International Journal of Information Technology & Decision Making

EMAIL SPAM DETECTION: A SYMBIOTIC FEATURE SELECTION APPROACH FOSTERED BY EVOLUTIONARY COMPUTATION

PEDRO SOUSA
Centro Algoritmi/Department of Informatics, Universidade do Minho, Braga, Portugal
pns@di.uminho.pt

PAULO CORTEZ
Centro Algoritmi/Department of Information Systems, Universidade do Minho, Guimarães, Portugal
pcortez@dsi.uminho.pt

RUI VAZ
Department of Information Systems, Universidade do Minho, Guimarães, Portugal
a48052@alunos.uminho.pt

MIGUEL ROCHA
CCTC/Department of Informatics, Universidade do Minho, Braga, Portugal
mrocha@di.uminho.pt

MIGUEL RIO
Department of Electric and Electronic Engineering, University College London, Torrington Place, London, U.K.
m.rio@ee.ucl.ac.uk

Received Day Month Year
Revised Day Month Year

The electronic mail (email) is nowadays an essential communication service being widely used by most Internet users. One of the main problems affecting this service is the proliferation of unsolicited messages (usually denoted by spam) which, despite the efforts made by the research community, still remains as an inherent problem affecting this Internet service. In this perspective, this work proposes and explores the concept of a novel symbiotic feature selection approach allowing the exchange of relevant features among distinct collaborating users, in order to improve the behavior of anti-spam filters. For such purpose, several Evolutionary Algorithms (EA) are explored as optimization engines able to enhance feature selection strategies within the anti-spam area. The proposed mechanisms are tested using a realistic incremental retraining evaluation procedure and resorting to a novel corpus based on the well known Enron datasets mixed with recent spam data. The obtained results show that the proposed symbiotic approach is competitive also having the advantage of preserving end-users privacy.

Keywords: Spam Detection; Content-Based Filtering; Evolutionary Algorithms; Naïve Bayes; Feature Selection

1. Introduction

According to the study presented by the Messaging Anti-Abuse Working Group (MAAWG), the percentage of unsolicited email, also known as spam, was around 90% of all worldwide messages sent in the first 9 months of 2011.[1] In this perspective, spam is a problem that affects both individuals and organizations and unsolicited messages are an intrusion of privacy, with problematic content, such as online fraud, phishing attacks or viruses.[2,3] Moreover, spam has costs in terms of Internet traffic fees and time that users spend reading unwanted messages. Due to its tiny cost to reach a high number of potential consumers, spam is widely spread and even criminal organizations have access to millions of infected computers (e.g., botnets), which might be used for spam proliferation.[4]

Collaborative Filtering (CF) and Content-Based Filtering (CBF) are the two main approaches used currently to fight spam.[5,6] The CF approach is based on sharing general identification information about spam messages, while CBF uses a Data Mining (DM)[7] classifier that learns to discriminate spam from specific message characteristics (e.g., common spam words). As an example, the CF based approach uses information about spam messages that can be based on blacklists containing IP addresses of known spam senders or fingerprints extracted from spam messages. CBF techniques can be used for several purposes, such as the cases of spam detection or Internet content filtering.[8] As regards to spam detection, current research on CBF relies mainly on improving individual classifier (e.g., Naïve Bayes) performance, by a better preprocessing or enhancement of the learning algorithm. The two approaches can also be also combined to achieve more reliable methods. For example, a blacklist is often used at a server level to tag a large number of spam, while the remaining spam can be detected later by using a personalized CBF at the end user level.

Within this context, this work explores a novel approach within the anti-spam area, focusing on improving content based filtering mechanisms by adopting a symbiotic feature selection approach. Here, under a collaborative scheme, users share relevant email features and Evolutionary Algorithms (EA) are explored as the optimization engines to search for the optimal subset of attributes maximizing the performance of spam detection filters. Moreover, in the devised solution end-users privacy is preserved, which is an essential concern within many DM related applications[9] and a crucial requirement to allow the deployment of the proposed approach in real environments.

Given the aforementioned, in this paper we compare the proposed evolutionary symbiotic filtering approach with: other non sharing EA variants; the commonly used CBF filter that uses the simpler Information Gain criterion for feature selection; and a symbiotic feature sharing CBF variant. The experiments were conducted using a novel corpus that includes the mailboxes of five Enron users mixed with recent spam. The paper is organized as follows. Firstly, the related work is presented in Section 2. Section 3 presents the e-mail data, local and symbiotic filtering methods, and evaluation metrics. Next, the results are presented and discussed in Section 4. Finally, closing conclusions are drawn in Section 5.

2. Related Work

Given the continuous adaptation of the spam proliferation methods and the evolution on the spam content, there is a continuous need for more robust and adaptive anti-spam techniques. As previously mentioned, there are two main approaches to fight spam: CF and CBF.[5,6] CF strategies are based in sharing information about spam messages (e.g., spam message hash, source domain, spammer IP address, etc.) in a community of users. CBF filters use a DM classifier to analyze content (e.g., word frequencies) extracted from email messages. Both approaches have disadvantages. CF often suffers from sparsity of data, when users classify very few messages, and first-rater problem, where an e-mail cannot be classified unless a user has rated it before[10]. Also, people have personal views of what is spam and CF often discards this issue.[11] On the other hand, CBF requires several representative training examples and poor performances are often achieved for new users. Moreover, the CBF behavior is dependent not only on the classifier learning capabilities but also the type of feature selection method adopted.[5]

Within the CBF approach to fight spam, the Naïve Bayes (NB) classifier is the most popular learning algorithm, since it is very fast while often achieving high detection accuracies.[12] Most NB solutions are based on textual content (e.g., word frequencies) of email messages. This popular approach (e.g., Thunderbird filter) has the advantage of being generalizable to wider contexts, such as spam instant messaging (spim) detection. However, the CBF performance is dependent of the type of feature selection method used. For instance, in research works where several well-known filter feature selection methods (e.g., Information Gain) were combined with several types of Naïve Bayes classifiers, the obtained results showed that the choice of the correct feature selection method is a key issue in order to gain a high spam detection accuracy.[5]

Recently, a novel Symbiotic Filtering (SF) approach was proposed, which makes use of a CBF approach improved by end-users collaboration.[13] Under the Web 2.0 paradigm, the idea is to use the Internet to gather distinct users interested on similar but not identical goals, i.e., improve the spam detection at a personalized level. The aim of SF is to foster mutual relationships, where all or most members benefit. Rather than exchanging data that is extremely sensitive (e.g., normal mail messages), the goal of SF is to share information about what each local CBF has learned. Within SF there are two interesting sharing alternatives: content-based filtering models or relevant features. The former approach was addressed in a recent work[13] using an adaptive mechanism based on the exchange of CBF models. However, due to privacy concerns, such approach may also preclude end-users from participating in such collaborative environments. This paper focuses on the latter alternative, exploring an alternative solution which is less sensitive, since no spam/ham probability is associated with the exchanged features. Moreover, sharing features requires less communication overhead when compared with exchanging CBF models.

