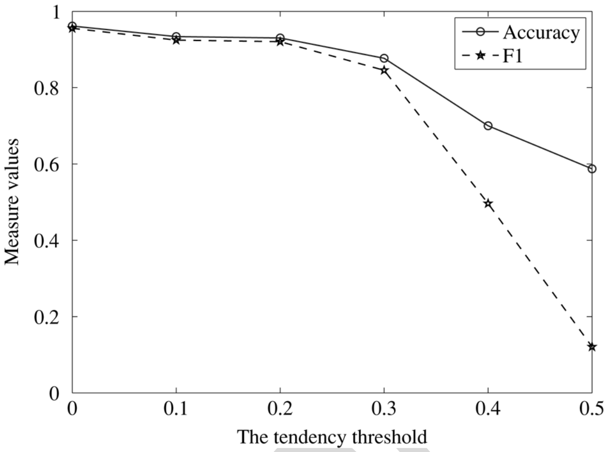


Fig. 2. Performance of the model with varied tendency threshold.

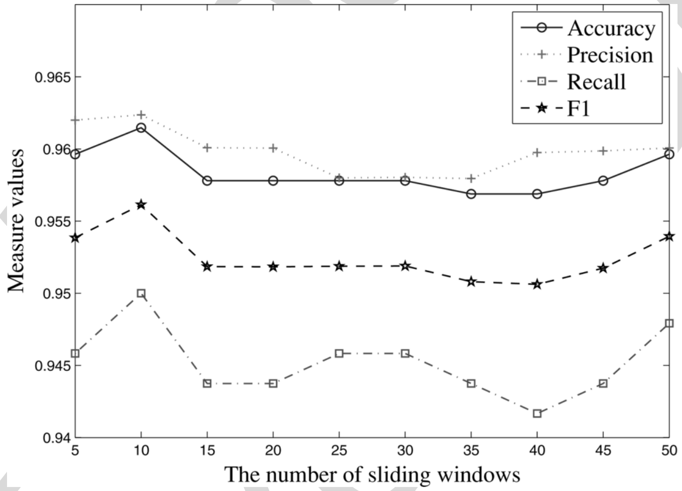


Fig. 3. Performance of the LC-FL model with different window numbers.

For the LC model using a variable-length sliding window (LC-VL), all the messages are directly transformed into feature vectors with the same dimensionality. However, there is still the necessity for determining the feature dimensionality, which corresponds to the number of local areas in a message. We conducted some preliminary experiments on PU1 and found that the LC-VL model performed optimally when the feature dimensionality was set to six or ten.

3) Selection of a Proper Sliding Window Size: For the LC-FL model, the sliding window size is quite essential as it defines the size of local area in a message. Only when the size of local area is properly defined can we calculate discriminative LC vectors for messages. Fig. 4 shows the performance of the LC-FL model under different values of sliding window size. When the size was set to 150 terms per window, the model performed best in terms of both accuracy and F_1 measure. It also can be seen that

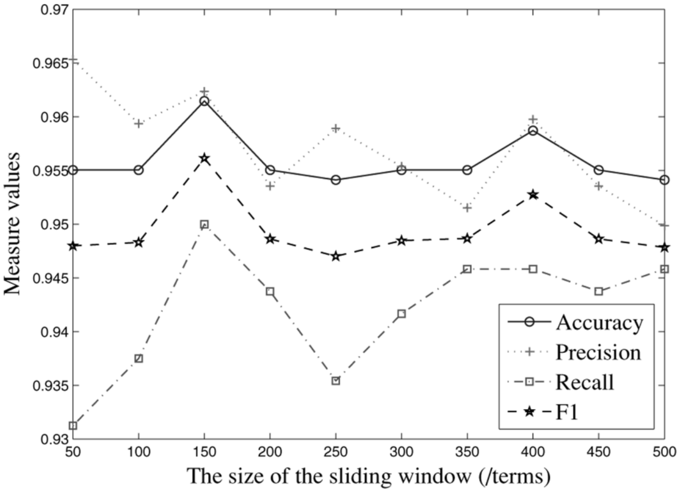


Fig. 4. Performance of the LC-FL model with different sliding window sizes.

