

# Quadcriteria Optimization of Binary Classifiers: Error Rates, Coverage, and Complexity

Vitor Basto-Fernandes<sup>1,2</sup>, Iryna Yevseyeva<sup>3</sup>, David Ruano-Ordás<sup>4</sup>,  
Jiaqi Zhao<sup>5</sup>, Florentino Fdez-Riverola<sup>4</sup>, José Ramón Méndez<sup>4</sup>,  
and Michael T.M. Emmerich<sup>6</sup>(✉)

<sup>1</sup> Instituto Universitario de Lisboa (ISCTE-IUL), University Institute of Lisbon,  
ISTAR-IUL, Av. Das Forças Armadas, 1649-026 Lisboa, Portugal

`vitor.basto.fernandes@iscte.pt`

<sup>2</sup> School of Technology and Management, Computer Science  
and Communications Research Centre, Polytechnic Institute of Leiria,  
2411-901 Leiria, Portugal

`vitor.fernandes@ipleiria.pt`

<sup>3</sup> School of Computer Science and Informatics, Faculty of Technology,  
Cyber Technology Institute,  
De Montfort University, Gateway House, The Gateway, Leicester LE1 9BH, UK  
`iryna@dmu.ac.uk`

<sup>4</sup> Informatics Engineering School, University of Vigo,  
Campus As Lagoas S/N, 32004 Ourense, Spain  
{`drordas,riverola,moncho.mendez`}@uvigo.es

<sup>5</sup> The School of Computer Science and Technology, China University of Mining and  
Technology, No 1, Daxue Road, Xuzhou 221116, Jiangsu, P.R. China  
`jiaqizhao88@126.com`

<sup>6</sup> Multicriteria Optimization, Design, and Analytics Group, LIACS,  
Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands  
`emmerich@liacs.nl`

**Abstract.** This paper presents a 4-objective evolutionary multiobjective optimization study for optimizing the error rates (false positives, false negatives), reliability, and complexity of binary classifiers. The example taken is the email anti-spam filtering problem.

The two major goals of the optimization is to minimize the error rates that is the false negative rate and the false positive rate. Our approach discusses three-way classification, that is the binary classifier can also not classify an instance in cases where there is not enough evidence to assign the instance to one of the two classes. In this case the instance is marked as suspicious but still presented to the user. The number of unclassified (suspicious) instances should be minimized, as long as this does not lead to errors. This will be termed the *coverage objective*. The set (ensemble) of rules needed for the anti-spam filter to operate in optimal conditions is addressed as a fourth objective. All objectives stated above are in general conflicting with each other and that is why we address the problem as a 4-objective (quadcriteria) optimization problem. We assess the

performance of a set of state-of-the-art evolutionary multiobjective optimization algorithms. These are NSGA-II, SPEA2, and the hypervolume indicator-based SMS-EMOA. Focusing on the anti-spam filter optimization, statistical comparisons on algorithm performance are provided on several benchmarks and a range of performance indicators. Moreover, the resulting 4-D Pareto hyper-surface is discussed in the context of binary classifier optimization.

**Keywords:** Binary classification · Three-way classification · Parsimony · Evolutionary multi-objective optimization · Parallel coordinates

## 1 Introduction

An email anti-spam system consists of a set of boolean filtering rules, that operate jointly and support spam messages detection. Discovering the relative importance of these rules and assigning the corresponding scores (weights) of each rule is a complex setup and maintenance process.

The need of frequent scores reassignment for existing rules and score settings for new rules, to keep the anti-spam filter updated and running, requires the adoption of machine learning and optimization techniques. In rule-based spam filtering, several binary classification techniques are combined in a filter, allowing for flexible creation and deployment of highly customized spam filtering. These techniques include intelligent analysis of email content, collaborative querying and information sharing on senders, deliveries and legitimacy verification. Each technique individually is not able to provide efficient classification, but their joint usage provide acceptable levels of classification quality.

Each rule corresponds to a logical test and has a score assigned for the filter operation. An email message is checked (binary classified) by each rule, and the final classification is done by comparing the sum of all the matching rule scores with a required threshold value. When the sum is above an upper threshold the message is classified as spam, when the sum is below a lower threshold the message is classified as legitimate. Otherwise it is not classified and marked as a boundary case.

