Applied Soft Computing 12 (2012) 3671–3682

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Rough sets for spam filtering: Selecting appropriate decision rules for boundary e-mail classification

Noemí Pérez-Díaz, David Ruano-Ordás, José R. Méndez, Juan F. Gálvez, Florentino Fdez-Riverola*

ESEI: Escuela Superior de Ingeniería Informática, University of Vigo, Edificio Politécnico, Campus Universitario As Lagoas s/n, 32004 Ourense, Spain

ARTICLE INFO

Article history:
Received 22 February 2011
Received in revised form 29 February 2012
Accepted 21 May 2012
Available online 29 June 2012

Keywords:
Spam classification
Rough sets
Rule execution schemes
Content-based techniques
Model evaluation

ABSTRACT

Nowadays, spam represents an extensive subset of the information delivered through Internet involving all unsolicited and disturbing communications received while using different services including e-mail, weblogs and forums. In this context, this paper reviews and brings together previous approaches and novel alternatives for applying rough set (RS) theory to the spam filtering domain by defining three different rule execution schemes: MFD (most frequent decision), LNO (largest number of objects) and LTS (largest total strength). With the goal of correctly assessing the suitability of the proposed algorithms, we specifically address and analyse significant questions for appropriate model validation like corpus selection, preprocessing and representational issues, as well as different specific benchmarking measures. From the experiments carried out using several execution schemes for selecting appropriate decision rules generated by rough sets, we conclude that the proposed algorithms can outperform other well-known anti-spam filtering techniques such as support vector machines (SVM), Adaboost and different types of Bayes classifiers.

1. Introduction

During the last century, the communication capabilities of humanity have been widely improved. In particular, one of the most important steps forward in the human communication domain was the genesis and popularization of Internet, emerged from a packet switching military network [1]. From its origins, Internet speed access has been growing steadily as well as the contents shared through Internet. Nowadays, there are millions of new successful web sites sharing multimedia contents (e.g. videos, photos, slides, etc.), popular peer-to-peer networks and crowded social communities. However, malware and spamming activities keep also growing at the same time.

In this context, spam is a term used to designate all forms of unsolicited commercial communication and can be formally defined as an electronic message satisfying the following two conditions: (i) the recipient's personal identity and context are irrelevant because the message is equally applicable to many other potential recipients and (ii) the recipient has not verifiably granted deliberate, explicit, and still-revocable permission for it to be sent [2].

Nowadays, spam is distributed in a widely variety of forms including blog comments, forum entries, e-mail, social networks, video comments, instant messaging bots, pop-up windows, etc. In such a situation, latest studies have shown that the growth of spam traffic is becoming a worrying problem hindering the trouble-free usage of the newest communication technologies [3,4]. In this context, the research community is making valuable efforts in order to filter spam contents during SMTP time [5].

Despite the existence of both general reviews of machine learning (ML) approaches to spam filtering and a good number of related works showing the successful application of well-known classifiers to the spam problem domain [6,7], the specific utilization of rough set theory for spam filtering has not been widely and thoroughly analysed yet. Intuitively, rule-based systems such as those generated through the utilization of rough set theory seem suitable for addressing disjoint concepts as spam and ham (legitimate) classes on spam filtering [8]. As an example, spam about illegal drugs has little in common with spam selling forged university diplomas. In this sense, each different rule should easily address the identification and labelling of each type of message. From another point of view, existing rules can be periodically regenerated from most recent messages to fight against concept drift [8,9]. Moreover, RS-derived rules taking into consideration the existence of specific words inside the body of the messages can be easily translated to different languages to get a multi-language accurate classifier. Finally, the whole rule matching process can also be quickly executed even on slow computers. All these reasons support the idea

* Corresponding author. Tel.: +34 988 387015; fax: +34 988 387001.
E-mail addresses: npdiaz@uvigo.es (N. Pérez-Díaz), drordas@uvigo.es (D. Ruano-Ordás), moncho.mendez@uvigo.es (J.R. Méndez), galvez@uvigo.es (J.F. Gálvez), riverola@uvigo.es (F. Fdez-Riverola).

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.05.024

that RS approaches are especially suitable to address spam filtering related issues.

In this contribution we bring together previous work and some novel alternatives for applying rough sets in the spam filtering domain and carry out a comparative analysis using existing techniques such as SVM, Adaboost and different types of Bayes classifiers. The rest of the paper is structured as follows: Section 2 reviews previous successful content-based techniques applied to the spam filtering domain; Section 3 establishes the mathematical background of rough set theory, summarizes the utilization of previous RS approaches and formalizes three alternatives for applying RS-generated classification rules; Section 4 introduces the protocol used for accomplishing the empirical experimentation while Section 5 presents and discusses the results achieved during the execution of the experiments. Finally, in Section 6 the main conclusions are summarized and future work is outlined.

2. Related work on content-based spam filters

In order to minimize the inconveniences continually imposed by spam on individuals, companies and Internet Service Providers, advances are taking place following three different but complementary alternatives: (i) domain authentication schemes, including support for both designating the servers authorized to send e-mail from each Internet domain and validating these authorizations from e-mail clients, (ii) collaborative approaches, designed to share relevant information about delivered spam messages through networks with the goal of quickly identifying spam e-mails, and (iii) content-based techniques, generally built on the top of successful ML algorithms.

Moreover, the cooperative application of different ML techniques together with their hybridization with domain authentication schemes and collaborative approaches is also an interesting research field because spam filtering could be handled at many points in the network, from the MTA (*Mail Transfer Agent*) to the MUA (*Mail User Agent*). From this perspective, in this section we focus our attention on some previously successful techniques used in the spam filtering industry (i.e. existing Bayes approaches) as well as different classifiers and hybrid alternatives adopted by the scientific community including SVM, artificial neural networks (ANN), artificial immune systems (AIS), *k*-nearest neighbour (K-NN), boosting strategies and case-based reasoning (CBR) systems.

In the spam filtering domain, the Naïve Bayes (NB) classifier is perhaps the most widely used algorithm. Although its independence assumption is over-simplistic, studies in anti-spam filtering have found NB to be very effective [10–12]. In NB-based approaches, information about tokens is stored in a vector of attributes describing the target e-mail. According to several representational issues and the classification criterion, there are different available techniques based on Naïve Bayes including (i) Multivariate Bernoulli NB, (ii) Multinomial NB, (iii) Multinomial NB with term frequency attributes, Multinomial NB with boolean attributes, (iv) Flexible Bayes and (v) the specific implementation of NB from SpamAssassin [13].

Although NB-based classifiers are very popular in the spam filtering industry, the scientific community has applied other classification techniques with different levels of success [5]. In this context, the main characteristic of an SVM is its ability for learning and classification regardless of the number of features (terms) present in each message. This classifier transforms the examples into points in an *n*-dimensional space using non-linear transformations. Then, SVM is able to find the hyperplane that divides points from positive and negative classes maximizing the distance between them [14]. Previous works showing the adequacy of this approach to the spam filtering domain are those by Drucker et al. [15] and Lai [12].

Another well-known technique is the utilization of different types and topologies of ANNs comprising several layers with artificial neurons and weighted directional connections between them. The usage of this approach involves the execution of a training stage in which existing connections update their associated weights [16]. The final output of the model is determined by an activation function which value depends on the adjusted connection weights. As shown in [6,17] the application of ANNs to e-mail classification attained poorer results than Bayesian alternatives.

In addition to ANNs, AIS are models inspired in the biological immune system that use pattern recognition schemes to solve unknown situations [18,19]. Although there are some previous work applying this approach to the spam filtering domain, in general this technique produce a large number of false positive (type I) errors [20,16].

From another perspective, simplicity is the main advantage of K-NN algorithm, in which the classification process is only based on voting schemes from closest training instances. As commented in the work of Lai [12], K-NN alternatives achieved worse performance than other methods for spam classification.

With the goal of taking advantage of base algorithms that learn with an error rate close to 50%, boosting approaches represent ensemble methods used to build accurate classifiers by combining weak learners. From the different available boosting alternatives, Adaboost can be successfully used to classify spam e-mails as reported by the work of Carreras and Márquez [21].

Finally, CBR systems are able to retrieve previous solutions from past problems (stored in the case base) with the goal of adapting them to solve new situations. In this context, some authors have also demonstrated the suitability of applying this method to the spam filtering domain obtaining accurate results [8,9,22].

