Variable Length Concentration based Feature Construction Method for Spam Detection

Yang Gao, Guyue Mi and Ying Tan
Key Laboratory of Machine Perception (MOE), Peking University
Department of Machine Intelligence, School of Electronics
Engineering and Computer Science, Peking University, Beijing, 100871, China
Email: {gaoyang0115, miguyue, ytan}@pku.edu.cn

***Abstract*—In the field of spam detection, concentration methods have been proposed for feature construction in recent years, which convert emails into fixed length feature vectors. This paper presents a novel method aiming to break through the limit of feature vector's length. Specifically, the method uses a fixed-length sliding window to divide each email into several sections. The number of sections depends on the length of each email. Consequently, length of feature vectors varies from each other and this paper names them variable length concentrations (VLC). This method can acquire adaptive feature vectors according to different lengths of emails. However, general classifiers are not suitable for this kind of feature vectors, because they are not able to handle fixed-length inputs. As a result, this paper applies recurrent neural networks (RNNs), whose inputs are not restricted by the length, to achieve spam detection. Recall, precision, accuracy and F_1 measure are taken to evaluate the method's performance. Experimental results on the classic corpora, PU1, PU2, PU3 and PUA, show that VLC performs significantly better than previously proposed methods, which provides support to the effectiveness of our method.**

I. Introduction

Spam emails have always been considered as an increasingly serious problem to the development of Internet. According to the CYREN Internet Threats Trend Report, 55 billion emails are produced every day in the second quarter of 2014[1]. The statistics from Symantec intelligence report [2] demonstrate that averagely 61.3% of the global emails are spam in the past twelve months. Numerous spams not only occupy great resources of Internet, but also endanger the network security when they carry viruses and malicious codes. Moreover, spam takes much time of people to tackle with them, decreasing productivity considerably[3].

In the field of spam filtering, many approaches have been proposed to distinguish spam from email traffic. Among these approaches, intelligent detection methods are the most effective ways[4], [5], [6], [7], [8], [9]. There are three main related research fields for intelligent anti-spam, which are term selection, feature extraction and classifier design. Among these fields, feature extraction is crucial to the process of spam filtering, because it can directly affect performance of classifiers.

In our previous research, we extracted global or local concentrations from emails and analyzed their performance[10], [11], [12], [13]. However, like many other methods, the lengths of feature vectors are fixed. As a result, the feature vectors may have redundant information when am email is short, and they may have information missing when an email is long. To solve this problem, we propose a variable length concentration (VLC) based feature construction method for anti-spam system, which acquires feature vectors adaptively according to different lengths of emails. Similarly to previously proposed LC approach, the implement of VLC approach is also designed using a sliding window. But the difference between LC approach and VLC approach is that the length of VLC feature vectors can be variable, which never increases redundancy or intercepts information. After converting each email into a corresponding VLC feature, we can obtain feature vectors reflecting position-related information. At the same time, we apply the variable-length vectors into RNN, which has the ability of memory to remember and deal with emails' information. The performance of the VLC is investigated on four classic corpora namely PU1, PU2, PU3 and PUA. Meanwhile, accuracy and F_1 measure are mainly utilized to evaluate the experimental results.

In Section II, we introduce the related works. In Section III, the proposed VLC-based feature extraction approach is presented in detail. Section IV introduces the detailed experimental setup and results. Finally, we conclude the paper with a detailed discussion.

II. Related Works

This section introduces concentration related methods [14][15], such as global concentration, local concentration, and adaptive concentration method, all of which have close relationship with our work.

A. Global Concentration Method (GC)

Inspired from the human immune system, Tan and Ruan[10][11] proposed the global concentration method, in which self and non-self concentrations were calculated by evaluating terms in self and non-self libraries. The terms in the two libraries were selected based on the tendencies of them. If a term tends to appear in legitimate emails, it would be added to the self library. To the contrary, terms tending to appear in spam would be added to the non-self library. With the help of the two libraries, each email is transferred into a 2-dimensional feature vector by calculating the self and non-self concentrations of the email.

B. Local Concentration Method (LC)

The local concentration method proposed by Zhu[12], [13] aims at extracting position-correlated information from mes-

Prof. Ying Tan is the corresponding author

978-1-4799-1959-8/15/$31.00

sages effectively. Similar to the GC, two kinds of libraries are generated after term selection. But the next stage is different between GC and LC. Compared with GCs transforming each message to a 2-dimensional feature, the LC method uses a fixed-length or a variable-length sliding window to divide the message into individual areas. Finally, each area of a message is converted to a corresponding LC feature and emails are transformed to multi dimensional feature vectors.