Continuous creation of new ways to distribute spam, leads to the need of continuous creation of new anti-spam filtering rules and corresponding scores setting, in addition of scores updating of existing rules. Rules creation and scores setting is mainly performed manually by system administrators based on experience, applying a try-and-error approach. It is a complex process that has to take into account the existing rules knowledge base and the relative importance of rules to assign individual scores.

This process involves the analysis of thousands of rules and scores to create a complex highly customized anti-spam filter. The anti-spam filter is highly dependent on the type of organization, business or leisure domains, location, language, culture and other user specific criteria. The importance of different objectives can change depending on the context. For instance, in leisure domains false positive classifications may have less serious consequences than they would have in business domains.

A survey of literature on assisted/automatic configuration proposals is found in the literature for the SpamAssassin anti-spam filtering in [3].

The general framework of the evolutionary multiobjective optimization approach for automatic anti-spam filtering scores tuning presented here was presented by some authors of this paper in recent works [4–6]. So far the framework was tested with at most three objective functions at a time. Now, the approach is extended to a four-objective problem formulation, including three-way classification (instead of binary classification) and parsimony (or complexity reduction of the rule base system), in addition to false positive and false negative minimization objectives.

This work is structured as follows. Section 2 presents the anti-spam filter multiobjective optimization problem formulation and its relation to the machine learning perspective. In Sect. 3, the 4-objective multiobjective optimization problem formulation is presented. Section 4 details the experiments design and protocol. Then Sect. 5 presents the results analysis and discussion. Finally, in Sect. 6 the authors present the conclusions and future research directions.

## 2 Multiobjective Problem Formulation

Machine learning problems often can be formulated as multiobjective optimization problems. Maximizing classification performance metrics such as true positive rates and true negative rates, parsimony at the same time, trade-off new information and forgetting outdated one and learning details while performing model abstracting, are examples of typical machine learning trade-offs to be achieved [7].

In multiobjective optimization problems  $m$  objective functions  $f = (f_1, f_2, \dots, f_m)$  must be optimized simultaneously, such that  $f_k$ ,  $k \in \{1, \dots, m\}$  are real-valued functions evaluated for points in some decision/input space (e.g., the weights of the rules and values of thresholds to be found). Each point in the decision space maps to an  $m$  dimensional vector in the so-called objective space, containing the objective function values (e.g., false positive rate, false negative rate, complexity and coverage of a classifier).

Solutions in the objective space are only partially ordered, that is two solutions can either be in a dominance relation or they can be incomparable to each other. A solution Pareto dominates another solution if it is better (lower for minimization) or equal on all, and better on at least one objective. If for two solutions neither the first dominates the second nor the second dominates the first they are said to be incomparable. The solutions in the decision space that are not dominated by other solutions are called *efficient solutions* and together form the efficient set. The projections of these solutions into the objective space are called Pareto optimal points and together form the *Pareto front*. In general, the Pareto front of a problem with  $m$  objective functions is at most of dimension  $m - 1$ . This means that for the 4-objective problems considered in this work, the Pareto fronts will be at most three dimensional.

Multiobjective optimization following the *a posteriori approach* results in the *efficient set* and *Pareto front* – or an approximation to these sets. Selection of a

single solution among Pareto optimal ones is done by or on behalf of the decision maker according to his/her preferences [15]. The knowledge of the Pareto front also reveals insights into the structure of the essential conflict between different objective functions. This information can be useful for the classification system's designer in order to assess problem inherent trade-off or to see limitations on what can possibly be achieved with a certain classification software by tuning parameters.

In the following we seek to find approximations to Pareto fronts that arise in the context of binary classifier tuning, and, more specifically, in tuning spam classifiers. The following objective functions will be considered:

- *Minimize false negative rates*: A spam detection system's major purpose is to detect all spam messages. If a spam email is not detected this error is called a false negative.
- *Minimize false positive rates*: A spam detection system should not mistakenly classify legitimate emails, also called *ham*, as being spam emails. If an email is classified as spam but it is not a spam email, we call this error a false positive.
- *Minimize the number of unclassified samples*: Ideally a spam filter should classify all instances. However, as discussed in [8, 14], binary classification of instances as positive or negative is sometimes too strict and can result in high misclassification costs. Three-way classification can also leave an email unclassified in case of low confidence in classification. In this case only a warning or suspicious flag is provided and it is assumed that the user will then correctly classify the email. This way, especially in badly supported cases significant improvements on error rates can be achieved, albeit at the cost of additional work for the user. Advantages of the so called three-way classification approach are described in the literature as the ability to provide a more complete feedback to the users, and in this sense reducing qualitatively and/or quantitatively the misclassification rate.