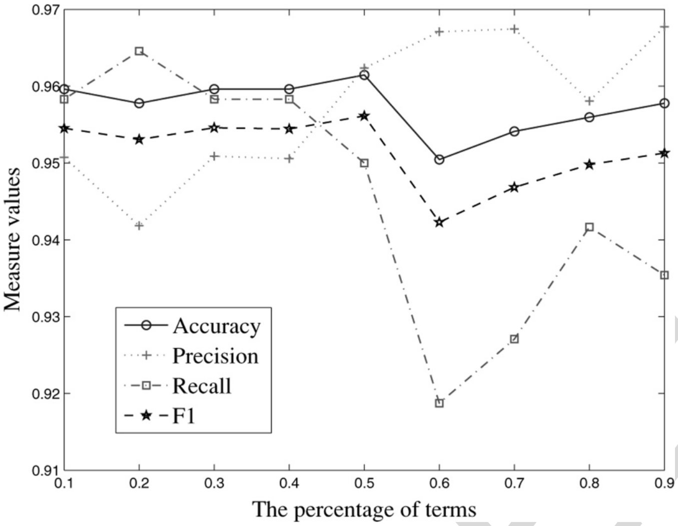


Fig. 5. Performance of the model with different percentage of terms.

the model performed acceptably when the parameter was set to other values.

4) Selection of Optimal Terms Percentage: The phase of term selection plays an important role in the LC model. The removal of less informative terms can reduce computational complexity and improve the overall performance of the model. We conducted experiments to determine the percentage of terms reserved after the phase of term selection. Therefore, the removal of uninformative terms can be maximized while avoiding removing informative ones.

Fig. 5 gives the results of the LC-FL model. When 50% terms were reserved after term selection, the model performed best in terms of both accuracy and F_1 measure. In the following experiments, we set the parameter to 50% for both the LC-FL model and the LC-VL model. We should pay attention to the fact that the model performed quite well when only 10% terms were reserved. This configuration can be applied to cost-sensitive situations.

D. Experiments of the Model With Three Term Selection Methods

To construct discriminative feature vectors for messages, both a term selection method and a feature extraction approach play quite essential roles. To some extent, a feature extraction approach depends on a proper term selection method. Therefore, it is necessary to verify whether the proposed LC approach can be incorporated with prevalent term selection methods.

We conducted comparison experiments of the model with three term selection methods IG, TFV, and DF. All these experiments were conducted on corpora PU1, PU2, PU3, and PUA using ten-fold cross-validation, and on corpus Enron-Spam using six-fold cross-validation. The performances of the LC-FL strategy and the LC-VL strategy are listed in Tables II and III, respectively. The two strategies performed quite well incorporated with any of these term selection methods. On one hand, the experiments showed that the proposed LC strategies could be incorporated with different term selection methods. On the other hand, the experiments had also reflected the effectiveness of the three term selection methods.

E. Comparison Between the LC Model and Current Approaches

In this section, we compared the two LC strategies with some prevalent approaches through the experiments on four PU corpora using ten-fold cross-validation and on corpus Enron-Spam using six-fold cross-validation. The approaches utilized in comparison are Naive Bayes-BoW, SVM-BoW [26], SVM-Global Concentration (SVM-GC), SVM-LC-FL, and SVM-LC-VL.

In Naive Bayes-BoW and SVM-BoW, Naive Bayes and SVM are utilized as their classifiers, respectively, BoW is utilized as the feature extraction approach, and IG is used as the term selection method [26]. In both SVM-LC-FL and SVM-LC-VL, SVM is utilized as their classifier. SVM-GC is a specific configuration of SVM-LC, in which the sliding window size is set to infinite. In such a case, each message is recognized as a whole window, and a two-dimensional feature (including a spam genes concentration and a legitimate genes concentration) is constructed for each message. In this way, it is similar to the CFC approach [18], [19]. The results of these experiments are shown in Table IV.

The comparison with Naive Bayes-BoW and SVM-BoW is mainly to compare the two LC strategies with the prevalent BoW approach. The results show that both of the two LC strategies outperformed the BoW approach in accuracy and F_1 measure. As mentioned before, we take accuracy and F_1 measure as comparison criteria without focusing on precision and recall. Because they are incorporated into the calculation of F_1 measure, and can be reflected by the value of F_1 measure.

