Journal of Computers Vol. 28, No. 3, 2017, pp. 174-183
doi:10.3966/199115592017062803014

Optimizing Naïve Bayes Algorithm for SMS Spam Filtering on Mobile Phone to Reduce the Consumption of Resources

Li-qun Bao[1*], Lin-xia LV[2], and Jin-Long Li[1]

[1] College of Electronics and Information Engineering, Lanzhou Institute of Technology
Lanzhou, China
baoliqun1983@163.com

[2] College of Software Engineering, Lanzhou institute of technology
Lanzhou, China

Received 25 May 2016; Revised 09 December 2016; Accepted 09 December 2016

Abstract. Mobile phone is a embedded system with limited energy, memory and computing ability. The traditional SMS spam filtering algorithms only focused on the accuracy rate of spam filters and did not consider the consumption of mobile phone's hardware resources. This paper proposes an optimized Naïve Bayes SMS Spam filtering algorithm, the main purpose is to reduce the consumption of hardware resources and reflect user's individual preference. A feature library is designed to save weight of features, conditional probabilities and prior probabilities. The filtering algorithm is divided into a feature library updating algorithm and an online SMS filtering algorithm. The feature library is updated regularly by the feature library updating algorithm. The feature library updating algorithm does not need to run immediately, and can be performed in free time of mobile application or copied to PC client asynchronously. The online SMS filtering algorithm running on a mobile phone performs only the minimum amount of work which does not require much resources. It can be quickly adapted to the changing of user's preference. Experimental results show that the algorithm can improve the speed of classification while maintaining a high classification accuracy. it takes up less storage space and can be used on common mobile platforms.

Keywords: feature library, Naïve Bayes, online SMS spam filtering algorithm, short messages, updating algorithm

1 Introduction

With the development of wireless communication, the number of mobile phone users in the world increased rapidly. According to the Forrester's forecasts, mobile phone users will reach 48 billion in the world by the end of 2016 [1]. Short Message Service (SMS) is a popular means of mobile communication. Due to huge success of Short Message Service, SMS has now become the preferred [2] and cheapest medium for advertisers to reach masses especially in developing countries like China. SMS traffic has brought huge economic losses [3]. In August 2015, Baidu mobile guards released the "first half of 2015, China Mobile Internet Security Report" [4]. According to the report, in the first half of 2015, advertising is the largest kind of spam messages, accounting for 30% of the total, followed by telecommunications business promotion, financial Services, news broadcasts, etc.

Spam can be defined as unsolicited email for a recipient or any messages that the users do not wanted to have in their inbox. The design of current mobile phone inbox did not provide flexibility to delete a spam SMS without looking at it. Due to unnecessary notifications and content, spam SMS has become a source of annoyance, a waste of time and frequent disturbance for the receiver, and become a widespread

* Corresponding Author

social problem.

Operators of short messages have no perfect spam message monitoring system so far. They still use spam filtering methods. SMS spam filters generally use four techniques: Black and white list filtering, keyword matching filtering, filtering based on rules and content based SMS spam filtering. The first three techniques are simple, efficient and easy to implement and require little computing resources [5]. Black and white list filtering, keyword matching filtering all need manual setting of phone numbers or key words, such as "Free", "Account", etc. Users preset telephone numbers of spammers and keywords frequently appear in spam messages. Thus, if a short message comes from those telephone numbers or contains those keywords, it will be saved into the spam folder and the user will not be alerted by this spam message. Furthermore, Manual updating of black list is always later than the emergence of new SMS spam senders. The SMS spam senders will feel the rule. They are very weak when it comes to intentional modification of keywords or phone numbers. Rule-based filtering has the advantages of simple, fast, and needing no training phases, but fixed rules can't adapt to the new characteristics of spam messages. If the user's interest changes or the categories of spam vary, these rules need to be changed accordingly. So the accuracy rate of these filters are often poor. On the other hand, content based SMS spam filtering techniques show better accuracy results compared to other methods. However, theses methods require much more computation resources.