3. Rough sets for spam filtering

The rough set theory proposed by Pawlak [23], is an attempt to provide a formal framework for the automated transformation of data into knowledge [23,24]. It is based on the idea that any inexact concept (e.g. denoted by a class label) can be approximated from below and from above using an indiscernibility relationship. Pawlak points out that one of the most important and fundamental notions related to the RS philosophy is the need to discover redundancy and dependencies between features [25].

In the context of the spam filtering domain, the main advantages of rough set theory are that it (i) provides efficient algorithms for discovering hidden patterns in data, (ii) identifies relationships that would not be found using statistical methods, (iii) allows the use of both qualitative and quantitative data, (iv) finds the minimal sets of data that can be used for classification tasks, (v) evaluates the significance of data and (vi) generates sets of decision rules from data. This last characteristic will be exploited in the present study.

3.1. Mathematical background

In order to allow the straightforward application of rough set theory to analyse a given e-mail corpus, content from each message should be split into tokens (also referred in this work as terms or words). Accordingly, each e-mail is represented by a set of condition attributes $C = \{a_1, \ldots, a_{n-1}\}$ together with its corresponding message class or decision attribute $D = \{a_n\}$. Therefore, this feature vector containing all the terms existing in the corpus plus the class attribute stands for the attribute set $A = C \cup D = \{a_1, a_2, \ldots, a_{n-1}, a_n\}$. For illustrative purposes, Table 1 shows an example corpus

Table 1
Example corpus representation containing six e-mails ($e_1, \ldots, e_6$) and nine attributes ($a_1, \ldots, a_9$).

		A								
		a_1 Cash	a_2 Address	a_3 Increase	a_4 Guarantee	a_5 Financial	a_6 Date	a_7 Viagra	a_8 Cialis	a_9 Class
U	e_1	0	0	0	1	1	0	1	0	1
	e_2	0	1	1	0	0	0	0	0	0
	e_3	1	1	1	0	0	1	0	0	0
	e_4	0	0	0	1	0	0	0	1	1
	e_5	0	0	0	0	0	0	1	1	1
	e_6	1	1	0	1	0	1	0	0	0

containing a total count of six e-mails ($m = 6$) and nine features ($n = 9$).

In Table 1, e-mails are represented as a feature vector in which the value assigned to each attribute a_i belonging to $\{a_1, \ldots, a_{n-1}\}$ is 1 when the message contains the term a_i, and 0 otherwise. Likewise, the value for the decision attribute (class), a_n, is 1 for spam messages and 0 for legitimate ones.

Therefore, a decision table is a pair $S = (U, A)$, where U is a non-empty and finite set called the *universe* (e.g. all the messages included in the corpus represented in Table 1), and A is the non-empty and finite set of features previously defined. Using the example previously introduced in Table 1, $A = C \cup D = \{cash, address, increase, guarantee, financial, date, viagra, cialis\} \cup \{\text{class}\}$. By means of this characterization we define an equivalence relation, called indiscernibility relation, associated with every subset of attributes $P \subseteq A$. This relation is defined as shown in Expression (1).

$$IND(P) = \{(x, y) \in U^2 : \forall a \in P, a(x) = a(y)\} \tag{1}$$

Expression (1) establishes that, considering the attributes included in P, an email x is indistinguishable from another one y ($x, y \in U$) if, and only if, they share the same values for all the attributes a_i included in P.

By using the indiscernibility relation $IND(P)$ from the set of attributes P, we can define the set of equivalence classes (basic categories) denoted by $U/IND(P)$ or U/P. For instance, and considering $P = \{a_6, a_7, a_8\}$ from Table 1, $U/IND(P) = \{\{e_1\}, \{e_2\}, \{e_3, e_6\}, \{e_4\}, \{e_5\}\}$. Equivalence classes defined through $IND(P)$ are called basic categories of knowledge P, and are denoted by $[x]_{IND(P)}$. Therefore, emails e_3 and e_6 in our example are indistinguishable.

Given a decision table $S = (U, P)$, any set $X \subseteq U$ can be defined by the use of two sets, called lower and upper approximations. The lower approximation, denoted by $\underline{P}X$, is the set of elements in U which can be classified with full certainty as elements of X using the set of attributes P, and is formally represented in Expression (2).

$$\underline{P}X = \cup\{Y \in U/IND(P) : Y \subseteq X\} \tag{2}$$

In the example of Table 1 and using $P = \{a_7, a_7, a_8\}$, the lower approximation of set $X = \{e_4, e_6\}$ is $\underline{P}X = \{e_4\}$. Alternatively, the upper approximation, denoted by $\overline{P}X$, is the set of elements in U which can be possibly classified as elements in X. Expression (3) contains the definition of this concept.

$$\overline{P}X = \cup\{Y \in U/IND(P) : Y \cap X \neq \varnothing\} \tag{3}$$

In the example showed in Table 1, $\overline{P}X = \{e_3, e_6, e_4\}$. A set X is rough regarding P if, and only if, $\overline{P}X \neq \underline{P}X$.

Through the utilization of upper and lower approximations, we can define the positive, negative and borderline regions for a set X as respectively shown in Expression (4).

$$\begin{aligned} POS_P(X) &= \underline{P}X \\ NEG_P(X) &= U - \overline{P}X \\ BND_P(X) &= \overline{P}X - \underline{P}X \end{aligned} \tag{4}$$

Borderline region of X regarding P contains the objects (messages) that cannot be classified as members of X or $-X$ (i.e. spam e-mails) using the attributes of P. In the previous example, the borderline region is $\{e_3, e_6\}$, the positive region is $\{e_4\}$ and the negative region is $\{e_1, e_2, e_5\}$.

In order to build a rule-based information system, we should (i) compute all relative reducts of C (condition attribute set) respect to decision attributes (D) which is equivalent to drop unhelpful columns from the decision table, (ii) remove duplicate rows from decision table and (iii) calculate the relative reducts of categories which is equivalent to remove superfluous values of condition attributes.

Formally, a relative reduct of C regarding D is a subset of attributes $RED \subseteq C$ having the following properties: (i) the classification induced by RED is the same as computed using C, therefore $POS_{RED}(D) = POS_C(D)$ and (ii) the attribute set RED is minimal, therefore $POS_{RED-\{a\}}(D) \neq POS_{RED}(D)$ for any attribute a included in RED, where $POS_C(D)$ is defined in Expression (5).

$$POS_C(D) = \bigcup_{X \in U/IND(D)} \underline{C}X \tag{5}$$

For instance, using $C = \{a_6, a_7, a_8\}$, there is only one reduct $RED = \{a_7, a_8\}$. This step is useful to remove superfluous features from input data, therefore reducing the computational overhead.

During the second stage, we delete rows having the same attribute values. In the third stage, we try to find the lowest amount of values for attribute conditions to successfully classify samples. To this end, for each row (decision rule) from the dataset (represented in Table 1), we define $F = \{x_1, \ldots, x_n\}$ as a family of sets where $x_i \subseteq U$ and a subset $Y \subseteq U$ such that $\cap F \subseteq Y$. For each family F, we should search relative reducts of categories considering that the family $H \subseteq F$ is a Y-reduct of $\cap F$, if H is minimal in $\cap F$ and $\cap H \subseteq Y$, where Y is the family associated to the values of decision attribute (D). For instance, considering the reduct $RED = \{a_7, a_8\}$, the family F for the first row is $F = \{[1]a_7, [1]a_8\} = \{\{e_1, e_5\}, \{e_1, e_2, e_3, e_6\}\}$. Using $F = \{[1]a_7, [1]a_8\}$, and considering $Y = [1]a_9 = \{e_1, e_4, e_5\}$, the Y-reduct computed is $H = \{[1]a_7\}$ and therefore, the value of attribute a_8 for the first rule (row) is superfluous. As a result, we can generate the rule "IF $a_7 = 1$ THEN $a_9 = 1$".

By the execution of this three-stage process from the decision table showed in Table 1, and considering all the attributes included in it, we can generate the rules showed in Table 2.

In Table 2, attributes without an assigned value for a given rule (marked with a hyphen) indicate the irrelevance of their values with respect to the target consequent. As shown in Table 2, only

Table 2
Discriminatory rules generated using rough sets applied to the input data included in Table 1.

	a_1 Cash	a_2 Address	a_3 Increase	a_4 Guarantee	a_5 Financial	a_6 Date	a_7 Viagra	a_8 Cialis	a_9 Class
r_1	–	–	–	–	–	–	1	–	1
r_2	–	–	1	–	–	–	–	–	0
r_3	–	–	–	–	–	1	–	–	0
r_4	–	–	–	–	–	–	–	1	1

four rules using four attributes are finally generated from the data included in Table 1.