C. Adaptive Concentration Method

In our previous work, we have proposed a method which can adaptively choose GC or LC according to length of each message [16]. This method considers information loss of GC and redundance of LC, and then it aims at taking both advantages of the two concentration approach. The main point of Adaptive Concentration Method is evaluating each email with the help of GC firstly. And then according to each email's evaluating result, it determines whether GC or LC can describe the email better.

D. Feature Extraction Approaches

1) Bag-of-Words (BoW): In spam filtering, BoW is one of the most commonly used feature extraction methods [17]. It converts a message to a d-dimensional-vector with the qualifying clause, which has been selected by a selecting method. In the vector, x_i indicates the occurrence function in the message. There are two main types of x_i: Boolean type and frequency type. In the Boolean case, x_i is set to 0 if it doesn't appear in the message, or it is set to 1. While in the frequency type, x_i is calculated as the frequency of term t_i in the message.

2) Sparse Binary Polynomial Hashing (SBPH): SBPH is an feature extraction method which extracts large numbers of different features with the help of an N-term-length sliding window [18]. The sliding window shifts over the incoming message stepped by one term. Features are extracted from the window at each step. The newest term is retained in the window, and the others are retained or moved in order to mapping the window to different features. SBPH is a promising method in consideration of classification accuracy. Nevertheless, it produces a lot of features so that the computational complexity is a heavy burden.

3) Orthogonal Sparse Bigrams (OSB): OSB was proposed by Siefkes et al. [19] to extract smaller size of features. It also use an N-term-length sliding window to extraction features. But different from SBPH, OSB only considers term-pairs with a common term. For each movement of window, the newest term is retained and one of others is also retained, while other terms are wiped off. As a result, feature is mapped from the remaining term-pair. OSB performs slightly better than SBPH in the experiments in [19].

III. VLC-based Feature Extraction Method

A. Background

Concentration method belongs to artificial immune system (AIS), which was proposed in the 1990s as a novel computational intelligence model [20]. AIS was inspired by biological immune system (BIS), which has the ability of distinguishing 'self cells' and 'non-self cells'. As a result, it can protect bodies from assaults of pathogens. Similarly, one main point of AIS is to distinguish between 'self' and 'non-self'. And there are large numbers of AIS models having been proposed for spam detection [21][22][23].

By far, global and local concentration method have been proposed for detecting spam emails. In global concentration (GC) method, each message is transformed into a two-dimensional feature vector, regardless of the length of the message. As for local concentration (LC) method, it extracts position-correlated information from messages, which overcomes the defect of GC. Specifically, the extraction uses fixed-length sliding window or variable-length sliding window to divide each message into different areas.

In this paper, we propose a VLC model, transforming messages into different dimensional feature vector, according to different length of the messages. In the VLC model for spam detection, feature vectors are constructed from messages through term selection methods and tendency decisions. After the feature construction, we get a series of feature vectors with different dimensionality. And finally, we combine those feature vectors of different length with recurrent neural networks, in order to making full use of the variable length concentration vectors.

B. Generation of Gene Libraries

Inspired from biological immune system, if a term mostly tents to occur in spam emails, it belongs to the non-self library. On the contrary, if a term tends to appear in legitimate emails, it is most likely to belong to the self library. However, the amount of terms is so large that if we intend to make use of all terms, it will lead to high computation complexity. As a result, in this paper, we use information gain to calculate and sort the importance of each term and discard 95% unimportant terms appearing in all emails. Algorithm 1 shows the detailed algorithm to build gene libraries, where $P(t_i|c_l)$ means the probability that term t_i belongs to legitimate emails and $P(t_i|c_s)$ means the probability that term t_i belongs to spam emails. And tendency threshold θ refers to the difference between $P(t_i|c_l)$ and $P(t_i|c_s)$.

