Ensemble Decision for Spam Detection Using Term Space Partition Approach

Ying Tan, *Senior Member, IEEE*, Quanbin Wang, and Guyue Mi

***Abstract*—This paper proposes an ensemble decision approach which combines global and local features of e-mails together to detect spam effectively. In the proposed method, a special feature construction method named term space partition (TSP) is utilized to divide the whole term space into several subspaces and adopt different feature construction strategies on each of them, respectively. This method can make each term play a distinct and important role when conducting detection. This method is utilized and extended by introducing the sliding window technique to extract local features from e-mails. The global classifier and local classifiers are constructed on a global feature vector set and local feature vector sets, respectively, and together make the ensemble decision by adopting the voting technique. The principles of the TSP-based approach and mechanism of the ensemble decision method are presented in detail. Five different and standard benchmark corpora are applied to experiments for performance evaluation of this proposed method. Comprehensive experimental results show that the proposed method brings significant performance improvement and better robustness on the basis of the TSP-based approach. In addition, the proposed method outperforms the current prevalent and state-of-the-art approaches, especially when a comprehensive consideration of performance, efficiency, and robustness is taken. This endows it with flexible capability and adaptivity in the real-world applications.**

***Index Terms*—Ensemble decision, feature construction, machine learning, spam detection, term space partition (TSP).**

I. Introduction

E-MAIL becomes a necessary means of communication because of its convenience and high efficiency. But the number of spam is increasing since it can make big profits with a small spending by spreading advertisement or other disgust news to mail users. Some lawbreakers even send computer virus with an e-mail which results in a huge threat of computer. Spam, usually considered as unsolicited bulk e-mail or unsolicited commercial e-mail [1], has brought many troubles to our normal communication by e-mail. Ferris Research Group [2] indicated that the number of spam was so large that a majority of network bandwidth and mailbox server's storage are unable to be used in other important applications. The huge amount of spam also brought much interference to users and had very severe influences for people to work effectively. Moreover, the spam always had threats once it carrying malicious codes secretly which would affected the safety of computer and personal information. It can be seen from the Symantec Internet Security Threat Report 2015 [3] that there are nearly 60% of e-mails are spam in 2014 and the report of Cyren Internet Threats Trend [4] revealed a more serious statistical result with the spam rate more than 68% in the third quarter of 2014. In a word, spam detection is still a severe challenge.

In order to reduce economical losses caused by spam and improve working efficiency, people from different fields had proposed many anti-spam methods in diversiform perspective, including changing the protocol of e-mail sending [5] and simple keywords filtering [6], address protection [7], and so on [8], [9]. With the rapid improvement of artificial intelligence, more and more intelligent classification methods are adopted to cope with them, the most popular approach is with supervised learning methods [6], [10]–[12]. In addition, with its robustness and flexibility, automatic intelligent detection methods are widely used in spam e-mail filtering.

Similar to other classification tasks, intelligent spam detection can be decomposed into three important research steps, commonly called feature selection, feature construction, and classifier design, The purpose of feature selection [13]–[18] lies in selecting features which are much more important in the further processed steps and resulting in a lower dimensionality which is useful to save computation resource and improve accuracy of the classification model. Feature construction methods [19]–[24] discover the inner relationship among all existing features and transform them into a new set first, then use this set of features to construct sample vectors. Supervised machine learning methods [16], [25]–[29] are very useful in pattern recognition and have been proved to be effective in spam detection domain. The prevalent and commonly used approaches and techniques are introduced in Section II.

In this paper, an ensemble decision method for spam detection is proposed by utilizing both global features and local features of an e-mail in the process of decision making.

Manuscript received June 9, 2018; revised August 8, 2018; accepted August 27, 2018. This work was supported in part by the Natural Science Foundation of China under Grant 61673025 and Grant 61375119, in part by the Beijing Natural Science Foundation under Grant 4162029, and in part by the National Key Basic Research and Development Plan (973 Plan) Project of China under Grant 2015CB352302. This paper was recommended by Associate Editor M. Last. *(Corresponding author: Ying Tan.)*

Y. Tan and Q. Wang 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; qbwang362@pku.edu.cn).

G. Mi was 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. He is now with the Technology Department, Pingnan County (e-mail: gymi@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/TCYB.2018.2868794

Corresponding features are constructed by further exploiting and improving the term space partition (TSP)-based feature construction approach [30]. Sliding window technique with different length of windows is introduced to extract local features of the e-mail, as well as position correlated information. Global and local classifiers are constructed, respectively, based on corresponding feature vector sets and make ensemble decision with voting techniques. Five different benchmark corpora named PU1, PU2, PU3, PUA, and Enron-Spam are employed in our experiments to evaluate the performance of our novel spam detection method. As in standard classification performance analysis, we adopted accuracy and F_1 measure as the main criteria to compare our results with others.

The organization of other contents in this paper are as follows. Section II describes details of some widely used measurements of feature selection and some methods for constructing feature vectors. In addition, machine learning algorithms-based spam detection models are also introduced. The TSP-based feature construction approach is described in Section III. The proposed ensemble decision method is presented in Section IV. Section V shows the results and comparison of experiments. Finally, the conclusion of this paper is presented in Section VI.

II. Related Works

A. Feature Selection Metrics

Document frequency (DF) [14] is a common method for feature reduction and the simplest way for feature selection. For a certain word or term, it counts the number of documents in which the term appears. Once the frequency of each term in our corpora is obtained, we only selected the most informative terms between a minimum threshold and a maximal value to compose the feature sets. It is a method with lower computation since we always use the DF in training data to approximate a very large document corpora. In spam detection, DF of term t_i is calculated as

$$\mathrm{DF}(t_i) = \left|\left\{m_j | m_j \in M, t_i \in m_j\right\}\right| \tag{1}$$

where m_j indicates an e-mail from the training set M.

Term strength (TS) [14] is an estimation of term's importance which is judged by conditional probability that represents how widely a specific term like to appear in "closely related" documents. Professionally, it indicates how possible that a specific term will occur in the second document of two related enough samples if it has occurred in the first one already. We call these two samples as closely related documents provided that there exist many common words or phrases in them. What is more, we also consider those overlapping terms are much more informative than others. In spam detection, TS of term t_i is calculated as

$$\mathrm{TS}(t_i) = P(t_i \in y | t_i \in x) \tag{2}$$

where x and y indicates the two related samples from training set M.

Information gain (IG) [15] is an important evaluation criterion of feature selection. It is defined as a quantitative measure of the effect that a feature would brings to classification model. The more significant the effect, the more useful the feature, and it also resulted in a greater IG quantitatively. In spam filtering, term t_i's value of IG can be defined as

$$\mathrm{IG}(t_i) = \sum_{c\in(s,h)} \sum_{t\in(t_i,\bar{t}_i)} P(t,c)\log\frac{P(t,c)}{P(t)P(c)} \tag{3}$$

where c represents e-mail's label which belongs to spam (s) and ham(h), t_i and $\bar{t}_i$ indicates whether a term t_i is present or absent in each situation.

Term frequency variance (TFV) [16] is proposed based on DF, the difference to DF is the way how it select terms. We choose specific terms which occur frequently in each individual class but not with an overall threshold. In order to get the variance, it is necessary to calculate the term frequency in each independent category first. TFV of term t_i can be defined as (4) for spam detection

$$\mathrm{TFV}(t_i) = \sum_{c\in(s,h)} \left(T_f(t_i,c) - T_f^{\mu}(t_i)\right)^2 \tag{4}$$

where $T_f(t_i, c)$ represents the term frequency of t_i in a certain class c, $T_f^{\mu}(t_i)$ denotes the mean value of t_i's frequency in classes s and h.

Chi square (χ^2) [14] is a widely used hypothesis testing method. It is applied in statistical inference of categorical data, and including two rates or two constituent ratios, multiple rates or multiple constituent ratios, and correlation analysis of classification data. We use this to measure the lack of independence between term t_i and class c. If term t_i is useless for categorization, the Chi value is near to 0. To detect spam, term t_i's value of χ^2 can be calculated as

$$\chi^2(t_i) = \sum_{c\in(s,h)} P(c)\chi^2(t_i,c) \tag{5}$$

$$\chi^2(t_i,c) = \frac{|M|\left(P(t_i,c)P(\bar{t}_i,\bar{c}) - P(\bar{t}_i,c)P(t_i,\bar{c})\right)^2}{P(t_i)P(\bar{t}_i)P(c)P(\bar{c})}. \tag{6}$$

Odds ratio (OR) [17] is one of the three main methods of quantifying the relationship between the feature A and the feature B in the feature sets. In this paper, we calculate the value of OR by comparing the two values of odds that a feature appearing in the two opposite classes. In spam detection, since it is a binary classification problem, we can define OR of term t_i with class c as

$$\mathrm{OR}(t_i,c) = \frac{P(t_i|c)}{1-P(t_i|c)}\frac{1-P(t_i|\bar{c})}{P(t_i|\bar{c})}. \tag{7}$$

As normally, we calculate the log value of a specific term's OR for all classes and add them together to obtain the measurement of this term.

