Incremental Information Gain Analysis of Input Attribute Impact on RBF-Kernel SVM Spam Detection

Hongmei He*, Ashutosh Tiwari*, Jörn Mehnen*, Tim Watson †, Carsten Maple †, Yaochu Jin ‡, Bogdan Gabrys§
*School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield, MK43 0AL, UK
Email: h.he@cranifeld.ac.uk
†Cyber Security Centre - WMG, University of Warwick, UK
‡Department of Computer Science, University of Surrey, UK
§School of Design, Engineering & Computing, Bournemouth University, UK

***Abstract*—The massive increase of spam is posing a very serious threat to email and SMS, which have become an important means of communication. Not only do spams annoy users, but they also become a security threat. Machine learning techniques have been widely used for spam detection. Email spams can be detected through detecting senders' behaviour, the contents of an email, subject and source address, etc, while SMS spam detection usually is based on the tokens or features of messages due to short content. However, a comprehensive analysis of email/SMS content may provide cures for users to aware of email/SMS spams. We cannot completely depend on automatic tools to identify all spams. In this paper, we propose an analysis approach based on information entropy and incremental learning to see how various features affect the performance of an RBF-based SVM spam detector, so that to increase our awareness of a spam by sensing the features of a spam. The experiments were carried out on the spambase and SMSSpemCollection databases in UCI machine learning repository. The results show that some features have significant impacts on spam detection, of which users should be aware, and there exists a feature space that achieves Pareto efficiency in True Positive Rate and True Negative Rate.**

I. Introduction

Spam is an ever-increasing problem. It pervades any information system through e-mail or web, social, blog or reviews platform [20], and is increasingly being used to distribute virus, spyware, links to phishing web sites, etc. Email spam detection has been an important part of correspondence since email became an essential part of our daily lives. The growth of mobile phone users has led to a dramatic increasing of SMS spam messages [1]. The following facts further tell that why spam detection is critical:

(1) Spammers use various methods to get user's email address, so that they can flood user's inboxes;

(2) Spammers attempt to acquire sensitive information through phishing, such as bank account information, credit card numbers or other confidential information, and they are becoming more sophisticated and are constantly managing to outsmart 'static' methods of fighting spam.

(3) It is reported that a malicious actor had infiltrated a German steel facility in 2014. The adversary used a spear phishing email to gain access to the corporate network and then moved into the plant network. The adversary showed knowledge in ICS and was able to cause multiple components of the system to fail. This specifically impacted critical process components to become unregulated, which resulted in massive physical damage [9].

(4) Even now it becomes more critical. Recently news in Nov. 2015 shows that Cybercriminals spoof law enforcement officials in Dubai, Bahrain, Turkey, and Canada to send terror-alert spear-phishing emails containing back door. [12].

Therefore, the problem of spam is not only an annoyance, but has also become a security threat. It has attracted much attention of researchers for decades. Usually, page having high PageRank is more likely to be a spam if it has no relationship with a set of trusted pages. PageRank is an algorithm used by Google Search to rank websites in their search engine results. Page et al [11] proposed the PageRank algorithm to estimate the global importance (authority or reputation) score of a webpage on the web. By the end of 2001, the Google search engine introduced a new kind of penalty for websites that use questionable search engine optimisation tactics: A PageRank of 0 (called PR0), which is assessed with a measure of BadRank as the opposite of PageRank [18]. The well-known link-based detection algorithm TrustRank [5] uses a small set of trustworthy pages that are carefully select by human experts, and random walk with a restart to the seed set is executed for a small fixed set of iterations. Krishnan and Rashmi Raj [8] improved the TrustRank algorithm with "Anti-Trust Rank". Another well-known link-based spam detector is SpamRank [3], in which, the SpamRank is defined by penalising pages that originate a suspicious PageRank share and personalising PageRank on the penalties.

It was shown that machine learning is superior to the PageRank algorithm for static page ranking [13]. Yamakami and Almeida [17] compared most classical machine learning methods, such as Artificial Neural Networks (ANN), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), K-Nearest Neighbors (KNN), Adaptive Boosting, Bagging and LogitBoost for web spams. Recently, Wang et al. [24] investigated social spam detection on Twitter with Bernoulli Naive Bayes (NB), KNN, SVM, DT and RF, and the RF

978-1-5090-0623-6/16/$31.00 ©2016 IEEE

algorithm obtained the best F_1-measure up to 0.946 on the social Honeypot dataset.

Kolari et al [7] used SVM-based approach for blog spams. The online classified advertisement domain is a target for spammers. Tran et al. [21] proposed a domain-feature based approach to detecting advertisement spam. Another domain of spams is customer review. Spammers may create false reviews (e.g. making fake, untruthful, or deceptive reviews) to artificially promote or devalue products and services for profit or gain [4]. Such review spams are very difficult to be detected. Jindal and Liu [6] investigated review spam by detecting duplicated review and classifying review with machine learning technology. Shirani-Mehr [16] investigated SMS Spams with NB algorithm. Sharifi et al. [15] used Logistic Regression approach to detecting Internet scam, and the precision and recall are 98% on the set of data with 43 characteristic statistics of the 837 websites from 11 online sources. Verma and Dhawan [23] investigated spam detection in social networks with clustered KNN technique.

Web spam can be categorised to: content spam, link spam (outgoing link spam, incoming link spam), cloaking & redirection, and click spam. Content spam is probably the first and most widespread form of web spam [20]. Email spam could also have content spam and click spam. While web spams usually target public users or some specific user groups, email spams may have clearer targets for specific purposes. Much research on email spam detection is based on content spam. Like using in web spam detection, machine learning technologies is widely applied for email spam detection. The family of NB classifiers [14], [19] is one of the most commonly implemented, which is also embedded in many popular email clients. Tretyakov [22] used the combination of the most classic machine learning methods (Bayesian classification, KNN, ANNs, SVMs) for the problem of email spam-filtering. The combination of Baysian classification and SVM obtained the precision of 94.4%.