At present, studies on SMS spam filtering are mainly focused on content based spam filtering algorithms. Another issue needing to be concerned about is the difference between short messages and e-mail. Short messages are short and lack of characteristic data which are different from emails. A email consists of certain structured information such as subject, mail header, sender's address etc, but a short message does not have such structured information and often consists of few words, which bring difficulties to SMS spam filtering. The traditional content based spam filtering algorithms cannot be directly applied to SMS spam filtering, feature selection is particularly important [6]. It is necessary to develop an efficient SMS Spam filtering method which classify short messages into spam and non-spam SMS efficiently.

The scientific contributions of this research work include:

(1) Designning SMS feature library, the content-based SMS spam filtering algorithm is divided into feature library updating algorithm and real-time SMS classification algorithm.

(2) The feature library updating algorithm adopts the method of asynchronous updating, which reduces the amount of calculation of client real-time processing.

(3) Improving the Naive Bayesian classification algorithm, and applied to real-time SMS filtering, it can immediately adapt to the changing content of short messages and filtering needs of the user.

The rest of the paper is organized as follows. Section II present the related work on the SMS Spam filtering. Section III specifies the design and implementation of the system including the system architecture, feature library, updating algorithm and SMS classification algorithm. The experiments, testing and results are reported and discussed in Section IV. Finally, the conclusion is presented in Section V.

2 Related Work

2.1 Content Based SMS Spam Filtering

Content based spam filtering algorithms include artificial immune [7], support vector machine, decision tree [8], Neural Networks and Naïve Bayes algorithm. The immune system is a complex network of organs and cells which is responsible for the organism's defense against alien particles. The capacity of distinguishing between self and nonself genes is the main features of the immune system. Support Vector Machines is a linear maximal margin binary classifier. It produces (linear) vectors that try to maximally separate the target classes (spam versus legitimate). It has shown excellent results in text classification applications. Decision tree is a classifier in the form of a tree structure to show the reasoning process. Each node in decision tree structures is either a leaf node or a decision node. Artificial Neural Networks is also a widely used machine learning based filter which are computer models that try to simulate the same tasks that the brain carries out [9]. Naïve Bayes classifier is a simple probabilistic classifier. The main advantage is that naïve Bayes classifiers can be trained very efficiently in a supervised learning.

Al-Hasan and El-Alfy [10] developed a novel approach based on DCA that fuses the output from two

classifiers. The empirical results showed significant improvement can be achieved when applying the proposed approach. Joe [11] proposed a SVM spam filter model which selects meaningful features among thousands of original ones by chi-square statistics after preprocessing. Healy et al. [12] studied the problems of performing spam classification on short messages by comparing the performance of the well-known K-Nearest-Neighbor (KNN), Support Vector Machines (SVM), and Naive Bayes classifiers. They conclude that, for short messages, the SVM and Naïve Bayes classifiers substantially outperform the KNN classifier. Hidalgo et al. [13] also accomplished content filtering experiments on English and Spanish spam SMS corpora, which proved that Bayesian filtering methods are still effective against spam SMS messages. Shahi [14] also believes that Naïve Bayes and Support Vector Machines (SVM) based classifications are the two most successful techniques.

These methods all require training samples, but because of privacy issues involved, lack of training sample library is a major problem at present, which has brought difficulties to researches on SMS spam filtering [15]. In addition, for existing two categories-based Bayesian spam filtering, if a user changes the category of some messages, the prior probability of spam messages and normal SMS, the class conditional probability of features all need to be recalculated, which will increase the burden of client processing. In this paper, a feature library is designed and on which an optimized Naïve Bayes algorithm is proposed.

2.2 The Weaknesses of Earlier Study

The filtering algorithm made no consideration of mobile phone's hardware resources. Mobile phone is a embedded system with limited hardware resources. It bring difficulties to run complex SMS spam filtering system. Most of previous researches did not consider the actual running environment and the algorithm testing were conducted on personal computers. They only focused on the accuracy, precision and recall of spam filters from the view point of the algorithm used [10, 14], but a spam filter on a mobile phone must consider the consumption of resources, because energy, memory and computing ability all are limited.

