A supervised approach for spam detection using text-based semantic representation

N. Saidani[1], K. Adi[1], and M.S. Allili[1]

[1] Department of Computer Science and Engineering, University of Quebec in Outaouais, Canada
sain06@uqo.ca,Kamel.Adi@uqo.ca
MohandSaid.allili@uqo.ca

Abstract. In this paper, we propose an approach for email spam detection based on text semantic analysis at two levels. The first level allows categorization of emails by specific domains (e.g., health, education, finance, etc.). The second level uses semantic rules for spam detection in each specific domain. We show that the proposed method provides an efficient representation of internal semantic structure of email content which allows for more precise and interpretable spam filtering results compared to existing methods.

Keywords: Email spam detection, domain categorization, semantic rule induction.

1 Introduction

Email is one of the most used services on the Internet given the advantages it offers in terms of transmission speed, the ability to handle multimedia documents and broadcast emails to groups of people. By its popularity, the email generates its biggest disadvantage, which is the overload of the mailboxes with unwanted messages called spam. Among others, it favors the fast distribution of false information and malicious codes. According to a study of Kaspersky Lab in 2014, spam represented a share of over 70% of unwanted email in the traffic. These emails not only cause a waste of time and resources (storage: disk and transmission: bandwidth) but can also cause a major problem in computer security with billions of dollars and productivity loss [1].

Several spam detection methods have been proposed in the literature [2]. Among recent approaches, learning-based methods using text mining have gained more and more popularity [10]. These methods generally represent email content using text features such as words, n-grams, etc., which aim to distinguish between spam and a legitimate emails (ham). These methods have proven good efficiency compared with other methods relying only on network analysis or black lists, for example [18]. However, these methods have also their limitations since textual features that are extracted independently can usually miss features correlation and semantic content description of the email. Indeed, spam content can

be very dependent on the domain and the targeted users. For example, in health, spams can be targeted to medicine or false therapy campaigns advertisement, whereas in finance, spams can carry advertisement for dubious financial services and products. Therefore, using a specific spam discrimination for each domain can be more efficient than a general-purpose spam filter. Moreover, using more semantic cues for each domain in addition to raw text features can offer better discrimination between legitimate and spam emails. For example, emails advertising health products can be of interest if they are only informative and not targeting money extortion.

In this paper, we propose a general approach for incorporating semantic analysis for spam detection. Semantic analysis of email content is carried out at two levels. These levels consist respectively on first categorizing emails by domains and then extracting explicit semantic concepts within each domain to classify emails into spam and ham categories. For email domain categorization, without loss of generality, we have considered five domains: 1) Computer, 2) Adult, 3) Education, 4) Finance, and 5) Health. These categories are among the most targeted by spams [8, 9, 17]. To assign emails to these domains, we train a supervised classifier on labeled data after operating feature selection on the vocabulary of email text content using information gain. Semantic rules are then generated automatically for each domain using CN2-SD method [14], which will serve as weak spam classifiers with outputs combined in a more general and robust classifier for discriminating spams from legitimate emails. Experiments on a large corpus of emails have shown that our approach yields very good spam classification results compared to recent text-based filtering techniques.

The rest of the paper is organized as follows: in Section 2 we give a brief overview of related work. In Section 3 we discuss our contribution to spam detection using text-based semantic representation. We report our evaluation results in section 4. Finally, Section 5 presents our conclusions and some directions for future work.

2 Related Work

Various spam filtering methods have been proposed in the literature for spam detection. Most of these methods have some success for filtering specific spams, but fail to provide effective procedures that solve definitely the problem. Recently, machine learning algorithms were widely used for spam filtering, after their indisputable success in text categorization [10]. In fact, spam filtering can be seen as a text categorization with two classes $\{spam, ham\}$ and several classifiers were applied such as support vector machines (SVM) [19], naïve Bayes (NB)[20], artificial neural networks (ANN) [15], etc.