Difficult instances to be classified (boundary cases) are marked as *unclassified* and forwarded to the user for further examination. This way the maximization of coverage is formulated as the minimization of the number of unclassified samples.

- *Minimize complexity*: In previous work on anti-spam filter optimization [8], it was observed that many rules were not participating in the classification process and those with very small scores only marginally influenced the classification results. This observation suggests that in addition to minimizing the occurrence of false positives and false negatives, the complexity of the anti-spam filter (or its parsimony) can also be optimized. For the anti-spam filtering case, we measure parsimony as the minimum number of rules with score different from zero, that support a specific classification quality.

*Several conflicts* between these four objectives can be identified: Firstly, there is a conflict between the false positive rate and the false negative rate. In extreme cases of anti-spam systems tuning/configuration the system can always have a zero false positive rate, that is a zero rate of legitimate messages lost.

This rate is obtained for instance if no instance is marked as spam. Usually this comes at the expense of higher false negative classifications. In the other extreme, system tuning may classify a large number of instances as being spam. In the most extreme case it might even classify all emails as spam. This will minimize the false negative rate but comes at the expense of more legitimate emails being dismissed (high false positive rate).

Secondly, there is typically a conflict between a high coverage of the classifier and the error rates. Obviously, every instance that is not examined by the user but instead automatically classified can potentially lead to misclassification costs (increment of false positive and false negative rates). On the other hand the effort of the user and his/her exposure to suspicious emails should be minimized.

Finally, there is a conflict between the error rates and the complexity of a classifier. It goes without saying that extremely simple classifiers might have a low accuracy while more complex classifiers can capture more complex rules and therefore potentially yield classifiers with lower error rates. Here, it should be noted that from a certain level onwards adding complexity to the classifiers might not anymore yield to improvements in terms of error rates and even can be counterproductive due to overfitting. To identify this critical complexity level which will be obtained as an upper bound of the Pareto front, can be a valuable output of multiobjective optimization.

The conflict between coverage and complexity seems to be of a somewhat more complex nature and we will assess it by means of empirical results. In our multiobjective problem formulation, 4-objectives are considered to be minimized, false positive rate, false negative rate, unknown classifications rate and parsimony.

### 3 Quadcriteria Optimization Methods

The optimization or tuning of anti-spam filtering systems using rule ensembles as classifiers is difficult to be accomplished with derivative-based deterministic optimization techniques, because the behaviour of rules can be highly complex, non-smooth and non-linear. This is why instead metaheuristics are used for this task. Among the metaheuristics used for multiobjective optimization, evolutionary multiobjective optimization algorithms (EMOAs) are allegedly the most frequently and best studied approaches.

In our study, state-of-the-art EMOAs were selected for the anti-spam filtering optimization problem, namely, NSGA-II [12], SPEA2 [16] and SMS-EMOA [17]. These methods are representative for the recently most relevant generational Pareto-based, steady state Pareto-based and indicator-based EMOA approaches. Testing representative methods from different optimization strategies groups, allow the study of their behaviour and performance for the anti-spam filtering problem.

In the current study the following formulation for the anti-spam filtering optimization is adopted. A four-objective ( $fnr$  - false negative rate,  $fpr$  - false positive rate,  $ur$  - unclassified rate,  $cr$  - complexity rate) binary-real representation decisions variable formulation (a real valued scores vector and a binary string/vector

for rules activation/deactivation), aiming at classifier performance and complexity rate improvement is proposed. Minimizing these objectives, means reducing the number of spam messages not identified by anti-spam filtering techniques, reducing the number of legitimate messages classified as spam by mistake, reducing the number of unclassified messages, and reducing the number of rules used in the classifier.

The anti-spam filtering problem is formulated here as a multiobjective optimization problem on a mixed-integer decision space and with normalized objective function values in the range  $[0; 1]$  that are all to be minimized. These are false negative rate ( $f_1$ ), false positive rate ( $f_2$ ), number of unclassified instances divided by the number of instances ( $f_3$ ) and number of rules used by the classifier divided by the total number of available rules ( $f_4$ ).

Minimization is assumed for all objectives, evaluated in decision space with a vector of decision variables,  $w = (w_1, w_2, \dots, w_n)$ , with  $n$  being the total number of rules. The output for all  $n$  rules of a filter is weighted by these in order to compute a final score. It is also possible to assign a negative score to a rule. The individuals of initial population are generated randomly with scores in the  $[-5; 5]$  range. New individuals are also generated by variation operations in the same range.