Feature extraction is also important in spam detection. Wang et al. [24] used four different types of feature sets (user features, content features, Uni-Bi features, and sentiment features) and their combinations to validate Random Forests for Twitter spam detection, and the experimental results show feature combination outperformed a single type of features on decision making. The study of Alqatawna et al. showed that adding malicious related features to training data significantly improved the detection of spam emails [2].

It is believed that false negative is less important than false positive, as it is unacceptable, if an important email gets lost. Therefore, true positive always gives way to true negative. This requires users have high awareness of email spams, even if client systems have embedded automatic spam detection. The purpose of the research is to study the feature impact on spam detection, rather than to develop a new spam detector. In this paper, we use RBF-kernel SVM, a well-known binary classifier, for spam detection, and based on information theory, analyse what kinds of factors have significant impact on the performance of the SVM spam detection in accuracy, true positive and true negative. Hence, the anaysis results could provide some clues to email/SMS users, thus to increase the sensitivity of users to email/SMS spams.

II. Methodology

Here, we discuss two problems: spam detection and feature analysis.

A. RBF-kernel SVM for spam detection

Spam detection can be a function mapping between input features (attributes) and the decision variable (spam, or non-spam). Namely $y = f(x)$, where y is decision variable with the two states of spam and non-spam, and x is the features, retrieving from emails/SMS. Identifying spams from massive emails/SMSs is not a linear separable problem. RBF-kernel SVM, equivalent to a specific three-layer forward neural network, is powerful for non-linear binary classification problems, and it is easy to repeat the assessment with the RBF-kernel SVM in MatLab. Therefore, a RBF-kernel SVM is employed for spam detection. The primary principle of RBF-kernel SVM is to transfer the problem space into higher dimension space, so that the data becomes linear separable, then we can use linear SVM to solve the problem in higher dimension space. Therefore, a RBF-kernel SVM is effective in high dimensional spaces, even if the number of dimensions is greater than the number of samples. Assume $\Phi(x)$ is a feature map, where x is mapped to, the kernel function, $k(x_j, x_i) = \Phi(x_j)^T \Phi(x_i)$, and the data becomes separable. The kernel-based SVM can be expressed as Eq. (1):

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i k(x_j, x_i) + b. \tag{1}$$

Correspondingly, learning to maximise:

$$\sum \alpha_i - \frac{1}{2} \sum_{jk} \alpha_j \alpha_k y_j y_k k(x_j, x_i)$$
$$subject\ to\ \alpha_i \geq 0, \forall i\ and\ \sum_i \alpha_i y_i = 0. \tag{2}$$

The (Gaussian) Radio-based function (RBF) kernel (Eq. (3)) is commonly used as the kernel of a SVM.

$$k(x, \tilde{x}) = e^{\frac{-||x-\tilde{x}||^2}{2\sigma^2}}. \tag{3}$$

The RBF-kernel SVM:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i\ e^{\frac{-||x-\tilde{x}||^2}{2\sigma^2}} + b. \tag{4}$$

B. Information entropy based feature analysis

Usually, the main goal of spam detection is to improve the detection accuracy. But the reason mentioned before, the loss of important emails will not acceptable. Hence, the true negative seems more important than true positive. Moreover, as such an automatic spam detection is embedded in the email/SMS system, the level of computing complexity should be as low as possible, so that it will not affect the real-time performance of the system. Information theory provides a good approach to quantifying the pureness of information. A key measure in information theory is 'entropy'. Entropy quantifies the uncertainty involved in predicting the value of a random variable. For a random variable with two outcomes, Information Entropy is the binary entropy function, usually

taken to the logarithmic base 2, thus having the Shannon as unit:

$$E(y) = -p(+)\log_2 p(+) - p(-)\log_2(p(-)). \quad (5)$$

where, $p(+)$ is the probability of some samples $y \in \{+\}$, and $p(-)$ is the probability of $y \in \{-\}$. Therefore, first we use 'entropy' to quantify the uncertainty of each feature involved decision making. When an attribute x is involved in decision making, the conditional entropy is:

$$\begin{aligned} E(Y|X) &= \sum_{x\in X} p(x)E(Y|X=x) \\ &= -\sum_{x\in X} p(x)\sum_{y\in Y} p(y|x)\log_2 p(y|x) \\ &= -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log_2 p(y|x) \end{aligned} \quad (6)$$

Note: $\lim_{x\to 0} x\log_2(x) = 0$.

Correspondingly, the information gain (IG) is:

$$IG(x) = E(y) - E(y|x). \quad (7)$$

The information gain that an attribute contributes to decision making is different to that another attribute does. Therefore, we can sort the attributes in terms of their contribution for decision making.

C. Increasingly search input space of SVM

Assume $I = a_1, ..., a_n$ is the sorted attribute order in terms of information gains. In order to further observe the contribution of an attribute for decision making, the RBF-kernel SVM as a decision maker is used with increasing input space. Namely, the input space gradually increases from 1 to n dimensions, where n is the total number of attributes. The RBF-kernel SVM detection was validated with ten-folder crossing validation for the increasing input feature space. When an attribute is added to the input space of RBF-kernel SVM, if the average accuracy of spam detection for the ten-folder crossing validation is worse than previous average accuracy, the attribute will be removed from the input space. The input space search procedure is described in Algorithm 1.