For spam filters, features are usually obtained from the body, the subject or the header of emails. Thus, email text-content play an important role in any categorization process. One of the most popular representation for email categorization is the vector-space model (VSM), also called bag-of-words (BoW)

[10]. This model, describes the text of an email as a vector of words where each word represents an individual feature of the email. However, this representation is very high-dimensional and may incur an important loss in the semantic of the email since words are taken independently. To overcome this issue, some works has used n-gram models instead of individual words [6]. In this representation, emails are represented by sequences of words which leads to more refined models. However, this approach increases exponentially the size of the vocabulary which leads to highly sparse spaces for representing email documents. Several works have proved that complex representations of texts do not always improve the efficiency of the classification and sometimes may even deteriorate it [5].

Given the limitations of above mentioned methods, some researchers have recently investigated semantic-based approaches for improving emails classification [2]. Here, we refer to the semantic approach as the ability to depict and capture, in an explicit way, the information conveyed by emails. Authors in [13] explored the use of semantic manipulations for spam filtering by introducing a word sense disambiguation as a pre-processing step. The task of disambiguating words senses is the process of identifying the most appropriate meaning of a polysemous word for a specific context. In [16], the authors introduce a model called "enhanced topic-based vector space" (eTVSM) which uses a semantic ontology to deal with synonyms. These methods can have some success in narrowing the semantic meaning of words depending on the email content. However, they do not extract higher level semantic concepts of emails which can be helpful for discriminating legitimate and spam emails.

3 Our approach

In our approach for spam detection, a fundamental step is the semantic characterization of the considered specific domains (computer, adult, education, finance and health), each described by a set of semantic rules. Each set of semantic-rules is then used to provide basic features for building a domain specific classifier. Figure 1 summarizes our overall approach. As we can see in the figure, we proceed by two levels of semantic analysis. In the first level, we use a classification algorithm to automatically partition a global training dataset (emails) into the considered five domains. In the second level, we automatically extract a set of semantic rules (or semantic concepts) from the dataset in each domain. The semantic concepts are then used as semantic attributes to build specialized classifiers for detecting domain specific spams.

3.1 Email categorization by domain

Today, spam is used in a myriad of goals, through the unsolicited advertising, phishing, to the dissemination of malicious code. Targeted domains are many: medicine, education, finance, etc. The annual reports on spam by Kaspersky [8,

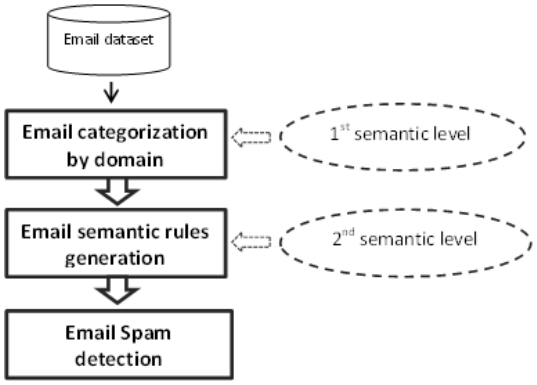

Fig. 1: General model of our approach.

9] and symantec [17] contain a deep analysis of the new targeted domains with the different techniques used to send spams. Those reports have been used, in our work, to fix the spammers' most targeted domains and five domains were considered.

In the domain of health, typical spams include advertisements for weight loss, skin care, improved posture, nutritional supplements, alternative medicine etc. In the domain of finance, we can found offers for insurance, debt reduction services, loans at competitive interest rates etc. The computer domain includes software, cheap equipment and services for website owners such as accommodation, registration areas, optimizing websites, etc. In the adult domain, typical spams are those offering products to increase or improve sexual ability, links to pornographic sites or pornographic advertising, etc. The education domain includes offers for seminars, training courses, evening classes, etc. It is worth to mention that, spammers are constantly seeking to penetrate new domains and develop new techniques and some sectors are quickly evolving and should be monitored closely (e.g.: politics).

For email categorization by domain, we assign a category to each email of the global training dataset. Notice that an appropriate email preprocessing steps are required before we can efficiently exploit the information contained in the subject and the body of emails for the categorization. We present, in Figure 2, the global process for email categorization.