For the task of selecting relevant features we propose the use of Evolutionary Algorithms (EAs).[14] EAs are good candidates for SF feature selection, since they perform a global multi-point search, quickly locating areas of high quality, even when the search space is very complex. Additionally, since a population of solutions is used, it is easy to

share relevant features among distinct users. In particular, the EA provides the optimization engine in the search for the optimal subset of attributes that maximizes an evaluation metric of spam detection. For this task, the EA uses a variable-sized set representation to encode a set of attributes used by the local classifier.

In other related contexts, some researchers have also used Evolutionary Computation to improve spam filtering solutions, and multi-objective EAs were also used to achieve a set of filtering rules with different profiles.[15] The filtering rules were encoded as expression syntax trees and the Non-dominated Sorting Genetic Algorithm (NSGA-II) was applied to maximize two evaluation criteria, i.e., precision and recall. In the same year, other authors proposed the use of an EA to analyze different configurations for SpamAssassin, a widely-used open source spam filter.[16] Their approach consisted in using an EA to achieve an optimal setup, at a personalized level, for the set of weights that is used to infer if a given message is spam. In this case, the EA minimized the number of false positives and false negatives. Other works also described a genetic programming approach to feature extraction for a cost-sensitive classification task of spam, where the used fitness comprised three objectives: an approximation to the bayes error, misclassification cost and number of tree nodes used to encode a particular solution.[17]

The work presented in this paper pursue the efforts of exploring the use of EAs within the anti-spam context, but focusing now on to the challenge of devising of a novel evolutionary feature selection approach, where relevant features are shared among multiple users interested in similar goals. Thus, within the present proposal, the devised symbiotic filtering approach is a natural distributed form of feature selection, also taking into account privacy issues.

3. Materials and Methods

This section starts by describing the corpus used to test the proposed mechanisms (Section 3.1) along with the adopted evaluation strategy (Section3.2). The following sections focus on content-based filtering subjects and alternative evolutionary feature selection approaches (Sections 3.3 and 3.4). Finally, Section 3.5 describes in detail the proposed symbiotic approach taking advantage of end-users collaboration.

3.1. *Spam Data*

While there are several public benchmark datasets created to evaluate anti-spam filters (e.g., Ling-spam, Spambase), most of these datasets are not fitted for personalized filtering.[18] To evaluate SF, ideally there should be real mailboxes collected from distinct users (possibly from a social network) during a given time period. Yet, due to logistic and privacy issues, it is quite difficult to obtain such data. Therefore, we created a novel corpus based on a realist and synthetic mixture of real ham and very recent spam messages.

The ham messages are from the popular Enron email collection, related with the year of 2001 (`http://www.cs.cmu.edu/~enron/`). From the total of 158 Enron users, we selected the five users that had an higher time overlap: **mar**tin-t, **pla**tter-p, **sai**bi-e, **scho**ltes-d and **smi**th-m. Since these employees worked at the same organization, it is reasonable

to assume that they could be somehow connected in the context of a professional social network environment.

The spam set consists in 19196 messages that were retrieved from the Bruce Guenter collection (`http://untroubled.org/spam/`), which is based in fake emails published in the Web, during the year of 2010 (our dataset was built in 2011). Only messages with Latin character sets were selected, because the ham messages use this type of character coding and non-Latin mails would be easy to detect. The choice for adopting this spam set is motivated by two main reasons. First, the original Enron email collection did not contain a clear spam/ham distinction and only a very small amount of messages (5%) were detected as spam when applying distinct filters (e.g., SpamAssassin) and human assessments[19]. Second, spam content evolves thought time due to new spam campaigns and also due to the intention of confusing CBF filters. Thus, it is more challenging and relevant to address fresh spam in this research area. In effect, several studies[19,18,13] propose the addition of recent spam to the original Enron ham messages for assessing the quality of spam filters.

In this work, the mixture of spam and ham is based on the time each message was received, as proposed in[18,13]. First, 9 years were added to the date field of all ham emails. Then, for each user, a random spam/ham ratio, uniform within $[0.5, 3]$, was initially set. Next, the corresponding amount of spam messages were randomly selected, within the same time period as defined by the ham data, from the whole spam set and mixed with the ham data. While the overall spam/ham ratio is set by a random and fixed value, it should be noted that under the proposed time ordered mixture, the spam/ham ratios fluctuate through time. Typically, in most batches this fluctuation is close to the overall spam/ham ratio (concept drift), although in a few cases there are higher changes (concept shift), as shown in Figure 1 for two users. Hence, the adopted SF corpus presents both concept drift and shift effects that are common in real mailboxes, where spam/ham ratios change over time due to several phenomena[20]. For example, the number of ham messages often varies thought time due to seasonal effects (e.g., holidays, weekends). Also, new spam campaigns or viruses can exploit security flaws, hugely increasing the amount of spam received, while novel anti-spam measures, applied at a network level (e.g., ISP blocking a known spammer), might heavily reduce spam on a temporary basis. Table 1 shows a summary of the adopted SF corpus, which is publicly available at `http://www3.dsi.uminho.pt/pcortez/sf-corpus/`.