We have described some metrics of feature selection above, in which DF and TS are irrelevant to class information, but all the others need this information along with a certain term. The experiments with these different feature selection methods [14] demonstrate that IG and χ^2 are the most useful and effective metrics when doing feature dimension reduction. DF also performs as good as IG and χ^2, so it is a better alternative of IG or χ^2 when the time consumption of calculating this two measures is too heavy. In addition, some other feature selection methods can also be used for spam detection, such as negative selection algorithm [31], [32] of which had proved to be effective in malware detection task and distance measure-based approach [33].

B. Feature Construction Approaches

Bag-of-words (BoW), a coarse representation of text is used commonly in natural language processing that disregarding grammar and word order of the text. It also usually applied for feature construction in spam filtering on account of its simpleness [19]. This approach uses a feature vector $\vec{x} = [x_1, x_2, \ldots, x_n]$ with n-dimension which called frequency representation to represent an e-mail m. Each value x_i of this vector is indicates the number of each term in a preselected term set $T = [t_1, t_2, \ldots, t_n]$ of which appear in e-mail m. Particularly, the binary representation which only consider the occurrence of each term can also be used to form the vector and experimental results in [34] show it can perform as well as former version.

Tan *et al.* [21], [22] proposed a feature construction (CFC) method for spam detection based on concentration what is similar as immunoreaction of human beings do for virus. It calculates "self" and "nonself" concentrations with respect to self and nonself gene libraries which is established based on training data. The performance of artificial immune system for classification had been proven to be good enough on virus detection [35]. After the process of CFC method, a 2-D feature vector will be used to represent the e-mail so as to reduce the space complexity greatly. Experimental result shows the CFC approach performs much better than BoW in not only classification accuracy but also efficiency.

Zhu and Tan [23], [24] proposed a feature extraction method based on immune local-concentration (LC), and [36] had successfully used them for virus detection. By combining immune LC theory with statistical methods, the LC method is endowed with several distinctive characters. Term selection methods are utilized to filter out noise and reduce computational complexity. After that, they built the detector sets according to a gene-tendency function, which endows the LC method with capability of robustness and noise-resistance. In addition, position-correlated LC features are extracted from messages by using variable-length and fixed-length sliding windows. Experimental results shown that the LC method can extract statistically critical features for classification, and obtained better performance on the two criterions we adopted.

C. Prevalent Machine Learning Methods

In this section, we will introduce several commonly used learning-based methods in detail.

As we all know, Bayes methods intuitively calculate the posterior probability $P(C = c_k|X = x)$ based on prior and joint probability. It indicates the possibility that a sample x belongs to each class c_k directly. As a representative of Bayes methods, naïve Bayes (NB) is commonly used model in many fields with a strong assumption of which all features of a sample are mutually independent. It had been adopted to solve recommendation [37] and classification [38] tasks for a long time. On account of its efficiency and effectiveness, researchers had put much emphasis on NB to cope with spam detection problem, [25] successfully used it for spam filtering and led to many other achievements [39]–[41].

Support vector machine (SVM), a famous model in machine learning domain, which aims at finding the optimal hyperplane that maximizes the classification margin. Weight vectors of the optimal hyperplane are obtained by calculation on the training set. Combining SVM with other machine learning algorithms can significantly improve the power of pure SVM, such as Gu *et al.* [42] employed it with K-nearest neighbor (KNN) for discriminant analysis and Tan *et al.* [43] applied neural networks to train SVM in order to eliminated quadratic programming and obtained a powerful pattern recognition model. Drucker *et al.* [26] attempted to detect spam by SVM and obtained a better result than former works. From then on, SVM has also been widely used in spam detection [24], [44], [45].

Decision tree (DT), a commonly used discriminative model which constructs a tree with each attribute as node and the final predictive label as leaf, the edges among nodes indicate the value of the corresponding attributes. For a specific sample, we use rules generated from the root to each leaves to determine the final prediction, and the value of each feature is considered to choose the path from top to bottom. ID3 and C4.5 are famous DT algorithms, they are used to choose attributes for each node when constructing the tree. Carreras and Marquez [27] had adopted DT to filter spam but researches always apply them in a Boosting way on account of its mediocre performance.

Boosting can be simply considered as "two heads are better than one," it obtains a more powerful model by combining each weak learner together with some special rules and improves the overall capacity. Some also regarded it as a voting strategy [46]. A canonical combination rule resulted in a famous and effective Boosting method which named adaptive boosting (AdaBoost). The most important idea of AdaBoost is putting more emphasis on samples of which are error classified [47] and increasing the weights of these samples dynamically based on the training performance by each weak classifier. The next weak learner needs to focus on those samples which are hard to identify [48]. Each weak model is combined by weight of which is computed based on its individual performance to form the final classifier.

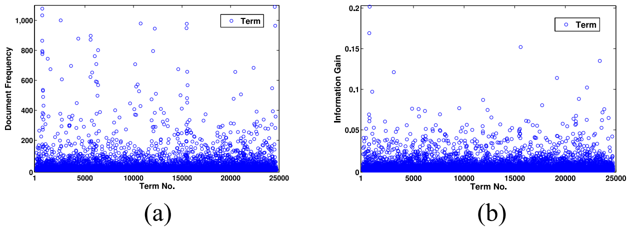
Random forest (RF) is an ensemble method based on weak learners too, but it work in a different style from Boosting. It resampling subset of training set for several times and constructs a DT with each subset, respectively. Each tree is combined together to do prediction for a new sample in a voting approach [49].

Some spam detection technologies which based on weak learners such as DT had shown their great ability in classification task. Carreras and Marquez [27] adopted DT as a weak model for AdaBoost and outperformed other detection approaches, such as DT and NB. Koprinska *et al.* [16] used RF based on simple DT to deal with spam filtering task and got much better experimental results compared with DT, SVM, and NB, even though the single DT and SVM were more complex than RF. In addition, Chakraborty *et al.* [50] figured out that spams had brought much trouble to social platforms, such as Facebook and Microblog, some attempts based on RF had been adopted to detect these social spams including [51]–[53] and some others also provided a promising results. Liu *et al.* [54] proposed a redistribution and asymmetric sampling method to solve imbalanced data problem in drifting Twitter spam detection task, they experimented on real time data and synthetic data independently and combined them with ensemble technique to make decision, their method made RF and C4.5-based ensemble model performs better.

Some machine learning methods had been combined with swarm intelligence algorithms to deal with classification problem, like work in [56] and other methods based on [55]. These kinds of approaches can be speed up by GPU-based parallel [57] to meet the requirements of application in real scene.

Inspired by the mechanism of biological neural networks, artificial neural network (ANN) is proposed to mimic its architecture. A large amount of artificial neurons are interconnected by weights which need to be learned to make the net working, and back-propagation is commonly used for this learning process. Clark *et al.* [28] utilized a fully connected ANN to cope with e-mail classification problem and result in a superior model than NB and KNN. More applications of ANN for spam detection could be found in [22], [58], and [59].

Different from former work with shallow networks, deep learning (DL) is good at learning more complex attributes with multiple levels and attracts more attention these years. In order to mitigate the problem of gradient vanishing and overfitting, Barushka and Hájek [60] adopted rectified linear units in multilayer perceptron (MLP) neural networks to detect spam. We applied one of the widely used deep neural networks named stacked auto-encoder (SAE) [61] for spam detection in previous work [29], and experimental results demonstrated that SAE performed better than the prevalent machine learning techniques introduced above on most of the corpus. Based on the success of some natural language tasks, in recent years, convolutional neural network (CNN) and recurrent neural network (RNN) are adopted to filter spam widely. Ren and Zhang [62] explored CNN and gated RNN with attention mechanism [63] to learn document level representation which is used to filter deceptive opinion spam, while Zhao *et al.* [64] used CNN with auxiliary word order characteristics to deal with this task. Jain *et al.* [65] proposed a semantic CNN which composed by CNN and a semantic layer to detect spam from social media, with the help of WordNet and ConceptNet, they obtained state-of-the-art results on two datasets. Ma *et al.* [66] detected rumors from Microblog with RNN and achieved more accurate results with quicker detecting speed. Detecting spam from social platforms had attracted more attention recently. Except some work we mentioned above, [67] and [68] are also focused on social media spam filtering. What is more, Sedhai and Sun [69] presented a semi-supervised model to detect Twitter spam which was useful for stream data. Semi-supervised methods were used frequently for text classification these years, the work in [70] obtained excellent performance for text sentiment classification, and it may also be useful for spam detection. Adversarial learning-based methods had been proved to be effective for malware detection [71] and it can also be adopted to filter spam.