Starting from this basic conceptualization initially proposed by Pawlak, several extensions have been later defined. Among these alternatives, the most outstanding is the VPRS (*Variable Precision Rough Set*) model, which is a generalization that introduces a controlled degree of uncertainty within its formalism by an additional parameter ϕ [26,27].

3.2. Preliminary studies

Following the abstraction described in the previous section, some VPRS model implementations have been initially tested in the spam filtering domain reporting promising results. In this context, the work of Glymin and Ziarko [28] presented a preliminary feasibility study of using a VPRS model to classify spam e-mails. Their proposal utilizes the rough set model for processing a set of pre-classified messages and constructing a decision table. For experimental purposes, the authors work with a corpus built by compiling private messages from distinct hotmail accounts. Using a two-year period e-mail corpus for the training stage, they use the resulting decision table to predict e-mail categories.

In [29], Zhao and Zhu proposed a novel scheme to perform e-mail classification by using another VPRS model able to label incoming messages as spam, legitimate or suspicious. Their proposal is compared with the popular Naïve Bayes classifier commonly used in the spam filtering industry. The dimensionality of the training corpus was reduced to eleven attributes by using the forward selection method [30].

Along the same lines, the works of [31–33] investigated new RS-based schemes for e-mail classification using three message categories (i.e.: spam, no spam and suspicious). Genetic algorithms were used in [31,32] for computing the rough sets reducts. Zhao and Zhang [31] also employed the forward selection method for limiting the number of attributes in their experiments. The authors in [33] finally stated that messages labelled as undecided needed further examination. In all the approaches, results using rough sets were compared with those obtained by the Naïve Bayes algorithm in order to determine the effectiveness of their proposals.

From a different perspective, the works [34–36] offer another view of performing e-mail classification. In their respective proposals, the authors considered separate methods based on sharing rules between servers that are always updated. In such a situation, when a rule becomes obsolete it may be definitely eliminated. Moreover, every time a rule is duplicated the scores are subsequently updated. In [34] rules are generated by applying the classical rough set model, but [35] performs statistical operations with data mining and [36] combines rough set and genetic algorithms for automatic rule generation.

Taking into consideration the state-of-the-art in spam filtering domain using rough sets, Table 3 presents a comparative study of the different strategies used in previous works for validating the proposed models. As Table 3 shows, most of them do not provide a comparison with other techniques nor use stratified fold-cross validation schemes for sustaining achieved results. Moreover, several works try to argue the benefits of classifying a message as suspicious. Nevertheless, assigning this label to a given message is equivalent to classifying the e-mail as legitimate, because the final user should read the text and take a decision manually.

From the analysis of Table 3, it is clear that a comprehensive study has not been carried out yet in order to systematically test the suitability of using rough sets in the spam filtering domain. In the present work we will compare the performance achieved by different rough sets approaches and other well-known content-based filtering techniques from the spam filtering industry and the scientific community, by using a raw and large e-mail corpus working with different dimensionalities for the input dataset and following a fold-cross validation scheme. Next, we introduce and formalize three different rule execution schemes commonly used in the rough sets domain for classification purposes.

3.3. Strategies for spam filtering using rough sets

The main differences between the work reported in the previous section (using VPRS models) and the different alternatives explored in the present work are based on the utilization of uncertainty and, specifically, the classification of e-mails belonging to the boundary region (messages that do not match any rule).

To this extent, in the work of [28] those messages from the boundary region remain unclassified while in [29,31,32,36] the final classification uses a third category, named *suspicious*. In [33] a further exploration is required to classify ambiguous messages and finally, in [34] those e-mails are automatically classified as legitimate. From a final and conservative user perspective, the last two approximations are the most reliable because *suspicious* e-mails should be manually catalogued (which is equivalent to classify them as legitimate) but also an accurate classification could be achieved by performing further computation.

In order to cope with this situation, in this work we propose a novel alternative to address the classification of e-mails belonging to the boundary region (messages that do not match any rule). Instead of using the above mentioned approaches, we have selected those rules that partially match the target message and implemented three different inference schemes to use them for further classifying ambiguous e-mails. Despite the simplicity of these methods to propose a final classification for e-mails belonging to the boundary region, they have not been tested in any previous work. In this sense, we believe that their utilization could provide a simple scheme to classify ambiguous messages preventing the utilization of other complex techniques with high computational requirements.

Our proposal is based on finding the best matching rules (those having the greatest amount of matches between rule and e-mail attribute values) and applying them using three different heuristics: (i) MFD, able to find the *most frequent decision* among rules covering the target e-mail with a minimal distance (Fig. 1), (ii) LNO, returning a decision that covers the *largest number of objects* with the minimal distance among rules from a given message (Fig. 2) and (iii) LTS, providing the decision with the *largest total strength* among rules with minimal distances from a given e-mail (Fig. 3).

As we can see from Fig. 1, we introduce for this pseudo-code the function *dissimilarity_attributes* that stands for the number of

```
00 FUNCTION MFD (INPUT rules: ARRAY OF rule, INPUT message,
01      INPUT possible_solutions: ARRAY OF T_SOLUTION): T_SOLUTION;
02
03 VARIABLES
04     ARRAY OF INTEGER diffs[1..size(rules)];
05     ARRAY OF INTEGER counts[1..size(possible_solutions)];
06     INTEGER minDiff, index, i;
07
08 BEGIN
09
10     /*card: finds the number of elements in a set
11       size: get the size of the array */
12     FOR i:=1 TO size(rules) DO
13        diffs[i] := card(dissimilarity_attributes(rules[i],message));
14
15     minDiff := attribute_number(message);
16     FOR i:=1 TO size(rules) DO
17       IF (diffs[i] < minDiff) THEN minDiff := diffs[i];
18
19     FOR i:=1 TO size(possible_solutions) DO
20       counts[i] := 0;
21
22     FOR i:=1 TO size(rules) DO
23       IF diffs[i] = minDiff THEN
24       BEGIN
25          index := find(rules[i].solution, possible_solutions);
26          counts[index] := counts[index]+1;
27       END;
28
29     /*find_max: finds the greatest value in a vector*/
30      RETURN possible_solutions[find_max(counts)];
31
32 END MFD;
```

Fig. 1. Pseudo-code representation of the MFD algorithm.

different attributes between a rule and a given e-mail, and the *attribute_number* function that computes the number of attributes in a message or a rule.

In the case of the LNO algorithm (Fig. 2), we use the function *cover* that computes the cover set of a rule (set of e-mails from training data that matches with the rule).

As it can be seen from Fig. 3, the LTS algorithm is a simplification of the LNO scheme in which there is no need to work with cover sets but an e-mail can be added more than once to compute the rule strength.

In order to provide a better description of these algorithms, they can be easily introduced as voting schemes. Thus, MFD simulates a voting heuristic, where each selected rule uses its class to emit a vote. Nevertheless, using LNO and LTS schemes, the vote is emitted by training messages instead of rules. Following the LNO scheme, each learned message included in any of the coverage sets from

```
00 FUNCTION LNO (INPUT rules: ARRAY OF rule, INPUT message,
01      INPUT possible_solutions: ARRAY OF T_SOLUTION): T_SOLUTION;
02
03 VARIABLES
04     ARRAY OF INTEGER diffs[1..size(rules)];
05     ARRAY OF SET covers[1..size(possible_solutions)];
06     ARRAY OF INTEGER cover_sizes[1..size(possible_solutions)];
07     INTEGER minDiff, index, i;
08
09 BEGIN
10     FOR i:=1 TO size(rules) DO
11        diffs[i] := card(dissimilarity_attributes(rules[i],message));
12
13     minDiff := attribute_number(message);
14     FOR i:=1 TO size(rules) DO
15       IF (diffs[i] < minDiff) THEN minDiff := diffs[i];
16
17     /*create_empty_set: returns an empty set*/
18     FOR i:=1 TO size(possible_solutions) DO
19     BEGIN
20       covers[i] := create_empty_set();
21       cover_sizes[i] := 0;
22     END;
23
24     /*union: computes the union of two sets*/
25     FOR i:=1 TO size(rules) DO
26       IF diffs[i] = minDiff THEN
27       BEGIN
28         index := find(rules[i].solution, possible_solutions);
29         covers[index] := set_union(covers[index],cover(rules[i]));
30         cover_sizes[i] := card(covers[index]);
31       END;
32
33     RETURN possible_solutions[find_max(cover_sizes)];
34
35 END LNO;
```