C. Construction of Variable Length Feature Vectors

After gene libraries generation, we can construct the variable length feature vectors. Assuming that spam detective set DS_s stands for terms from spam library and legitimate detective set DS_l stands for terms from legitimate library, then with the help of DS_s and DS_l, we can convert each email into its corresponding feature vector. In detail, during the conversion, we use a sliding window to split each message into several locations, and then calculate concentration in each window. Algorithm 2 shows procedure of the construction of variable length feature vectors. In the procedure, M(t_j, DS_s) means the matching degree between term t_j and detective set DS_s. Besides, SC_j means spam concentration of window j and LC_j mean legitimate concentration of window j. And N_t is the total number of terms in message t.

$$SC_j = \frac{\sum_{j=1}^{\omega_n} M(t_j, DS_s)}{N_t} \tag{1}$$

Algorithm 1 Generation of Gene Libraries

```
Initialize gene libraries, detector $DS_s$ and $DS_l$ to the empty;
Initialize tendency threshold $\theta$ to predefined value;
Tokenization about the emails;

for each word $t_k$ separated do
    According to the term selection method, calculate the importance of $t_k$ and the amount of information I($t_k$);
end for

Sort the terms based on the I(t);
Expand the gene library with the top m% terms;

for each term $t_i$ in the gene library do
    if $\|P(t_i|c_l) - P(t_i|c_s)\| > \theta$, $\theta \geq 0$ then
        if $P(t_i|c_l) - P(t_i|c_s) < 0$ then
            add term $t_i$ to the spam detector set $DS_s$;
        else
            add term $t_i$ to the legitimate detector set $DS_l$;
        end if
    else
        abandon this term, because it contains little information about those emails;
    end if
end for
```

Algorithm 2 Construction of Variable Length Feature Vectors

```
Choose $\omega_n$, which indicates the number of terms in each sliding window;
Move the $\omega_n$-term sliding window to separate each email, without overlap.

for each moving window do
    for each term in the moving window do
        calculate the matching M($t_j$, $DS_s$) between term $t_j$ with $DS_s$;
    end for
    According to (1), calculate the concentration of spam terms $SC_j$;
    According to (4), calculate the concentration of legitimate terms $LC_j$;
end for

Combine local concentration in each sliding window to construct the variable-length concentration feature vector: $<(SC_1, LC_1), (SC_2, LC_2), \ldots, (SC_K, LC_K)>$
```

$$DS_s = \{d_1, d_2, ..., d_m\} \tag{2}$$

$$LC_j = \frac{\sum_{j=1}^{\omega_n} M(t_j, DS_l)}{N_t} \tag{3}$$

$$LC_j = \{d_1, d_2, ..., d_n\} \tag{4}$$

D. Structure of VLC Model

To implement spam detection based on our VLC model, a general structure of the VLC model is designed, which is shown in Fig. 1. The tokenization is just a simple step to tokenize messages into words (terms) by examine blanks or other delimiters. And terms selection, VLC calculation and RNN training are described as follow:

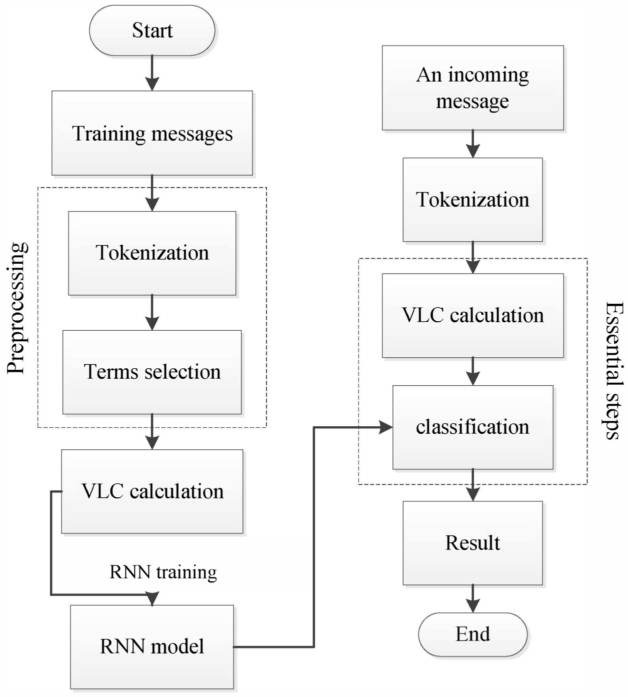

Fig. 1. Training and classification steps of the VLC model

1) Terms selection: After tokenization, messages are divided into large numbers of terms, which cause high computational complexity. As a result, the terms selection is necessary to remove some less informative terms, in order to reduce computational complexity. In our experiments, information gain (IG) [24] is applied to the model to calculate importance of the terms. The calculation formula of information gain is defined as (5).

$$I(t_i) = \sum_{C\in(C_s,C_l)} \sum_{T\in(t_i,\bar{t_i})} P(T,C)\log\frac{P(T,C)}{P(T)P(C)} \tag{5}$$

where C indicates an email's class (C_s and C_l are the spam and legitimate email classes) and T denotes that whether term t_i appears in the email or not. And all the probabilities are estimated from the whole data set.