The comparison between the two LC strategies and SVM-GC is to verify whether the two LC strategies can extract useful position-correlated information from messages. Both the two LC strategies correspond different parts of a message to different dimensions of the feature vector, while SVM-GC extracts position independent feature vectors from messages. As shown in Table IV, both the two LC strategies outperformed SVM-GC in accuracy and F_1 measure, which verified that the proposed

TABLE II
Experiments of the LC-FL Model With Three Different Term Selection Methods on Corpora PU1, PU2, PU3, PUA, and Enron-Spam, Utilizing Cross Validation

Corpus	Feature sel.	Precision(%)	Recall(%)	Accuracy(%)	F_1(%)	Feature dim.
PU1	IG	96.04	95.42	96.24	95.73	20
	TFV	95.12	96.88	**96.42**	**95.99**	20
	DF	94.38	96.67	95.96	95.51	20
PU2	IG	95.74	75.71	94.37	84.55	20
	TFV	93.37	74.29	93.80	82.74	20
	DF	90.86	82.86	**94.79**	**86.67**	20
PU3	IG	95.99	95.33	**96.13**	**95.66**	20
	TFV	95.80	95.05	95.91	95.43	20
	DF	95.15	95.99	96.00	95.57	20
PUA	IG	96.01	94.74	**95.26**	**95.37**	20
	TFV	95.83	94.39	94.91	95.10	20
	DF	95.25	94.56	94.74	94.90	20
Enron-Spam	IG	94.07	98.00	96.79	**95.94**	20
	TFV	93.73	98.10	**96.80**	95.79	20
	DF	93.67	98.10	96.68	95.77	20

TABLE III
Experiments of the LC-VL Model With Three Different Term Selection Methods on Corpora PU1, PU2, PU3, PUA, and Enron-Spam, Utilizing Cross Validation

Corpus	Feature sel.	Precision(%)	Recall(%)	Accuracy(%)	F_1(%)	Feature dim.
PU1	IG	94.85	95.63	95.78	95.21	6
	TFV	95.48	96.04	**96.24**	**95.72**	6
	DF	95.07	96.25	96.15	95.63	6
PU2	IG	95.74	77.86	94.79	85.16	6
	TFV	94.43	79.29	94.79	85.47	6
	DF	92.06	86.43	**95.63**	**88.65**	6
PU3	IG	96.68	94.34	96.03	95.45	6
	TFV	96.46	94.29	95.91	95.32	6
	DF	95.64	95.77	**96.15**	**95.67**	6
PUA	IG	95.60	94.56	**94.91**	**94.94**	6
	TFV	95.22	94.39	94.65	94.67	6
	DF	95.95	93.33	94.56	94.52	6
Enron-Spam	IG	92.44	97.81	**96.02**	**94.94**	6
	TFV	92.07	97.88	95.90	94.77	6
	DF	92.11	97.93	95.95	94.82	6

LC approach (including the LC-FL strategy and the LC-VL strategy) could effectively extract position-correlated information from messages.

Compared to BoW, the proposed LC strategies can greatly reduce feature vector dimensionality, and have advantages in processing speed. As shown in Table V, the two LC strategies outperformed the BoW approach significantly in terms of feature dimensionality and processing speed. However, BoW obtained poor performance when feature dimensionality was greatly reduced [26], while LC strategies performed quite promisingly with a feature dimensionality of 20.

F. Discussion

In Section V-E, it is shown that both the LC-FL strategy and the LC-VL strategy outperform the GC approach on all the corpora. The success of the LC strategies is considered to lie in two aspects. First, the LC strategies can extract position-correlated information from a message by transforming each area of a message to a corresponding feature dimension. Second, the LC strategies can extract more information from messages, compared to the GC approach. As the window size can be acquired when the parameter of the LC strategies are determined, the Global Concentration can be approximately expressed by the weighted sum of Local Concentration, and the weights are correlated with the window size. However, the Local Concentration cannot be deduced from Global Concentration. Thus, the Local Concentration contains more information than the Global concentration does.