Fig. 2. Pseudo-code representation of the LNO algorithm.

```
00 FUNCTION LTS (INPUT rules: ARRAY OF rule, INPUT message,
01      INPUT possible_solutions: ARRAY OF T_SOLUTION): T_SOLUTION;
02
03 VARIABLES
04     ARRAY OF INTEGER diffs[1..size(rules)];
05     ARRAY OF INTEGER total_strengths[1..size(possible_solutions)];
06     INTEGER minDiff, index, i;
07
08 BEGIN
09     FOR i:=1 TO size(rules) DO
10        diffs[i] := card(dissimilarity_attributes(rules[i],message));
11
12     minDiff := attribute_number(message);
13     FOR i:=1 TO size(rules) DO
14       IF (diffs[i] < minDiff) THEN minDiff := diffs[i];
15
16     FOR i:=1 TO size(possible_solutions) DO
17       total_strengths[i]=0;
18
19     FOR i:=1 TO size(rules) DO
20       IF diffs[i] = minDiff THEN
21       BEGIN
22          index := find(rules[i].solution, possible_solutions);
23          total_strengths[index] := total_strengths[index] +
24                card(cover(rules[i]));
25       END;
26
27     RETURN possible_solutions[find_max(total_strengths)];
28
29 END LTS;
```

Fig. 3. Pseudo-code representation of the LTS algorithm.

selected rules votes one time. However, using LTS alternative, each learned message votes as many times as found in the coverage sets of the selected rules.

By using the previous conceptualization, we include a practical example showing the usage of MFD, LNO and LTS schemes applied to a fragment of text extracted from a real e-mail "*Your source online for Viagra, Cialis, Levitra and much more! See yourself saving* $$ *and hassle. Discreet and customer satisfaction is guaranteed*". As starting point, we have generated the rules included in Table 4 by applying RS theory.

As we can see from our example, there is no rule from Table 4 that full matches the original text. Thus, those rules with the shortest distance (highlighted in Table 4) will be applied to compute a final classification using MFD, LNO and LTS heuristics as following:

i. From the set of closest rules, two of them designate ham contents and three indicate spam topics. Using the MFD heuristic and taking into consideration that majority of votes are spam, the target e-mail will be classified as spam.
ii. Following the LNO heuristic, each different message from the coverage of rules should be taken into consideration. This method leads to the following situation:
    set_messages_from_coverare_of_rules[spam = 1] = $\{e_1, e_2, e_3, e_8\}$
    set_messages_from_coverage_of_rules[spam = 0] = $\{e_4, e_5, e_6\}$
    As there is more spam messages than legitimate ones, the target e-mail will be classified as spam.
iii. Finally, using the LTS heuristic we have the following situation in which messages e_4 and e_5 vote twice because they are covered by two different rules.
    list_messages_from_coverare_of_rules[spam = 1] = $[e_1, e_2, e_3, e_8]$
    list_messages_from_coverage_of_rules[spam = 0] = $[e_4, e_5, e_6, e_4, e_5]$
    Therefore, under this scheme, the target e-mail will be classified as legitimate.

4. Experimental protocol

With the goal of testing the three proposed rule execution schemes that use rough sets for spam filtering, we have developed a specific protocol to measure their global accuracy. In this section, we include a detailed description concerning the corpus selection, several preprocessing and representational issues, the validation strategy and different measures for evaluating the final performance.

4.1. Corpus selection, preprocessing and representational issues

A relevant decision when addressing the execution of experiments in the spam filtering domain is the selection of an appropriate corpus for benchmarking purposes. Despite privacy issues, an appreciable number of corpora like SpamAssassin,[1] LingSpam[2] or Spambase[3] can be freely downloaded from Internet. However, most of the publicly available corpuses have been firstly tokenized and later preprocessed by assigning a unique ID for each term in the corpus. Therefore, some relevant information included in these messages can be lost during this preprocessing step.

Moreover, in several works on spam filtering using rough sets [28,29,31] Spambase was selected as input dataset due to its low dimensionality (see Table 5). Nevertheless, we believe that working with such a small corpus (containing only 57 attributes from 4601 messages) is inappropriate and can lead to misleading or inconsistent conclusions.

In this work, we use the well-known SpamAssassin corpus containing 9332 messages from January 2002 up to and including December 2003 (see Table 5). This decision is supported by a good number of previously successful research papers on spam filtering domain [37–39].

From another point of view, the internal structure of messages used during both training and classification stages remains a relevant question when designing an experimental protocol. Messages are usually denoted as a vector $\vec{t} = \langle t_1, t_2, \ldots, t_p \rangle$ containing numerical values that stand for certain message attributes. The selected features usually represent the presence or absence of a term in the message. This idea has been taken from the vector space model used in information retrieval studies [40,41].

[1] Available at http://www.spamassassin.org/publiccorpus/.
[2] Available at http://labs-repos.iit.demokritos.gr/skel/i-config/downloads/.
[3] Available at http://www.ics.uci.edu/~mlearn/MLRepository.html.

Table 3
Comparative study of previous successful approaches using rough sets.

	Corpus description			Train-test balance	Number of considered features	Fold-cross validation	Assigned labels S: spam L: legitimate SS: suspicious	Comparative level	
	Repository	#Spam	#Ham					Other techniques	Metrics
Zhao and Zhang [31]	Spambase	1813	2788	2/3–1/3 4/5–1/5	11	No	S|L|SS	NB	Recall, precision and accuracy
Zhao and Zhu [32]	Spambase	1813	2788	2/3–1/3 4/5–1/5	10	No	S|L|SS	NB	Effectiveness in classification
Zhao and Zhu [29]	Spambase	1813	2788	2/3–1/3	11	No	S|L|SS	NB	Effectiveness in classification
Glymin and Ziarco [28]	Email server	2940	1260	–	58	No	S|L	No	Accuracy and error
Chiu et al. [36]	Email server	500	3794	–	11	No	S|L|SS	No	Recall, precision accuracy and miss rate
Lai et al. [35]	Email server	54602	5921	–	–	No	S|L	No	Precision and miss rate
Zhoy et al. [33]	Spambase	1813	2788	3834–767	58	No	S|L|SS	NB	Recall, precision and accuracy
Lai et al. [34]	Email server	24546	2331	4794	–	No	S|L	No	Recall, precision, accuracy and miss rate

An important procedure for obtaining an informative message representation is feature extraction. In this context, feature identification can be carried out by using a variety of common lexical tools, like the tokenization of the text extracted from e-mails into words. At first glance, it could appear to be a simple tokenizing task guided by certain characters working as word separators. However, at least the following specific cases have to be considered with special care: hyphens, punctuation marks and the case of the letters (lower and upper case) [42]. In the spam domain, punctuation marks together with hyphenated words are among the best discriminating attributes for any given corpus, partly because they are more common in spam messages than legitimate ones.

In our experimentation, text for tokenizing was extracted from both the e-mail body and attachments. In order to correctly handle the diverse formats existing for the attached files, we used different techniques in each case taking into account the *content-type* header information. Therefore, HTML code was translated into text/plain using the HTMLParser[4] tool, images were processed using the Asprise OCR[5] software and the text inside pdf documents was extracted using the PDFBox[6] package. We tokenized the text extracted from e-mails using only blank spaces in order to preserve the helpful noise originally contained in the messages. Moreover, all identified words were converted to lower case.

Once the tokenizing step has been accomplished, stopword removal (which drops articles, connectives and other words without semantic content) and/or stemming (which reduces distinct words to their common grammatical root) are commonly applied to refine the final list of identified terms [43]. In the present work we have used only stopword removal as it has been shown to be the best choice for the majority of systems [43].

In accordance with the results of previous work in the same domain [40,41], we have selected Information Gain (IG) as the appropriate feature selection method for spam filtering. This method is based on firstly computing the IG measure for each identified term by using the equation given in Expression (6), and then selecting those terms having the highest value.

$$IG(t) = \sum_{c \in \langle l,s \rangle} P(t \wedge c) \cdot \log \frac{P(t \wedge c)}{P(t) \cdot P(c)} \tag{6}$$

For the experiments carried out in the present work, a binary representation has been used in which a feature with an assigned value of 1 means that the word is part of the content of the e-mail, and a value of 0 denotes that the term is not in the e-mail when using Adaboost, SVM, Naïve Bayes and RS-based models. In the case of Flexible Bayes classifier, we use a frequency based representation.