The filtering algorithm did not take into account the user's individualized requirements. All users are given the same classification results [16]. Due to the special nature of SMS, everyone has different opinion. First of all, different users give different classification results to the same message, for example, a real estate sales based SMS is an important message to users interested in buying houses, but others may consider it to be a spam message. Secondly, user's preferences change over time. At one stage, a message is considered as a spam message. At another stage, it may be determined to be a normal SMS.

2.3 User's Expectations from a SMS Spam Filtering Solution

From the view of implementation, there are two kinds of technological solutions used for SMS spam filtering, which include Short Message Service Center (SMSC) based filtering and filter by installing related software in the mobile phone [17-18]. For SMSC based filtering, all the spam SMSes will be filtered by mobile operators. Once the classification error occurs at the service center, a normal message will be lost and could not reach the mobile phone user. It also involves the interests of the operators, so this kind of filter is not easy to implement.

In order to understand users' requirements, behavior and perceptions related to spam SMS, Yadav et al. [19] conducted a survey from 458 participants of different age groups and cities in India, the result show that there is a need for personalized SMS spam filtering which takes into account with user preferences and information requirements. Due to ineffectiveness of available solutions, mobile users showed strong willingness to use a technological solution for SMS spam filtering.

The purpose of this paper is to propose an optimized Naïve Bayes classifier for SMS Spam filtering on mobile devices in order to reduce the consumption of resources, that means the filter can run on common mobile terminals with limited energy, memory and computing ability. Even more, we expect our approach to block mobile spam messages and have higher accuracy and operating efficiency. The filter can also adapted to the preference of individuals. The key research problem of this paper includes: (1) How to design the feature library. (2) How to update the feature library without affecting the efficiency of filtering system. (3) How to design the real-time SMS spam filtering algorithm.

3 Proposed System

3.1 System description

This paper proposed an optimized Naïve Bayes classifier for SMS Spam Filtering on mobile phones for the purpose of reducing the consumption of resources. As shown in Fig. 1, the filtering system consists of feature library, feature library updating algorithm and online SMS spam filtering algorithm. The feature library is generated on the PC side by a large number of training samples after being preprocessed and selected. The feature library is updated regularly by the feature library updating algorithm, and the updating cycle can be set by users. The feature library updating algorithm does not need to perform online, but performs in free time of mobile application or be copied to PC side asynchronously. Features of a new short message are sent to online SMS spam filtering algorithm. Meanwhile, the user can get real-time classification results.

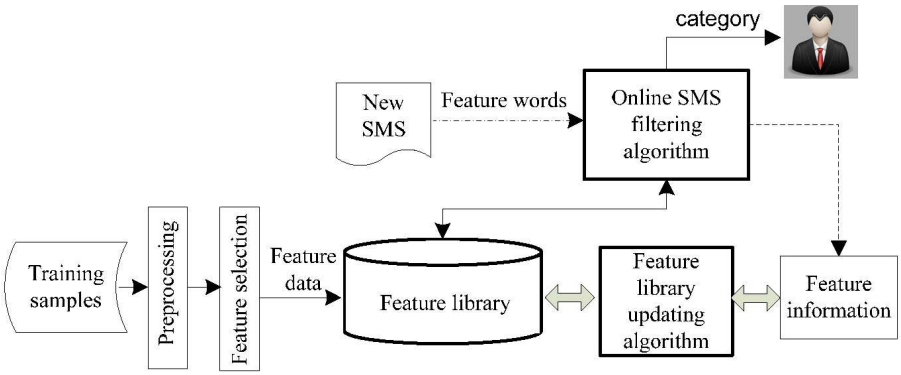

Fig. 1. System configuration

The updating algorithm takes on the more complex and time consuming jobs, such as computing and updating of conditional probabilities, while the online SMS spam filtering algorithm performs only the minimum amount of work for filtering which gets the data from the feature library. The online SMS filtering algorithm requires minimal resources which can run on mobile phones and can adapt itself quickly to the changing of the user's preference.