Table 1: Summary of the SF corpus

user	size	features	time period	spam/ham
mar	888	5057	[3/10,12/10]	1.51
pla	672	2303	[4/10,12/10]	2.15
sai	1688	4476	[3/10,12/10]	1.38
sch	765	2833	[4/10,12/10]	1.50
smi	941	3460	[3/10,12/10]	0.94

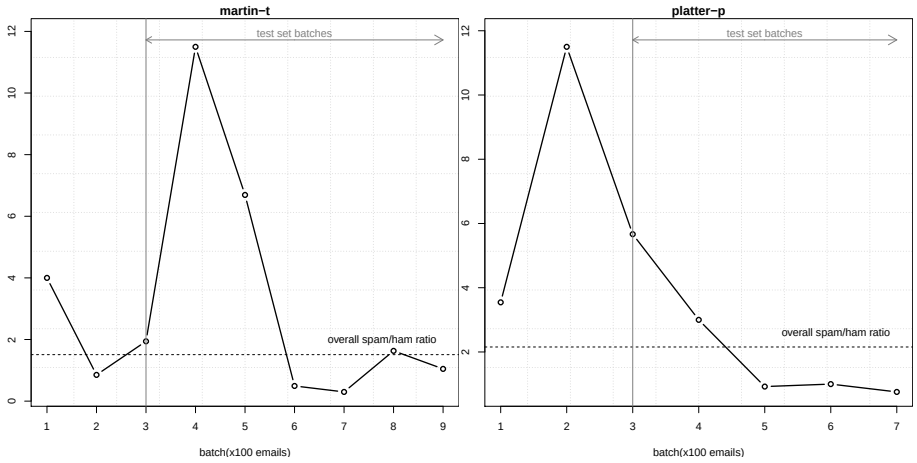

Fig. 1: Evolution of the spam/ham ratio for **mar**tin-t and **pla**tter-t mailboxes.

3.2. *Evaluation*

Spam detection is a process that evolves through time (i.e., there is a concept drift or shift) [20]. Given that, we adopt the more realistic incremental retraining evaluation procedure[18] that allows testing different train/test splits while preserving the time order in which the messages arrive. Under such procedure, a mailbox is split into batches $b_1, ..., b_n$ of K adjacent messages ($|b_n|$ may be less than K). Figure 1 shows an example the evolution of the spam/ham ratio over different batches for users **mar** and **pla**, with $K = 100$. For $i \in \{2, ..., n-1\}$, the spam filter is trained with $\mathcal{D}_u = b_1 \cup \ldots \cup b_i$ and tested with the messages from b_{i+1}, where $\mathcal{D}_u$ denotes the training data for user u.

An illustrative representation of the used incremental retraining procedure is presented in Figure 2. It should be noted that the minimum $\mathcal{D}_u$ size is set to $2K$, since the EA algorithms use the last batch of the training data as the validation set, to compute the fitness value.

For a given probabilistic filter, the predicted class for message $\mathbf{x}_j$ is given by: spam if $p(\text{spam}|\mathbf{x}_j, \mathcal{D}_u) > D$, where $D \in [0.0, 1.0]$ is a decision threshold. For a given D and test set, it is possible to compute the true (TPR) and false (FPR) positive rates, as expressed by Eq. 3.1, where TP, FP, TN and FN denote the number of true positives, false positives, true negatives and false negatives, respectively.

$$\begin{aligned} TPR &= TP/(TP + FN) \\ FPR &= FP/(TN + FP) \end{aligned} \tag{3.1}$$

The receiver operating characteristic (ROC) curve shows the performance of a two class classifier across the range of possible threshold (D) values, plotting FPR (x-axis) versus TPR (y-axis).[21] The global accuracy is given by the area under the curve (AUC) metric (see Eq. 3.2). In this perspective, a random classifier will have an AUC of 0.5, while the

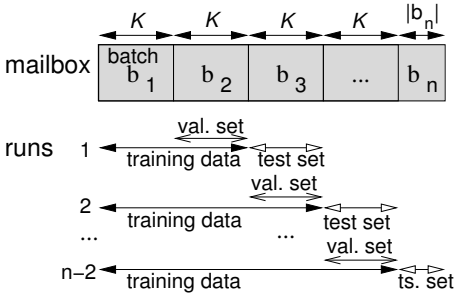

Fig. 2: Example of the incremental retraining procedure.

ideal value is 1.0.

$$AUC = \int_0^1 ROC \;\; dD \tag{3.2}$$

We adopt the AUC metric for the evaluation of the distinct spam detection methods, which is estimated using the algorithm proposed in[21]. Such algorithm only requires as inputs the spam/ham target values and classifier predictions and thus is not dependent on a particular choice of the decision threshold (D). Following the same procedure proposed in[18 13], one ROC is computed for each b_{i+1} test set batch and the overall result is presented by adopting an average of the AUC values computed for over all test set batches. In case of the EA algorithms, several runs are executed for each method and results are presented as the average of these runs with statistical significance given in terms of t-student confidence intervals at the 95% confidence level.[22]

3.3. *Content-Based Filtering*

The adopted CBF filter uses only textual content (i.e., word frequencies) of email messages and is based on the popular NB classifier. The preprocessing used follows the steps proposed by Metsis *et al.*[18] The word frequencies were extracted from the subject and body of the message. All HTML tags and non numeric or alphabetic characters were removed. Then, all capital characters were converted into lowercase letters. Next, words with two ore less characters were removed from the text. Each message j was then encoded into a vector $\mathbf{x}_j = (x_{1j}, \ldots, x_{mj})$, where x_{ij} is the number of occurrences of token X_i in the text. As an initial feature selection, any words with very small frequency (e.g., $x_{ij} < 5$) in the whole mailbox were removed. This preprocessing was performed using scripts written in the Perl language. In Table 1, which describes the used corpus, the column features denotes the total number of distinct words present in each mailbox of the analyzed corpus.

For the simpler CBF, the feature selection method is based on the Information Gain criterion,[5] which is applied to the training set in order to select the F_{IG} most relevant features. This popular approach is used as a comparison term for devised EA feature selection

techniques studied in this work.

Given its popularity for spam filtering, there are several NB versions that have been successfully applied within this domain.[18] In this paper, we adopt the Multinomial NB variant, as implemented in the popular open source RapidMiner tool and when using a sparse representation, which heavily reduces the computational memory requirements.[23] In Metsis *et al.*[18] such variant obtained a high quality spam detection accuracy, outperforming other NB versions, such as the Multivariate Gaussian.

3.4. *Evolutionary Feature Selection*

Generally there are two main approaches for feature selection: filters and wrappers.[24] Filters methods are independent of the learning algorithm and are applied in the preprocessing stage (e.g., Information Gain). Wrappers test several combinations of features and each testing requires the training of a given classifier. Wrapper methods tend to be more accurate than filters, although they require more computation and the results are specific to a particular classifier.

The evolutionary approach for feature selection adopts the Multinomial NB variant mentioned in Section 3.3. In order to reduce the search space to a reasonable size, the information gain filter is first applied to the training data, in order to select the F_{IG} most relevant features. Then, an EA is applied as a wrapper method, requiring the training of several NB classifiers.