Algorithm 1 IncreasingSVM($\mathscr{D}$, I, $nFolders$)

```
Ntst = floor(Size(D)/nFolders);
Ntrn = size(D)-Ntst;
P= zeros(N);
Space = [];
for (i = 1 to N) do
    Space = addDim(Space, I(i));
    X = extractData(D,Space);
    for (k = 1 to nFolders) do
        Xtst = randomTestData(X,Ntst);
        Xtrn = X - Xtst;
        svmStruct = trainSVM(Xtrn);
        Y = svmClassifier(Xtst);
        A(k) = assessment(Y, Ttst);
    end for
    A_av = means (A);
    if (i¿1 and (A_av < P(i-1)) then
        Space = RemoveAttr(Space,i);
    end if
    P(i) = A_av;
end for
```

III. Features

A. Email spams

The email spam database was created by Hopkins et. al in 1999, and published on UCI machine learning repository [10]. It provides a set of unsolicited commercial e-mails for spam detection. There are 4601 instances, of which, there are 1813 spam, accounting for 39.4% in total instances. Frequency of some key words of an email could indicate whether the email is a spam or not. The features of the email set have been extracted, and represented by 57 continuous attributes. The set of emails have been labeled with spam (1) and non-spam (0), represented by the target variable.

Most of the attributes indicate whether a particular word or a character was frequently occurring in the e-mail. The 48 continuous real [0,100] attributes are the frequency of a WORD, which is measured with the percentage of words in the e-mail that match the WORD, i.e.

$$x = 100 \times \frac{number\ of\ WORD\ occurences}{total\ number\ of\ WORDs\ in\ email}. \quad (8)$$

A WORD in this case is any string of alphanumeric characters bounded by non-alphanumeric characters or end-of-string. The 6 continuous real [0,100] attributes are the frequency of a CHAR, which is the percentage of characters in the e-mail that match the CHAR, i.e.

$$x = 100 \times \frac{number\ of\ CHAR\ occurences}{total\ number\ of\ characters\ in\ email} \quad (9)$$

The attributes $(x_{55} - x_{57})$ measure the length of sequences of consecutive capital letters. x_{55}, a continuous real [1,...] attribute, is the average length of uninterrupted sequences of capital letters. x_{56}, a continuous integer [1,...] attribute, is the length of longest uninterrupted sequence of capital letters. x_{57}, a continuous integer [1,...] attribute, is the sum of length of uninterrupted sequences of capital letters, i.e. the total number of capital letters in the email.

B. SMS spams

Unlike an email, SMS messages are fairly short, content-based spam filters may have their performance degraded [1]. This requires a careful analysis to spam messages. Most of spam messages intend to induce mobile users to make some actions through some benefits or rewards. Therefore, some key words frequently occur in spam messages. The SMSSpamCollection database, created by Almeida et al. [1], published on UCI machine learning [10], provides 5574 raw messages, of which 747 spams, accounting for 13.4% of total messages. As SMS messages are short, the features extracted from the SMS message data are binary value. If a KEY-WORD or a kind of behaviour exists in an SMS message, the corresponding feature is set to one, otherwise, it is set to zero, and key-words are not case-sensitive. For example, key-word, "free" indicates if any words, in which 'free' is partial phase (e.g. Free, FREE,

Freedom, free), then x_1 will be set to 1, otherwise, 0. x_{17} indicates all spams of noisy advertisements, which claim cheap service price, for instances, 1p/MIN, 1.5p/MIN, ... 2.5p/call, etc; x_{19} intends to find SMS messages, which include money values with different currency units. Totally 20 features are extracted in Table I. Although we do not expect that some words (e.g. "Freedom") are included in the searched targets, the extraction of each feature is completed in one condition function in excel, $IF(ISERR(SEARCH("free",s_1,1)),0,1)$.

TABLE I. FEATURE DEFINITION FOR THE SMS MESSAGE DATA

x	key word	x	key-word
1	free	11	send
2	urgent	12	stop
3	Congrat	13	click
4	WIN	14	sex
5	WON	15	girl
6	Offer	16	cash
7	Award	17	0p, 1p, ..., 9p
8	Prize	18	half price
9	Call	19	EURO, GBP, pound,£, $
10	Reply	20	Text, Txt

IV. EXPERIMENTS AND ANALYSIS

A. Experiment setup

The experiments are conducted on the spambase and SMSSpamCollection databases from UCI machine learning repository [10]. Firstly, the information gain of each attribute on decision making is calculated for the two databases, respectively, and then attributes are sorted in the order of decreasing information gains. In order to observe the impacts of attributes on the performance of decision making, four experiments are performed on two attribute orders:

I_1: a random attribute order (i.e. original attribute order from $x_1,...,x_{57}$);

I_2: an order, sorted in the decreasing information gain when each attribute involves decision making.

The four experiments are:

Ex1: the input attribute space of SVMs are increased in the order of I_1. Algorithm 1 without the phase of removing attribute in the procedure of increasing space is applied.

Ex2: repeat the Ex1, but the input space will be increased in the order of I_2.

Ex3: the input attribute space of SVMs are searched in the order of I_1. Algorithm 1 is applied. When an attribute is added to the input space, it does not have larger impact on the performance of decision making than last attribute does, the attribute will be removed during the evolvement.

Ex4: Repeat Ex3, but the input space will be searched in the order of I_2.

The accuracy and confusion matrix for each SVM are recorded during the evolvement for each experiment. The test platform is a laptop with Windows 10 and Intel (R) Core (TM) i5-3337U CPU @1.8GHZ 6GB memory. The RBF-kernel SVM uses the default Gaussian in MatLab.