Let $D = \{d_1, \cdots, d_n\}$ be the set of email documents, $C = \{c_1, \cdots, c_p\}$ be the set of categories and $T = \{t_1, \cdots, t_m\}$ be the set of characteristics called terms. Note that in the present work, we consider five categories ($p = 5$) and each document $d \in D$ is associated to a unique category $c(d) \in C$. For the categorization process, we consider three main steps. The first step allows texts preprocessing on documents in D, where each document is represented by

a vector of terms. The second step is used to extract the relevant features. A special attention is paid for the reduction of the data dimensionality to avoid the deterioration of the resolution system in the presence of noisy data. The third step consists of a learning phase on different classification algorithms to a build the better classification model for categorizing email documents.

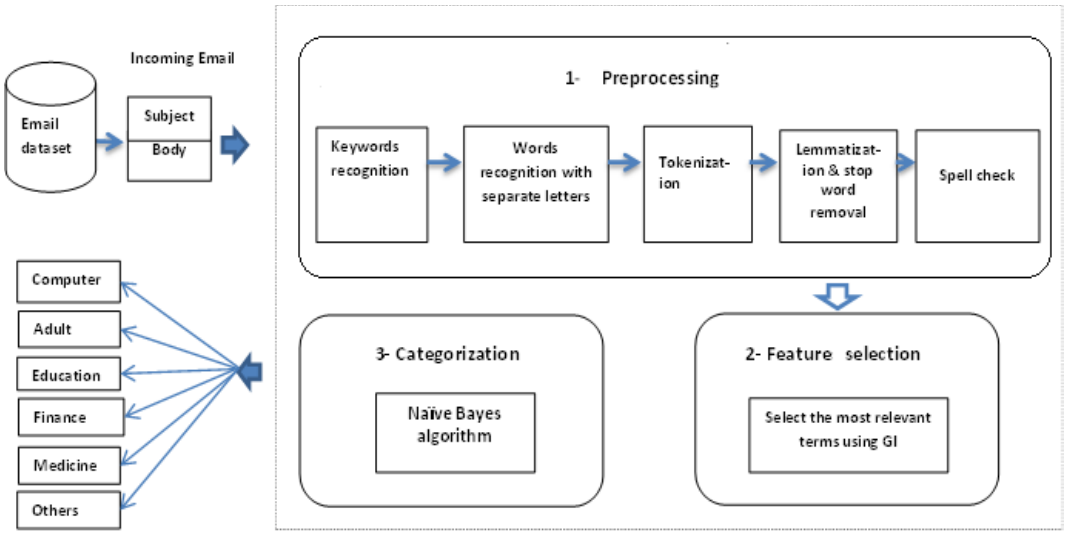

Fig. 2: System for categorizing emails by domains.

3.1.1 Preprocessing

The preprocessing phase includes the following steps:

- *Keywords recognition*: used to recognize keywords and abbreviations in the emails with the help of a dictionary of keywords for each category. For instance, in the computer science category, we identified the abbreviations VLC, CD, PDF etc.; in the adult category, we spot the taboo words, etc. Keywords are coded by regular expressions.

- *Words recognition with separate letters*: this step is important for text segmentation phase (tokenization), which is the next phase of our process. The goal of this phase is to avoid alphabet letters in the feature vectors at the end of this process. Indeed, while referring to a dictionary of natural language, we implemented a tree search algorithm of the longest word on segments of text strings.

- *Tokenization*: is the step of text segmentation to extract the words or the terms of the email. Our algorithm performs a text division into tokens by using white space as a separator. So each email is coded as a vector of tokens representing the vocabulary used in the email.

- *Stop-word removal*: allows to eliminate some words that often occur in messages (e.g., "to", "a", "for").

- *Lemmatization*: allows to reduce words to their root forms (e.g., "extracting" to "extract"), at this stage we use the algorithm PorterStemmer [1].
- *Spell check Step*: allows to check the spelling of words in the vocabulary vector (vector of characteristics), we used a function to call Microsoft Word spell checker in order to correct misspellings especially those of spammers. This step enhances the recognition of key words in each category.