Each EA individual is represented as a variable-sized set of strings, which allows the definition of a maximum and minimum number of words. This representation is a more natural form that is closer to the problem to be solved and has the advantage of not requiring a mapping function, when compared with the popular binary representation, since each individual contains the explicit words used by the CBF. The representation approach adopted in this work is depicted in Figure 3, considering a merely illustrative dataset example including a given number of features that are used for email detection, in terms of the (H)am/(S)pam classes.

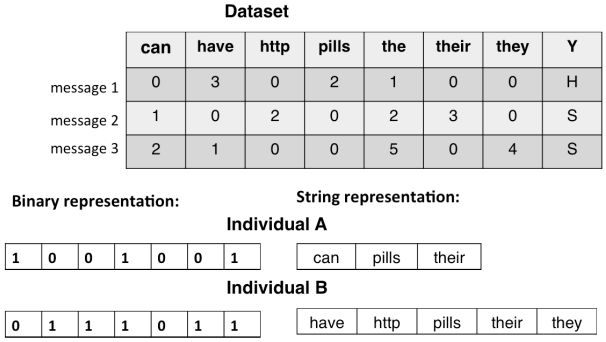

Fig. 3: Example of binary versus variable-sized set of strings representations.

The EAs are set to maximize the AUC metric (previously described in Section 3.2). The computation of the respective fitness is obtained as follows. For a given run of the incremental training (depicted in Figure 2), the training data is divided into training (with all cases except the validation samples) and validation sets (with the last K emails). The features that appear in a given chromosome are fed into the CBF model, which is fit using all training samples. The NB predictions over the validation set are then used to compute the AUC value. After the EA termination criteria, the best individual is selected and the respective features are used to feed a new NB that is fit by using all training data.

Regarding the EA engine, we adopted a general EA, as implemented in the JECoLi Java library.[25] First, there is an initial population with P individuals. New solutions are bred through the use of random respectful recombination[26] and random mutation operators:

- The recombination method creates two lists of features: first, with common features between the two progenitors and; second, with the remaining features. The descendants contain all features from the first set plus a random number of words from the second list.
- The mutation operator replaces a random number of features from the chromosome.

In both operators, the minimum and maximum number of features is always preserved. The genetic operators are used (with 50% probability each) to create a new population of size P. Both the original and new populations are evaluated and then a tournament selection is adopted (with a tournament size of 2) to select the P individuals that will survive to the next generation. Finally, the EA is stopped after G generations. Figure 4 shows the schematic of the EA engine adopted.

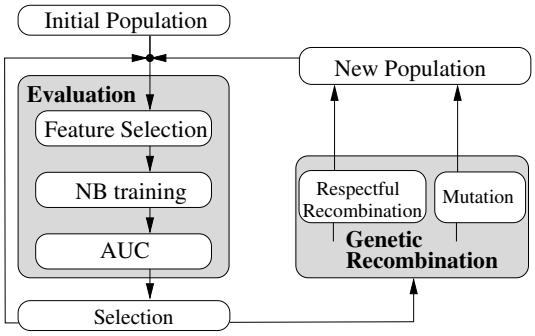

Fig. 4: Schematic of the EA engine.

Each EA is executed $n-2$ times, according to the incremental training approach (Section 3.2), where each EA run is applied over the training data available at the i-th iteration of the incremental procedure. When creating a random population, P individuals are generated, such that each individual contains a random size, between the minimum and

maximum threshold, with randomly selected words from the set of F_{IG} features. In this paper, we start by exploring two local EA variants, which are dependent on the type of initial population used:

- The EA with reinitialization (EAR) - this approach uses a random initial population for each run of the incremental training procedure, thus reseting past optimizations.
- The EA with memory (EAM) - this approach only uses a random population in the first iteration of the incremental batch (i.e., when the training set is equal to b_1). When a new batch of messages is included in the analysis, the EA restarts with the last population, thus reusing the best features from the previous EA execution.

In addition to the aforementioned local techniques, a third novel symbiotic EA based approach for feature selection will be described in the following section.

3.5. *Symbiotic Filtering*

In this section, the Evolutionary Algorithm that performs a Symbiotic filtering approach is described (the approach is from this point on identified as EAS). In counterpoint with the previously explained local EA variants, the proposed collaborative filtering approach assumes now a symbiotic collaboration within a group of distinct users, which share the most relevant email features among the group. In the devised solution, we assume that the featuring sharing process does not arise privacy concerns, given that no spam or ham probability is assigned to these exchanged features. Still, if required, additional secure data transfer mechanisms could also be used to assure confidentiality in the feature exchanges, such as using an anonymous sharing of encrypted features.

The EAS works similarly to EAM except that the initial population includes a percentage of p_s individuals, with features shared from other users, and $1 - p_s$ of the best individuals from the previous EAS batch. It is assumed that the symbiotic group has a size of n and each user runs a EAS and during the same time period. To reduce communication costs and computational effort, the exchange of features is asynchronous and occurs only when a new CBF is trained. In this work, this occurs every time a new batch of messages is analyzed. It should be noted that while the same batch size of K messages is used for all users, the messages included in each batch may be related with distinct dates.

To respect the chronological order of the distinct EAS, the last message date of the training set (t) is used to synchronize the exchange of features. Thus, the sharing is performed among the best individuals from the distinct EAS that were available at time t. For each iteration of the incremental retraining, a given user receives a total of $S = p_s \times P$ individuals, such that the S solutions are equitably retrieved from the other members of the symbiotic group (i.e., each user shares $S/(n-1)$ individuals). To simulate the distributed execution of the EAS, the JECoLi library was adapted to include a different thread for each user. The distinct threads were synchronized, in order to preserve the temporal order.

In some situations, a given user A may receive external features from user B that are not included in the mailbox of A (i.e., mapped in the matrix of word frequencies of A). To

increase the diversity of the shared features, we opted for searching for additional features that are extracted from the the best individuals from user B and that appear in the mailbox of A. This procedure is executed until the number of exchanged features is equivalent to the ones contained in the desired $S/(n-1)$ individuals exchange.

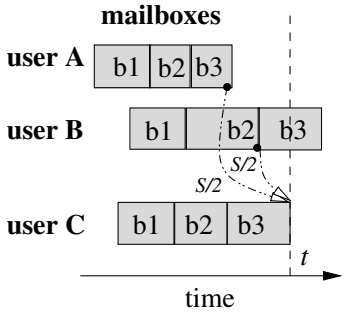

Fig. 5: Example of the time ordered exchange of individuals from users A and B with C.