B. Result evaluation on the spambase data

Fig. 1 - Fig. 4 show the results of the four experiments on the spambase data, respectively. From Fig.1, we can see the performances are randomly up-and-down, as we increased input space in random order of attributes. But we can still see the difference between the TPR and TNR when input space is less than 12 dimension attributes is larger than that when input space is not less than 12 dimensions.

Experiment 2 (Ex2) is based on I_2. The information gain based attribute order (I_2) and attribute names are listed in Table II. It can be seen that the frequency of those features, related to Capital run length, are most important in all features. Also the frequency of symbol '!' is obviously very important, since such email spams want to raise users' attention.

TABLE II. THE ATTRIBUTE ORDER BASED ON INFORMATION GAIN ON THE SPAMBASE DATA

order 1-13	names	order 14-32	names	order 33-57	names
x_{55}	capital len average	x_{23}	w freq 000	x_{49}	w freq ;
x_{52}	char freq !	x_3	w freq all	x_{15}	w freq addresses
x_{57}	capital len total	x_{17}	w freq business	x_{46}	w freq edu
x_{53}	char freq $	x_{27}	w freq george	x_{30}	w freq labs
x_{56}	capital len longest	x_{10}	w freq mail	x_{35}	w freq 85
x_{21}	w freq your	x_2	w freq address	x_{28}	w freq 650
x_{19}	w freq you	x_{26}	w freq hpl	x_{29}	w freq lab
x_{16}	w freq free	x_{12}	w freq will	x_{51}	c freq [
x_7	w freq remove	x_{54}	c freq #	x_{36}	w freq technology
x_{24}	w freq money	x_8	w freq internet	x_{14}	w freq report
x_5	w freq our	x_{11}	w freq receive	x_{42}	w freq meeting
x_{25}	w freq hp	x_{18}	w freq email	x_{31}	w freq telnet
x_{50}	c freq (	x_6	w freq over	x_{43}	w freq original
		x_9	w freq order	x_{33}	w freq data
		x_{20}	w freq credit	x_{39}	w freq pm
		x_{45}	w freq re	x_{44}	w freq project
		x_{13}	w freq people	x_{40}	w freq direct
		x_{37}	w freq 1999	x_{34}	w freq 415
		x_1	w freq make	x_{32}	w freq 857
				x_{22}	w freq font
				x_{48}	w freq conference
				x_{41}	w freq cs
				x_{38}	w freq parts
				x_4	w freq 3d
				x_{47}	w freq table

From Fig.2, it can be seen that the variation of performances is divided to three stages. The attribute order and names of each stage are listed in each two columns of Table II. The first stage is when the number of input dimensions is not higher 13, and in this stage, the Accuracy and TPR has the increasing trend. The second stage is when the number of input dimensions is between 14 and 32 (inclusive), and in this stage, the Accuracy, TPR and TNR roughly hold a similar level. The third stage is when the number of input dimensions is larger than 32. At the early third stage, TNR jump to a higher level, while TPR drop to a lower level, and they have a large difference, and then the accuracy tends to be stable, while TNR is gradually increasing and TPR is gradually decreasing.

Namely, when x_{55}, x_{52}, x_{57}, x_{53}, x_{56}, x_{21}, x_{19}, x_{16}, x_7, x_{24}, x_5, x_{25}, x_{50} are added to input space one by one, the TPR is gradually increased. This indicates these attributes have positive impact to spam. These features in the first stage are most important for spam detection. When x_{23}, x_3, x_{17}, x_{27}, x_{10}, x_2, x_{26}, x_{12}, x_{54}, x_8, x_{11}, x_{18}, x_6, x_9, x_{20}, x_{45}, x_{13}, x_{37}, x_1 are added to input space one by one, the TPR, TNR and Accuracy basically hold a similar level, this indicates these attributes hold a similar impact to spam and non-spam emails.

However, x_{49}, x_{15}, x_{46}, x_{30}, x_{35}, x_{28}, x_{29}, x_{51}, x_{36}, x_{14}, x_{42}, x_{31}, x_{43}, x_{33}, x_{39}, x_{44}, x_{40}, x_{34}, x_{32}, x_{22}, x_{48}, x_{41}, x_{38}, x_4, x_{47} have different impact to spam and non-spam emails. They have positive impact on non-spam emails, and negative impact on spam emails.

Fig. 3 shows that the Ex3 for the random order search, does not improve the performance, the the performance of TPR and TNR have a large difference. Fig. 4 shows that the Ex4 for the search on the information gain based order (I_2), does not improve the performance much, but the difference between TPR and TNR for order I_2 is less than that for randomly search.

From Fig. 3 and Fig. 4, although we remove some attributes that do not have positive impact on decision making, the performance does not improve much. This may imply that some attributes may not have independent positive impact on decision making, but they combine with other attributes may produce positive impact on decision making. Table III lists the average, maximum and standard deviation of Accuracy, TPR and TNR for the spambase data.