While a real decision variables vector is used for the representation of the anti-spam rules scores settings in the interval  $[-5; 5]$ , a binary vector  $b = (b_1, \dots, b_n) \in \{0, 1\}^n$  of decision variable, with each bit representing one rule for the algorithm to activate (or deactivated) rules according to variation and selection operators along the evolutionary process, allows the algorithm to activate/deactivate rules and assess their relevance in the classification process. In addition a lower and an upper threshold  $t_1$  and  $t_2$  are optimized by the algorithm. Both thresholds are set in the interval  $[0; 1]$ , with lower threshold being always lower than the upper threshold. If the sum of rules' score that match an email message is below  $t_1$  it is classified as ham, if it is above  $t_2$  it is classified as spam, and otherwise it remains unclassified. Therefore, the values of  $t_1$  and  $t_2$  will have a direct impact on the number of unclassified samples but also on the misclassification costs.

A normalized counting of false negatives, false positives, unknown messages and number of active rules is adopted, leading all four objectives to assume values in the range of  $[0; 1]$ , as described in Eqs. 1, 2, 3 and 4.

$$fnr(w, b, t_1, t_2) = fn(w, b, t_1, t_2) / TotalNumberOfSpamMessages \rightarrow \min \quad (1)$$

$$fpr(w, b, t_1, t_2) = fp(w, b, t_1, t_2) / TotalNumberOfHamMessages \rightarrow \min \quad (2)$$

$$ur(w, b, t_1, t_2) = \#unclassified(w, b, t_1, t_2) / TotalNumberOfMessages \rightarrow \min \quad (3)$$

$$cr(w, b, t_1, t_2) = \sum_{i=1}^n b_i / TotalNumberOfRules \rightarrow \min \quad (4)$$

with  $w \in [-5, 5]^n$  being the rules weights,  $b \in \{0, 1\}^n$  being the rule activation variables ( $n = TotalNumberOfRules$ ), and  $(t_1, t_2)$  being the lower and upper

threshold. Finally, in order to ensure that the thresholds are feasible we introduce the constraints:

$$0 \leq t_1 \leq t_2 \leq 1 \tag{5}$$

## 4 Experimental Setup

In the next section, we describe the experimental setup and the performance evaluation metrics used for the experiments results analysis. SpamAssassin is the anti-spam filtering system adopted in our experiments due to wide adoption by the open source community, the research community on anti-spam systems, its wide commercial usage, and available email corpora. The SpamAssassin corpus used in our experiments is composed of 9349 email messages, 2398 spam and 6951 legitimate messages [9].

The experiments phase was performed following the default spam filter configuration present in the Debian GNU/Linux Squeeze distribution running SpamAssassin 3.3.1 [10]. Filtering rules scores range fall under the interval  $[-5; 5]$ .

From the 2440 rules available in SpamAssassin distribution, only those fitting at least one message in the dataset are considered in the optimization experiments. Actually, only 330 rules fit email messages and only those have been considered in the optimization process.

Experiments were performed with jMetal [11] version 4.5, an optimization framework for the development of multiobjective metaheuristics in Java. A jMetal RealBinary encoding decision variables scheme was used where the chromosome is constituted by an array of real values in the interval  $[-5; 5]$  and a bit string. The length of the chromosome is determined by the number of anti-spam filtering rules effectively used (330). Each rule is associated with a real value score in the  $[-5; 5]$  interval and a one bit in the chromosome. If the  $i^{\text{th}}$  bit is 0 the  $i^{\text{th}}$  rule is ignored, and otherwise the rule is active and considered by the spam classifier with the  $i^{\text{th}}$  corresponding real value score. Messages are classified as spam when the sum of scores of active rules that match the message is equal or greater than the defined threshold value.

NSGA-II, SPEA2 and SMS-EMOA algorithms stopping criteria are set to a maximum of 25000 function evaluations. The SBX single point crossover and polynomial bit flip mutation operators are used as the variation operators in the experiments, with crossover probability 0.9 and mutation probability  $1/n$ , where  $n$  is the number of anti-spam filtering rules. Population size is 100 individuals for all algorithms, archive size 100 for SPEA2 and offset set to 100 for SMS-EMOA. In order to obtain robust performance statistics, all stochastic algorithms perform 30 independent runs.