The essence of the LC strategies is the definition of local areas for a message. As the local areas may vary with message length, we conducted experimental analysis to see whether drift of message length would affect the performance of the LC strategies. The average message length of corpora PU1, PU2, PU3, PUA, and Enron-Spam are 776 terms, 669 terms, 624 terms, 697 terms, and 311 terms, respectively. It can be seen that the average message length of Enron-Spam is quite shorter than the other four PU corpora. To further demonstrate the difference between Enron-Spam corpus and the PU corpora, the

TABLE IV
COMPARISON BETWEEN THE LC MODEL AND CURRENT APPROACHES

Corpus	Approach	**Precision(%)**	**Recall(%)**	**Accuracy(%)**	F_1(%)	**Feature dim.**
PU1	Naive Bayes-BoW	89.58	99.38	94.59	94.23	600
	SVM-BoW	93.96	95.63	95.32	94.79	600
	SVM-GC	94.97	95.00	95.60	94.99	2
	SVM-LC-FL	95.12	96.88	**96.42**	**95.99**	20
	SVM-LC-VL	95.48	96.04	96.24	95.72	6
PU2	Naive Bayes-BoW	80.77	90.00	93.66	85.14	600
	SVM-BoW	88.71	79.29	93.66	83.74	600
	SVM-GC	95.12	76.43	94.37	84.76	2
	SVM-LC-FL	90.86	82.86	94.79	86.67	20
	SVM-LC-VL	92.06	86.43	**95.63**	**88.65**	6
PU3	Naive Bayes-BoW	93.59	94.84	94.79	94.21	600
	SVM-BoW	96.48	94.67	96.08	95.57	600
	SVM-GC	96.24	94.95	96.05	95.59	2
	SVM-LC-FL	95.99	95.33	96.13	95.66	20
	SVM-LC-VL	95.64	95.77	**96.15**	**95.67**	6
PUA	Naive Bayes-BoW	95.11	94.04	94.47	94.57	600
	SVM-BoW	92.83	93.33	92.89	93.08	600
	SVM-GC	96.03	93.86	94.82	94.93	2
	SVM-LC-FL	96.01	94.74	**95.26**	**95.37**	20
	SVM-LC-VL	95.60	94.56	94.91	94.94	6
Enron-Spam	Naive Bayes-BoW	79.34	99.17	88.41	87.32	600
	SVM-BoW	90.88	98.87	95.13	94.62	600
	SVM-GC	91.48	97.81	95.62	94.39	2
	SVM-LC-FL	94.07	98.00	**96.79**	**95.94**	20
	SVM-LC-VL	92.44	97.81	96.02	94.94	6

TABLE V
PROCESSING SPEED OF THE APPROACHES

Approach	Naive Bayes-BoW	SVM-BoW	Naive Bayes-BoW	SVM-BoW	SVM-GC	SVM-LC-FL	SVM-LC-VL
Seconds/email	0.2	0.5	3	5	0.06	0.07	0.06
Feature dim.	120	120	600	600	2	20	6

Cumulative Distribution Function (CDF) of the message length in PU1 corpus and Enron-Spam corpus are depicted in Fig. 6.