TABLE III. THE AVERAGE, MAXIMUM AND STANDARD DEVIATION OF ACCURACY, TNR AND TPR FOR FOUR EXPERIMENTS ON THE SPAMBASE DATA

assessment	statistics	Ex1	Ex2	Ex3	Ex4
	average	0.7409	0.8555	0.6390	0.6740
A	max	0.8443	0.9219	0.6779	0.8033
	stdev	0.0655	0.0471	0.0205	0.0500
	average	0.9679	0.9437	0.9625	0.9532
TNR	max	1	0.9907	1	0.9946
	stdev	0.0259	0.0368	0.0243	0.0192
	average	0.3897	0.7191	0.1343	0.2277
TPR	max	0.6901	0.9282	0.3453	0.6155
	stdev	0.1798	0.1515	0.0896	0.1550

Table III shows:

(1) For accuracy, increasing space (Ex1 and Ex2) obtains high accuracy than increasing searching space (Ex3 and Ex4) does, and results of experiments on Information Gain (IG) based order are better than that on the random order. Increasing space on IG-based order obtained the best performance, when Ex2 goes to the step 13. The best solution is a 13-dimension space with features $\{x_{55}, x_{52}, x_{57}, x_{53}, x_{56}, x_{21}, x_{19}, x_{16}, x_7, x_{24}, x_5, x_{25}, x_{50}\}$.

(2) For TNR, experiments (Ex1 and Ex3) on the random order obtains better performance than the experiments on IG-based order, and the maximum TNR for both experiments can reach 1, while the experiments on IG-based order cannot. For Ex1, when TNR reaches 100%, TPR is dropped to 0, and the input space is $\{x_1, x_2, x_3, x_4\}$. For Ex3, when TNR reaches to 100%, TPR is 13.81%. The input space is $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$. The Performance obtained by Ex3 is slightly better than that by Ex1. However, the TPR is too low to be acceptable.

(3) For TPR, experiments on the IG-based order obtains better performance than the experiments on the random order. Increasing space on IG-based order obtains the best performance. The best TPR reaches 92.82%, when Ex2 searches to the step 25. However, at this step, corresponding TNR is dropped to 89.11%. During the search process, features x_{52}, x_{56}, x_{16}, x_{24}, x_{23}, x_2, x_{12}, x_{54}, x_8, are removed. Therefore, the best solution is a 16-dimension space, $\{x_{55}, x_{57}, x_{53}, x_{21}, x_{19}, x_7, x_5, x_{25}, x_{50}, x_3, x_{17}, x_{27}, x_{10}, x_{26}, x_{11}, x_{18}\}$.

C. Result evaluation on the SMSSpamCollection data

The IG-based order of attributes for SMSSpamCollection data is listed in Table IV. Obviously, features x_{19} and x_{17}, related to money, have most significance on decision making.

TABLE IV. INFORMATION GAIN BASED ORDER OF ATTRIBUTES FOR THE SMS MESSAGE DATA

x	key word	x	key-word
x_{19}	EURO, GBP, pound,£, $	x_2	urgent
x_{17}	0p, 1p, ..., 9p	x_4	WIN
x_{20}	Text, Txt	x_6	Offer
x_9	Call	x_5	WON
x_1	free	x_{14}	sex
x_8	Prize	x_{11}	send
x_{12}	stop	x_3	Congrat
x_{10}	Reply	x_{18}	half price
x_7	Award	x_{13}	click
x_{16}	cash	x_{15}	girl

Fig. 5 - Fig. 8 show the results of the four experiments on the SMSSpamCollection data, respectively. In Fig. 5 and Fig. 6, the three types of performance converge to a similar level. TNR gradually decreases a little, TPR gradually increases, and accuracy has slight undulation during the evolvement. Performance obtained by Ex2 converges faster than that obtained by Ex1. For Ex2, when the input space is increased to 5 dimensions, the three types of performance reach to the stable level for Ex2, while for Ex1, when the input space is increased to 14 dimensions, they reach to the stable level. Fig. 7 and Fig. 8 show that the difference between TNR and TPR for the increasing search space approach on the random order is lager than that on IG-based order. For both experiments, the accuracy has slight undulation as well. For Ex4, when TPR reaches the highest, TNR gets the lowest. Table V lists the average, maximum and standard deviation of Accuracy, TPR and TNR.

TABLE V. THE AVERAGE, MAXIMUM AND STANDARD DEVIATION OF ACCURACY, TNR AND TPR FOR FOUR EXPERIMENTS ON THE SMSSPAMCOLLECTION DATA

assessment	statistics	Ex1	Ex2	Ex3	Ex4
	average	0.8898	0.9051	0.9083	0.9417
A	max	0.9054	0.9418	0.9453	0.9491
	stdev	0.0129	0.0124	0.0158	0.0090
	average	0.9201	0.9116	0.9698	0.9749
TNR	max	0.9872	0.9948	0.9880	0.9950
	stdev	0.0398	0.0319	0.0154	0.0119
	average	0.6916	0.8626	0.5064	0.7247
TPR	max	0.9081	0.9392	0.8122	0.8014
	stdev	0.2035	0.1304	0.1404	0.0812

The statistic results in Table V show that:

(1) For accuracy, the increasing search space approach obtains better performance than the increasing space approach for the same order. The performance of the experiment on IG-based order is better than the random order for the same approach. Increasing search space approach achieves the best performance (94.91%), when Ex4 searches the input space up to step 14. During the searching process, features, x_{20}, x_9, x_8, x_{12}, x_{10}, x_{16}, x_2, x_4, are removed. Therefore,

the best solution is a 6-dimension input space, $\{x_{19}, x_{17}, x_1, x_7, x_6, x_5\}$.

(2) For the maximum of TNR, experiments on the IG-based order obtained better performance than that on the random order. For the average of TNR, the increasing search space approach obtains better performance than that the increasing space approach. The best performance 99.5% is obtained by the increasing search space approach on the IG-based order, when Ex4 searches at the first step. Namely, feature x_{19} determines 99.5% SMS non-spams, but 40.54% SMS spams.