The removal of the stop words and the standardization (lemmatization) are very useful as they allow the dimensionality reduction of the characteristics vector. However, this still remain insufficient and to solve the problem, we used a statistical method for selecting relevant features.

3.1.2 Feature selection

Feature selection is an important step and aims to reduce the number of features for improving classifier performance. To select the most representative terms we used the Information Gain (IG) which is widely recognized in the spam detection literature [10]. It measures the discriminating power of a word, i.e.: the information amount provided by the knowledge of the appearance or not of a term in the decision process. Given the set T containing all email's terms obtained during the preprocessing phase from D and a set of categories C, the IG provided by a term $t \in T$ for a category $c \in C$ is defined as follows:

$$IG(t,c) = \sum_{\acute{c} \in \{c,\bar{c}\}} \sum_{\acute{t} \in \{t,\bar{t}\}} p(\acute{t},\acute{c}) log(\frac{p(\acute{t},\acute{c})}{p(\acute{t})p(\acute{c})}) \quad (1)$$

Where $\bar{t}$ indicates the absence of a term t and $\bar{c} \in C \setminus \{c\}$ and:

- (t,c): represent the presence of t and a membership in c.
- $(t,\bar{c})$: represent the presence of t and a non-membership in c.
- $(\bar{t},c)$: represent the absence of t and a membership in c.
- $(\bar{t},\bar{c})$: represent the absence of t and a non-membership in c.

The first and last tuples represent the positive dependencies between t and c, while the other two represent the negative dependencies and $p(\acute{t},\acute{c})$ is the occurrence frequencies of the four-tuples in the corpus D. $p(\acute{t}$ and $p(\acute{c})$ represent term and class probabilities in the collection. Finally, we chose the 500 terms having the highest information gain for each category.

[1] http ://tartarus.org/martin/PorterStemmer/index-old.html

3.1.3 Categorization

After performing feature relevance analysis, various classification algorithms are applied to categorize email documents by domain. We compared the following classification methods in our experiments:

- **Bayesian classifier** is based on the theorem of Bayes. For a set of training data D, the classifier calculates for each category, the probability that a document $d \in D$ represented as a vector of m terms $d = (t_1, \cdots, t_m)$, belongs to a category $c \in C$. This calculation is done for each category, and we consider the highest probability to select the category of an email.

- **K-nearest neighbor** is one of the most popular used methods for text classification. The approach looks at the K email documents in the training dataset that are the closest to the email under classification: it is classified according to the class to which the majority of the K-nearest neighbors belong.

- **Decision tree** is a structure that includes a root node, branche with internal nodes and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label.

3.2 Email semantic rules generation

This step aims at extracting semantic meanings for email text. As semantics of emails, we mean a set of hidden concepts describing the email's content. The final goal is to create an very precise semantic representation for an efficient detection of spams. In this regard, we applied CN2-SD algorithm [14] for an automatic extraction of semantic rules.

The CN2-SD algorithm is built on top of two algorithms: CN2 [3, 4] and SD [12]. SD is used to discover subgroups on a set of data and CN2 is used to induce classification rules. The main difference between a classification and a discovery of subgroups is that classification is a predictive task, while discovering subgroups is a descriptive task. The main reasons behind our choice of CN2-SD algorithm are as follows:

- allows an efficient and automatic induction of rules.
- allows an automatic generation of population description. This is particulary useful for the extraction of the hidden semantic concepts.
- ensures precise discrimination between populations.

CN2-SD algorithm:
The idea of CN2-SD is to adapt the classifier CN2 to the task of subgroup discovery. CN2 sequentially build a set of classification rules from a training

dataset. At each iteration, generating a rule, CN2 removes the rule covered subset from the training dataset. Rule candidates are constructed based on a beam search strategy and a selection metric. One of the most used selection metrics is the accuracy, it is defined as follows:

$$Acc(Cond \rightarrow Class) = p(Class|Cond) = \frac{p(Class.Cond)}{p(Cond)} \tag{2}$$

where $p(Cond)$ represents the number of examples (emails) covered by the rule $Cond \rightarrow Class$ and $p(Class.Cond)$ is the number of correctly classified examples (true positives).