(a) (b)

Fig. 1. Distribution of terms in PU1 with respect to feature selection metrics. (a) DF. (b) IG.

III. Term Space Partition-Based Feature Construction Approach

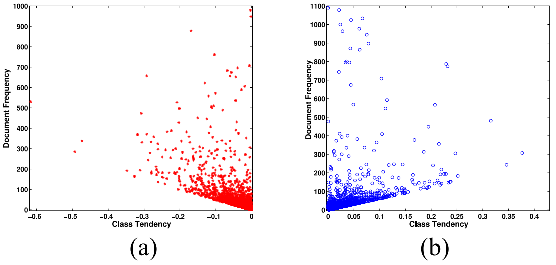
A. Motivation

As we all know, feature selection is indispensable when doing classification tasks of which the feature dimensionality is very high. In spam detection, we also need to design effective algorithms to cope with this challenge and make the model much more superior and stable. In other words, we should choose the better ones which are informative and distinct from the whole feature set. The better terms we select, the more higher performance we will obtain from the same spam detection model. On the other hand, features with less useful information for classification not only bring excess computation resource consumption, but also have severe disturbance since they always act as noisy terms. Fig. 1(a) and (b) shows the distribution of terms with two different select metrics DF and IG which we had described in the last section as goodness evaluation in PU1 corpus. PU1 is a standard benchmark corpus of spam detection which contains nearly 25 000 distinct terms. We can conclude that rare terms are much more discriminative and informative while the great majority of them are less useful, no matter what metrics will be adopted for feature selection because the distributions are similar.

After feature selection with respect to effective selection metrics we obtained terms which would be used for constructing feature vector to represent each specific sample. From the conclusion we got above, only a small part of feature set is obviously superior than others and can be used to generate feature vector with strong confidence. As in commonly used methods, several hundred terms are consider better than others based on the value of select metrics, each of them is used as an individual dimension of the feature vector which will be used in the following steps. Different from this approach, more terms (empirically more than 50% in [21], [22], and [24]) would be reserved and feature vectors are constructed by computing gene concentrations in some heuristic methods, all predominant terms are given equal emphasis with much lower scores, making the contributions of superior terms weakened. There are much little loss of information since more features are included.

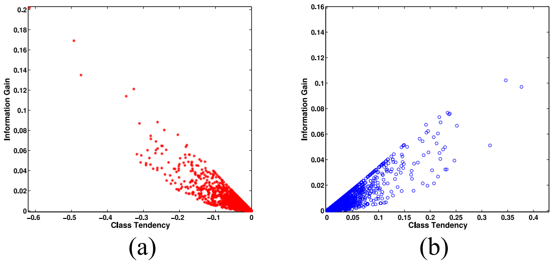
B. Principle

The feature construction method proposed in this paper which named TSP is designed for a reasonable selecting mechanism when choosing the components of feature vector. In



(a) (b)

Fig. 2. Distribution of terms in PU1 with respect to DF. (a) Spam terms. (b) Ham terms.



(a) (b)

Fig. 3. Distribution of terms in PU1 with respect to IG. (a) Spam terms. (b) Ham terms.

order to obtain sufficient and rational terms for spam detection, we split the whole feature set into several subsets and apply different feature construction methods to each of them, respectively, the dividing strategy results in a more efficient and better performance detection system.

As what we have mentioned above, feature selection metrics defined the ways how to evaluate the quality of each term. We adopted a simple vertical partition method based on the distribution characteristics of terms value with different selection metrics, the *Dominant Terms* are separated from *General Terms* in such a intuitive strategy. In addition, we call terms with high metric as dominant terms, these terms are more important and play dominant roles when classifying spam examples. The terms in dominant set have some common characters while others in the rest part are much less informative and discriminative. This part of terms also have positive impact when doing classification if we use them in a crafty approach.

In order to obtain features which are more discriminative, we defined the tendency of a feature appear in e-mails belonging to a certain class as *Class Tendency*, and we use this metric to partition the term space in a transverse way. As result, we split the whole terms set into *Spam Terms* from *Ham Terms*. We defined the calculation of class tendency as

$$\text{tendency}(t_i) = P(t_i|c_h) - P(t_i|c_s) \tag{8}$$

where $P(t_i|c_h)$ is the probability that term t_i appear if the e-mail is ham, and $P(t_i|c_s)$ is the probability when given a spam. Spam terms mean that their tendency value is negative and they are more likely to occur in spam while ham terms vice versa.

As a result, we can represent each term by a vector $\vec{t} = \langle\text{tendency}, \text{goodness}\rangle$ with 2-dimension. Afterward, we can obtain the new distribution of terms extracted from PU1 with different select metrics such as what shown in Figs. 2 and 3. And we decompose the four independent and nonoverlapping subspaces from original term space based on DF and IG. We call each of them as spam-dominant, ham-dominant, spam-general, and ham-general.

We set a crafted threshold manually when separate dominant terms from general terms. From the characteristics of each term subspace, we defined different features for different part. *Spam Term Ratio* and *Ham Term Ratio* for dominant terms, while *Spam Term Density* and *Ham Term Density* are corresponding to general terms. Detailed description is in Section III-C.

C. TSP Approach

The TSP approach that we adopted can be decomposed into three essential steps.

1) Preprocessing: Once we have to decide whether an e-mail is spam or not, the first and necessary operation is construct the representation vector of the e-mail. This operation called preprocessing is based on a technique named tokenization of which transforming text to a list of terms according to blank spaces and delimiters.

2) Term Space Partition: As we have described above, the detail of TSP contains two main steps which are term selection and TSP shown in Algorithm 1. In order to reduce the occupation of resource and redundancy of information, term selection is essential and we use parameter p to control the degree of selection.

After term selection, we need to separate dominant terms from general ones in a way called vertical partition of which is manually designed by a threshold. The value of threshold is determined based on the corresponding selection metrics, we defined the evaluation of it as in

$$\theta_{\text{dg}} = \frac{1}{r}(\tau_{\max} - \tau_{\min}) + \tau_{\min} \tag{9}$$

where $\tau_{\max}$ and $\tau_{\min}$ indicate the highest and lowest value of selecting metric among all terms in the training set, respectively, and r here is used to control the restriction level of dominant terms. If $\tau(t_i) \geq \theta_{dg}$, term t_i is regarded as a dominant term and belongs to general term set otherwise. All terms of which with $\text{tendency}(t_i) = 0$ are discarded when doing transverse partition process because of their uselessness.

3) Feature Construction: With aim to construct feature vectors for e-mails that waiting for detection, in order to make the vector be discriminative and effective, we put forward two kinds of features named *Term Ratio* and *Term Density* for dominant and general terms, respectively. The features we used are described as follows.

Dominant terms are in a very small quantity but play leader role in classification task. We should pay more attention to each of the terms in this set. We calculate term ration for spam and ham from each dominant term, respectively. The definitions of these two features are shown as

$$\text{TR}_s = \frac{n_{\text{sd}}}{N_{\text{sd}}} \tag{10}$$

where n_{sd} represents the number of independent terms in the e-mail we are considering which are also included in spam dominant term sets TS_{sd}, and N_{sd} indicates the number of

Algorithm 1 TSP

```
Input: All terms in training set.
Output: Partitioned term sets: $TS_{sd}$, $TS_{sg}$, $TS_{hd}$, $TS_{hg}$.
initialize preselected term set $TS_p$, spam-dominant term set $TS_{sd}$, ham-dominant term set $TS_{hd}$, spam-general term set $TS_{sg}$ and ham-general term set $TS_{hg}$ as empty sets
for each term $t_i$ occurs in the training set do
    calculate goodness evaluation $\tau(t_i)$ based on the adopted feature selection metrics
end for
sort the terms in descending order of evaluation
add the front $p\%$ terms to $TS_p$
calculate partition threshold $\theta_{dg}$ based on Eq. (9)
for each term $t_i$ in $TS_p$ do
    calculate $tendency(t_i)$ based on Eq. (8)
    if $tendency(t_i) < 0$ then
        if $\tau(t_i) \geq \theta_{dg}$ then
            add $t_i$ to $TS_{sd}$
        else
            add $t_i$ to $TS_{sg}$
        end if
    else
        if $tendency(t_i) > 0$ then
            if $\tau(t_i) \geq \theta_{dg}$ then
                add $t_i$ to $TS_{hd}$
            else
                add $t_i$ to $TS_{hg}$
            end if
        end if
    end if
end for
```

different terms in $\mathrm{TS}_{\mathrm{sd}}$. Ham term ratio is defined in the same way as

$$\mathrm{TR}_h = \frac{n_{\mathrm{hd}}}{N_{\mathrm{hd}}}. \tag{11}$$

The meaning of each component is similar as we talked in (10).