2) VLC calculation: As we mentioned above, LC can reflect area-correlating information about messages, which improves the performance of spam detection. However, during feature construction, LC may lose some information or increase some redundancy. Consequently, we propose VLC method to calculate variable-length feature vectors with the help of fixed length sliding window, whose length equals a certain number of terms in the window. And in our experiments, we set different values of the length of sliding window to compare their performances, just as Fig. 3 to Fig. 6 show to us, aiming to find the most suitable length of sliding window for different corpora.

3) Recurrent Neural Networks (RNNs): Recurrent neural networks are inspired by the cyclical connectivity of neurons in brain, which introduce iterative function loops to store information [25]. One of the difference between a multilayered perceptron (MLP) and an RNN is that an MLP maps inputs to output vectors directly, whereas an RNN can map whole previous inputs to each output. In other words, the RNNs allow a "memory" of previous inputs which stay in the networks and have effect on the outputs.

In this paper, we focus on a simple RNN containing a single, self connected hidden layer, as shown in Fig. 2. Although it is similar to a multilayered perceptron, there are also big improvements between them. An MLP just simply maps from input to output vectors, whereas an RNN allows a "memory" of previous inputs to stay in the network, and thereby influences the network output.

However, for standard RNN architectures, the networks' ability to hold contexts is quite limited. In other words, the influence of a given input on the hidden layer, and therefore on the network output, either decays or blows up exponentially as it cycles around the network's recurrent connections, which is referred as the vanishing gradient problem [26][27]. As a result, in Fig. 2, we can see that the hidden layer is composed of Long Short-Term Memorys(LSTMs), which are used to tackle the vanishing gradient problem to enhance the "memory" of the network [28].

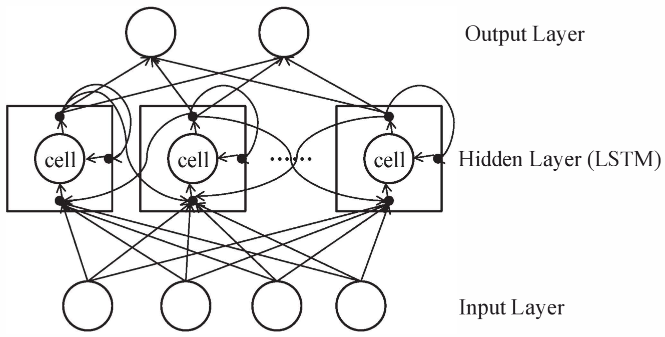

Fig. 2. A Recurrent Neural Network with LSTM

The following function (6) shows that the forward pass of an RNN is similar to MLP, except that the activations arriving at the hidden layer are from both the current external input and the hidden layer activations from the previous timesteps.

$$\mathrm{a}_{\mathrm{h}}^{\mathrm{t}} = \sum_{i=1}^{\mathrm{I}} \omega_{\mathrm{ih}} \mathrm{x}_{\mathrm{i}}^{\mathrm{t}} + \sum_{h=1}^{\mathrm{H}} \omega_{\mathrm{h'h}} \mathrm{b}_{\mathrm{h'}}^{\mathrm{t}-1} \tag{6}$$

$$\mathrm{b}_{\mathrm{h}}^{\mathrm{t}} = \theta_{\mathrm{h}}(\mathrm{a}_{\mathrm{h}}^{\mathrm{t}}) \tag{7}$$

where I means input units and H means hidden units. Let $\mathrm{x}_{\mathrm{i}}^{\mathrm{t}}$ be the value of input i at time t, and $\mathrm{a}_{\mathrm{h}}^{\mathrm{t}}$ and $\mathrm{b}_{\mathrm{h}}^{\mathrm{t}}$ be the input to unit h at time t and the activation of unit h at time t. And ω_{ih} means weights between input i and unit h, as well as $\omega_{\mathrm{h'h}}$ means weights between unit $\mathrm{h}^{'}$ and unit h. Function (7) shows the activation of unit h at time t.