3.2 Features of Proposed System

Taking up less mobile storage capacity. After preprocessing and feature extraction, removing the SMS message's interference and irrelevant information, the feature library only store the features and the number of feature occurrence in the sample database, not the actual content of short messages, which can reduce the storage capacity. The feature library can also be shared between filtering systems.
Consuming less computing resources. Although the number of training samples increases, the amount of work that a mobile phone has to deal with should almost be little. The filter consumes less time and memory capacity, which can be used on mobile phones with less computing and storage resources, that means the filter on the mobile phone is free from burden of training process.
Classifying SMS messages according to user's preferences. This paper establishes feature library on mobile phone side, the filtering system classify short messages according to user's preferences to avoid the limitations of traditional filtering system by which all users are given the same classification results.

4 Implementation

4.1 Feature Selection

Since having a good feature representation is one of the most important parts for getting a good classifier, we have to face the fact that SMS messages don't have the same structure and characteristics than email messages, and that could be a problem because fewer words extracted from SMS means less information to work with. Many feature selection techniques are used in the area of text classification such as mutual information, CHI statistics, Information Gain, Term strength, document frequency, etc. The purpose of feature selection in this paper is to select a subset of features from all the features in SMS samples which should contain as much classification information as possible. Some of the features appear rarely in short messages, but they are very important for classification, such as "gun". Some of the features appear frequently in all short messages, such as "good", but it does little to classification.

Mutual information function describes the relevance of a feature to a category. In SMS classification, if the value of mutual information function of feature t_k to categories C_i is high, that is not to say the value of mutual information function of feature t_k to categories C_j is low. So if t_k frequently appears in multiple classes, its effect to the classification result is little. It even produces the interference in the process of probability calculation, which will reduce the filter's performance. In this paper, the mutual information function is improved, If the feature t_k is frequently appear in only one category, it has a higher weight value. $w(t_k)$ is the value of feat_val. It is calculated according to the formula (1), (2), and (3).

$$w(t_k) = \sqrt{\sum_{i=1}^{i=n} (MI(t_k, C_i) - MI_{avg}(t_k))^2} \quad \textbf{(1)}$$

$$MI(t_k, C_i) = \log_2 \frac{P(t_k \mid C_i)}{P(t_k)P(C_i)} \quad \textbf{(2)}$$

$$MI_{avg}(t_k) = \sum_{i=1}^{n} P(C_i) MI(t_k, C_i) \quad \textbf{(3)}$$

$P(t_k)$ represents the quotient which the occurence of t_k belongs to the occurence of all features. $P(C_i)$ represents the quotient which the number of messages belong to type C_i divide the the number of messages of the training set. It can be calculated according to the formula (4). The online SMS spam filtering algorithm reads its value from table SMS_category. $P(t_k \mid C_i)$ represents the quotient which the number of occurrences of feature t_k in class C_i divide the number of occurrences of all features in class C_i (t_k is the probability of term t given presence of class C_i), the value of it can be read form table Feature_cat.

$w(t_k)$ is the weight of feature t_k , it describes the ability of feature t_k to distinguish C_i from other categories. Some of the features is selected and inserted into the feature library according to the value of $w(t_k)$. The larger value of $w(t_k)$, the higher priority for being selected.

4.2 Design of the Feature Library

In order to reduce the CPU consumption on the execution of SMS spam filtering algorithm and reduce memory capacity occupied by characteristic data of SMS samples, this paper designs the feature library to save the characteristic data of the SMS samples, which include SMS categories table "SMS_category", feature information table "Feature_info" and feature categories table "Feature_cat".