Although performance assessment of multiobjective optimization algorithms constitutes a complex task, involving outcome quality assessment, computing resources usage, analysis of several runs of the stochastic based algorithms, in addition of not having an absolute optimal for comparison purposes, a set of well established performance indicators can be used and provide guidance for multi-objective optimization algorithms performance assessment. General performance

criteria include accuracy (or convergence) which measures how close the solutions are to some known optimal, coverage (or uniformity) which measures how many non-dominated solutions are generated and how well they are distributed, and variance (or spread) which measures, for every objective, the maximum range of non-dominated front covered by the generated solutions.

In order to evaluate and compare approximation sets from multiple runs of two or more stochastic multiobjective optimizers, complementary techniques must be combined.

Firstly, we use the 4-D hypervolume (HV) indicator as a measure for the size of the subspace that is dominated by the Pareto front approximation. This dominance compliant quality indicator has favorable theoretical properties (Pareto compliant) and high values indicate a diverse set of solutions located close to the true Pareto front. For the HV indicator a reference point is required. This choice is problem specific and it is important that all points in the Pareto front approximation are always dominated by the reference point.

The hypervolume indicator is accompanied by the SPREAD indicator. This indicator shows how far Pareto front, spreads in objective space, or decision space. The larger the spread of the Pareto front is, the wider range of values on objectives it covers. This indicator is not Pareto compliant but is useful to understand the geometry of the obtained Pareto fronts, in particular whether also extreme parts are well covered.

A graphical representations of the 4D Pareto front is done by means of graphical parallel coordinates. The objective functions determine the four parallel axis of this diagram. Each solution in the Pareto front representation is represented as a polyline. The diagram we used was made available by the statistical software **System R**. The Pareto front that we visualized is the reference Pareto front, that is the Pareto front obtained from the non-dominated subset of the union of all Pareto front approximations obtained in the independent runs. The main purpose of this graphic is to provide insights into the trade-offs of the anti-spam filtering problem and to obtain a good compromise solution.