In contrast with dominant terms, general terms are in a large amount of number but redundant and less useful. As a result, we always despise those terms with much lower weights. The term density for each category based on each of the specific general term sets are calculated as (12) and (13)

$$\mathrm{TD}_s = \frac{n_{\mathrm{sg}}}{N_e}. \tag{12}$$

The meanings of elements n_{sg}, n_{hd} of each formula are similar as in (10) too, but with respect to general term space, while N_e represent distinct term number of an e-mail

$$\mathrm{TD}_h = \frac{n_{\mathrm{hg}}}{N_e}. \tag{13}$$

Term ratio and term density are two crucial but very different ideas used in feature construction step. Term ration represents the proportion of dominant terms appeared in a specific e-mail but term density indicates the ratio that how many terms of this e-mail are included in the general terms. The difference between these two concepts determines the roles they played in classification.

Finally, we compute TR_s, TR_h, TD_s, and TD_h according to (10)–(13) and combined these four items together to obtain the feature vector, i.e., $\vec{v} = <\mathrm{TR}_s, \mathrm{TR}_h, \mathrm{TD}_s, \mathrm{TD}_h>$.

IV. Ensemble Decision Using TSP Approach

A. Global and Local Features

In research of image recognition, global features are used to describe the overall characteristics of samples, while local features are used to express the detailed characteristics of samples [72], [73]. Based on the different functions and content as described, utilizing both global features and local features in characterizing images has been extensively studied and makes apparent performance improvements [74], [75].

After the feature construction process as what we talked above, the distribution of features from e-mails is obtained, and if the distribution of features for spam e-mail was significant different from that of ham ones, we regarded the method we adopted to construct feature vectors are useful. As mentioned above, a 4-D feature vector is utilized to represent each e-mail in our TSP method, this vector depicts the form of distribution about each specific e-mail with respect to four separate term spaces. Since each component of the feature vector is based on the whole e-mail, this vector is named global features and it is distinguishable and effective for a majority of spam and legitimate e-mails. But for some special samples of which the distribution is nearly similar with only different in some local parts, global features are not enough and we need to explore other approach to handle them.

We proposed a technique called sliding window in this paper to defeat such difficult situation. The sliding window is used to extract local areas and we implement the same TSP approach on these areas to obtain local features in contrast to the global ones. It is apparent that local features are focusing on only local parts and representing some details of an e-mail, it is obviously different from global features. In addition, local features could not only represent the difference of local areas of which always be concealed by global vectors, but also sensitive to the information of position differences which are always indispensable since spam e-mails are usually with odd beginning or ending.

B. Extraction of Local Features With Sliding Windows

We obtain local features of each local areas which defined by a sliding window strategy from the whole e-mails. The TSP feature vector in local domain is established in the same way as in global mode, we obtain a vector for each local area with regards to the four subspace as described above. Fig. 4 shows the process of how a independent local TSP (L-TSP) feature vector is constructed. In practice, we use different length of windows for samples with different sizes and result in equal and sufficient number of local feature vectors for each of them. And these vectors are used to train local classification model.

Intuitively, in order to obtain equal number of vectors for e-mails in different sizes, we defined the length of sliding

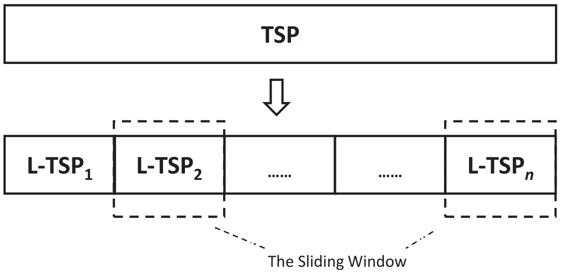


Fig. 4. Construction of local features with sliding window.

Algorithm 2 Local Feature Vectors Construction With TSP

```
define the number of local areas that an email sample is to be divide into as $n$
move a sliding window of $\frac{N_t}{n}$ terms over the given sample with a step of $\frac{N_t}{n}$ terms
for each position $i$ of the sliding window do
    construct TSP feature vector on the current local area
end for
```

window based on the number of terms of each e-mail. For example, if an e-mail with N_t terms and we need n independent vectors for each sample, the window's size is defined as (N_t/n). In addition, n is an important hyper-parameter that leverage the granularity of our experiments. Algorithm 2 described the details of how we get local feature vectors of each e-mails.

C. Ensemble Decision on TSP

As their names indicate, global and local feature vectors represent an e-mail in distinct and complementary aspects. One for holistic characteristics while the other focusing on local details. In consideration of the importance both of global and local features, we adopt ensemble decision approach when doing classification based on TSP feature vectors. The information brought by global classifier what is trained on global features and local models from local features are both being considered in this method. The details of a voting strategy utilized in ensemble decision based on global and local models are shown in Algorithm 3.

V. Experiments

A. Corpora

The e-mail datasets we used in this paper, including PU1, PU2, PU3, PUA [76], and Enron-Spam [77], all of them are standard benchmark corpora that commonly used in spam detection to measure the performance of classification models. The statistical information of these five corpus are shown in Table I, with EN and SN represent all e-mail number and spam e-mail number, respectively, in that corpora, AL indicates the average lengths of e-mails and NDT is the number of distinct terms of a specific data set. To guarantee the accuracy and objectivity of the experimental results, we implement tenfold and sixfold cross validation on the PU corpora and Enron-Spam, respectively, since they are being divided into different number of subsets.

Algorithm 3 Ensemble Decision in Real-World Scenario Using TSP-Based Feature Construction Approach

```
Input: Training set and new email which need to be classified.
Output: Detection model and label of the new email.
construct global feature vectors with TSP on the training sample set $FV_g$
construct local feature vectors with TSP on the training sample set $FV_l$
construct global classifier $classifier_g$ on $FV_g$
for each position $i$ of the sliding window do
    construct local classifier $classifier_l(i)$ on $FV_l(i)$
end for
construct global feature vector with TSP on the given sample
classify the given sample with $classifier_g$
for each position $j$ of the sliding window do
    construct local feature vector with TSP on the given sample
    classify the given sample with $classifier_l(j)$
end for
all the above classifiers vote to form the final decision
```

TABLE I
Statistical Information of Five Corpus

Corpora	EN	SN	AL	NDT
PU1	1099	481	776	24729
PU2	721	142	669	16340
PU3	4139	1826	624	69415
PUA	1142	572	697	24522
Enron-Spam	33716	17171	311	159833

TABLE II
Expressions of Evaluation Criteria

Criterion	Expression
Spam Recall	$R_s = \frac{n_{s,s}}{n_{s,s}+n_{s,h}}$
Spam Precision	$P_s = \frac{n_{s,s}}{n_{s,s}+n_{h,s}}$
Accuracy	$A = \frac{n_{s,s}+n_{h,h}}{n_s+n_h}$
F_β	$F_\beta = (1+\beta^2)\frac{R_s P_s}{\beta^2 P_s + R_s}$

B. Evaluation Criteria

As in almost all classification problem, we also use accuracy, precision, recall, and F_β measure [19] as performance evaluation criteria in spam detection. The definition for each of them is shown in Table II and the meaning of each components is as follows. Specially, we regard spam e-mails as positive samples here $n_{s,s}$ represents the number of spam e-mails classified into spam class as true positive and $n_{s,h}$ denotes false negative with the meaning of spam samples are distinguished as hams, $n_{h,h}$ and $n_{h,s}$ are in the similar definitions as true negative and false positive. In addition, n_h and n_s indicate the total number of samples of spam and ham, respectively. As for F_β measure, we usually set β equal to 1 since it can reflects the overall performance properly.

C. Experimental Setup

In the experiments, we choose SVM as basic classification model for the proposed ensemble method and two widely used tools named WEKA [78] and LIBSVM [79] were adopted for model implementation. The kernel type was radial basis function, which is default in WEKA. The corresponding γ and cost were set to 0.7 and 2.0 for all of our experiments since the values of these two parameters slightly affect the model performance based on the result of our hyper parameter selection process.