SMS_category (cat_id, category, samp_nu, flag, PCk). Cat_id is an identifier of a SMS category, and is the primary key of table SMS_category. Category is the name of a SMS category. Short messages can be set by users into different categories. For example, messages are divided into categories of merchandising, pornographic and illegal information, daily greetings, business communication, real

estate sales, insurance. Wherein merchandising, pornographic and illegal information categories are of spam messages. Daily greetings and business communication categories are of normal messages. Real estate sales and insurance categories can be set as normal messages or spam messages by users according to their preferences. Samp_nu is the number of the samples belong to the SMS category identified by the cat_id. Flag represents whether this category is spam messages, where '1' represents the spam category and '0' represents the normal category. PCk is the prior probability of type C_k. $P(C_k)$ is the value of PCk which can be calculated according to the formula (4). If a user changes the flag attribute of a SMS category, the probability of a message belonging to that category does not need to change accordingly, so does other relevant data in the feature library, they can be automatically updated. SMS spam filtering system can also adapt to the new category settings on real-time.

$$P(C_k) = \frac{S_{C_k}}{S} \tag{4}$$

Where S_{C_K} is the number of messages of class C_k. S is the number of all the messages.

Feature_info(feat_id, feat, feat_val, rec_time). Feat_id is the identifier of a feature, and is the primary key of table Feature_info. Feat is the name of the feature identified by the feat_id and feat_val is its weight value of classification. Rec_time records the last time of the feature being accessed by the on line SMS spam filtering system. In order to control growth rate of the storage space occupied by the feature library and reduce the storage burden on the client, this article remove features from the feature library periodically according to the value of rec_time.

Feature_cat(feat_id, Ck, feat_nu, feat_Ck). Feat_id is the identifier of a feature. Ck is the identifier of a SMS category. Feat_nu is the number of times a feature characterized by feat_id appears in short messages of class CK. Feat_Ck is the probability the feature appears in short messages of CK class. The value of feat_Ck is $P(t_i \mid C_k)$ which is calculated according to the formula (5). Feat_id reference feat_id property in table Feature_info. Ck reference cat_id property in table SMS_category. Word_id plus cat_id constitute the primary key of this table. The value of feat_Ck is saved in the feature library in order to avoid the repeated calculation during the execution of the SMS spam filtering algorithm. It can be automatically updated by the feature library updating algorithm described in section 4.2.

$$P(t_i \mid C_k) = \frac{N_{t_i\_c_K} + 1}{N_{c_K} + m} \tag{5}$$

N_{c_K} is the total number of occurrence of all features belong to C_k class. $N_{t_i\_c_K}$ is the number of occurrence of t_i belongs to C_k class, m is the total number of features in the feature library and repeated features are calculated only once.

4.3 Feature Library Updating Algorithm

The feature library updating algorithm is described as follows:

Algorithm FLUP(X)

Input: New SMS features table X, feature library L

Output: Updated feature library L

Read category information from X, update samp_nu, flag, PCk($P(C_k)$) in SMS_category table.

if class($csms$) = C_k

$$P(C_k) = \frac{N_{c_k} + 1}{N_{all} + 1} = \frac{N_{all}}{N_{all} + 1} \cdot P(C_k) + \frac{1}{N_{all} + 1} \quad (C = C_k) \tag{6}$$

if class($csms$) ≠ C_k

$$P(C_k) = \frac{N_{c_k}}{N_{all} + 1} = \frac{N_{all}}{N_{all} + 1} \cdot P(C_k) \quad (C \neq C_k) \tag{7}$$

For each feature in X:
if it exists in the Feature_info table
then update feat_val in Feature_info table;
update feat_nu, feat_Ck($P(t_i \mid C_k)$) in Feature_cat table;
if class($csms$) = C_k and $t_i \in csms$

$$P(t_i \mid C_k) = \frac{N_{t_i\_c_K} + N_{ti\_csms}}{N_{c_K} + N_{all\_csms}} \quad (C = C_k \text{ 並且 } t_i \in csms)$$

$$= \frac{N_{c_k}}{N_{c_k} + N_{all\_csms}} \times P(t_i \mid C_k) + \frac{N_{ti\_csms}}{N_{c_k} + N_{all\_csms}} \tag{8}$$

if class($csms$) = C_k and $t_i \notin csms$

$$P(t_i \mid C_k) = \frac{N_{c_k}}{N_{c_k} + N_{all\_csms}} \times P(t_i \mid C_k) \ (C = C_k \text{ 並且 } t_i \notin csms) \tag{9}$$

else inset it into Feature_info table and Feature_cat table;
If the number of features N_{all} is greater than $N_{\max}$
Remove N features of lowest weight value from Feature_info table in L;
Cascading delete the data from Feature_cat table.