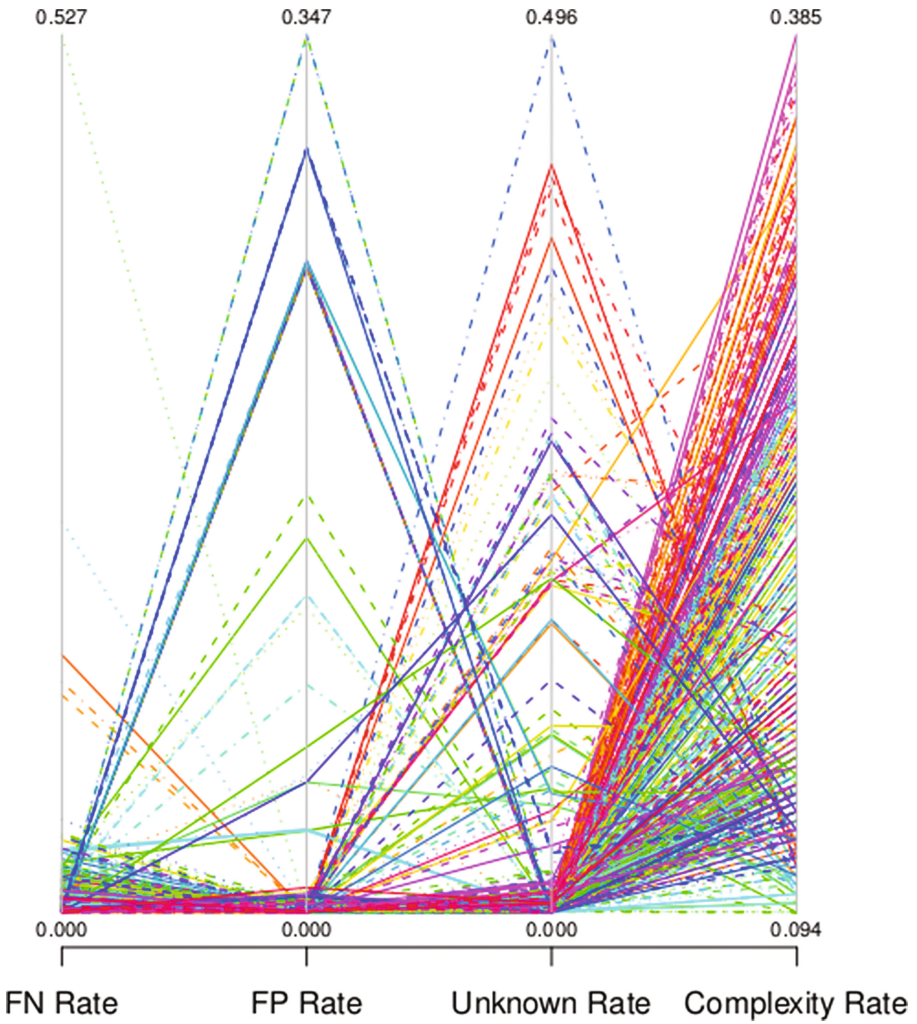
## 5 Results Analysis

The results presented in this section outline NSGA-II, SPEA2 and SMS-EMOA algorithms behavior and performance on the 4-objectives anti-spam filtering problem formulation.

Figure 1 shows the reference Pareto front, in other words, the best individuals of all algorithms of all runs. It is clear from this figure that the anti-spam filtering system may have very low  $fnr$ ,  $fpr$  and  $ur$ , while using less than 38.5% of the 330 rules that match some message in the email corpus. Some solutions could reach  $fnr$ ,  $fpr$ , and  $ur$ , all falling under 1%, using around 16% of the anti-spam filtering rules. The finding is confirmed by the plot in the scatterplot matrix in Fig. 2, showing clearly the range in which there is a conflict between  $fnr$  and  $complexity$ .

The experiments results confirm the trend of increasing the number of rules in the anti-spam filtering system (due to the need of dealing with new spam

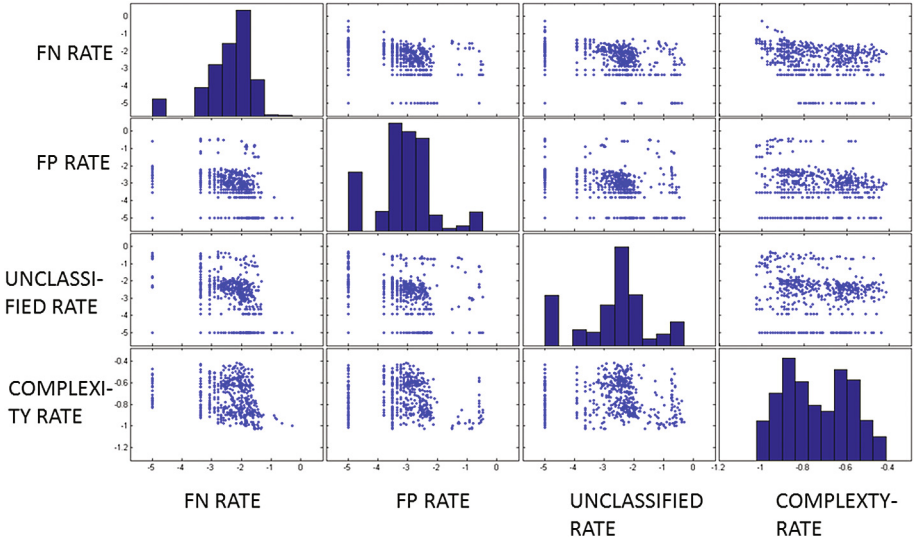




**Fig. 1.** Reference front parallel coordinates plotting for four objectives anti-spam problem formulation.

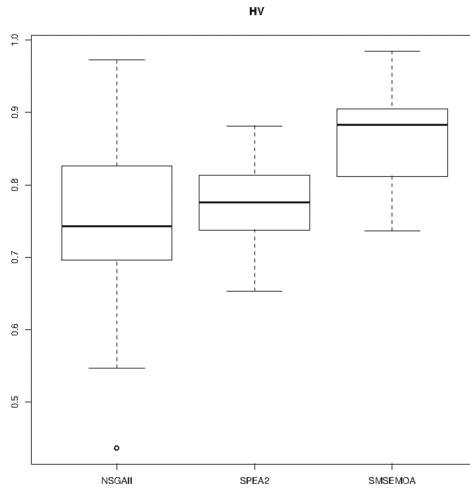
message), has a marginal effect on progressively decreasing the relevance of the rules set in the classifier quality, while having a considerable impact on classifier complexity, and therefore on the computational effort. This suggests that in order to introduce new rules, multiobjective optimization should be used to avoid redundancy or near-redundant rules in a rule ensemble.