In addition, the approaches for comparison also adopted NB, C4.5, AdaBoost, RF, MLP, and SAE for classification. MLP and SAE were implemented by MATLAB with the help of DL toolbox [80], we constructed a neural network with six layer and 2000, 500, 250, 125, 10, and 1 nodes for each layer, respectively, for both of them. NB, C4.5, AdaBoost, and RF were implemented by using WEKA toolkit. The base learner for AdaBoost was C4.5, and 100 trees was used for RF.

D. Investigation of Parameters

Independent experiments were conducted on PU1, which is a relatively small corpus, to investigate the effect of core parameters on performance of the proposed method. Tenfold cross validation is also utilized. The selected group of parameter values on PU1 is applied on all the five benchmark corpora in the following performance comparison experiments.

As mentioned above, parameter n in the proposed ensemble decision based on TSP feature vectors (EDTSP) method determines not only the number of local classifiers to be trained but also the granularity of local areas. It further affect both the effectiveness and efficiency of the method. Besides n, p, and r in the process of TSP feature vector construction are also important parameters.

We tune the values of parameters p and r according to the experiments on PU1 in our previous work [30]. Since the feature selection metrics are adopted for vertical partition in the TSP process, we study the influence of parameter n under two metrics of feature selection in different style, i.e., unsupervised and supervised setting. DF and IG are selected as the representatives of these two types of metrics for selection, respectively. Fig. 5 shows the performance of the EDTSP with respect to DF under varied n, where p is set to 30 and r is set to 7 according to [30]. As expected, the EDTSP method always achieves better results with different n and the performance getting better with n increasing in the former interval. This could be intuitively explained: small n always extracts local area coarsely and results in less local eigenvector, so as to ignore some important information which is necessary for detection in local area; when n getting larger, the number of local feature vectors are enough, but each local area is divided in an improper way which leads to a worse representation of local information used for classification, some of them are redundant. As a tradeoff between coarseness and redundancy, we choose $n = 3$ in our experiments with the EDTSP since it obtains a relatively high precision and recall.

The performance of the EDTSP with respect to IG under varied n is shown in Fig. 6, where p is set to 30 and r is set to 3 according to [30]. As we can see, similar experimental results are achieved and $n = 3$ is considered as a suitable selection of parameter n under this certain condition as well.

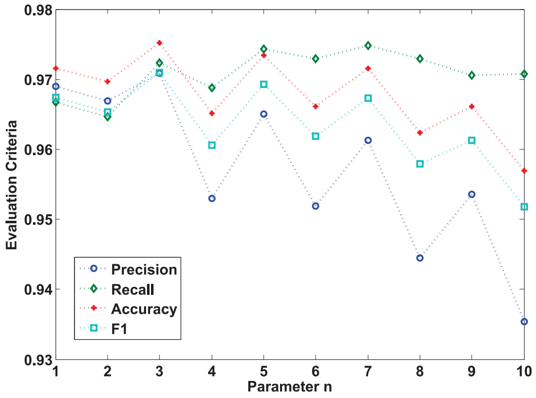


Fig. 5. Performance of the EDTSP with respect to DF under varied n.

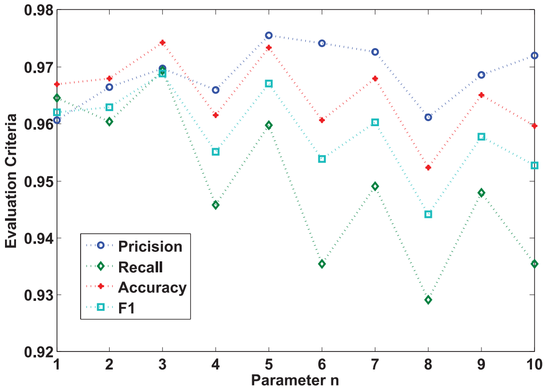


Fig. 6. Performance of the EDTSP with respect to IG under varied n.

E. Performance Improvement With Different Feature Selection Metrics

A appropriate metric of feature selection is important in the EDTSP for a better detection result, since feature selection metrics are adopted for vertical partition in the TSP process. Therefore, it is a necessary step to test whether the feature selection metrics are appropriate or not to work well with the EDTSP. We verify different kinds of metrics such as unsupervised and supervised ones, and investigate the performance improvement of the EDTSP compared with the original TSP (global TSP) and L-TSP under these two kinds of feature selection metrics.

As what we did above, we also use DF and IG to conduct the verification and investigation experiments to verify the effectiveness of unsupervised and supervised metrics, respectively. Performance comparisons of the ensemble method with global and local methods under DF and IG are shown in Tables III and IV, respectively. We can conclude that both the two style of selection metrics are appropriate for the EDTSP to achieve satisfactory performance. What need to be noted is that the EDTSP method with DF outperforms that with IG on majority of the selected benchmark corpora in experiments, other than the reverse in the past comparative study of text categorization [14]. This demonstrates that the EDTSP method itself could successfully construct term-class associations and effectively utilize this part of information into spam detection when unsupervised feature selection metrics are employed.

TABLE III
PERFORMANCE COMPARISON OF THE EDTSP AND TSP WITH RESPECT TO DF

Corpus	Method	Precision(%)	Recall(%)	Accuracy(%)	F_1(%)
PU1	G-TSP	96.90	96.67	97.16	96.74
	L-TSP	94.00	96.25	95.60	95.01
	EDTSP	97.09	97.23	**97.52**	**97.10**
PU2	G-TSP	94.09	83.57	95.63	88.12
	L-TSP	97.33	75.71	94.65	84.61
	EDTSP	97.86	84.16	**96.90**	**90.00**
PU3	G-TSP	95.69	95.88	96.20	95.73
	L-TSP	95.35	94.84	95.62	95.04
	EDTSP	96.83	96.45	**97.07**	**96.61**
PUA	G-TSP	95.91	96.49	96.05	96.11
	L-TSP	95.01	96.14	95.44	95.50
	EDTSP	96.51	96.96	**96.67**	**96.67**
Enron-Spam	G-TSP	94.29	98.21	97.02	96.14
	L-TSP	91.00	97.98	95.48	94.22
	EDTSP	94.74	98.88	**97.58**	**96.71**

TABLE IV
PERFORMANCE COMPARISON OF THE EDTSP AND TSP WITH RESPECT TO IG

Corpus	Method	Precision(%)	Recall(%)	Accuracy(%)	F_1(%)
PU1	G-TSP	96.07	96.46	96.70	96.21
	L-TSP	95.98	93.33	95.32	94.53
	EDTSP	96.97	96.93	**97.43**	**96.88**
PU2	G-TSP	96.32	80.00	95.35	87.09
	L-TSP	92.63	75.00	93.66	81.85
	EDTSP	96.06	81.07	**96.06**	**87.34**
PU3	G-TSP	96.37	97.09	97.05	96.69
	L-TSP	95.53	95.38	95.93	95.41
	EDTSP	97.23	97.41	**97.65**	**97.29**
PUA	G-TSP	95.62	94.74	95.00	95.06
	L-TSP	95.19	93.86	94.39	94.41
	EDTSP	96.77	95.15	**95.96**	**95.88**
Enron-Spam	G-TSP	94.18	98.23	96.90	96.12
	L-TSP	91.90	97.84	95.67	94.68
	EDTSP	94.89	98.83	**97.53**	**96.79**

As the experimental results indicate, though the original TSP approach could achieve outstanding performance, performance improvements of the EDTSP method are notable on all the selected benchmark corpora in experiments and with both the two kinds of feature selection metrics. The EDTSP method also outperforms L-TSP comprehensively and significantly. This reveals the necessity and importance of constructing local and global features at the same time in spam detection, and verifies the effectiveness of the ensemble mechanism of local features and global features established by the proposed EDTSP method from a perspective of decision. More importantly, the EDTSP method obtains not only better performance but also good robustness. As we can see, the EDTSP method achieves more balanced performance on different benchmark corpora compared with global and L-TSP, especially when cooperating with the unsupervised feature selection metrics. This further shows the superiority of considering global and local features together when filtering spam.

It is worth mentioning that the purpose of constructing local features is to avoid the dilution of local differences in a global perspective and capture the position correlated information, e.g., spam terms are more likely to appear at the beginning or the end of e-mails. The construction of local features could be a useful complement to the global features, and this also determines that the performance improvement of the ensemble method is limited on corpus with less samples having apparent local differences or position correlated information. What the ensemble method does is to guarantee performance improvement in a certain range and better robustness.