Csms is a newly classified short message waiting to be inserted into the feature library. N_{c_k} is the number of the features of class C_k . N_{all} is the total number of the features in the feature library. N_{ti_csms} is the number of occurrences of t_i in csms. N_{all_csms} is the number of occurrences of all features in csms.

4.4 Classification Algorithm

The steps of optimized Naïve Bayes online SMS spam filtering algorithm is as follows.
Algorithm IMNB(nsms)
Input: a new short message nsms, feature library L
Output:the category of nsms(Spam message or normal message)
Extract the feature $t_1\ t_2\ ...t_n$ from nsms
For each C_k = cat_id in SMS_category table:
Read $P(C_k)$, $P(t_i \mid C_k)$ from L, compute

$$P(C_k \mid nsms) = \frac{P(C_k)\prod_{i=1}^{n} P(t_i \mid C_k)}{\sum P(C_k)\prod_{i=1}^{n} P(t_i \mid C_k)}$$

Find C_i, C_j where $P(C_i \mid \text{nsms}) = \arg\max_{k \in cat\_id} P(C_k \mid nsms)$ and

$$P(C_j \mid \text{nsms}) = \arg\max_{k \in cat\_id \&\& k \neq i} PcC_k \mid nsms)$$

If ($C_i \in$ nospam and $C_j \in$ nospam)
The class of nsms is nospam;
If ($C_i \in$ spam and $C_j \in$ spam)
The class of nsms is spam;
If (($C_i \in$ spam and $C_j \in$ nospam)&&($P(C_i \mid nsms) / P(C_j \mid nsms)$)>R)
The class of nsms is spam;
else the class of nsms is nospam;
If ($C_i \in$ nospam and $C_j \in$ spam)
The class of nsms is nospam.

Nsms is a new short message waiting to be classified. n is the number of features extracted from the nsms.

5 Experiments and Analysis

5.1 Experimental Environment

The experiment use Tiny210 development board as the hardware platform which is based on ARM CortexTM-A8. Samsung S5PV210 is the main processor. It's running speed can reach 1 GHZ, its memory is 512 MB. The SMS spam filtering algorithm is running on the embedded Linux operating system, SQLite 3.7.13 is used as the embedded database to save the feature library. The host machine install Linux operating system and arm-linux-gcc cross compilation environment.

5.2 Testing of the Feature Library size

Because of limited storage capacity and processing power, it is important to known the storage space occupied by the feature library. This experiment test the size of feature library file during the increasement of features. The results are shown in Fig. 2. When the number of features is 25, the size of feature library file is 5120 bytes. When the number of features is 54, the size of feature library file is 6003 bytes. When the number of features is 82, the size of feature library file is 7168 bytes. When the number of features reaches 170, the size of feature library file is up to 10218 bytes. If 10 features are extracted from each of short messages, the growth of the feature library file is about 351.59 bytes at every increase of one SMS sample. When 2000 short messages are inserted into the feature library, the file size of it is about 686.70 K. It is completely feasible to run on the general mobile platforms. This is acceptable because it can completely run on a mobile platform even the cheapest mobile phones.