The main modifications of the CN2 algorithm, making it appropriate for SD, involve the implementation of the weighted covering algorithm by incorporating example weights into the Weighted Relative Accuracy (WRAcc) heuristic.

In the first iteration of the algorithm, all examples are assigned the same weight: $w(d_i,\ 0) = 1$, which means the email d_i have not been covered by any rule. In the following iterations, weights of emails covered by one or more rules will decrease according to a weighting scheme. Two weighting schemes can be used in CN2-SD, the additive weights and the multiplicative weights. In our spam detection framework, we used the additives weight method, whose equation is defined as follows:

$$w(d_i,\ j) = \frac{1}{j+1} \tag{3}$$

the WRAcc is defined as follows:

$$\text{WRAcc}(Cond \rightarrow Class) = \frac{s(Cond)}{S}\left(\frac{s(Class.Cond)}{s(Cond)} - \frac{s(Class)}{S}\right) \tag{4}$$

Where S is the sum of the weights of all the examples, $s(Cond)$ represents the sum of the weights of all the examples covered by the induced rule, and $s(Class.Cond)$ is the sum of the weights of all correctly covered examples by the rule.

The algorithme CN2-SD uses the metric WRAcc to select rule candidates. It also yields unordered sets of rules, but combines them in terms of a uniform weighting scheme. Furthermore, covered examples at each iteration are not removed, but only re-weighted.

3.3 Generation of domain specific classifiers

For each specific domain, the set of semantic rules generated from the previous step is used as learning attributes to build a domain specific classifier. To this end, we use a labeled training set of emails in each domain and perform a

supervised learning of candidate classifiers such as naive Bayes, decision trees and KNN.

Whence domain specific classifiers are generated, a newly coming email should be first automatically assigned to one of the considered domains. The domain specific classifier is, then, used to classify the e-mail as legitimate or spam. See Figure 3 for the process of spam detection.

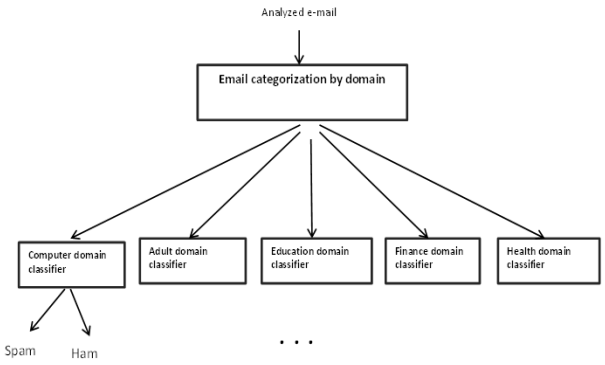

Fig. 3: The process of spam detection using domain specific classifiers.

4 Experimental results

To evaluate our approach, we collected a dataset of emails from several sources. The spam emails come from two datasets, Enron[2] and Ling-spam[3]. The ham emails come from specialized discussion forums and the two datasets Enron and Ling-spam. The use of the forums for ham collection is made necessary due to the insufficiency hams in the Enron and Ling-spam databases. We used a total of 6679 emails distributed according to their content in the categories health, adult, finance, education and computer. The global collection includes 3475 ham emails and 3204 spam emails.

We compared three classifiers: Knn, naïve Bayes and decision tree to categorize emails by domains. Then we applied the same classifiers to separate spam from legitimate emails in each of the considered domains. In order to evaluate the performance of the machine learning classifiers, we apply the k-fold cross validation model with $k = 10$, which randomly divides the dataset into k subsets.