From a different perspective, in [44] we exhaustively studied the effects of changing the input vector dimensionality for different well-known anti-spam filters. In the present work, we adopt the optimum dimensionality for each classifier according to this study: 2000 features for SVM and Flexible Bayes, 1400 features for Naïve Bayes and 700 features for Adaboost. In order to test the suitability of RS-based models we have initially selected 100 and 200 binary features. This decision was adopted because greater dimensionalities for rough sets consume a lot of CPU and computation time.

4.2. Measures for analysing the performance of classifiers

Given the particular nature of the spam filtering problem, several specific measures have been applied with the goal of assessing

[4] HTMLParser is available for download at http://htmlparser.sourceforge.net/.
[5] Asprise OCR can be downloaded at http://asprise.com/product/ocr/.
[6] PDFBox is available for download at http://www.pdfbox.org/.

Table 4
Example rules generated using RS theory extracted from a real e-mail.

Rule	Distance	Cover.	#Cover.
viagra = 1 ∧ cialis = 1 ∧ buy = 1 ⇒ spam = 1	1	{e1}	1
levitra = 1 ∧ cialis = 1 ∧ drugs = 1 ∧ employment = 0 ⇒ spam = 1	1	{e2,e3}	2
satisfaction = 1 ∧ source = 1 ∧ buy = 0 ∧ employment = 1 ⇒ spam = 0	1	{e4,e5,e6}	3
cialis = 1 ∧ research = 1 ∧ buy = 0 ∧ employment = 0 ⇒ spam = 0	1	{e4,e5}	2
cialis = 1 ∧ online = 1 ∧ research = 1 ∧ guaranteed = 1 ⇒ spam = 1	1	{e8}	1
research = 1 ∧ job = 1 ∧ online = 0 ∧ guaranteed = 1 ⇒ spam = 0	3	{e6,e7}	2
research = 1 ∧ cialis = 1 ∧ online = 0 ∧ guaranteed = 1 ∧ satisfaction = 0 ⇒ spam = 0	3	{e8,e9}	2

the relevance of existing classifiers. Taking into consideration previous work, we have selected some known measures including the percentage of false positives (FP), false negatives (FN) and correct classifications.

Moreover, in order to evaluate the classifiers from a more realistic perspective, we have also used *recall* and *precision* measures. *Recall* is able to estimate the ability of identifying spam e-mails (higher values imply more spam detected) while *precision* evaluates the ability of preventing positive errors (higher values imply lower false positive rates). The equations that govern *recall* and *precision* are introduced in Expression (7).

$$recall = \frac{nspam - fn}{nspam}, \quad precision = \frac{nspam - fn}{nspam - fn + fp} \tag{7}$$

where *fn* and *fp* represent the number of false negatives and false positives respectively, and *nspam* stands for the number of spam e-mails in the training corpus.

Previous work on spam filtering [45] also suggests the usage of *batting average* in order to better determine the connection between *recall* and *precision*. This measure is defined taking into consideration *hitrate*/*strikerate* ratio, where the former represents the number of detected spam messages and the latter, the false positive average. This criterion has been also included in our experimental protocol.

In the same line, *f-score* represents another interesting measure able to combine *recall* and *precision* values [46]. *F-score* ranges in the interval [0, 1] and its value is 1 only if the number of FP and FN errors generated by the filter is 0. Expression (8) shows how *f-score* is calculated.

$$f\text{-}score = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{8}$$

More recently, a new measure called *balanced f-score* or $f\text{-}score_\beta$ has also been introduced [47]. As in the previous case, *balanced f-score* combines *precision* and *recall* but considers their different importance. If $\beta = 1$, then *precision* and *recall* have the same weight, so $f\text{-}score = f\text{-}score_\beta$. If $\beta < 1$, then *precision* is more important than *recall*. Otherwise, *recall* has more weight. $f\text{-}score_\beta$ can be computed as Expression (9) shows. In the present work we have included $f\text{-}score_\beta$ using 1, 1.5 and 2 as possible values for the β parameter, therefore also representing *f-score* measure.

$$f\text{-}score_\beta = \frac{(\beta^2 + 1) \cdot precision \cdot recall}{\beta^2 \cdot precision + recall} \tag{9}$$

In the work of [11] the *Total Cost Ratio* (TCR) measure was also used. This defines the λ parameter to ponder false positive and false negative costs. In this context, an FP error is λ times more costly than a FN one, representing the problems caused to the end-user when a legitimate e-mail is incorrectly labelled as spam and deleted. Default values of λ are 1, 9 and 999. Expression (10) shows how to compute TCR.

$$TCR = \frac{nspam}{\lambda \cdot fp + fn} \tag{10}$$

where *fn* and *fp* represent the number of false negatives and false positives, respectively, and *nspam* stands for the number of spam e-mails in the training corpus. In the present work we have computed TCR using λ values of 1, 9 and 999.

5. Experimental results

With the goal of correctly assessing the effectiveness of the three different strategies for spam filtering using rough sets, we designed and executed the previous experimental protocol comparing the results with those obtained by four well-known successful anti-spam techniques: SVM (using Sequential Minimum Optimization implementation with a degree-1 polynomial kernel), Adaboost (using Decision Stumps as weak classifier with 150 boosting iterations), Multinomial Naïve Bayes (with boolean attributes) and Flexible Bayes. In order to use a publicly available implementation of the selected rule execution schemes, we have used the rough set library provided by Warsaw University of Technology [48]. Moreover, to guarantee the quality of our experimental results, we have used a 10-fold stratified cross-validation scheme [49] for all the experiments over the SpamAssassin corpus.

We have structured the discussion of achieved results from two different but complementary points of view: on one hand, presenting and comparing previously commented measures and, on the other hand, exploring relevant issues related with the real deployment process of the analysed alternatives.

5.1. Model comparison

This subsection analyses the accuracy achieved while using all the filtering techniques. First, in order to obtain a preliminary impression, Fig. 4 shows the percentage of correct classifications (%OK), false positive (%FP) and false negative (%FN) errors.

As can be easily seen from Fig. 4, rough set approaches (specially the MFD algorithm) achieve the highest number of correct classifications (98.7% when using 100 features and 98.8% with 200 attributes) finding the most frequent decision among rules that cover target e-mails with a minimal distance. Fig. 4 also shows that the percentage of FP errors generated by RS alternatives is only improved by SVM and Flexible Bayes classifiers. Finally, Fig. 4 also evidences that rough set approaches present the lowest FN error rate (always 0%).

Table 5
Description of publicly available corpora.

Corpus	Spam messages	Ham messages	Total messages	S:L ratio	Format	Preprocessing
SpamAssassin [34]	2381 (25.51%)	6951 (74.49%)	9332	0.35	RFC 822	Not preprocessed: all text available
Spambase [35]	1813 (39.40%)	2788 (60.60%)	4601	0.65	Feature vectors	Tokenized + feature selection: only 57 available attributes

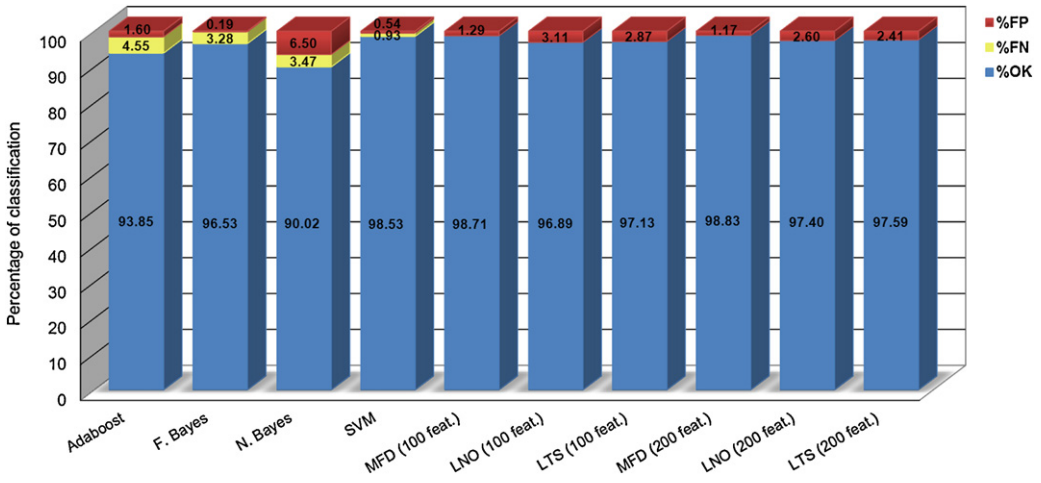

Fig. 4. Percentage of correct classifications, FP errors and FN errors from validation over the SpamAssassin corpus.