(3) For TPR, the increasing space approach obtains better performance than the increasing search space approach on the same order. The best performance 93.92% is obtained by the increasing space approach on the IG-based order, when Ex2 runs to step 10. Therefore, the best solution is a 10-dimension input space, $\{x_{19}, x_{17}, x_{20}, x_9, x_1, x_8, x_{12}, x_{10}, x_7, x_{16}\}$. At this step, the TNR is only 89.77%.

V. Conclusions

Automatic spam analysis and detection in email or mobile information systems is of the essence for the security insurance of our email and SMS communication. While improving true positive rate of the automatic spam detector, we might lose some important emails or messages, as the true negative rate of the automatic spam detector is usually decreased. The tradeoff of TNR and TPR is worthy of consideration. Experiments show that TPR is always very low when TNR reaches the maximum, but Fig. 2 and Fig. 6 show that there exists a feature space that achieves Pareto efficiency in TPR and TNR.

According to the information gains of attributes, it can been seen that the frequency of those features related to Capital run length and symbol '!' are the most important in all features for spam email detection. Similarly, the two features of money units and price units are the most important in spam SMS detection.

The results show that experiments on IG-based order obtained higher accuracy than the experiments on the random order did for both spam emails and spam SMS detection. For spam SMS detection, the increasing search space approach obtained higher accuracy than the increasing space approach, but for spam email detection, the conclusion is opposite. It might indicate that some features retrieved from the email database could have synthetic impact on the decision making. Therefore, appropriate features retrieving is important.

From Fig. 1 - Fig. 8, it can be seen that the impact of an added attribute on the detection performance could be either positive or negative. Due to synthetic effect of features on decision making, simply removing an attribute in the increasing search approach may not a very good strategy. Hence, the optimisation of feature space for Pareto efficiency in TPR and TNR will be our future work.

References

[1] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami. Contributions to the study of sms spam filtering:new collection and results. In *DocEng'11*, Mountain View, California, USA, 19-22 September 2011.

[2] J. Alqatawna, H. Faris, K. Jaradat, M. Al-Zewairi, and O. Adwan. Improving knowledge based spam detection methods: The effect of malicious related features in imbalance data distribution. *International Journal of Communications, Network and System Sciences*, 8:118–129, 2015.

[3] A. A. Benczúr, K. Csalogány, T. Sarlás, and M. Uher. Spamrank - fully automatic link spam detection work in progress. In *The first international workshop on adversarial information retrieval on the web (AIRWeb'05)*, Chiba, Japan, 2005.

[4] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. A. Najada. Survey of review spam detection using machine learning techniques. *Journal of Big Data*, 2(23):Online open assess, Dec 2015.

[5] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *30th International Conference on Very Large Data Bases (VLDB)*, pages 576–587, Morgan Kaufmann, August 2004. Morgan Kaufmann.

[6] N. Jindal and B. Liu. Review spam detection. In *WWW 2007*, ACM 978-1-59593-654-7/07/0005, Banff, Alberta, Canada, 8-12 May 2007.

[7] P. Kolari, A. Java, T. Finin, T. Oates, and A. Joshi. Detecting spam blogs: a machine learning approach. In *Proceeding of AAAI'06 proceedings of the 21st national conference on Artificial intelligence*, volume 2, pages 1351–1356, 2006.

[8] V. Krishnan and R. Raj. Web spam detection with anti-trust rank. In *In AIRWEB*, pages 37–40, 2006.

[9] R. M. Lee, M. J. Assante, and T. Conway. Ics defense use case (duc) : German steel mill cyber attack. Report, SANS, DEC 2014.

[10] M. Lichman. UCI machine learning repository, 2013.

[11] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Techincal report, Stanford University, 1998.

[12] L. Payet. Terror-alert spam targets the middle east, canada to spread malware. Symantec Official Blog, Nov 2015.

[13] M. Richardson, A. Prakash, and E. Brill. Beyond pagerank: Machine learning for static ranking. In *15th International Conference on World Wide Web (WWW)*, pages 707–715, NY, USA: ACM Press, May 2006.

[14] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk e-mail. In *In AAAI-98 Workshop on Learning for Text Categorization*, Madison, Wisconsin, USA, 1998.

[15] M. Sharifi, E. Fink, and J. G. Carbonell. Detection of internet scam using logistic regression. In *2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2168 – 2172, Oct 2011.

[16] H. Shirani-Mehr. Sms spam detection using machine learning approach.

[17] R. M. Silva, A. Yamakami, and T. A. Almeida. An analysis of machine learning methods for spam host detection. In *2012 11th International Conference on Machine Learning and Applications (ICMLA)*, volume 2 of *10.1109/ICMLA.2012.161*, Boca Raton, FL, Dec 2012.

[18] M. Sobek. Pr0-google's pagerank 0 penalty. pr.efactory.de/e-pr0.shtml, 2002.

[19] GFI Software. Why bayesian filtering is the most effective anti-spam technology. GFI White Paper, 2011.

[20] N. Spirin and J. Han. Survey on web spam detection: Principles and algorithms. *ACM SIGKDD Explorations Newsletter archive,ACM New York, NY, USA*, 13(2), 12 2011.

[21] H. Tran, T. Hornbeck, V. Ha-Thuc, J. Cremer, and P. Srinivasan. Spam detection in online classified advertisements. In *Proceedings of the 2011 Joint WICOW/AIRWeb Workshop on Web Quality (WebQuality'11)*, ISBN: 978-1-4503-0706-2, pages 35–41. ACM NY, USA, 2011.

[22] K. Tretyakov. Machine learning techniques in spam filtering. In *Data Mining Problem-oriented Seminar, MTAT.03.177*, pages 60–79, May 2004.