From the experiments it is also possible to conclude that the 4-objectives classifier is able to reach high levels of accuracy with respect to the false negative rate, false positive rate and unclassified rate, even when only a small fraction of filtering rules are activated.

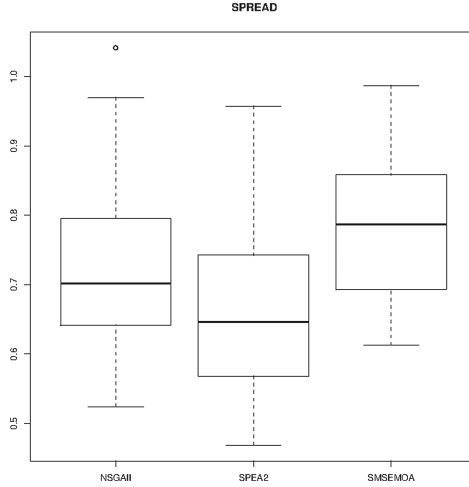


**Fig. 2.** Scatterplot matrix of combined set of results. Of all values the logarithm with basis 10 is taken.

Besides, our results provide first trends on the question which algorithms are most suitable for the quadcriteria optimization of binary classifiers. Boxplots depicting statistics on the multicriteria performance indicators hypervolume and spread, are shown for the algorithms under consideration, namely NSGA-II, SPEA2 and SMS-EMOA (Figs. 3 and 4). The boxplots represent graphically the



**Fig. 3.** Hypervolume boxplot for four objectives anti-spam problem formulation.



**Fig. 4.** SPREAD boxplot for four objectives anti-spam problem formulation.

median, quartiles, and outliers of the statistics on each algorithm (30 runs, each). The comparison of NSGA-II, SPEA2 and SMS-EMOA is done with respect to the reference Pareto front.

From the boxplots we conclude that SMS-EMOA is clearly the best performing algorithm of all with respect to both hypervolume and spread indicators. The second best performing algorithm with respect to hypervolume indicator is SPEA2, and the second best algorithm with respect to spread indicator is NSGA-II. It should be noted, however, that SMS-EMOA is also the most demanding algorithm in terms of required CPU-time. For the same computation resources used in our experiments to run all algorithms, a SPEA2 run duration could be measured in a scale of seconds, NSGA-II in a scale of minutes and SMS-EMOA in a scale of dozens of minutes. The hypervolume calculation of jMetal SMS-EMOA implementation used in our experiments, follows traditional algorithms approaches, which are relatively slow for 4-D hypervolume [18]. Recent faster implementations are now available [19, 20] and will be adopted in the future for 4-D hypervolume calculations.

## 6 Conclusions and Future Work

Introducing three-way classification and parsimony as additional and simultaneous optimization objectives, in addition to false negatives and false positives minimization, revealed important improvements that can be achieved in modern anti-spam filtering systems.

For the proposed 4-objectives problem formulation, it was found that a good performance can be achieved with a small number of rules being used by the anti-spam filtering classifier. It was observed that from 330 rules that match

messages in the SpamAssassin data corpus, only 16% to 38.5% of rules are needed to achieve classification error rates on all the other three objectives (fnr, fpr, ur) under 1%.

The authors will take this results as the basis for further anti-spam filtering and classifier optimization research, heading in two main directions. First, other multiobjective and many-objectives optimization algorithms of high potential for the anti-spam filtering type of problems will be studied and explored (e.g. MOEA/D and NSGA-III), and also tailor made approaches for classification such as CH-EMOA [1,8] or mixed integer optimization [2]. Secondly, analysis of the rules that reveal highest contributions for the classification process will be addressed, in order to assess not only quantitative classifier complexity, but also to explore the nature of the rules most frequently present in the best solutions. Knowledge exploration raised by this analysis was introduced by the authors in [13]. There the authors point research hypothesis and directions with respect to rules relative relevance analysis, and anti-spam rules automatic generation guided by knowledge extracted by the means of multiobjective optimization techniques.

Finally, it will be of interest to consider newly proposed visualization techniques for 4-D Pareto fronts by Tuša and Filipič [21], in order to gain additional insight into the structure of the Pareto front data.

**Acknowledgements.** This work was partially funded by the [14VI05] Contract-Programme from the University of Vigo. Iryna Yevseyeva acknowledges Engineering and Physical Sciences Research Council (EPSRC), UK, and Government Communications Headquarters (GCHQ), UK, for funding Choice Architecture for Information Security (ChAISE) project EP/K006568/1 as a part of Cyber Research Institute.