For demonstration purposes, Figure 5 plots an example of the symbiotic exchange of individuals. In the example, two users (A and B) share $S/2$ individuals each with user C. It should be noted that while the distinct EAS are run within the same time period, the exchange of individuals is performed using different EAS evolution stages. In the example, the best individuals from user A were searched using all data until batch 3, while the exchange from B was performed over a EA that included only batches 1 and 2.

The devised symbiotic approach for feature selection is expected to produce more fruitful results if the involved users are somehow related, e.g., sharing some common characteristics, subjects of interest, etc. Moreover, due to the inherent EAS optimization process (which defines the best individuals from a given population), irrelevant shared features should tend to be progressively discarded.

For measuring the added value of using an EA optimization for the symbiotic share of features, in this work we also test a CBF Symbiotic (CBFS) variant. Such CBFS works similarly to the EAS method (e.g., with a synchronized exchange of features and same p_s parameter), except that no EA is used to select the best features. Instead, the Information Gain criterion is used first to rank all F_{IG} individual features and the best $S/(n-1)$ of these features are shared with other users.

4. Experiments and Results

All experiments were conducted in Java programming environments under the Linux operating system. Following previous works in the area[18,13], we set $K = 100$ for all users, as a reasonable balance between the computational effort required by the EA optimization and achieving a robust evaluation of the tested methods. For example, in Section 4.1 we compare the accuracy and computational effort of the simplest method (CBF) under $K = 1$

Table 2: Parameters set for the EA methods

parameter	**value**
population size (P)	20
minimum individual size (#features)	300
maximum individual size (#features)	400
elitism value	2
stopping criterion (G)	100
shared percentage (p_s)	0.6
symbiotic group size (n)	5

(classical approach) and $K = 100$, showing that there is only a slight gain in performance while computational effort highly increases. Moreover, as argued in [18], the $K = 100$ setup can make easier future experiments with other more computationally heavier learning algorithms, such as Support Vector Machines.

For each iteration of the incremental training, the number of information gain selected features was set $F_{IG} = 500$. The configuration parameters used by the EA versions are listed in Table 2. The values related with the last two rows of Table 2 are only used by the symbiotic approaches (CBFS and EAS). It should be noted that since the EAs work as a second order optimization procedure (i.e., optimizing the features of NB fitted model), the tuning of its internal parameters is not a critical issue (e.g., using a population size of 18 does not does not substantially change the results). Each EA algorithm was executed a total of 10 runs, with the results presented as the average of these runs and respective confidence intervals, as given by the t-student test at the 95% confidence level, while statistical significance is measured using non-parametric Mann-Whitney paired tests.[22]

In the next sections we will overview the obtained results, starting firstly with an illustrative analysis from a single user perspective (Section 4.1), followed by a general overview of the EA based methods performance in the analyzed corpus (Section 4.2). Finally, Section 4.3 will debate some deployment issues of the proposed symbiotic approach.

4.1. *Illustrative Analysis of an email user*

For demonstrative purposes we start the results analysis by selecting a single email user, considering in this case the user **mar**. The results discussion within this user context can also be applied to most of the other users as will be later summarized in Section 4.2.

The results obtained for each batch of the user **mar** are presented under two representations, expressed in Figure 6 and in a more detailed analysis in Table 3, expressing the obtained AUC values in the experiments and for each one of the batches. In Table 3, the last row presents the average AUC results (over all batches). This last row also includes the average of the confidence intervals (in brackets) for the EA methods. In the same table, the first column ($\text{CBF}_{K=1}$) presents the results of the Naïve Bayes continuous learning variant,

which is included here for comparison purposes with the batch learning $K = 100$ method (CBF).

As observed in Table 3, in general, the obtained results favor the symbiotic approach (EAS), which outperforms the non sharing EA variants (EAR and EAM), the CBF local NB filter and its symbiotic CBFS version. In effect, the last row of Table 3 presents the average AUC value over all batches and the highest value is achieved for EAS, i.e., the previously explained symbiotic feature selection strategy. In addition, for this user, it is also important to note that the remaining methods (EAR, EAM, CBF and CBFS) achieve considerable worst performances for two of the analyzed batches (5 and 9).

When comparing CBF with $\text{CBF}_{K=1}$, there is only a slight increase in performance when using a continuous learning method (i.e. overall improvement of 0.005 points). Also, while using more training messages, the overall performance of $\text{CBF}_{K=1}$ is worse when compared with the EA or sharing methods. Also importantly, the computational effort required by the continuous learning method is much higher than its batch variant. For example, under the tested simulation setup (e.g. use of Information Gain criterion, incremental retraining procedure, RapidMiner implementation of the Naïve Bayes method, 2.66 GHz Intel Core i7 processor) the execution time for $\text{CBF}_{K=1}$ is around K times more heavier than the batch CBF version, requiring more than four hours of computation for user **mar**. Similar results were achieved for other users. Given such performance versus computational effort tradeoff and taking into account the rationale explained in Section 4, we use $K = 100$ in the remaining of this paper.

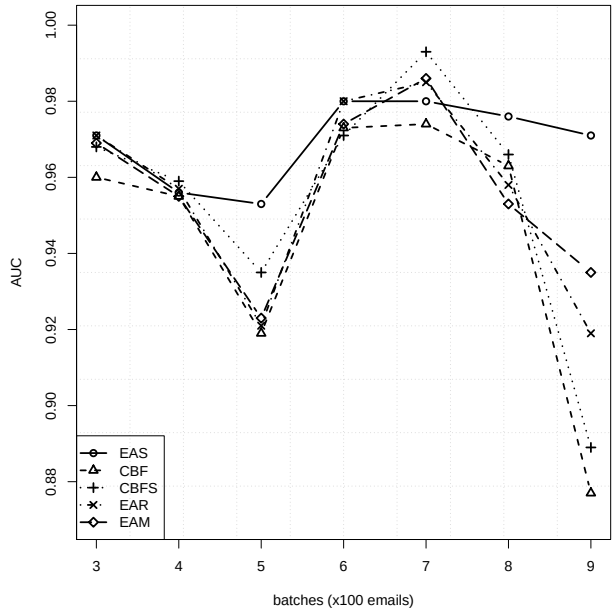

Fig. 6: Predictive results for user **mar** (AUC evolution over the test set batches)