[23] J. Verma and S. Dhawan. Detection of spam in social networks using clustered k-nearest neighbour. *International Journal of Advanced Research in Computer Science and Software Engineering*, 5(3):1120–1123, Mar 2015.

[24] B. Wang, A. Zubiaga, M.Liakata, and R. Procter. Making the most of tweet-inherent features for social spam detection on twitter. In *Proceedings of the 5th Workshop on Making Sense of Microposts(Microposts2015) @WWW2015*, volume CEUR 1395, pages 10–16, Florence, Italy, 18 May 2015.

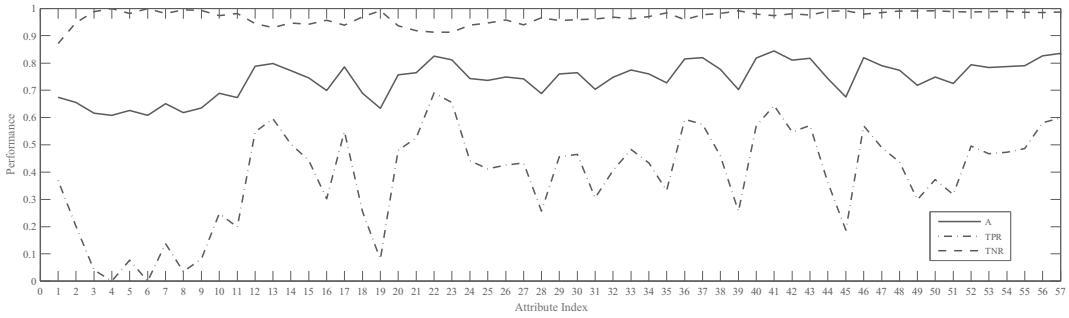

Fig. 1. The evolved performance of accuracy, TPR and TNR for Ex1 on the spambase data

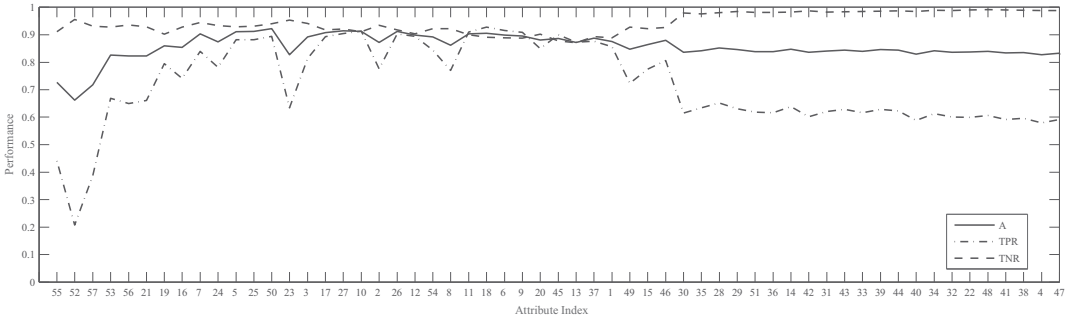

Fig. 2. The evolved performance of accuracy, TPR and TNR for Ex2 on the spambase data

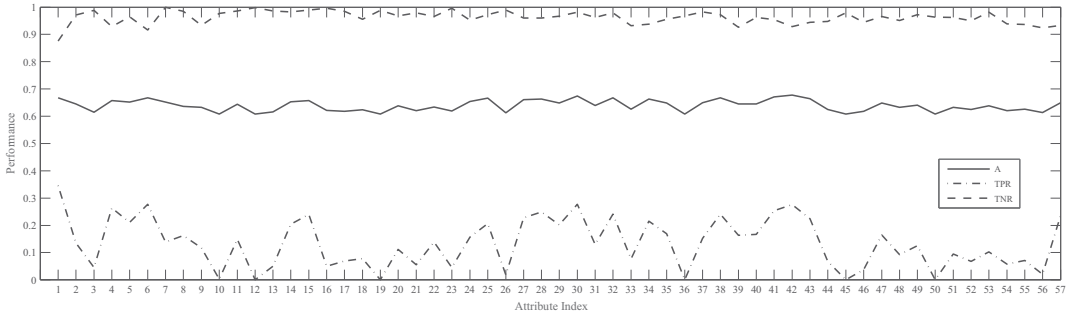

Fig. 3. The evolved performance of accuracy, TPR and TNR for Ex3 on the spambase data

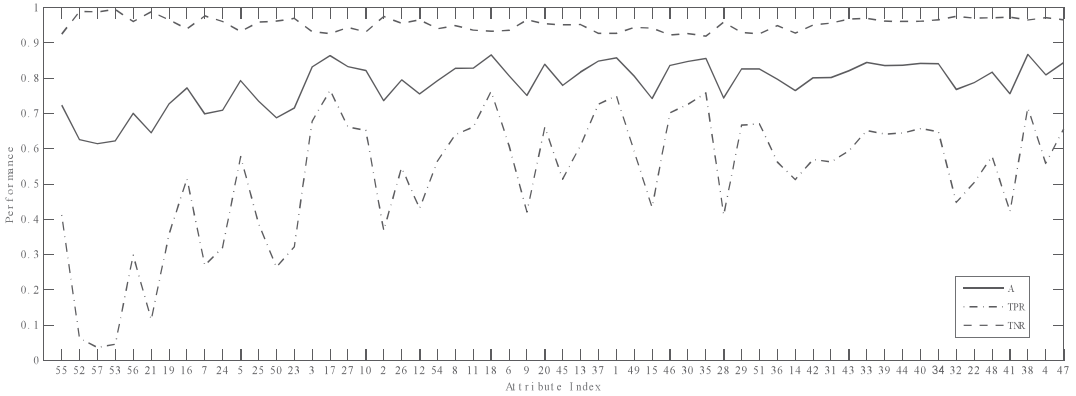

Fig. 4. The evolved performance of accuracy, TPR and TNR for Ex4 on the spambase data