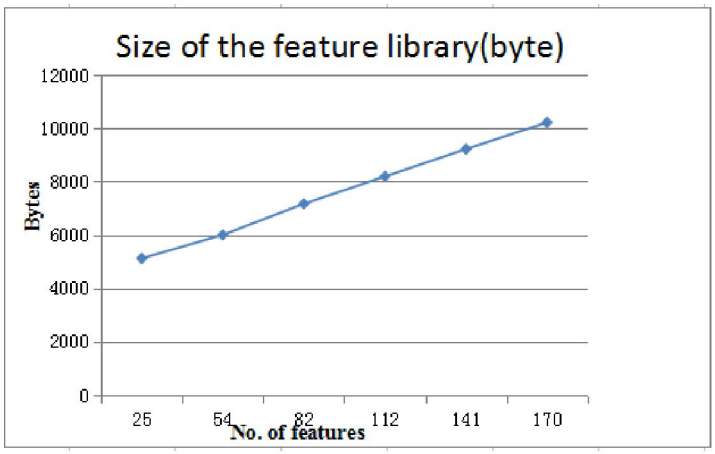

Fig. 2. The feature library size versus feature number

5.3 Testing of the Accuracy Rate of the Algorithm

There is no public corpus for Chinese SMS spam filtering. The experiment uses the SMS samples collected by ourselves, including 706 spam messages and 813 normal messages. The SMS samples is divided into training set and testing set. The training set includes 400 spam messages and 430 normal messages. The testing set consists of 306 spam messages and 383 normal messages. This article uses an evaluation index widely used in spam filtering. The filtering accuracy is calculated by the formula (10).

$$OA = (TP + TN) \Big/ (TP + FP + FN + TN) \tag{10}$$

TP is the number of spam messages correctly classified.
TN is the number of normal messages correctly classified.
FP is the number of normal messages misclassified as spam messages.

FN is the number of spam messages misclassified as normal SMS.

The accuracy rate of the optimized Naïve Bayes SMS spam filtering algorithm proposed in this paper is tested. The result is shown in Fig. 3. When the number of the features extracted from each of messages is between 9 to 13, the average accuracy rates of the algorithm are all over 90%, the accuracy rate reaches a maximum when the number of features is 11. When the number of features is more than 13, the accuracy rate shows a downward trend, which means that features with low weight value reduce the performance of the filtering algorithm.

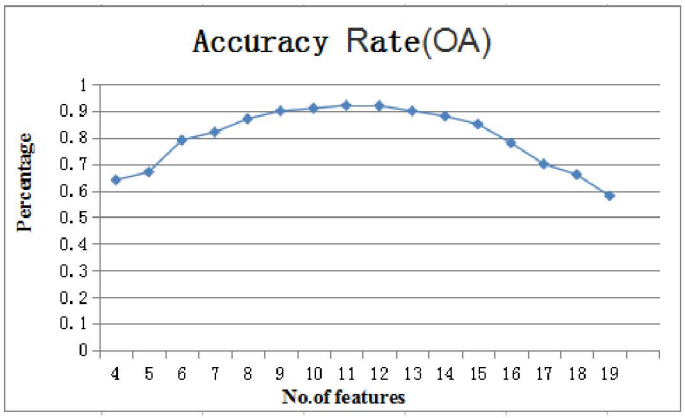

Fig. 3. Accuracy rate vs. number of features

5.4 Testing of Time Performance of the Filtering System

The 689 short messages in the testing set are classified using the algorithm proposed in this paper. The classification time of each message is recorded. Its value range from 0.05 seconds to 0.18 seconds, with an average time of 0.1 seconds. For comparison, the traditional Bayesian algorithm is also tested on tiny210 platform, the average classification time of a messages is 1.1 seconds. Because the improved algorithm omits a lot of calculation, the processing speed is greatly improved. We ignore the running time of the feature library updating algorithm because it is assumed that the updating could happen offline and hence the time spent by the updating algorithm is not relevant.

6 Conclusion and Future Work

This paper proposed a SMS spam filtering algorithm that requires minimal resources and reflects user's individual preference on an independent mobile phone which does not depend on another server or mobile operators. It obtained the reasonable accuracy, low storage consumption and quick filtering speed. In addition to these, our approach ensures security and privacy because the filtering algorithm runs on a mobile phone and each user has its own feature library, spammers have no way to access to the feature library. The limitations of the algorithm proposed in this paper is that it applies only to the text information filtering. Feature selection plays a very important role in short message classification, the more effective feature extraction and feature selection approaches remain as interesting future works.