At the beginning of training, messages have been transformed into feature vectors with different length through terms selection and VLC calculation. Then we take these vectors as inputs of RNN, because RNN can handle input sequences with different length. Whats more, RNN is an effective structure for sequence learning tasks where the data is strongly correlated along a single axis and it has achieve good performance in the field of speech recognition and image recognition. Similarly, an email message can also be taken as a text sequence because of its content terms. As a result, RNN is taken as the classifier and experiments in Section IV compares performance among RNN and other classifiers.

E. Evaluation Criteria

In spam detection, many evaluation criteria have been proposed to evaluate performance of different spam filters [17][29]. Among them, we adopt recall, precision, accuracy and F_β measure to evaluate the comparison of filtering effect between the VLC method and some other prevalent approaches. Among them, accuracy and F_β measure are the most important because recall and precision are the common component of F_β measure. And their calculation functions are described as follow.

1) Recall: It reflects the ability that email filters find spam emails. The higher recall is, the more spam emails that cannot be detected. It is defined as follows:

$$R_s = \frac{n_{ss}}{n_{ss} + n_{sl}} \tag{8}$$

where n_{ss} means the number of spam emails that are classified correctly, and n_{sl} is the number of spam emails that are classified as legitimate ones by error.

2) precision: It measures that when classified as spam, how many emails are truly spam ones. The higher precision is, the fewer legitimate emails is classified as spam mistakenly. It is defined as follows:

$$P_s = \frac{n_{ss}}{n_{ss} + n_{ls}} \tag{9}$$

where n_{ls} means the number of legitimate emails mistakenly classified as spam ones.

3) accuracy: It is a kind of criterion that can reflect overall performance of filters. The higher accuracy is, the more emails are classified correctly. It is defined as follows:

$$P_s = \frac{n_{ss} + n_{ll}}{n_s + n_l} \tag{10}$$

where n_{ss} is the number of spam emails correctly classified, n_{ll} is the number of legitimate emails correctly classified, n_s is the number of spam emails and n_l is the number of legitimate emails.

4) F_β measure: It is composed of recall and precision, which can also reflect overall performance of filters in another aspect. It is defined as follows:

$$F_\beta = (1+\beta^2)\frac{R_s P_s}{\beta^2 P_s + R_s} \tag{11}$$

Same with the paper [17], we adopt $\beta = 1$. As a result, it is referred to as F_1 measure.

IV. Experiments

A. Experimental setup

In 2004, Androutsopoulos and his colleagues [30] collected and published the series of PU data sets namely PU1, PU2, PU3 and PUA, which are widely used in spam detection research. Among them, PU1 and PU2 are all English emails, while PU3 and PUA consist of English and non-English ones. And in this paper, our experiments are organized on the four data sets. To ensure objectivity, experiments are conducted with 10-cross-fold validation. In addition, recalls, precision, accuracy and F_1 measure are used to evaluate the results. And we take F1 measure as the comprehensive evaluation, which is the most important indicator. All experiments are conducted on a PC with Intel P7450 CPU and 2G RAM.

B. Parameter selection of experiments

1) Proportion of term selection: During term selection phase, all terms need to be filtered so as to reduce the size of gene libraries. When choosing terms, we need to consider cutting off noise terms and retaining important terms. And finally only $q\%$ of the terms is preserved.In practice, this parameter can be adjusted according to the time and space complexity.

According to [12], [13], the performance of experiments achieve best on PU data sets when q is set to 50. In the same way, we also choose 50% of terms to create the gene libraries.

2) Dimension of feature vectors: Because of the sliding window, each message is divided into several parts and converted into corresponding feature vector. In this paper, we fix the length of each sliding window to N, and a message containing m terms can be transformed into $\lfloor m/N \rfloor$ dimensional feature vector.

In our experiments, we set the parameter N to 5, 10, 15, $\ldots$, 30 to study the best feature vector dimension for the PU series corpora. And TABLE I to TABLE IV show the performances on different lengths of sliding window. At the same time, Fig. 3 to Fig. 6 shows us the performance comparisons between our proposed method which combines RNN with VLC (RNN-VLC) and other spam detection methods.

3) Parameters of RNN: In the RNNs, we use long short-term memory (LSTM) to enhance the memory ability of the network. In detail, the network have only one hidden layer which is consist of ten LSTM blocks. What's more, the learning rate of RNN is set to $1e^{-4}$ and the momentum is set to 0.9.