[2] https ://www.cs.cmu.edu/ ./enron/
[3] http ://csmining.org/index.php/lingspam-datasets.html

Each classifier is trained on $k-1$ sets and evaluated on the remaining set. The final estimation of the classifier is the average of the k results from the subsets. We consider the following metrics: *Precision*, *Recall* and *Accuracy* metrics to evaluate the generated classifiers. These are defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

where TP, TN, FP and FN are the obtained true positives, true negatives, false positives and false negatives after classification.

Table 1: Evaluation results of machine-learning classifiers for categorizing emails by domains.

Classifier	Accuracy	Recall	Precision
Knn	0.8182	0.8758	0.5482
Naïve Bayes	0.9684	0.9684	0.9745
Decision tree	0.9416	0.8994	0.9718

Table 1 shows the obtained evaluation results for email categorization by domains. Knn, Decision tree and Naïve Bayes classifiers were capable of achieving more than 80 % of good prediction. Naïve Bayes gives the best results with an accuracy of 0.9684, a recall of 0.9684 and a precision of 0.9745.

Table 2 summarizes the results for spam detection on each considered domain. We can see that Naïve Bayes almost achieved a higher accuracy, recall and precision in all the specified domains (computer, adult, education, finance, health). The last row of the table "Average" represents the overall evaluation results of our approach which are given by computing the average of the obtained results in the five domains. According to the results, Naïve Bayes classifier clearly outperform the other algorithms. This achieved an accuracy of 0.9772, a recall of 0.9771 and a considerably high precision of 0.9705.

To prove that domain-based detection allows better prediction of spam, we used a direct approach, where the semantic attributes of all the domains are used to detect spam without going through specific domains. The obtained results are shown in Table 3. We can see clearly that the spam detection after categorization by domain outperform the direct approach.

We compare the filtering capabilities of our approach with two different approaches, eTVSM and VSM. The approach eTVSM use semantic relationships between terms and the approach VSM uses terms as a bag of words. As it can be seen in Table 4, our method could produce significant performance compared

Table 2: Evaluation results of machine-learning classifiers based on our semantic approach for spam detection in each domain.

	Classifier	Accuracy	Recall	Precision
Computer	Knn	0.9424	0.9448	0.9278
	Naïve Bayes	0.9633	0.9611	0.9722
	Decision tree	0.9677	0.9725	0.9692
Adult	Knn	0.9782	0.9710	0.9678
	Naïve Bayes	0.9759	0.9745	0.9736
	Decision tree	0.8763	0.8613	0.8042
Education	Knn	0.9565	0.9696	0.9759
	Naïve Bayes	0.9779	0.9954	0.9774
	Decision tree	0.9786	0.9963	0.9774
Finance	Knn	0.9669	0.9888	0.9566
	Naïve Bayes	0.9707	0.9888	0.9626
	Decision tree	0.9698	0.9856	0.9639
Health	Knn	0.9729	0.9735	0.9620
	Naïve Bayes	0.9718	0.9655	0.9668
	Decision tree	0.9751	0.9655	0.9746
Average	Knn	0.9634	0.9695	0.9580
	Naïve Bayes	0.9772	0.9771	0.9705
	Decision tree	0.9554	0.9562	0.9378

Table 3: Evaluation results for spam detection without categorization by domains

Classifier	Accuracy	Recall	Precision
Knn	0.9070	0.9301	0.8953
Naïve Bayes	0.8534	0.9494	0.8043
Decision tree	0.9175	0.9197	0.9216

to other methods. Additionally, our approach does not extract high dimensional feature dataset which makes the system more efficient.

Table 4: Comparative evaluation results

Model	Classifier	Accuracy	Recall	Precision
eTVSM [16]	Knn	0.9355	0.9067	0.9565
	Naïve Bayes	0.9739	0.9650	0.9674
	Decision tree	0.9657	0.9683	0.9658
VSM [10]	Knn	0.8512	0.7709	0.9312
	Naïve Bayes	0.9361	0.9784	0.9062
	Decision tree	0.9088	0.9102	0.9142

5 Conclusion

We have proposed a new approach for exploiting semantic information for spam detection. This is achieved by extracting semantic features specific to email domains. For this purpose, emails are first assigned to their domains by training a supervised classifier. Then, semantic features in form of induction rules are extracted for each domain and combined to form a more general and robust spam classifier specific to each domain. Conducted experiments have shown that our approach yields better results in terms of spam detection in comparison with approaches based on bag-of-words and/or extracting word-based semantic information (eTVSM). Future work will focus on enhancing our model using more elaborated semantic features like ontologies and word sense disambiguation.