## References

1. Wang, P., Emmerich, M., Li, R., Tang, K., Bäck, T., Yao, X.: Convex hull-based multi-objective genetic programming for maximizing receiver operating characteristic performance. *IEEE Trans. Evol. Comput.* **19**(2), 188–200 (2015)
2. Li, R., Emmerich, M.T., Eggermont, J., Bäck, T., Schütz, M., Dijkstra, J., Reiber, J.H.: Mixed integer evolution strategies for parameter optimization. *Evolu. Comput.* **21**(1), 29–64 (2013)
3. Basto-Fernandes, V., Yevseyeva, I., Méndez, J.R.: Anti-spam multiobjective genetic algorithms optimization analysis. *Int. Resour. Manage. J.* **26**(1), 54–67 (2012)
4. Yevseyeva, I., Basto-Fernandes, V., Méndez, J.R.: Survey on anti-spam single and multi-objective optimization. In: Cruz-Cunha, M.M., Varajo, J., Powell, P., Martinho, R. (eds.), *ENTERprise Information Systems. Communications in Computer and Information Science*, vol. 220, pp. 120–129. Springer, Heidelberg (2011)
5. Basto-Fernandes, V., Yevseyeva, I., Méndez, J.R.: Optimization of anti-spam systems with multiobjective evolutionary algorithms. *Int. Resour. Manage. J.* **26**, 54–67 (2012)
6. Yevseyeva, I., Basto-Fernandes, V., Ruano-Ordás, D., Méndez, J.R.: Optimising anti-spam filters with evolutionary algorithms. *Expert Syst. Appl.* **40**(10), 4010–4021 (2013)

7. Jin, Y.: Multi-objective Machine Learning. Studies in Computational Intelligence. Springer, Heidelberg (2006)
8. Zhao, J., Basto-Fernandes, V., Jiao, L., Yevseyeva, L., Maulana, A., Li, R., Bäck, T., Emmerich, M.T.M.: Multiobjective optimization of classifiers by means of 3-d convex hull based evolutionary algorithm, ARXIV Computer Science abs/1412.5710 (2014). <http://arxiv.org/abs/1412.5710>
9. The Apache SpamAssassin Project - SpamAssassin public corpus (2005). <http://spamassassin.apache.org/publiccorpus>
10. SpamAssassin Team: The apache spamassassin project (2011). <http://spamassassin.apache.org/>
11. Durillo, J.J., Nebro, A.J.: jMetal: a java framework for multi-objective optimization. *Adv. Eng. Softw.* **42**, 760–771 (2011)
12. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
13. Basto-Fernandes, V., Yevseyeva, I., Frantz, R.Z., Grilo, C., Daz, N.P., Emmerich, M.: An automatic generation of textual pattern rules for digital content filters proposal, using grammatical evolution genetic programming. *Procedia Technol.* **16**, 806–812 (2014)
14. Yao, Y.: The superiority of three-way decisions in probabilistic rough set models. *Inf. Sci.* **181**(6), 1080–1096 (2011)
15. Miettinen, K.: *Nonlinear Multiobjective Optimization*. Springer, New York (1999)
16. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength Pareto evolutionary algorithm. In: *Proceedings of EUROGEN 2001, Athens Greece*. CIMNE, Barcelona (2001)
17. Emmerich, M., Beume, N., Naujoks, B.: An EMO algorithm using the hypervolume measure as selection criterion. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) *EMO 2005*. LNCS, vol. 3410, pp. 62–76. Springer, Heidelberg (2005)
18. While, L., Bradstreet, L., Barone, L.: A fast way of calculating exact hypervolumes. *IEEE Trans. Evol. Comput.* **16**(1), 86–95 (2012)
19. Emmerich, M.T.M., Fonseca, C.M.: Computing hypervolume contributions in low dimensions: asymptotically optimal algorithm and complexity results. In: *Evolutionary Multi-Criterion Optimization*. Springer, Heidelberg (2011)
20. Guerreiro, A.P., Fonseca, C.M., Emmerich, M.T.: A fast dimension-sweep algorithm for the hypervolume indicator in four dimensions. In: *CCCG*, pp. 77–82 (2012)
21. Tušar, T., Filipič, B.: Visualizing 4D approximation sets of multiobjective optimizers with projections. In: *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, pp. 737–744. ACM (2011)