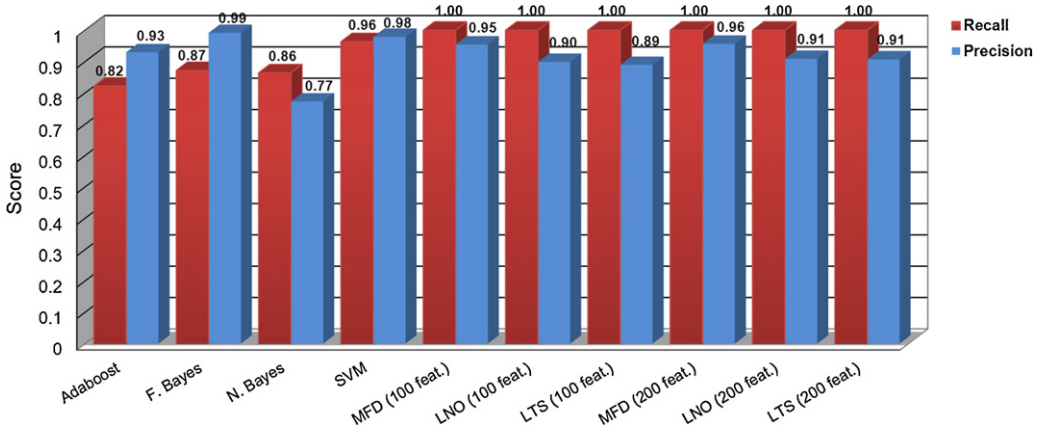

Fig. 5. Recall and precision values for the analysed models.

This preliminary model behaviour observed through the comparison of error percentage values can be further analysed using *recall* and *precision* measures. The results are included in Fig. 5.

From Fig. 5, it can be stated that the *recall* achieved by the rough sets alternatives is greater than other proposals, while maintaining an acceptable precision level (better than Naïve Bayes and Adaboost and slightly lower than SVM and Flexible Bayes).

Moreover, we have combined *recall* and *precision* using both *f-score* and *balanced f-score* in order to yield a unified measure of the filtering performance achieved by all the analysed techniques. The results are shown in Table 6.

As we can realize from Table 6, global results for the *balanced f-score* achieved by using rough sets are better than those obtained when using other popular anti-spam filters. Moreover, the advantage of applying RS-based techniques is greater when using higher values of the β-parameter, partly because the implicit increment of *recall* significance.

Table 6
Balanced *f*-score results using different β values.

	$\beta = 1$ (*f-score*)	$\beta = 1.5$	$\beta = 2$
Adaboost	0.87	0.85	0.84
Flexible Bayes	0.93	0.91	0.89
Naïve Bayes	0.82	0.83	0.84
SVM	0.97	0.97	0.97
MFD (100 features)	0.98	0.99	0.99
LNO (100 features)	0.95	0.97	0.98
LTS (100 features)	0.94	0.96	0.98
MFD (200 features)	0.98	0.99	0.99
LNO (200 features)	0.95	0.97	0.98
LTS (200 features)	0.95	0.97	0.98

In general, a high spam *recall* value indicates a low FN error rate, and a high spam *precision* value implies a low FP error rate. These two parameters are straightforward to understand model effectiveness regarding to different type of misclassifications, but do not provide a simple measure to compare proposals taking into consideration the associated cost generated by errors in a real environment. TCR score is therefore introduced to combine information about errors and their associated costs, where higher TCR values indicate better performance of the models.

Now, let us assume that FP errors are λ times more costly than FN errors, where λ depends on the usage scenario. Three different usage scenarios are used in our experiments. In the first one, the filter flags messages when it suspects them to be spam (without removal). In this case, $\lambda = 1$. The second scenario assumes that messages classified as spam are returned to the sender. In this scenario $\lambda = 9$ is considered, that is, mistakenly blocking a legitimate message was taken to be as bad as letting 9 spam messages pass the filter. In the third scenario messages classified as spam are deleted automatically without further processing. Now $\lambda = 999$ is used. Fig. 6 shows the results taking into account the TCR score and varying the λ parameter as commented above.

From Fig. 6 we can see that TCR values achieved by using rough sets are clearly higher when using low λ values. This fact reminds the existence of a small increment in FP errors when using RS

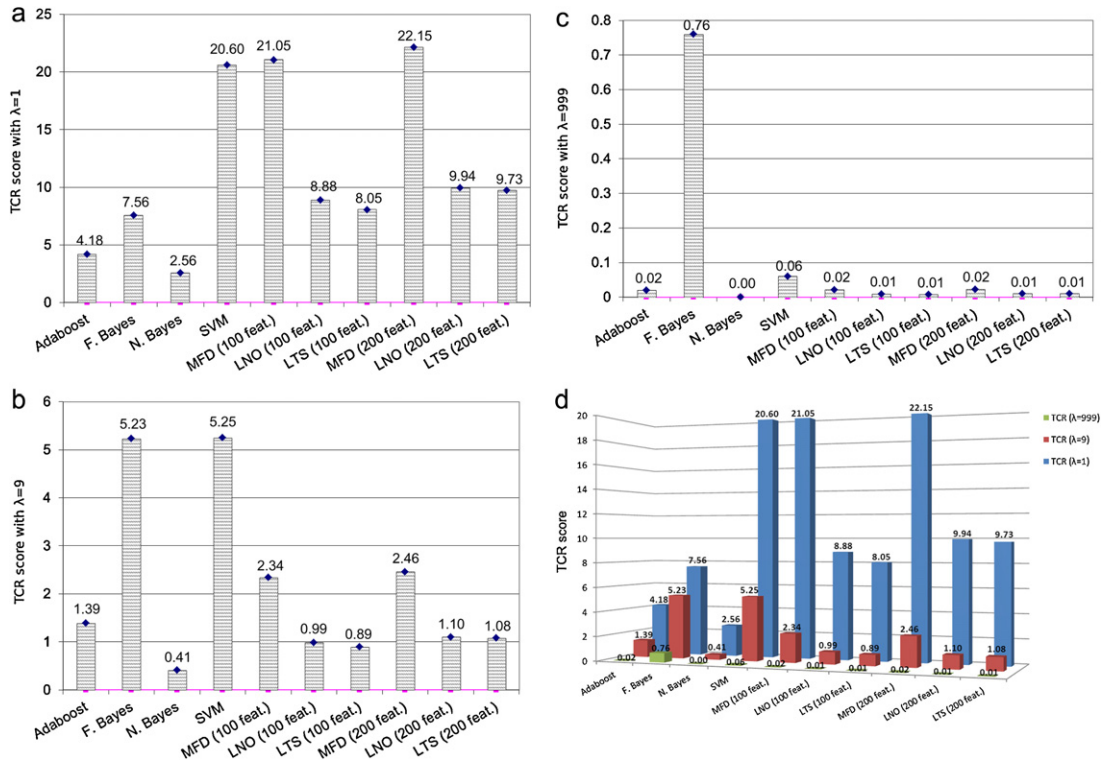

Fig. 6. TCR values for the analysed models varying the λ parameter over the SpamAssassin corpus.

approaches (less *precision*), which generates higher costs in environments that should prevent the occurrence of FP errors. This *precision* issue can be confirmed when analysing *strike rate* from *batting average* evaluations in Table 7.

As shown in Fig. 5 and also corroborated by Table 7, Flexible Bayes classifier achieves a first-class strike rate (ability to avoid FP errors) while rough sets alternatives are able to attain the highest sensibility (facility of detecting spam messages). This fact entails that the application of RS-based schemes should be limited to filtering scenarios, where FP errors are acceptable until they get improved.

5.2. Practical applications and limitations

During the experimental stage, we found some limitations while using RS-based approaches to filter spam. In order to introduce and comment these issues, this subsection presents a comprehensive explanation of practical application scenarios and limitations of the proposed methods.

The utilization of RS-based schemes to spam filtering purposes is suitable to address the large amount of subjects contained in spam messages. Current spam e-mails advertise watch replicas, different kinds of drugs to improve sexual function, fake university diplomas and, after Fukushima nuclear disaster, drugs to fight radiation side effects. Each of these themes can be easily addressed by a RS-generated rule able to handle most advertisements among spam.