Table 3: Predictive results for user **mar** (AUC on test sets, best value in **bold**)

b_i	**CBF**$_{K=1}$	**CBF**	**CBFS**	**EAR**	**EAM**	**EAS**
3	0.967	0.960	0.968	**0.971**$\pm$0.006	0.969$\pm$0.006	**0.971**$\pm$0.005†
4	**0.961**	0.955	0.959	0.957$\pm$0.008	0.955$\pm$0.011	0.956$\pm$0.019
5	0.919	0.919	0.935	0.921$\pm$0.006	0.923$\pm$0.005	**0.953**$\pm$0.012†*
6	0.956	0.973	0.971	**0.980**$\pm$0.004	0.974$\pm$0.006	**0.980**$\pm$0.004†
7	0.976	0.974	**0.993**	0.985$\pm$0.004	0.986$\pm$0.001	0.980$\pm$0.005†
8	0.960	0.963	0.966	0.958$\pm$0.008	0.953$\pm$0.008	**0.976**$\pm$0.010†*
9	0.916	0.877	0.889	0.919$\pm$0.009	0.935$\pm$0.007	**0.971**$\pm$0.007†*
$\bar{b}_i$	0.951	0.946	0.954	0.956 (0.006)	0.956 (0.006)	**0.970** (0.009)

$\dagger$ – average confidence interval does not overlap with CBF value.

$\star$ – statistically significant when compared with EAM and EAR.

In addition, and as observed in the CBF results column of Table 3, within this specific user the traditional CBF method obtained the worst performance. From the observed averaged values, one can point that the evolutionary symbiotic approach is competitive showing the ability to achieve improvements over some state-of-the-art mechanisms, such as CBF, and the other explored EA variants (EAR and EAM), and even the CBF symbiotic variant. The overall ROC curve of the EAS approach for user **mar** is also plotted in Figure 7.

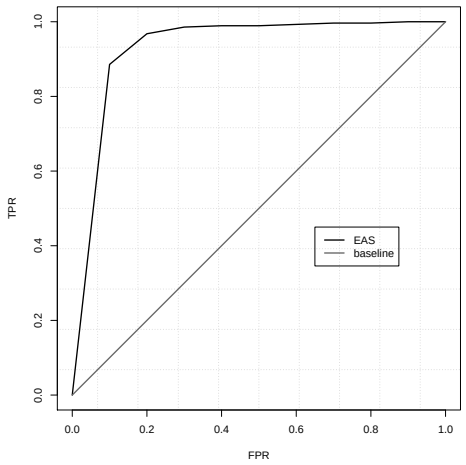

Fig. 7: Global ROC curve (over all test set batches) for user **mar** and EAS.

From the perspective of the optimization function, it is also interesting to assess the fitness behavior (as measured over the validation set) of the studied techniques. For demon-

strative purposes, the average (over all 10 runs) fitness evolution of the best individuals for user **mar** and methods EAS and EAM is shown in Figure 8. From the plot, it is clear that each new iteration of the incremental training procedure produces a disruption in the EA optimization. However, in most batches, EAS produces a much faster recovery when compared with EAM, in a clear demonstration of the benefit of using relevant features shared by other collaborating users, as observed in Figure 8.

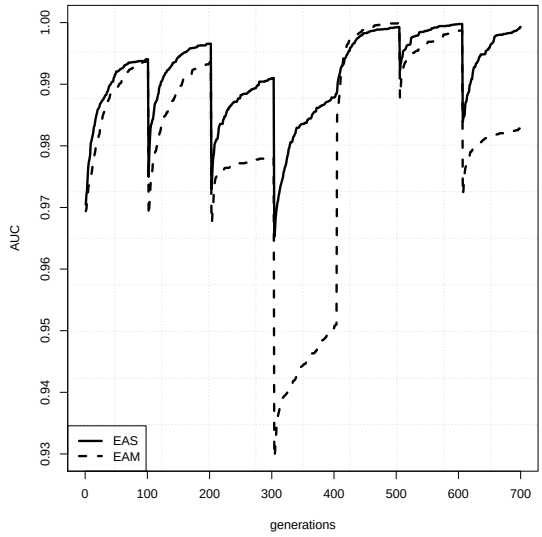

Fig. 8: Optimization search evolution of the best individuals for EAS and EAM methods and user **mar** (average AUC fitness computed over validation sets).

4.2. *Global Results of the analyzed corpus*

This section presents the overall results obtained by the studied techniques in the analyzed corpus. For that purpose, the global results are measured using two criteria: *i)* average AUC value, over of all batches $\overline{b}_i$ (shown in Table 4) and *ii)* percentage of test set batches where the method returns the best AUC value (shown in Table 5). For each user, the last criterion is computed using the formula w/n_{ts}, where w denotes the number of wins of the method and n_{ts} the number of test set batches. When t methods produce the same best AUC value (e.g., batch 3 for user **mar** as shown in Table 3), the value of w is increased with $1/t$, for all t methods related with ties. In Table 4, the average (over all test set batches) of the confidence intervals is also shown (in brackets) for the EA based methods. The last two rows of Tables 4 and 5 correspond to the aggregated result (over all users), as measured using the average and median value. When compared with the average aggregation method, the median is more robust with respect to outliers.

Overall, the best method is the symbiotic EA (EAS). It is the best option for three users

for both AUC and percentage of wins. Moreover, EAS also presents the best global average and median values for both AUC and percentage of wins criteria. The average confidence intervals for EAS are small and quite similar to the ones achieved by the EA methods, confirming that the EA search is robust, tending to achieve the similar optimum solutions within the distinct applied runs. This was an expected outcome, since (as previously explained) the EA performs a second order optimization procedure. Regarding the non sharing EAs, EAR and EAM approaches obtain a very similar performance, in terms of the AUC criterion (EAM average value is slightly above EAR, while median values are identical). When considering the percentage of batch wins, EAR is better than EAM for user mar, while EAM is ranked first for user smi. The CBFS method achieves the best results for two users (pla and sai), under the AUC criterion, and one user (sai), under the percentage of batch wins measure. When considering the percentage of wins global measures, this method is ranked at second place, although the AUC aggregate results rank CBFS at a worst position (fifth place for the average and fourth for the median). The best comparative CBF result is achieved for user sai (ranked at second place). Overall, its performance is worst than all EA methods, when considering AUC (average and median), and it is ranked in forth/third place (average/median), when considering the percentage of batch wins.