References

1. A. Bratko, G. V. Cormack, and al. Spam filtering using statistical data compression models. Journal of Machine Learning Research, vol. 7, no Dec, p. 2673-2698, 2006.
2. G. Caruana and M. Li. A survey of emerging approaches to spam filtering. ACM Computing Surveys (CSUR), vol. 44, no 2, p.1-27, 2012.
3. P. Clark and R. Boswell. Rule induction with CN2 : Some recent improvements. In Proceedings of the Fifth European Working Session on Learning, Springer Berlin Heidelberg, p. 151-163, 1991.
4. P. Clark and T. Niblett. The CN2 induction algorithm. Machine learning, vol. 3, no 4, p. 261-283, 1989.
5. G. V. Cormack. Email spam filtering : A systematic review. Foundations and Trends in Information Retrieval, vol. 1, no 4, p. 335-455, 2007.
6. A. Çiltik, and T.Güngör. Time-efficient spam e-mail filtering using n-gram models. Pattern Recognition Letters, vol. 29, no 1, p. 19-33, 2008.
7. J. Fürnkranz and D. Gamberger. Foundations of rule learning. Springer Science and Business Media, Heidelberg New York Dordrecht London, p. 199-207, 2012.

8. D. Gudkova, M. Vergelis and al. Spam and phishing in Q2 2016. Kaspersky Lab, p. 1-22, 2016.
9. D. Gudkova, M. Vergelis and N. Demidova. Spam and phishing in Q2 2015. Kaspersky Lab, p. 1-19, 2015.
10. T. S. Guzella and W. M. Caminhas. A review of machine learning approaches to spam filtering. Expert Systems with Applications, vol. 36, no. 7, p. 10206-10222, 2009.
11. L. O. Hall, N. Chawla and K. W. Bowyer. Combining decision trees learned in parallel. In Working Notes of the KDD-97 Workshop on Distributed Data Mining, p. 10-15, 1998.
12. F. Herrera, C. J. Carmona del Jesus and al. An overview on subgroup discovery : Foundations and applications. Knowledge and Information Systems Published online first, vol. 29, no 3, p. 495-525, 2010.
13. C. Laorden, I. Santos and al. word sense disambiguation for spam filtering. Electronic Commerce Research and Applications, vol. 11, no 3, p. 290-298, 2012.
14. N. Lavrac, B. Kavsek, P. Flach and L. Todorovski. Subgroup discovery with CN2-SD. The Journal of Machine Learning Research, vol. 5, no 2, p. 153-188, 2004.
15. D. K. Renuka, T. Hamsapriya and al. Spam classification based on supervised learning using machine learning techniques. In : Process Automation, Control and Computing (PACC), International Conference on. IEEE, p. 1-7, 2011.
16. I. Santos, C. Laorden, B. Sanz and P. G. Bringas. Enhanced topic-based vector space model for semantics aware spam filtering. Expert Systems with Applications, vol. 39, no 1, p. 437-444, 2012.
17. Symantec. Internet Security Threat Report. Vol 21, p. 1-77, April 2016.
18. G. Tang, J. Pei, W.S. Luk. Email mining: tasks, common techniques, and tools. Knowledge and Information Systems,vol. 41, no 1, p. 1-31, 2014.
19. Z. S. Torabi, M. H. Nadimi-Shahraki and al. Efficient support vector machines for spam detection : a survey. International Journal of Computer Science and Information Security, vol. 13, no 1, p. 11, 2015
20. H. Wang, G. Zheng and Y. He. The improved bayesian algorithm to spam filtering. In Proc. of the 4th International Conference on Computer Engineering and Networks. Springer International Publishing. p. 37-44, 2015.