F. Comparison With Current Approaches

This paper applies five standard benchmark corpora what we described before to make performance comparison among the EDTSP approach and other commonly used spam detection technologies. Those other methods contain BoW with different machine learning methods, including DL, hot topic of machine learning recently. CFC and LC with SVM are also selected [24]. Experimental results of the compared approaches are partly published in our previous work [29].

Tables V–IX show the performance comparisons of the EDTSP with the current approaches on five corpora, respectively. As mentioned before, we only compare the accuracy and F_1 measure but ignore the value of precision and recall among all approaches since F_1 measure is a overall evaluation of precision and recall.

BoW, as talked before, could cooperate well with different machine learning methods. From the comprehensive experimental results, the EDTSP method performs the best in terms of both accuracy and F_1 measure in most of the cases when

TABLE V
PERFORMANCE COMPARISON OF THE EDTSP WITH CURRENT APPROACHES ON PU1

Approach	Precision(%)	Recall(%)	Accuracy(%)	F_1(%)
BoW-NB	97.74	82.50	91.47	89.37
BoW-C4.5	91.24	89.17	91.28	90.01
BoW-SVM	96.19	95.62	96.33	95.77
BoW-AdaBoost	97.08	95.62	96.79	96.28
BoW-RF	98.36	97.92	**98.35**	**98.11**
BoW-MLP	97.99	97.50	97.98	97.68
BoW-SAE	98.16	97.71	98.17	97.89
CFC-SVM	94.97	95.00	95.60	94.99
LC-FL-SVM	95.12	96.88	96.42	95.99
LC-VL-SVM	95.48	96.04	96.24	95.72
EDTSP-SVM	97.09	97.23	97.52	97.10

TABLE VI
PERFORMANCE COMPARISON OF THE EDTSP WITH CURRENT APPROACHES ON PU2

Approach	Precision(%)	Recall(%)	Accuracy(%)	F_1(%)
BoW-NB	82.67	70.71	90.70	75.34
BoW-C4.5	79.26	71.43	89.72	73.96
BoW-SVM	90.99	78.57	94.08	83.92
BoW-AdaBoost	91.74	75.71	93.66	82.23
BoW-RF	97.46	65.00	92.68	76.83
BoW-MLP	90.85	89.29	95.91	89.57
BoW-SAE	93.29	88.57	96.34	**90.37**
CFC-SVM	95.12	76.43	94.37	84.76
LC-FL-SVM	90.86	82.86	94.79	86.67
LC-VL-SVM	92.06	86.43	95.63	88.65
EDTSP-SVM	97.86	84.16	**96.90**	90.00

TABLE VII
PERFORMANCE COMPARISON OF THE EDTSP WITH CURRENT APPROACHES ON PU3

Approach	Precision(%)	Recall(%)	Accuracy(%)	F_1(%)
BoW-NB	92.75	78.63	87.72	84.93
BoW-C4.5	91.10	91.76	92.25	91.34
BoW-SVM	95.44	93.96	95.33	94.67
BoW-AdaBoost	95.54	94.56	95.62	95.02
BoW-RF	97.50	95.66	96.97	96.55
BoW-MLP	96.72	95.66	96.63	96.17
BoW-SAE	96.77	97.14	97.24	96.91
CFC-SVM	96.24	94.95	96.05	95.59
LC-FL-SVM	95.99	95.33	96.13	95.66
LC-VL-SVM	95.64	95.77	96.15	95.67
EDTSP-SVM	97.23	97.41	**97.65**	**97.29**

TABLE VIII
PERFORMANCE COMPARISON OF THE EDTSP WITH CURRENT APPROACHES ON PUA

Approach	Precision(%)	Recall(%)	Accuracy(%)	F_1(%)
BoW-NB	95.39	94.21	94.65	94.63
BoW-C4.5	87.02	92.63	88.68	89.30
BoW-SVM	91.84	94.39	92.63	92.87
BoW-AdaBoost	91.08	97.02	93.42	93.80
BoW-RF	92.15	97.02	94.04	94.36
BoW-MLP	94.24	97.02	95.35	95.49
BoW-SAE	94.49	97.90	95.88	96.04
CFC-SVM	96.03	93.86	94.82	94.93
LC-FL-SVM	96.01	94.74	95.26	95.37
LC-VL-SVM	95.60	94.56	94.91	94.94
EDTSP-SVM	96.51	96.96	**96.67**	**96.67**

compared with the BoW-based spam detection approaches (i.e., spam detection approaches using BoW for feature construction). It is worth noting that the EDTSP method achieves similar or worse performance on PU1 compared with BoW-RF, BoW-MLP, and BoW-SAE, for RF, MLP neural networks, and SAE (one of the main types of deep neural networks) are absolutely excellent machine learning techniques currently. However, the training process of the above three approaches are really time consuming, and the proposed EDTSP method could achieve much higher efficiency compared with them.

CFC and LC, two kinds of our previous work, had achieved satisfactory results on spam detection both in accuracy and

TABLE IX
PERFORMANCE COMPARISON OF THE EDTSP WITH CURRENT APPROACHES ON ENRON-SPAM

Approach	Precision(%)	Recall(%)	Accuracy(%)	F_1(%)
BoW-NB	77.74	98.41	87.45	86.10
BoW-C4.5	82.88	97.07	90.33	89.02
BoW-SVM	89.64	98.74	94.63	93.86
BoW-AdaBoost	89.13	98.78	94.25	93.57
BoW-RF	91.46	99.28	96.06	95.11
BoW-MLP	92.70	98.71	96.23	95.54
BoW-SAE	94.90	98.95	97.49	**96.84**
CFC-SVM	91.48	97.81	95.62	94.39
LC-FL-SVM	94.07	98.00	96.79	95.94
LC-VL-SVM	92.44	97.81	96.02	94.94
EDTSP-SVM	94.74	98.88	**97.58**	96.71

TABLE X
EFFICIENCY COMPARISON OF THE EDTSP WITH CURRENT APPROACHES

Approach	BoW-NB	BoW-SVM	BoW-AdaBoost	BoW-RF	BoW-SAE	CFC-SVM	LC-VL-SVM	TSP-SVM	EDTSP-SVM
Seconds/email	$2.61e^{-3}$	$2.53e^{-3}$	$5.01e^{-3}$	$1.09e^{-2}$	$> 1.50e^{-2}$	$2.60e^{-4}$	$3.13e^{-4}$	$2.72e^{-4}$	$4.30e^{-4}$

F_1 measure with high efficiency. As the experimental results show, the EDTSP method far outperforms the CFC and LC approaches in both accuracy and F_1 measure on all the selected benchmark corpora.

Meanwhile, the EDTSP approach obtains not only much better performance but also more balanced performance compared with all the selected current approaches in the experiments. This reflects better robustness and further endows it with flexibility in real-world applications. Another wonderful character of the EDTSP is its practicability with higher and stabler precision when filtering spam since for e-mail users, it is better to receive a spam than to discard a normal e-mail.

In addition, since the EDTSP method takes TSP-based feature construction approach and SVM as basic elements and components, its computational complexity is in a linear relationship with that of the original TSP approach. Table X reveals the comparison of time spent among all approaches for processing one incoming e-mail on PU1, including the time for feature construction and classification. Since C4.5 is usually used as the base learner of boosting method and hardly give good performance, it is not included. MLP has the same time for processing one incoming e-mail with SAE due to the same network structure. As we can see, the EDTSP method could not only achieve much better performance in spam detection, but also inherit the high efficiency of the original TSP approach. This is mainly because the significant reduction on feature vector dimension.

VI. CONCLUSION

In this paper, a spam detection approach based on ensemble decision using TSP to construct feature vectors was proposed. It combines global and local classification models which are trained with corresponding features. We have carried out comprehensive experiments, and the results have shown that the EDTSP method has much better performance and advantages in some aspects.

1) Utilization of sliding window could not only extract local features from e-mail samples but also obtain position correlated information for the specific application of spam detection.
2) The EDTSP method successfully establishes an ensemble mechanism of global and local features from a decision perspective.
3) The EDTSP method could cooperate well with many metrics for feature selection in various types, which makes it adaptive to be utilized in real-world application.
4) The longitudinal comparison shows that the EDTSP method brings significant performance improvement compared to the original TSP approach, as well as better robustness, by considering global and local features together for classification.
5) The horizontal comparisons show that the EDTSP method far outperforms all the selected current approaches, especially when a comprehensive consideration of performance, efficiency, and robustness is taken.

REFERENCES

[1] L. F. Cranor and B. A. LaMacchia, "Spam!" *Commun. ACM*, vol. 41, no. 8, pp. 74–83, 1998.