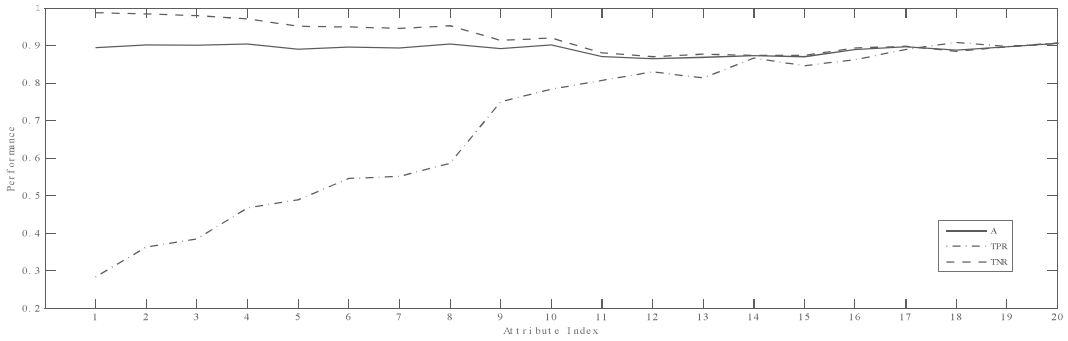

Fig. 5. The evolved performance of accuracy, TPR and TNR for Ex1 on the SMSSpamCollection data

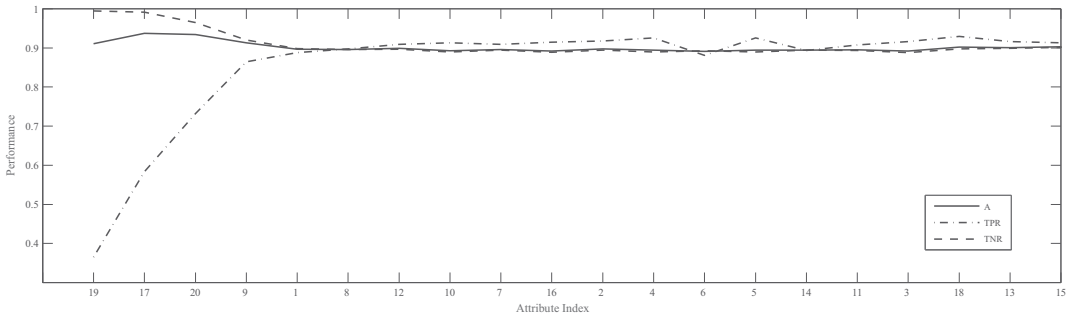

Fig. 6. The evolved performance of accuracy, TPR and TNR for Ex2 on the SMSSpamCollection data

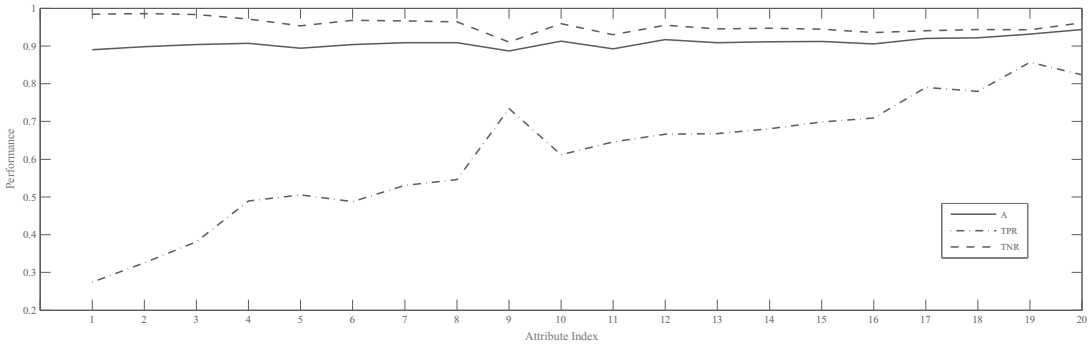

Fig. 7. The evolved performance of accuracy, TPR and TNR for Ex3 on the SMSSpamCollection data

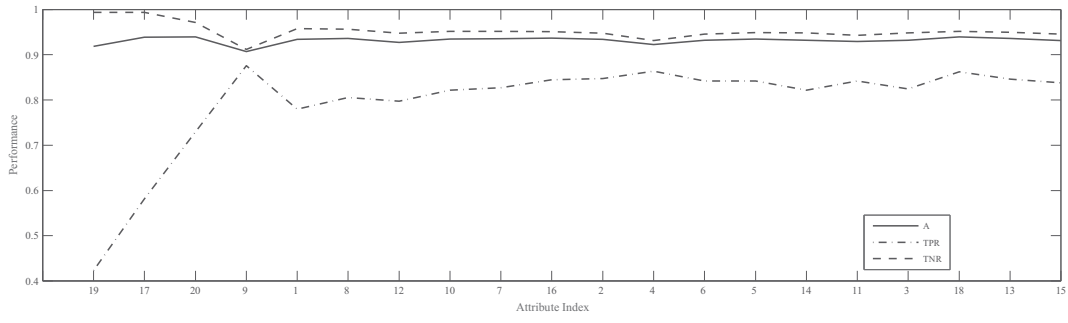

Fig. 8. The evolved performance of accuracy, TPR and TNR for Ex4 on the SMSSpamCollection data