Table 4: Overall results for all users (AUC values, best value in **bold**)

user	CBF	CBFS	EAR	EAM	EAS
mar	0.946	0.954	0.956 (0.006)	0.956 (0.006)	**0.970** (0.009)
pla	0.950	**0.955**	0.949 (0.007)	0.947 (0.007)	0.953 (0.010)
sai	0.983	**0.987**	0.975 (0.005)	0.980 (0.004)	0.974 (0.006)
sch	0.961	0.968	0.967 (0.005)	0.964 (0.005)	**0.970** (0.010)
smi	0.935	0.898	0.938 (0.010)	0.942 (0.007)	**0.943** (0.010)
average	0.955	0.952	0.957	0.958	**0.962**
median	0.950	0.955	0.956	0.956	**0.970**

Figure 9 shows the influence of the average (left) and median (right) AUC global values according to two factors: non symbiotic (CBF, EAR and EAM) versus symbiotic methods (CBFS and EAS); and non evolutionary (CBF and CBFS) versus evolutionary (EAR, EAM and EAS). From the plots, it is clear that the evolutionary optimization improves both average and median AUC values. Regarding the symbiotic factor, its effect is strongly positive for EAS but only slight positive for CBFS under the median metric, while producing a negative effect for CBFS under the average AUC. Such outcome was expected, since the increased value of receiving external features from other users should be dependent on the quality (or relevance) of these features. And the wrapper feature selection, performed by the EA, was expected to select more relevant features when compared with the Information Gain criterion.

Table 5: Percentage of batch wins for all methods and users (best value in **bold**)

user	CBF	CBFS	EAR	EAM	EAS
mar	0.0	28.6	28.6	0.0	**42.9**
pla	20.0	20.0	0.0	0.0	**60.0**
sai	20.0	**40.0**	13.3	13.3	13.3
sch	16.7	33.3	0.0	0.0	**50.0**
smi	0.0	25.0	25.0	**37.5**	12.5
average	11.3	29.4	13.4	10.2	**35.7**
median	16.7	28.6	13.3	0.0	**42.9**

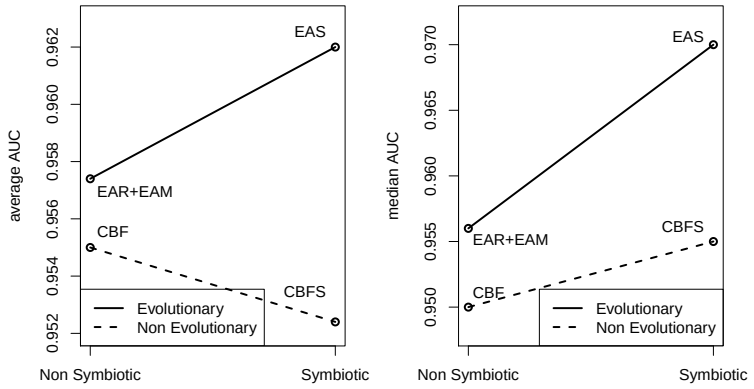

Fig. 9: Interaction plots for symbiotic/non symbiotic and EA/non EA factors

Globally, all spam detection methods achieved a high quality spam detection, with all average AUC values higher than 0.9. The differences between the distinct methods may seem small, with AUC improvements of 0.3 to 2.4 pp of EAS over CBF. Nevertheless, it should be noted that higher improvements may be achieved for particular batches. For example, the difference between EAS and CBF for user **mar** and batch 9 is 9.4 pp (Table 3). Also, as shown in Table 5, EAS tends to provide the best AUC values in most of the batches. Moreover, we stress that EAS uses less features (with a reduction that ranges from 20% to 40%) when compared with the CBF and CBFS methods, and such feature reduction can lead to other potential benefits (e.g., reduced storage requirements, better data understanding).

Spam classification is a very competitive area. Thus, any improvement achieved by a given method is important to be considered, as it leads to a considerable user added value, translating into a better spam email detection probability, which means less time reading unwanted messages and more immunity to virus, worms or phishing attacks. In this perspective, the results of the EAS method provided clear indications that it can constitute

a valuable approach to be considered in the area. Thus, the proposed symbiotic approach for feature selection could underpin and inspire the development of future research works within the field of spam detection.

Also in the context of the results analysis, it should be highlighted that the tested corpus included a small symbiotic group with five users. Thus, the fact the EAS achieved the best performance in this dataset is a preliminary indication that the proposed method could even produce better results when considering symbiotic environments with a higher number of users.

4.3. *Further Deployment Considerations*

As regards to deployment issues, the proposed symbiotic collaborative approach can be easily deployed in real scenarios. As a simple example, the devised anti-spam technique can take advantage of social network environments, where users are grouped according with their particular characteristics and interests. In such context, the collaboration in the symbiotic features exchanges can take the form of a possible add-on service that could be offered under the umbrella of such social network environments.

Moreover, the proposed method takes into account privacy issues of end-users, as no ham/spam classification information is exchanged among the participants, which could preclude end-users from participating in these collaborative approaches. In practice, this means that there is no need to protect data exchanges of the collaborative system with additional security/confidentiality mechanisms. This also constitutes a valuable add-on that clearly distinguish this approach from previous works in the area.

As related to the processing overhead, EAS requires more communication and computation when compared with the simpler CBF method. However, the increase in communication and computation is still affordable for a common user. The communication costs are low and we can point to a value of around the size of one email message for every batch (e.g., 100 messages). Moreover, under the proposed symbiotic approach, the execution of a batch is not computationally expensive. For example, under the tested computer (2.66 GHz Intel Core i7), the average execution times for 100 generations of the EAS were 11s for user **pla** and 41s for user **mar**. Furthermore, if required, in real implementations the EAS system could be run using a different thread, in the background, taking advantage of possible user idle periods.

It is worth to mention that other anti-spam systems can also take advantage of optimized feature selection approaches inspired in the mechanisms here proposed. In fact, systems such as SpamAssassin[27] often deal with very large databases of word features that are required by the Naïve Bayes filter. This may require significant computational resources, thus increasing the anti-spam system response time. In this perspective, such systems may also resort to EA based optimization processes to select the best features, thus improving scalability and response time.

In this perspective, taking into account both the results obtained by the EAS method in the experiments made and the aforementioned deployment considerations, we believe that EAS constitutes a valuable approach to be considered within this research field.

5. Conclusions

This work devised a novel distributed feature selection solution within the spam detection research area. The proposed approach makes use of Evolutionary Algorithms (EA) as the optimization engines for searching the best features. Under the proposed symbiotic solution (EAS) collaborating users share relevant features in order to improve spam detection at a personalized level.