[2] "Spam, spammers, and spam control: A white paper by Ferris research," Ferris Research, San Francisco, CA, USA, Rep., 2009.

[3] "Internet security threat report: 2015," Symantec Corporat., Mountain View, CA, USA, Rep., 2015.

[4] "Internet threats trend report: October 2014," Cyren, Jerusalem, Israel, Rep., 2014.

[5] Z. Duan, Y. Dong, and K. Gopalan, "DMTP: Controlling spam through message delivery differentiation," *Comput. Netw.*, vol. 51, no. 10, pp. 2616–2630, 2007.

[6] G. Cormack, "Email spam filtering: A systematic review," *Found. Trends Inf. Retrieval*, vol. 1, no. 4, pp. 335–455, 2007.

[7] S. Hershkop, "Behavior-based email analysis with application to spam detection," Ph.D. dissertation, Graduate School Arts Sci., Columbia Univ., New York, NY, USA, 2006.

[8] E. Sanz, J. G. Hidalgo, and J. C. Perez, "Email spam filtering," *Adv. Comput.*, vol. 74, pp. 45–114, Jan. 2008.

[9] E. Moustakas, C. Ranganathan, and P. Duquenoy, "Combating spam through legislation: A comparative analysis of U.S. and European approaches," in *Proc. 2nd Conf. Email Anti Spam*, 2005, pp. 1–8.

[10] J. Carpinter and R. Hunt, "Tightening the net: A review of current and next generation spam filtering tools," *Comput. Security*, vol. 25, no. 8, pp. 566–578, 2006.

[11] F. Benevenuto *et al.*, "Practical detection of spammers and content promoters in online video sharing systems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 688–701, Jun. 2012.

[12] F. Zhang, P. Chan, B. Biggio, D. S. Yeung, and F. Roli, "Adversarial feature selection against evasion attacks," *IEEE Trans. Cybern.*, no. 46, no. 3, pp. 766–777, Mar. 2016.

[13] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[14] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proc. 14th Int. Conf. Mach. Learn. (ICML)*, Nashville, TN, USA, Jul. 1997 pp. 412–420.

[15] Y. Yang, "Noise reduction in a statistical approach to text categorization," in *Proc. 18th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1995, pp. 256–263.

[16] I. Koprinska, J. Poon, J. Clark, and J. Chan, "Learning to classify e-mail," *Inf. Sci.*, vol. 177, no. 10, pp. 2167–2187, 2007.

[17] W. Shaw, "Term-relevance computations and perfect retrieval performance," *Inf. Process. Manag.*, vol. 31, no. 4, pp. 491–498, 1995.

[18] S. Li and D. Wei, "Extremely high-dimensional feature selection via feature generating samplings," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 737–747, Jun. 2014.

[19] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to spam filtering," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10206–10222, 2009.

[20] C. Siefkes, F. Assis, S. Chhabra, and W. S. Yerazunis, "Combining winnow and orthogonal sparse bigrams for incremental spam filtering," in *Proc. Knowl. Disc. Databases (PKDD)*, 2004, pp. 410–421.

[21] Y. Tan, C. Deng, and G. Ruan, "Concentration based feature construction approach for spam detection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2009, pp. 3088–3093.

[22] G. Ruan and Y. Tan, "A three-layer back-propagation neural network for spam detection using artificial immune concentration," *Soft Comput. A Fusion Found. Methodol. Appl.*, vol. 14, no. 2, pp. 139–150, 2010.

[23] Y. Zhu and Y. Tan, "Extracting discriminative information from e-mail for spam detection inspired by immune system," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2010, pp. 1–7.

[24] Y. Zhu and Y. Tan, "A local-concentration-based feature extraction approach for spam filtering," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 2, pp. 486–497, Jun. 2011.

[25] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian approach to filtering junk e-mail," in *Proc. Learn. Text Categorization Papers Workshop*, vol. 62, 1998, pp. 98–105.

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

[27] X. Carreras and L. Màrquez, "Boosting trees for anti-spam email filtering," *CoRR*, vol. cs.CL/0109015, 2001. [Online]. Available: http://arxiv.org/abs/cs.CL/0109015

[28] J. Clark, I. Koprinska, and J. Poon, "Linger-a smart personal assistant for e-mail classification," in *Proc. 13th Int. Conf. Artif. Neural Netw. (ICANN)*, Istanbul, Turkey, Jun. 2003, pp. 26–29.

[29] G. Mi, Y. Gao, and Y. Tan, "Apply stacked auto-encoder to spam detection," in *Advances in Swarm and Computational Intelligence*. Beijing, China: Springer, 2015, pp. 3–15.

[30] G. Mi, P. Zhang, and Y. Tan, "Feature construction approach for email categorization based on term space partition," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2013, pp. 1–8.

[31] P. Zhang, W. Wang, and Y. Tan, "A malware detection model based on a negative selection algorithm with penalty factor," *Sci. China*, vol. 53, no. 12, pp. 2461–2471, 2010.

[32] W. Luo, X. Wang, Y. Tan, and X. Wang, "A novel negative selection algorithm with an array of partial matching lengths for each detector," in *Proc. 9th Int. Conf. Parallel Problem Solving Nat. (PPSN IX)*, Sep. 2006, pp. 112–121. [Online]. Available: https://doi.org/10.1007/11844297_12

[33] S. Zheng and Y. Tan, "A unified distance measure scheme for orientation coding in identification," in *Proc. IEEE 3rd Int. Conf. Inf. Sci. Technol.*, 2013, pp. 979–985.

[34] K. Schneider, "A comparison of event models for naïve Bayes anti-spam e-mail filtering," in *Proc. 10th Conf. Eur. Ch. Assoc. Comput. Linguist.*, vol. 1, 2003, pp. 307–314.

[35] R. Chao and Y. Tan, "A virus detection system based on artificial immune system," in *Proc. Int. Conf. Comput. Intell. Security*, 2010, pp. 6–10.

[36] W. Wang, P.-T. Zhang, Y. Tan, and X.-G. He, "Animmune local concentration based virus detection approach," *J. Zhejiang Universityence C*, vol. 12, no. 6, pp. 443–454, 2011.

[37] K. Wang and Y. Tan, "A new collaborative filtering recommendation approach based on naïve Bayesian method," in *Proc. Adv. Swarm Intell. 2nd Int. Conf. (ICSI)*, Chongqing, China, Jun. 2011, pp. 218–227. [Online]. Available: https://doi.org/10.1007/978-3-642-21524-7_26

[38] A. McCallum and K. Nigam, "A comparison of event models for naïve Bayes text classification," in *Proc. AAAI Workshop Learn. Text Categorization*, vol. 752, 1998, pp. 41–48.

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

[40] Z. Zhong and K. Li, "Speed up statistical spam filter by approximation," *IEEE Trans. Comput.*, vol. 60, no. 1, pp. 120–134, Jan. 2011.

[41] S. K. Trivedi and S. Dey, "Interaction between feature subset selection techniques and machine learning classifiers for detecting unsolicited emails," *ACM SIGAPP Appl. Comput. Rev.*, vol. 14, no. 1, pp. 53–61, 2014.

[42] S. Gu, Y. Tan, and X. He, "Discriminant analysis via support vectors," *Neurocomputing*, vol. 73, nos. 10–12, pp. 1669–1675, 2010.

[43] Y. Tan, Y. Xia, and J. Wang, "Neural network realization of support vector methods for pattern classification," in *Proc. Neural Comput. New Challenges Perspectives New Millennium IEEE-INNS-ENNS Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 6, 2000, pp. 411–416.

[44] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 465–479, Mar. 2012.

[45] C. Li and M. Liu, "An ontology enhanced parallel SVM for scalable spam filter training," *Neurocomputing*, vol. 108, no. 5, pp. 45–57, 2013.

[46] J. Wu, S. Pan, X. Zhu, and Z. Cai, "Boosting for multi-graph classification," *IEEE Trans. Cybern.*, vol. 45, no. 3, pp. 416–429, Mar. 2015.

[47] L. Rokach, "Ensemble-based classifiers," *Artif. Intell. Rev.*, vol. 33, nos. 1–2, pp. 1–39, 2010.

[48] D. DeBarr and H. Wechsler, "Spam detection using random boost," *Pattern Recognit. Lett.*, vol. 33, no. 10, pp. 1237–1244, 2012.

[49] R. Amin, J. Ryan, and J. R. van Dorp, "Detecting targeted malicious email," *IEEE Security Privacy*, vol. 10, no. 3, pp. 64–71, May/Jun. 2012.