Despite the potential application of rough sets to manage different spam topics, we should keep in mind the awkward effects of concept drift. In this context, the purpose of spam distribution depends on the advertisement needs of each moment. As an example, spam deliveries related to drugs able to fight radiation took place only during a few months after the Fukushima nuclear disaster. In the same line, the use of different types of spam obfuscation techniques (e.g.: character substitutions i by ¡, use of spaces, etc.) introduces similar difficulties. This kind of workaday issues cannot be easily modelled. In such a situation, the regeneration of the rule sets is the most effective way to fix the adverse effects of concept drift. Therefore, rule sets should include an expiration date to limit an excessive performance decrement. Beyond the expiration date, rule sets should be computed again to incorporate the knowledge included in latest messages.

While executing initial experiments using all the features belonging to the SpamAssassin corpus, we were not able to generate a rule set after a reasonable time. Despite RS can be used to reduce the problem dimensionality, the computational requirements when using large attribute datasets are unaffordable, even with a moderate computer infrastructure. As we can see from Table 5, the dimensionality of the dataset used in previous works (Spambase) is under 100 features. In our experiments using the SpamAssassin corpus, after a stopword removal process applied to the text from e-mail subject and body, we found 128,972 different words (potential features). This situation suggested the possibility of applying a feature ranking algorithm to reduce the problem dimensionality. According with results from previous works [40,41] we executed an IG feature selection process. After

Table 7
Batting average evaluation for the different analysed techniques.

	Hit rate	Strike rate
Adaboost	0.82	0.02
Flexible Bayes	0.87	0
Naïve Bayes	0.86	0.09
SVM	0.96	0.01
MFD (100 features)	1	0.02
LNO (100 features)	1	0.04
LTS (100 features)	1	0.04
MFD (200 features)	1	0.01
LNO (200 features)	1	0.03
LTS (200 features)	1	0.03

several runs with different dimensionalities, we found that using more than 1000 features requires an unaffordable computational cost while using more than 200 does not contribute to achieve a significant performance increment. In a real deployment scenario, where filtering throughput is essential, rule set should not be computed in the same infrastructure used for executing the spam filtering software. Nevertheless, the execution of rules is a fast and lightweight process, so it can be successfully included in the spam filtering software.

Filtering rules are formed as a combination of simple conditions that stands for the presence of several terms in the incoming e-mail. This building scheme allows their easy translation into different languages and can be directly used to find misspelled words and noise introduced by spammers. Therefore, these possibilities should be addressed in future research works.

Finally, we found that RS-based approaches present a high sensibility to detect spam messages while preserving a reasonable specificity level (ability to avoid FP errors). In such a situation, they should be preferably applied in environments when the cost of FP errors is low as, for example, for hotmail accounts used to share jokes with friends.

6. Conclusions and future work

This work presents a comprehensive study about the utilization of rough sets as main classifier for industrial spam filtering. We have introduced and analysed several proposals using rough sets together with a practical discussion about their usage, weaknesses and strengths. Examining the state-of-the-art, we have found that most of previous works present only modest analyses using corpora with an inadequate preprocessing. In this context, we have carried out a discrimination analysis using a large, raw and unpreprocessed corpus provided by the SpamAssassin team.

From all the experiments carried out, we have reached some interesting conclusions discussed below.

i. As indicated by the results obtained, RS-based schemes are always a suitable replacement for Naïve Bayes classifier and Adaboost technique.
ii. Moreover, the proposed RS-based approaches are good substitutes for SVM and Flexible Bayes classifiers in environments in which the cost of FP errors is low.
iii. MFD heuristic achieves the best accuracy when compared to LNO and LTS.
iv. Despite the execution of rules is very fast, the utilization of additional hardware to compute rule set is recommended in order to perform an efficient deployment of RS-based filtering techniques.
v. In order to minimize the time required for the rule generation stage, we found that an adequate feature ranking and selection algorithm should be used.
vi. In order to fix the effects of concept drift, rules should be regularly (re)generated.
vii. As our benchmarking results show, RS-based alternatives make the most of knowledge extracted from the corpus by attaining very good results with a reduced dimensionality dataset (100–200 features). Nevertheless, some popular techniques require a large amount of features to achieve good accuracy (2000 for SVM and Flexible Bayes, 1400 for Naïve Bayes and 700 for Adaboost).
viii. Concerning the SpamAssassin corpus (used in our experiments), it includes 250 hard ham e-mails representing the 2.68% of all the available messages. This kind of e-mail is very close to spam, constituting the main cause of the existence of FP errors in the analysed RS alternatives. Caused by the low input dimensionality required for reducing the CPU consumption in the available rough set implementation, most hard ham messages were located at the boundary region, being classified as spam due to the low distance with spam e-mails. Nevertheless, we believe that high dimensionality input is possible by optimising the RS library in order to adapt it to the spam filtering domain (with one single decision attribute and only two possible values for all the features).

Although RS-based techniques show great performance in spam filtering, new reduction methods and rule execution schemes should be designed in order to achieve superior results. We also believe in the necessity of improving the rough sets library by including some domain constraints in order to better adapt this technique to the spam filtering domain.

Additionally, some work should be also carried out to address noise handling. Thus, we think that features should be regular expressions instead of words to handle noise and common misspellings. Keeping in mind this idea, some work from other domains (such as SNA pattern identification) can be successfully used to address this issue.

Finally, another interesting application would be the codification of a fully functional implementation of rough sets for use in conjunction with other successful techniques inside a filter development framework (e.g. SpamAssassin).

Acknowledgments

This work was partially funded by the projects *Optimización de sistemas antispam* (08TIC041E) and *Deseño e validación de filtro antispam intelixente baseado en análise contextual ponderado do contido das mensaxes* (09TIC028E) from Xunta de Galicia.

References

[1] L. Roberts, The evolution of packet switching, Proceedings of the IEEE 66 (11) (1978) 1307–1313, http://www.packet.cc/files/ev-packet-sw.html.
[2] SpamHaus Project Organization, The SpamHaus Project, 1998. http://www.spamhaus.org/.
[3] P. Bueno, T. Dirro, P. Greve, R. Kashyap, D. Marcus, S. Masiello, F. Paget, C. Schmugar, A. Wosotowsky, (McAfee Inc.), McAfee Threats Report, Third Quarter 2010. http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q3-2010.pdf.
[4] Message Labs Ltd., MessageLabs Intelligence: 2010 Annual Security Report. http://www.messagelabs.co.uk/intelligence.aspx.
[5] J. Klensin, Simple Mail Transform Protocol, RFC5321, 2008. http://www.rfc-editor.org/rfc/pdfrfc/rfc5321.txt.pdf.
[6] T.S. Guzella, W.M. Caminhas, A review of machine learning approaches to spam filtering, Expert Systems with Applications 36 (7) (2009) 10206–10222.
[7] B. Biggio, G. Fumera, I. Pillai, F. Roli, A survey an experimental evaluation of image spam filtering techniques, Pattern Recognition Letters 32 (10) (2011) 1436–1446.
[8] S.J. Delany, P. Cunningham, A. Tsymbal, L. Coyle, A case-based technique for tracking concept drift in spam filtering, Knowledge-Based Systems 18 (4–5) (2005) 187–195.
[9] F. Fdez-Riverola, E.L. Iglesias, F. Díaz, J.R. Méndez, J.M. Corchado, Applying lazy learning algorithms to tackle concept drift in spam filtering, Expert Systems with Applications 33 (1) (2007) 3–48.
[10] I. Androutsopoulos, V. Metsis, G. Paliouras, Spam filtering with Naïve Bayes—which Naïve Bayes? in: Third Conference on Email and Anti-Spam CEAS, 2006.
[11] P. Graham, A Plan for Spam, 2000. http://www.paulgraham.com/spam.html.
[12] C.-C. Lai, An empirical study of three machine learning methods for spam filtering Knowledge-Based Systems 20 (3) (2007) 249–254.
[13] SpamAssassin, The Apache SpamAssassin project. http://spamassassin.apache.org/.
[14] V. Mitra, C. Wang, S. Benerjee, Text classification: at least square support vector machine approach, Applied Soft Computing 7 (2007) 908–914.
[15] H. Drucker, D. Wu, V.N. Vapnik, Support vector machines for spam categorization, IEEE Transactions on Neural Networks 10 (5) (1999) 1048–1054.
[16] A.H. Mohammad, R.A. Zitar, Application of genetic optimized artificial immune system and neural networks in spam detection, Applied Soft Computing 11 (4) (2011) 3827–3845.