C. Experimental results on the VLC approach

In this section, we conducted comparison experiments to demonstrate the effectiveness of our VLC approach. Ten-fold-cross validation is adopted into these experiments to ensure objectivity. The average performance experiments are reported in TABLE I to TABLE IV, which show different classification results on different size of sliding windows. And Fig. 3 to Fig. 6 show comparisons among our models best results with other methods. According to these results, we can come to a conclusion that our proposed method performs better.

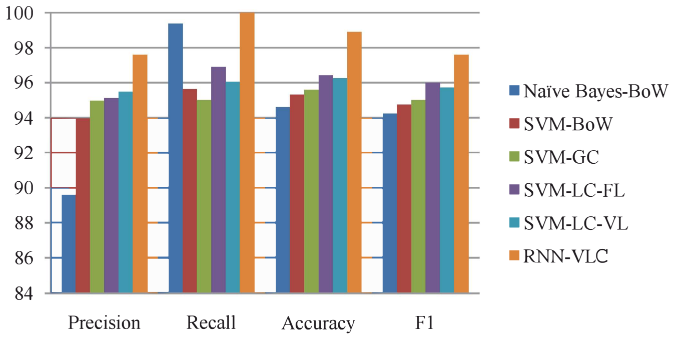

Fig. 3. Comparison of different methods results on corpus PU1

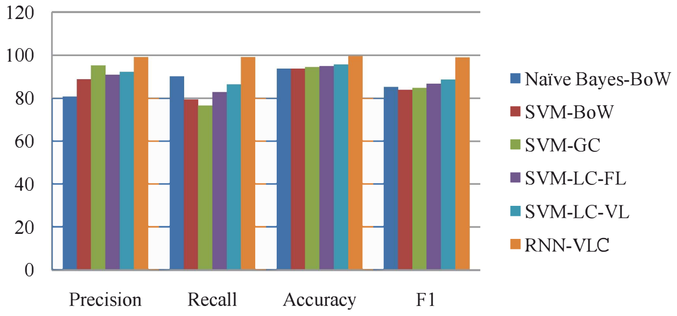

Fig. 4. Comparison of different methods results on corpus PU2

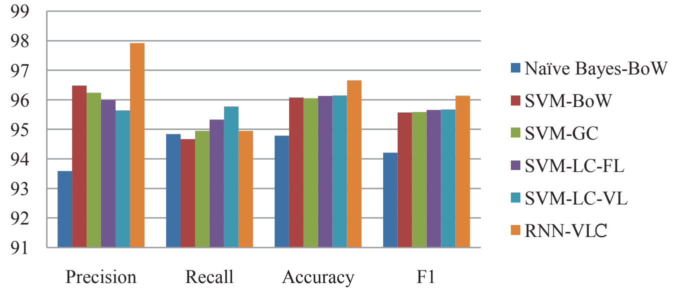

Fig. 5. Comparison of different methods results on corpus PU3

D. Discussion

We have proposed the VLC methods to adaptively construct feature vectors according to different length of messages. According to the experimental results, it is obvious that the proposed method can further enhance the effectiveness of immune concentration vectors. The disadvantage of GC is information loss and the disadvantage of LC is information redundancy, which mainly results from the fixed length of feature vectors. However, compared with GC and LC, VLC can adaptively convert each message into its corresponding feature vector which reduces information loss and redundancy.

In TABLE I to TABLE IV, we can see that different corpora have their unique best length of sliding window. For example,

TABLE I. PERFORMANCE OF VLC ON CORPUS PU1

Corpus	Size of Sliding Window	Accuracy(%)	Precision(%)	F_1(%)	Recall(%)
PU1	N=5	97.36	96.98	97.56	96.46
	N=10	**98.90**	97.58	**98.77**	**100**
	N=15	96.38	97.03	95.81	94.72
	N=20	94.48	97.82	93.42	89.61
	N=25	95.41	98.27	96.81	91.25
	N=30	94.70	97.64	96.13	91.16

TABLE II. PERFORMANCE OF VLC ON CORPUS PU2

Corpus	Size of Sliding Window	Accuracy(%)	Precision(%)	F_1(%)	Recall(%)
PU2	N=5	**99.40**	99.09	**98.86**	**98.69**
	N=10	99.01	99.33	97.38	95.71
	N=15	98.17	98.62	95.00	92.14
	N=20	96.23	97.76	88.99	83.17
	N=25	94.96	98.85	84.72	75.24
	N=30	92.25	98.89	75.03	61.43