The EAS approach and other non sharing EA alternatives were tested in a new corpus that performs a realistic mixture of ham messages from five Enron users with recent spam. The performance of EAS was compared with two local EA algorithms (EAR and EAM), as well as with two CBF methods (the standard CBF and a symbiotic variant) based on the information gain criterion. The presented results showed that even considering a small symbiotic group (i.e., with 5 users), EAS achieves the best spam detection performance, as measured by the AUC metric.

Moreover, the proposed symbiotic approach is simple to be implemented in real networked scenarios and preserves end-users privacy, which is a crucial aspect to be considered within the email context. Taking into account both the obtained results and the EAS deployment possibilities, we believe that this work will foster the research efforts and the development of innovative solutions within the email spam detection area.

In future work, we intend to study scalability issues of the proposed solution, addressing groups with a larger number of collaborating users. Furthermore, the devised EAS solution might be also potentially useful in other personalized filtering scenarios, such as the example of Web page selection of relevant documents[28] or blocking offensive content[29].

Acknowledgments

The work of P. Cortez and P. Sousa was funded by FEDER, through the program COMPETE and the Portuguese Foundation for Science and Technology (FCT), within the project FCOMP-01-0124-FEDER-022674. Also, the authors wish to thank the anonymous reviewers for their helpful comments.

References

1. MAAWG, Email metrics program: the network operators perspective, Report # 15 - First, second and third quarter 2011, *Technical Report, Messaging Anti-Abuse Working Group*, 2011.
2. Guido Schryen. *Anti-Spam Measures: Analysis and Design*, Springer, 2007.
3. OECD, OECD guidelines for the security of information systems and networks: Towards a culture of security, *Technical report*, 2006.
4. A. Ramachandran and N. Feamster, Understanding the Network-Level Behavior of Spammers, *ACM SIGCOMM 2006 Conference*, 2006, pp. 291–302.
5. J. Méndez, I. Cid, D. Glez-Peña, M. Rocha and F. Fdez-Riverola, A comparative impact study of attribute selection techniques on naive bayes spam filters, *8th industrial conference on Advances in Data Mining: Medical Applications, E-Commerce, Marketing, and Theoretical Aspects*, 2008, pp. 213–227.
6. E. Blanzieri and A. Bryl, A survey of learning-based techniques of email spam filtering, *Artificial Intelligence Review* **29**(1) (2008) 63–92.

7. Y. Peng and K. Gang and S. Young and Z. Chen, A descriptive framework for the field of data mining and knowledge discovery, *International Journal of Information Technology & Decision Making*, **7**(4) (2008) 639–682.
8. S. Ho, S. Lui and W. Ma, Acceptance of Internet Content Filters: An Empirical Study, *International Journal of Information Technology & Decision Making*, **2**(3) (2003) 477–496.
9. Q. Yang and X. Wu, 10 Challenging Problems in Data Mining Research, *International Journal of Information Technology & Decision Making*, **5**(4) (2006) 597–604.
10. A. Abdelwahab, H. Amira and I. Matsuba and Y. Horiuchi and S. Kuroiwa, Alleviating The Sparsity Problem Of Collaborative Filtering Using An Efficient Iterative Clustered Prediction Technique, *International Journal of Information Technology & Decision Making*, **11**(1) (2012) 33–53.
11. A. Gray and M. Haahr, Personalised, collaborative spam filtering, *1st Conference on E-Mail and Anti-Spam (CEAS)*, 2004.
12. S. Garriss, M. Kaminsky, M. Freedman, B. Karp, D. Mazières and H. Yu, RE: reliable email, *3rd conference on Networked Systems Design and Implementation (NSDI)*, 2006, pp. 297–310.
13. C. Lopes, P. Cortez, P. Sousa, M. Rocha and M. Rio, Symbiotic filtering for spam email detection, *Expert Systems with Applications* **38**(8) (2011) 9365–9372.
14. K. De Jong, *Evolutionary computation: a unified approach*, MIT Press, 2006.
15. A. Lopez-Herrera, E. Herrera-Viedma and F. Herrera, A multiobjective evolutionary algorithm for spam e-mail filtering, *3rd International Conference on Intelligent System and Knowledge Engineering*, 2008, pp. 366–371.
16. J. Dudley, L. Barone and L. While, Multi-objective spam filtering using an evolutionary algorithm, *IEEE Congress on Evolutionary Computation*, 2008, pp. 123–130.
17. Y. Zhang, H. Li, M. Niranjan and P. Rockett, Applying cost-sensitive multiobjective genetic programming to feature extraction for spam e-mail filtering, *11th European Conference on Genetic Programming*, 2008, pp. 325–336.
18. V. Metsis, I. Androutsopoulos and G. Paliouras, Spam filtering with naive bayes - which naive bayes?, *3rd Conference on Email and AntiSpam (CEAS)*, 2006, pp. 125–134.
19. G. Cormack and T. Lynam. TREC 2005 spam track overview, *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings*, 2005, pp. 1–17.
20. T. Fawcett, In vivo spam filtering: A challenge problem for KDD, *SIGKDD Explorations* **5**(2) 2003 140–148.
21. T. Fawcett, An introduction to ROC analysis, *Pattern Recognition Letters* **27**(8) (2006) 861–874.
22. A. Flexer, Statistical evaluation of neural networks experiments: minimum requirements and current practice, *13th European Meeting on Cybernetics and Systems Research*, 1996, pp. 1005–1008.
23. I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz and T. Euler, Yale: rapid prototyping for complex data mining tasks, *12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 935–940.
24. I. Guyon and A. Elisseeff, An introduction to variable and feature selection. *Journal of Machine Learning Research* **3** (2003) 1157–1182.
25. P. Evangelista, P. Maia and M. Rocha, Implementing metaheuristic optimization algorithms with JECoLi, *Ninth International Conference In Intelligent Systems Design and Applications*, 2009, pp. 505–510.
26. N. Radcliffe, Genetic set recombination, *Foundations of Genetic Algorithms* **2** (1993) 203–219.
27. A. Schwartz, *SpamAssassin*, O'Reilly, 2005.
28. Z. Palotai and B. Gábor and A. Lőrincz, Adaptive highlighting of links to assist surfing on the internet, *International Journal of Information Technology & Decision Making*, **4**(1) (2005) 117–139.
29. D. Canali and M. Cova and G. Vigna and C. Kruegel, Prophiler: A fast filter for the large-scale

detection of malicious web pages, *20th international conference on World Wide Web*, 2011, pp. 197–206.