Acknowledgement

The research is supported by higher school scientific research project of Gansu Province (Grant No.2013A-127). The authors would like to thank the Associate Editor and the reviewers for their valuable comments.

References

[1] Yesky YORK News, The global mobile phone users will amount to 4.8 billion by the end of 2016, <http://news.yesky.com/

478/98617478.shtml>, 2015.

[2] M.A. Balubaid, U. Manzoor, B. Zafar, A. Qureshi, N. Ghani, Ontology based SMS controller for smart phones, International Journal of Advanced Computer Science and Applications 6(1)(2015) 133-139.

[3] T.A. Almeida, J.M.G. Hidalgo, A. Yamakami, Contributions to the study of SMS spam filtering: new collection and results, in: Proc. the 11th ACM Symposium on Document Engineering in DocEng'11, 2011.

[4] Xinhua Net, China Mobile Internet security report in the first half of 2015.360 Internet security center, <http://news. xinhuanet.com/tech/ 2016-01/29/c_128684179.htm>, 2016.

[5] H. Zhang, W. Wang, Application of Bayesian method to spam SMS filtering, in: Proc. International Conference on Information Engineering and Computer Science, 2009.

[6] I. Santos, C. Laorden, B. Sanz, P. G. Bringas, Reversing the effects of tokenisation attacks against content-based spam filters, International Journal of Security and Networks 8(2)(2013) 106-116.

[7] T.M. Mahmoud, A.M. Mahfouz, SMS spam filtering technique based on artificial immune system, International Journal of Computer Science Issues 9(2)(2012) 589-597.

[8] L. Shi, Q. Wand, X. Ma, M. Weng, H. Qiao, Spam email classication using decision tree ensemble, Journal of Computational Information Systems 8(3)(2012) 949-956.

[9] A. Rodan, H. Faris, J. Alqatawna, Optimizing feedforward neural networks using biogeography based optimization for e-mail spam identification, International Journal Communications, Network and System Sciences 9(1)(2016) 19-28.

[10] A.A., Al-Hasan, E.-S.M. El-Alfy, Dendritic cell algorithm for mobile phone spam filtering, Procedia Computer Science 52(2015) 244-251.

[11] I. Joe, H. Shim, An SMS spam filtering system using support vector machine, in: Proc. Future Generation Information Technology, 2010.

[12] M. Healy, S. Delany, A. Zamolotskikh, An assessment of case-based reasoning for short text message classification, in: Proc. AICS'04, 2004.

[13] J.M.G. Hidalgo, G.C. Bringas, E.P. Sanz, F.C. Garc, Content based SMS spam filtering, in: Proc. ACM Symposium on Document Engineering, 2006.

[14] T.B Shahi, A. Yadav, Mobile SMS spam filtering for Nepali text using naïve Bayesian and support vector machine, International Journal of Intelligence Science 4(1)(2014) 24-28.

[15] T.A. Almeida, J.M.G. Hidalgo, A. Yamakami, Contributions to the study of SMS spam filtering: new collection and results, in: Proc. the 2011 ACM Symposium on Document Engineering (ACM DOCENG'11), 2011.

[16] H. Xia, P. Luo, M. Gao, Filtering algorithm for SMS spam based on design science, Computer Engineering and Applications 52(15)(2016) 111-116.

[17] Y.-H. Xu, M.-Y. Liu, Content-based junk short message filtering for mobile phone, Journal of Beijing Information Science and Technology University 28(1)(2013).

[18] H. Song, H. Kim, K. Han, J. Choi, et al. A Sub-2 dB NF dual-band CMOS LNA for CDMA/WCDMA applications, IEEE Microwave and Wireless Components Letter 18(3)(2008) 212-214.

[19] K. Yadav, A. Malhotra, P. Kumaraguru, R. Khurana, D.K. Singh, Take control over your SMSes: a real-world evaluation of a mobile-based spam SMS filtering system, <https://repository.iiitd.edu.in/jspui/handle/123456789/38>, 2012 (accessed 12.05.14).