TABLE III. PERFORMANCE OF VLC ON CORPUS PU3

Corpus	Size of Sliding Window	Accuracy(%)	Precision(%)	F_1(%)	Recall(%)
PU3	N=5	**96.66**	**97.92**	**96.14**	**94.45**
	N=10	95.37	96.87	94.64	92.55
	N=15	95.04	96.23	94.26	92.42
	N=20	94.65	96.64	93.74	91.15
	N=25	94.94	96.91	94.09	91.51
	N=30	94.43	97.23	93.43	89.97

TABLE IV. PERFORMANCE OF VLC ON CORPUS PUA

Corpus	Size of Sliding Window	Accuracy(%)	Precision(%)	F_1(%)	Recall(%)
PUA	N=5	89.12	87.19	89.08	91.60
	N=10	94.30	94.04	94.35	94.99
	N=15	**96.16**	**96.05**	**96.16**	**96.32**
	N=20	94.30	93.86	94.28	94.78
	N=25	94.21	93.68	94.19	94.86
	N=30	91.16	91.28	91.29	91.62

TABLE V. AVERAGE PERFORMANCE OF VLC ON PU CORPORA

Corpus	Size of Sliding Window	Accuracy(%)	Precision(%)	F_1(%)	Recall(%)
Average on PU corpora	N=5	95.64	95.30	95.41	95.30
	N=10	**96.90**	96.96	**96.29**	**95.81**
	N=15	96.44	**96.98**	95.31	93.90
	N=20	94.92	96.52	92.61	89.68
	N=25	94.88	96.93	92.45	88.22
	N=30	93.14	96.26	88.97	83.55

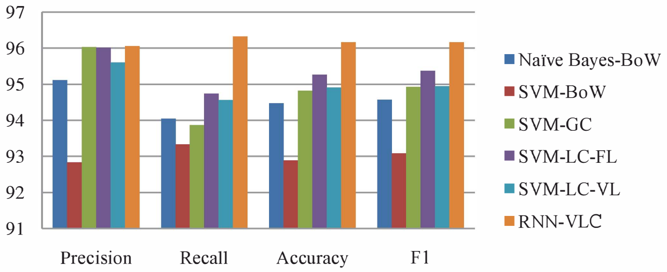

Fig. 6. Comparison of different methods results on corpus PUA

for corpus PU1, accuracy, precision, recall and F_1 measure all achieve peak value when N=10, which indicates that N=10 is the best length of sliding window for corpus PU1. In order to determine a common N-value on all corpora, we average performance results and show them in TABLE V. From the table, we can see that when N=10, accuracy and F_1 measure are both the best. As a result, we can determine N=10 as the common N-value. However, this is not always the most satisfying N-value. For PU3 and PUA, which contain not only English emails but also non-English ones, results are better when N-value isn't 10. As a result, we prefer that the best N-value depends on specific dataset.

And in Fig. 3 to Fig. 6, we choose the best performance of RNN-VLC on the four corpora to make a comparison with other methods. It is obvious that RNN-VLC achieves better than Naive Bayes, SVM-GC, SVM-LC and other methods on all the PU corpora. And considering F_1 measure, our RNN-VLC even improves the experiment performance by almost 18%, which is a significant improvement.

As a result, we come to a conclusion that different corpora have their suitable sliding window size and the proposed VLC method enhances the experimental effects to achieve better classification.

V. CONCLUSIONS

In this paper, we present a novel concentration vectors construction method to adaptively convert each email into its corresponding feature vector. During the phase of feature extraction, we use IG to evaluate and choose important terms. Then we use sliding window to convert emails into their corresponding feature vectors. To deal with these variable-length feature vectors, we choose RNN as the classifier to accomplish the final training and classification. And finally, the experimental results on PU corpora indicate that our proposed method is more effective and promising.

In the future, we intend to convert emails into variable length future vectors according to the length of emails messages and study its performance.

Acknowledgment

This work was supported by the Natural Science Foundation of China (NSFC) under grant no. 61375119, 61170057 and 60875080, and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China with grant no. 2015CB352302.

References

[1] CYREN, “Internet threats trend report: July 2014,” *Tech. rep.*, 2014.

[2] Symantec, “Symantec intelligence report: November 2014,” *Tech. rep.*, 2014.

[3] F. Research, “Spam, spammers, and spam control: A white paper by ferris research,” *Tech. rep.*, 2009.