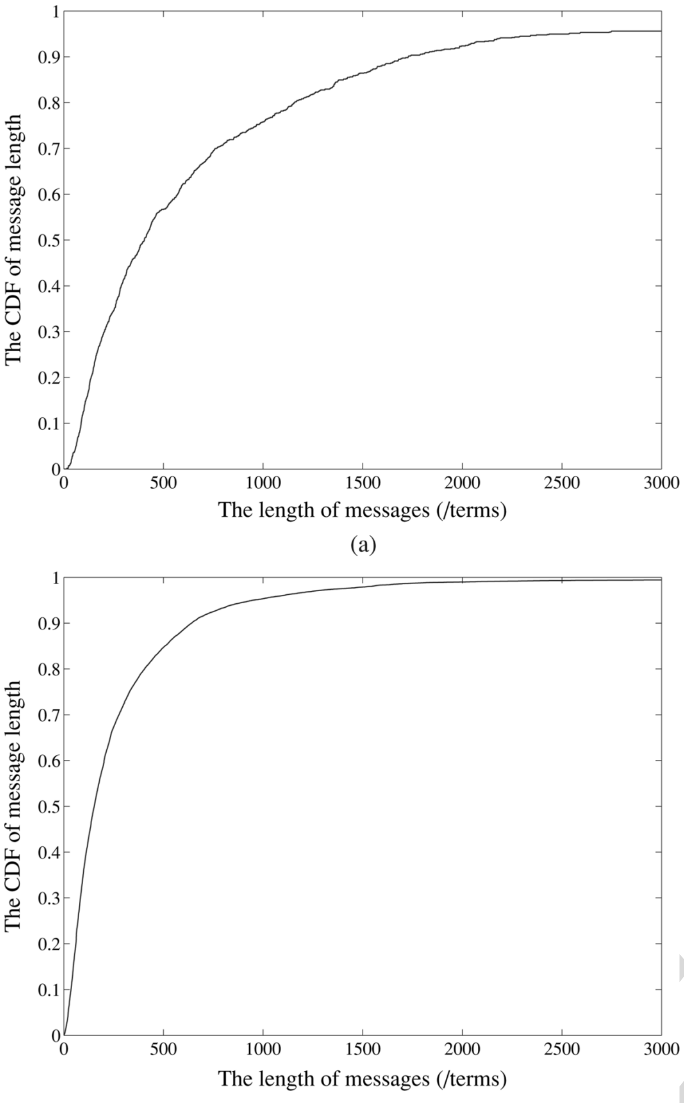
Even though the message length distribution in Enron-Spam corpus is quite different from that of PU corpora, it is shown in Section V-E that the LC strategies perform well on both Enron-Spam corpus and PU corpora. Thus, a preliminary conclusion can be drawn that the LC strategies are robust against variable message length, and the coexistence of short messages and long messages does not decrease the performance of the LC strategies. As long as the average message length is larger than the size of a window, the LC strategies can extract Local Concentration from messages. When almost all the messages become shorter than a window, the performance of the LC strategies would decay and become equivalent to that of the GC approach. However, the window size could be tuned accordingly when the message length changes too much. In that way, the LC strategies can still extract Local Concentration from messages with variable length. In future, we intend to focus on developing adaptive LC approaches, so that the definition of local area can be automatically adapted to the change of message length.

VI. CONCLUSION

We have proposed an LC approach for extracting local-concentration-features for messages. Two implementation strategies of the approach, namely the LC-FL strategy and the LC-VL strategy, have been designed. Extensive experiments have shown that the proposed LC strategies have quite promising performance and advantage in the following aspects.

1) Utilizing sliding windows, both the two LC strategies can effectively extract the position-correlated information for messages.
2) The LC strategies cooperate well with three term selection methods, which endows the LC strategies with flexible applicability in real world.
3) Compared to the prevalent BoW approach and the GC approach, the two LC strategies perform better in terms of both accuracy and F_1 measure.
4) The LC strategies can greatly reduce feature dimensionality and have much faster speed, compared to the BoW approach.
5) The LC strategies are robust against messages with variable message length.

In future work, we intend to incorporate other classifiers into the LC model and investigate their performance under these configurations. In addition, we hope the model can be developed as an adaptive anti-spam system, by taking into account the drift of spam content and the changing interests of users.



(a)

(b)

Fig. 6. CDF curves of message length in PU1 corpus and Enron-Spam corpus. (a) CDF curve of message length in PU1 corpus. (b) CDF curve of message length in Enron-Spam corpus.

Acknowledgment

The authors would like to thank the Associate Editor and anonymous reviewers for providing insightful comments and constructive suggestions that greatly helped improving the quality of this paper.

References

[1] Commtouch, Q4 2008 Internet Threats Trend Report Jan. 2009 [Online]. Available: http://www.pallas.com/fileadmin/img/content/publikationen/Commtouch-Pallas_2008_Q4_Internet_Threats_Trend_Report.pdf

[2] R. Jennings, Cost of Spam is Flattening—Our 2009 Predictions Ferris Research, Jan. 2009 [Online]. Available: http://www.ferris.com/2009/01/28/cost-of-spam-is-flattening-our-2009-predictions/

[3] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, A Baysian Approach to Filtering Junk e-mail AAAI Tech. Rep. WS-98-05, 1998, pp. 55–62.