[17] L. Özgür, T. Güngör, F. Gürgen, Adaptative anti-spam filtering for agglutinative languages: a special case for Turkish, Pattern Recognition Letters 25 (16) (2004) 1819–1831.
[18] A. Visconti, H. Tahayori, Artificial immune system based on interval type-2 fuzzy set paradigm, Applied Soft Computing 11 (6) (2011) 4055–4063.
[19] S.X. Wu, W. Banzhaf, The use of computational intelligence in intrusion detection systems: a review, Applied Soft Computing 10 (1) (2010) 1–35.
[20] T.S. Guzella, T.A. Mota-Santos, J.Q. Uchôa, W.M. Caminhas, Identification of SPAM messages using an approach inspired on the immune system, Biosystems 92 (3) (2008) 215–225.
[21] X. Carreras, L. Márquez, Boosting trees for anti-spam e-mail filtering, in: Proceedings of the 4th International Conference on Recent Advances in Natural Language Processing, 2001, pp. 58–64.
[22] F. Fdez-Riverola, E.L. Iglesias, F. Díaz, J.R. Méndez, J.M. Corchado, Spamhunting: an instance-based reasoning system for spam labelling and filtering, Decision Support Systems 43 (3) (2007) 722–736.
[23] Z. Pawlak, Rough Sets – Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishers, 1991.
[24] Z. Pawlak, Rough sets, International Journal of Computer and Information Sciences 11 (1982) 341–356.
[25] Z. Pawlak, Rough sets: present state and the future, Foundations of Computing and Decision Sciences 11 (3–4) (1993) 157–166.
[26] W. Ziarko, Variable precision rough set model, Journal of Computer and System Sciences 46 (1993) 39–59.
[27] J.F. Gálvez, F. Díaz, P. Carrión, A. García, An application for knowledge discovery based on a revision of VPRS model, RSCTC '00, in: Proceedings of 2nd International Conference of Rough Sets and Current Trends in Computing, 2001, pp. 296–303.
[28] M. Glymin, W. Ziarko, Rough set approach to spam filter learning, RSEISP '07, in: Proceedings of the International Conference of Rough Sets and Intelligent System Paradigms, 2007.
[29] W. Zhao, Y. Zhu, Classifying email using variable precision rough set approach, Lecture Notes in Artificial Intelligence 4062 (2006) 766–771.
[30] D.C. Whitley, M.G. Ford, D.J. Livingstone, Unsupervised forward selection: a method for eliminating redundant variables, Journal of Chemical Information and Computer Sciences 40 (5) (2000) 1160–1168.
[31] W. Zhao, Z. Zhang, An email classification model based on rough set theory, in: Proceedings of the International Conference on Active Media Technology, 2005, pp. 403–408.
[32] W. Zhao, Y. Zhu, An email classification scheme based on decision-theoretic rough set theory and analysis of email security, in: Proceedings of 2005 IEEE Region 10 TENCON, 2005, pp. 1–6.
[33] B. Zhoy, Y. Yao, J. Luo, A three-way decision approach to email spam filtering, in: Proceedings of the 23rd Canadian Conference of Artificial Intelligence, LNAI 6085, 2010, pp. 28–39.
[34] G. Lai, C. Chen, C. Laih, T. Chen, A collaborative anti-spam system, Expert System with Applications: An International Journal 36 (3) (2009) 6645–6653.
[35] G. Lai, C. Chou, C. Chen, Y. Ou, Anti-spam filter based on data mining and statistical test, Computer and Information Science 208 (2009) 179–192.
[36] Y. Chiu, C. Chen, B. Jeng, H. Lin, An alliance-based anti-spam approach, in: Proceedings of 3rd International Conference on Natural Computation, 2007.
[37] C. Liu, S. Stamm, Fighting unicode-obfuscated spam, in: Proceedings of the Anti-phishing Working Groups 2nd Annual eCrime Researchers Summit (eCrime '07), 2007.
[38] A. Taha, A. Hamdan, B. Al-Shargabi, M. Abu, Developing New Continuous Learning Approach for Spam Detection using Artificial Neural Network (CLA_ANN), European Journal of Scientific Research 42 (3) (2010) 525–535, http://www.eurojournals.com/ejsr_42_3_15.pdf.
[39] T.A. Meyer, B. Whateley, SpamBayes: effective open-source, Bayesian based, email classification system, in: CEAS 2004 – First Conference on Email and Anti-Spam, 2004, http://www.ceas.cc/papers-2004/136.pdf.
[40] J.R. Méndez, I. Cid, D. Glez-Peña, M. Rocha, F. Fdez-Riverola, A comparative impact study of attribute selection techniques on Naïve Bayes spam filters, in: Proceedings of the 8th Industrial Conference on Data Mining, 2008, pp. 213–227.
[41] J.R. Méndez, F. Fdez-Riverola, F. Díaz, E.L. Iglesias, J.M. Corchado, A comparative performance study of feature selection methods for the anti-spam filtering domain, Industrial Conference on Data Mining (2006) 106–120.
[42] J.R. Méndez, E.L. Iglesias, F. Fdez-Riverola, F. Díaz, J.M. Corchado, Analyzing the impact of corpus preprocessing on anti-spam filtering software, Research on Computing Science 17 (2005) 129–138.
[43] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison Wesley, 1999.
[44] J.R. Méndez, B. Corzo, D. Glez-Peña, F. Fdez-Riverola, F. Díaz, Analyzing the performance of spam filtering methods when dimensionality of input vector changes, in: Proceedings of the 5th Conference on Machine Learning and Data Mining, Leipzig, Germany, 2007, pp. 364–378.
[45] J. Graham-Cumming, Understanding spam filter accuracy, in: Jgc Spam and Anti-spam Newsletter, 2004, http://www.jgc.org/antispam/11162004-baafcd719ec31936296c1fb3d74d2cbd.pdf.
[46] C.J. Rijsbergen, Information Retrieval, Butterworth, London, 1979.
[47] W.M. Shaw, R. Burgin, P. Howell, Performance standards and evaluations in IR test collections: cluster-based retrieval models, Information Processing and Management 33 (1) (1997) 1–14.
[48] J. Sienkiewicz, The rough set library, in: Rough Sets in Knowledge Discovery: Applications, Case Studies, and Software Systems, Physica-Verlag, 1998, ISBN 3-7908-1120-3.
[49] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1995, pp. 1137–1143.

N. Pérez-Díaz, Ph.D. student from University of Vigo, Spain. She was born in Galicia – Spain (1985). She is a novel computer science engineer with high experience on spam filtering systems and Internet security. She collaborates as researcher with the SING group (*Computer Systems of New Generation*) belonging to the University of Vigo. Currently, her main topic of interest is the development of computer security tools including antiviral and spam filtering software.

D. Ruano, Ph.D. student from the University of Vigo, Spain. He was born in Galicia – Spain (1985). He is a junior computer science engineer with high experience on Linux administration and software development in the C language. He collaborates as researcher with the SING group belonging to the University of Vigo. He is mainly interested in the study of data mining techniques applied to different AI problems.

J.R. Méndez, Ph.D. from the University of Vigo, Spain. He was born in Galicia, Spain (1977). He has been working as system administrator, software developer and IT consultant in the civil service and IT Industry during the last 10 years. Moreover, he is joint author of several scientific papers in the domain of spam filtering and developed an innovative efficient technique known as SpamHunting. He is an active researcher from SING research group. He is mainly interested in the development and improvement of spam filters.

J.F. Gálvez, Ph.D. from the University of Vigo, Spain. He was born in Granada – Spain in 1968. He is a full time professor in the Computer Science Department of the University of Vigo. He collaborates as researcher with the SING research group and the LIA (*Laboratory of Applied Computer Science*) group, both belonging to the University of Vigo. Although his present research is related to the field of Rough Sets theory and its application to real problems, previous work was on topics related to classification and image analysis.

F. Fdez-Riverola, Ph.D. from the University of Vigo, Spain. He was born in Langen-Hessen, Germany, in 1973. He is Director of the CITI research Centre and a full time professor in the Computer Science Department of the University of Vigo. He is also the principal investigator of the SING group and collaborates with the BISITE (*Biomedicine, Intelligent Systems & Educational Technology*) group belonging to the University of Salamanca. He is joint author of several books and book chapters, as well as the author of numerous articles published by well-known houses such as the Springer-Verlag, Ios Press, Kluwer, etc. (http://sing.ei.uvigo.es/).