[4] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, “A bayesian approach to filtering junk e-mail,” in *Learning for Text Categorization: Papers from the 1998 workshop*, vol. 62. Madison, Wisconsin: AAAI Technical Report WS-98-05, 1998, pp. 98–105.

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

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

[7] C. Wu, “Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks,” *Expert Systems with Applications*, vol. 36, no. 3, pp. 4321–4330, 2009.

[8] H. Shen and Z. Li, “Leveraging social networks for effective spam filtering,” *Computers, IEEE Transactions on*, vol. 63, no. 11, pp. 2743–2759, Nov 2014.

[9] R. Shams and R. Mercer, “Classifying spam emails using text and readability features,” in *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, Dec 2013, pp. 657–666.

[10] Y. Tan, C. Deng, and G. Ruan, “Concentration based feature construction approach for spam detection,” in *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*. IEEE, 2009, pp. 3088–3093.

[11] G. Ruan and Y. Tan, “A three-layer back-propagation neural network for spam detection using artificial immune concentration,” *Soft Computing*, vol. 14, no. 2, pp. 139–150, 2010.

[12] Y. Zhu and Y. Tan, “Extracting discriminative information from e-mail for spam detection inspired by immune system,” in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, July 2010, pp. 1–7.

[13] ——, “A local-concentration-based feature extraction approach for spam filtering,” *Information Forensics and Security, IEEE Transactions on*, vol. 6, no. 2, pp. 486–497, 2011.

[14] G. Mi, P. Zhang, and Y. Tan, “Feature construction approach for email categorization based on term space partition,” in *Neural Networks (IJCNN), The 2013 International Joint Conference on*. IEEE, 2013, pp. 1–8.

[15] ——, “A multi-resolution-concentration based feature construction approach for spam filtering,” in *Neural Networks (IJCNN), The 2013 International Joint Conference on*. IEEE, 2013, pp. 1–8.

[16] Y. Gao, G. Mi, and Y. Tan, “An adaptive concentration selection model for spam detection,” in *Advances in Swarm Intelligence*. Springer, 2014, pp. 223–233.

[17] T. S. Guzella and W. M. Caminhas, “A review of machine learning approaches to spam filtering,” *Expert Systems with Applications*, vol. 36, no. 7, pp. 10 206–10 222, 2009.

[18] W. S. Yerazunis, “Sparse binary polynomial hashing and the crm114 discriminator,” in *2003 Cambridge Spam Conference Proceedings*, vol. 1, 2003.

[19] C. Siefkes, F. Assis, S. Chhabra, and W. S. Yerazunis, “Combining winnow and orthogonal sparse bigrams for incremental spam filtering,” in *Knowledge Discovery in Databases: PKDD 2004*. Springer, 2004, pp. 410–421.

[20] D. Dasgupta, “Advances in artificial immune systems,” *Computational Intelligence Magazine, IEEE*, vol. 1, no. 4, pp. 40–49, 2006.

[21] G. Ruan and Y. Tan, “Intelligent detection approaches for spam,” in *Natural Computation, 2007. ICNC 2007. Third International Conference on*, vol. 3. IEEE, 2007, pp. 672–676.

[22] T. Oda and T. White, “Developing an immunity to spam,” in *Genetic and Evolutionary ComputationłGECCO 2003*. Springer, 2003, pp. 231–242.

[23] A. Secker, A. A. Freitas, and J. Timmis, “Aisec: an artificial immune system for e-mail classification,” in *Evolutionary Computation, 2003. CEC’03. The 2003 Congress on*, vol. 1. IEEE, 2003, pp. 131–138.

[24] J. R. Quinlan, “Induction of decision trees,” *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[25] A. Graves, *Supervised sequence labelling with recurrent neural networks*. Springer, 2012, vol. 385.

[26] S. Hochreiter, “Untersuchungen zu dynamischen neuronalen netzen,” *Master’s thesis, Institut fur Informatik, Technische Universitat, Munchen*, 1991.

[27] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *Neural Networks, IEEE Transactions on*, vol. 5, no. 2, pp. 157–166, 1994.

[28] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] E. Blanzieri and A. Bryl, “A survey of learning-based techniques of email spam filtering,” *Artificial Intelligence Review*, vol. 29, no. 1, pp. 63–92, 2008.

[30] I. Androutsopoulos, G. Paliouras, and E. Michelakis, *Learning to filter unsolicited commercial e-mail.* ” DEMOKRITOS”, National Center for Scientific Research, 2004.