[4] R. Segal, "Combining global and personal anti-spam filtering," in *Proc. 4th Conf. Email and Anti-spam (CEAS' 07)*, 2007***[Please provide page range or location of conference]***.

[5] A. Ciltik and T. Gungor, "Time-efficient spam e-mail filtering using n-gram models," *Pattern Recognit. Lett.*, vol. 29, no. 1, pp. 19–33, 2008.

[6] H. Drucker, D. Wu, and V. Vapnik, "Support vector machines for spam categorization," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1048–1054, Sep. 1999.

[7] G. Ruan and Y. Tan, "Intelligent detection approaches for spam," in *Proc. Third Int. Conf. Natural Computation (ICNC07)*, Haikou, China, 2007, pp. 1–7.

[8] S. Bickel and T. Scheffer, "Dirichlet-enhanced spam filtering based on biased samples," *Adv. Neural Inf. Process. Syst.*, vol. 19, pp. 161–168, 2007.

[9] I. Kanaris, K. Kanaris, I. Houvardas, and E. Stamatatos, "Words versus character N-grams for anti-spam filtering," *Int. J. Artif. Intell. T.*, vol. 16, no. 6, pp. 1047–1067, 2007.

[10] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. D. Spyropoulos, and P. Stamatopoulos, "Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach," in *Proc. Workshop "Machine Learning and Textual Information Access," 4th Eur. Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD' 00)*, 2000, pp. 1–13.

[11] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. D. Spyropoulos, and P. Stamatopoulos, "A memory-based approach to anti-spam filtering for mailing lists," *Inform. Retrieval*, vol. 6, no. 1, pp. 49–73, 2003.

[12] J. Clark, I. Koprinska, and J. Poon, "A neural network based approach to automated e-mail classification," in *Proc. IEEE Int. Conf. Web Intelligence (WI' 03)*, Halifax, Canada, 2003, pp. 702–705.

[13] C.-H. Wu, "Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 4321–4330, Apr. 2009.

[14] X. Carreras and L. Márquez, "Boosting trees for anti-spam email filtering," in *Proc. 4th Int. Conf. Recent Advances in Natural Language Processing (RANLP' 01)*, 2001, pp. 58–64.

[15] J. R. He and B. Thiesson, "Asymmetric gradient boosting with application to spam filtering," in *Proc. 4th Conf. Email and Anti-spam (CEAS'07)*, 2007***[Please provide page range or location of conference]***.

[16] T. Oda and T. White, "Developing an immunity to spam," *Lecture Notes Comput. Sci. (LNCS)*, pp. 231–242, 2003.

[17] T. S. Guzella, T. A. Mota-Santos, J. Q. Uchôa, and W. M. Caminhas, "Identification of spam messages using an approach inspired on the immune system," *Biosystems*, vol. 92, no. 3, pp. 215–225, Jun. 2008.

[18] Y. Tan, C. Deng, and G. Ruan, "Concentration based feature construction approach for spam detection," in *Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN2009)*, Atlanta, GA, Jun. 14–19, 2009, pp. 3088–3093.

[19] G. Ruan and Y. Tan, "A three-layer back-propagation neural network for spam detection using artificial immune concentration," *Soft Comput.*, vol. 14, pp. 139–150, 2010.

[20] D. Dasgupta, "Advances in artificial immune systems," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 40–49, Nov. 2006.

[21] Wikipedia [Online]. Available: http://en.wikipedia.org/wiki/Information_gain

[22] I. Koprinska, J. Poon, J. Clark, and J. Chan, "Learning to classify e-mail," *Inform. Sci.*, vol. 177, pp. 2167–2187, 2007.

[23] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proc. Int. Conf. Machine Learning (ICML'97)*, 1997, pp. 412–420.

[24] T. S. Guzella and M. Caminhas, "A review of machine learning approaches to spam filtering," *Expert Syst. Appl.*, vol. 36, pp. 10206–10222, 2009.