[50] M. Chakraborty, S. Pal, R. Pramanik, and C. R. Chowdary, "Recent developments in social spam detection and combating techniques: A survey," *Inf. Process. Manag.*, vol. 52, no. 6, pp. 1053–1073, 2016.

[51] H. Fu, X. Xie, and Y. Rui, "Leveraging careful microblog users for spammer detection," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 419–429.

[52] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: Social honeypots + machine learning," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2010, pp. 435–442.

[53] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proc. Comput. Security Appl. Conf.*, 2010, pp. 1–9.

[54] S. Liu, J. Zhang, and Y. Xiang, "Statistical detection of online drifting Twitter spam: Invited paper," in *Proc. ACM Asia Conf. Comput. Commun. Security*, 2016, pp. 1–10.

[55] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," in *Proc. 1st Int. Conf. Adv. Swarm Intell. (ICSI)*, Beijing, China, Jun. 2010, pp. 355–364. [Online]. Available: https://doi.org/10.1007/978-3-642-13495-1_44

[56] W. Hu and Y. Tan, "Prototype generation using multiobjective particle swarm optimization for nearest neighbor classification," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2719–2731, Dec. 2016. [Online]. Available: https://doi.org/10.1109/TCYB.2015.2487318

[57] Y. Zhou and Y. Tan, "GPU-based parallel particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Trondheim, Norway, May 2009, pp. 1493–1500. [Online]. Available: https://doi.org/10.1109/CEC.2009.4983119

[58] 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, 2009.

[59] C. H. Li and J. X. Huang, "Spam filtering using semantic similarity approach and adaptive BPNN," *Neurocomputing*, vol. 92, pp. 88–97, Sep. 2012.

[60] A. Barushka and P. Hájek, "Spam filtering using regularized neural networks with rectified linear units," in *Proc. AI*IA Adv. Artif. Intell. 15th Int. Conf. Italian Assoc. Artif. Intell.*, Genoa, Italy, Nov./Dec. 2016, pp. 65–75.

[61] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Mar. 2010.

[62] Y. Ren and Y. Zhang, "Deceptive opinion spam detection using neural network," in *Proc. COLING 26th Int. Conf. Comput. Linguist. Conf. Tech. Papers*, Dec. 2016, Osaka, Japan, 2016, pp. 140–150. [Online]. Available: http://aclweb.org/anthology/C/C16/C16-1014.pdf

[63] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, Sep. 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[64] S. Zhao, Z. Xu, L. Liu, and M. Guo, "Towards accurate deceptive opinion spam detection based on word order-preserving CNN," *CoRR*, vol. abs/1711.09181, 2017. [Online]. Available: http://arxiv.org/abs/1711.09181

[65] G. Jain, M. Sharma, and B. Agarwal, "Spam detection on social media using semantic convolutional neural network," *Int. J. Knowl. Disc. Bioinformat.*, vol. 8, no. 1, pp. 12–26, 2018. [Online]. Available: https://doi.org/10.4018/IJKDB.2018010102

[66] J. Ma *et al.*, "Detecting rumors from microblogs with recurrent neural networks," in *Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI)*, New York, NY, USA, Jul. 2016, pp. 3818–3824. [Online]. Available: http://www.ijcai.org/Abstract/16/537

[67] T. Wu, S. Liu, J. Zhang, and Y. Xiang, "Twitter spam detection based on deep learning," in *Proc. Aust. Comput. Sci. Week Multiconf. (ACSW)*, Geelong, VIC, Australia, Jan./Feb. 2017, pp. 1–3. [Online]. Available: http://doi.acm.org/10.1145/3014812.3014815

[68] A. Singh and S. Batra, "Ensemble based spam detection in social IoT using probabilistic data structures," *Future Gener. Comput. Syst.*, vol. 81, pp. 359–371, Apr. 2018. [Online]. Available: https://doi.org/10.1016/j.future.2017.09.072

[69] S. Sedhai and A. Sun, "Semi-supervised spam detection in Twitter stream," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 1, pp. 169–175, Mar. 2018. [Online]. Available: https://doi.org/10.1109/TCSS.2017.2773581

[70] W. Xu, H. Sun, C. Deng, and Y. Tan, "Variational autoencoder for semi-supervised text classification," in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, Feb. 2017, pp. 3358–3364. [Online]. Available: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14299

[71] W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on GAN," *CoRR*, vol. abs/1702.05983, Feb. 2017. [Online]. Available: http://arxiv.org/abs/1702.05983

[72] P. S. Penev and J. J. Atick, "Local feature analysis: A general statistical theory for object representation," *Netw. Comput. Neural Syst.*, vol. 7, no. 3, pp. 477–500, 1996.

[73] N. Wang and X. Ding, "Face recognition based on global and local information," *Comput. Eng.*, vol. 30, no. 5, pp. 154–155, 2004.

[74] C. Kim, J. Oh, and C. Choi, "Combined subspace method using global and local features for face recognition," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 4, 2005, pp. 2030–2035.

[75] D. Lin and X. Tang, "Recognize high resolution faces: From macrocosm to microcosm," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 2006, pp. 1355–1362.

[76] I. Androutsopoulos, G. Paliouras, and E. Michelakis, *Learning to Filter Unsolicited Commercial E-mail*. Athens, Greece: Nat. Center Sci. Res., 2004.

[77] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naïve Bayes—Which naïve Bayes?" in *Proc. 3rd Conf. Email Anti Spam (CEAS)*, Mountain View, CA, USA, Jul. 2006. [Online]. Available: https://dblp.org/rec/bib/conf/ceas/MetsisAP06

[78] M. Hall *et al.*, "The WEKA data mining software: An update," *ACM SIGKDD Explorations Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.

[79] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.

[80] R. Palm, *Prediction As a Candidate for Learning Deep Hierarchical Models of Data*, vol. 25, Tech. Univ. Denmark, Lyngby, Denmark, 2012.

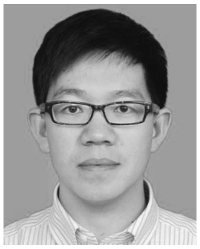
Ying Tan (SM'02) received the B.Eng., M.S., and Ph.D. degrees from Southeast University, Nanjing, China, in 1985, 1988, and 1997, respectively.

He is a Full Professor and a Ph.D. Advisor with the School of Electronics Engineering and Computer Science, and the Director of Computational Intelligence Laboratory, Peking University, Beijing, China. He was a Professor of the Faculty of Design, Kyushu University, Fukuoka, Japan, in 2018, Columbia University, New York, NY, USA, as a Senior Research Fellow in 2017, and the Chinese University of Hong Kong, Hong Kong, in 1999 and from 2004 to 2005 as a Research Fellow. His current research interests include machine learning, computational intelligence, swarm intelligence, data mining, and intelligent information processing for information security. He has published over 300 papers in refereed journals and conferences in the above areas, and authored/co-authored 13 books and 25 chapters in book, and received 4 invention patents.

Dr. Tan was a recipient of the Second-Class Natural Science Award of China in 2009. He is the Inventor of Fireworks Algorithm. He serves as the Editor-in-Chief of the *International Journal of Computational Intelligence* and *Pattern Recognition*, and an Associate Editor for the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON CYBERNETICS, and the *IEEE Transactions on Neural Networks and Learning Systems*. He also served as an Editor of Springers Lecture Notes on Computer Science for over 32 volumes, and a Guest Editor of several referred journals, including the IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, *Information Science*, *Neurocomputing*, *Natural Computing*, and *Softcomputing*. He is the Founder and the Chair of the ICSI International Conference series. He was the General Chair of joint general chair of first and second BRICS Congress of Computational Intelligence and the Program Committee Co-Chair of IEEE WCCI 2014.

Quanbin Wang received the Bachelor of Engineering degree in intelligent science and technology from Xidian University, Xi'an, China, in 2015. He is currently pursuing the Ph.D. degree in computer science 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, China.

His current research interests include machine learning, deep neural networks, and natural language process.



Guyue Mi received the Bachelor of Engineering degree in software engineering from Tianjin University, Tianjin, China, in 2011 and the Ph.D. degree in computer science from Peking University, Beijing, China, in 2015, under the guidance of Prof. Y. Tan, where he is currently pursuing the Ph.D. degree with the Key Laboratory of Machine Perception (Ministry of Education) and Department of Machine Intelligence, School of Electronics Engineering and Computer Science.

He is currently with Pingnan County as the Deputy County Head of technology. His current research interests include machine learning, swarm intelligence, AIS, information processing, and pattern recognition.