[25] E. Blanzieri and A. Bryl, A Survey of Learning-Based Techniques of e-mail Spam Filtering University of Trento, Information Engineering and Computer Science Department, Trento, Italy, Tech. Rep. DIT-06-065, Jan. 2008.

[26] I. Androutsopoulos, G. Paliouras, and E. Michelakis, Learning to Filter Unsolicited Commercial E-mail NCSR "Demokritos" Tech. Rep. 2004/2, Oct. 2006, minor corrections.

[27] K.-M. Schneider, "A comparison of event models for naive bayes anti-spam e-mail filtering," in *Proc. 10th Conf. Eur. Chapter of the Association for Computational Linguistics*, 2003, pp. 307–314.

[28] W. S. Yerazunis, "Sparse binary polynomial hashing and the CRM114 discriminator," in *Proc. 2003 Spam Conf.*, Cambrige, MA, 2003.

[29] C. Siefkes, F. Assis, S. Chhabra, and W. S. Yerazunis, "Combining winnow and orthogonal sparse bigrams for incremental spam filtering," *Lecture Notes Comput. Sci.*, vol. 3202/2004, pp. 410–421, 2004.

[30] G. V. Cormack, "Content-based web spam detection," in *Proc. 3rd Int. Workshop Adversarial Information Retrieval on the Web (AIRWeb'07)*, 2007***[Please provide page range or location of conference]***.

[31] D. Sculley, "Advances in Online Learning-Based Spam Filtering," Ph.D. dissertation, Tufts Univ., Somerville, MA, 2008.

[32] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes—Which naive bayes?," in *Proc. 3rd Conf. Email and Anti-Spam (CEAS'06)*, Mountain View, CA, 2006, pp. 125–134.

[33] C.-C. Chang and C.-J. Lin, LIBSVM: a Library for Support Vector Machines [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

Yuanchun Zhu received the B.S. degree in computer science and the B.A. degree in English from Jilin University, Jilin, China, in 2007. He is currently majoring in computer science and working towards the Ph.D. degree at Key Laboratory of Machine Perception (Ministry of Education) and Department of Machine Intelligence, EECS, Peking University, Beijing.

His research interests include machine learning, swarm intelligence, AIS, bioinformatics, information processing, and pattern recognition.

Ying Tan (M'98–SM'02) received the B.S., M.S., and Ph.D. degrees in signal and information processing from Southeast University, Nanjing, China, in 1985, 1988, and 1997, respectively.

Since then, he became a Postdoctoral Fellow and then an Associate Professor at the University of Science and Technology of China. He was a Full Professor, advisor of Ph.D. candidates, and Director of the Institute of Intelligent Information Science of his university. He worked with the Chinese University of Hong Kong in 1999 and in 2004–2005. He was an electee of 100 talent program of the Chinese Academy of Science in 2005. Now, he is a Full Professor, advisor of Ph.D. candidates at the Key Laboratory of Machine Perception (Ministry of Education), Peking University, and Department of Machine Intelligence, EECS, Peking University, and he is also the head of Computational Intelligence Laboratory (CIL) of Peking University. He has authored or coauthored more than 200 academic papers in refereed journals and conferences and several books and book chapters. His current research interests include computational intelligence, artificial immune system, swarm intelligence and data mining, signal and information processing, pattern recognition, and their applications.

Dr. Tan is Associate Editor of the *International Journal of Swarm Intelligence Research* and the *IES Journal B, Intelligent Devices and Systems*, and Associate Editor-in-Chief of the *International Journal of Intelligent Information Processing*. He is a member of the Advisory Board of the *International Journal on Knowledge Based Intelligent Engineering System* and the Editorial Board of the *Journal of Computer Science and Systems Biology* and *Applied Mathematical and Computational Sciences*. He is also the Editor of *Springer Lecture Notes on Computer Science, LNCS 5263, 5264, 6145,* and *6146*, and Guest Editor of special issues of several journals including *Information Science*, *Soft Computing*, *International Journal of Artificial Intelligence*, etc. He was the general chair of the *International Journal on Swarm Intelligence* (ICSI 2010, ICSI 2011) and the program committee chair of ISNN2008. He was honored the second-class National Natural Science Award of China in 2009.