Neural Comput & Applic (2016) 27:2315–2331
DOI 10.1007/s00521-015-2069-7

Supervised classification of spam emails with natural language stylometry

Rushdi Shams[1] · **Robert E. Mercer**[1]

Received: 12 March 2015 / Accepted: 26 September 2015 / Published online: 3 November 2015

Abstract Email spam is one of the biggest threats to today's Internet. To deal with this threat, there are long-established measures like supervised anti-spam filters. In this paper, we report the development and evaluation of SENTINEL—an anti-spam filter based on natural language and stylometry attributes. The performance of the filter is evaluated not only on non-personalized emails (i.e., emails collected randomly) but also on personalized emails (i.e., emails collected from particular individuals). Among the non-personalized datasets are CSDMC2010, SpamAssassin, and LingSpam, while the Enron-Spam collection comprises personalized emails. The proposed filter extracts natural language attributes from email text that are closely related to writer stylometry and generate classifiers using multiple learning algorithms. Experimental outcomes show that classifiers generated by meta-learning algorithms such as ADABOOSTM1 and BAGGING are the best, performing equally well and surpassing the performance of a number of filters proposed in previous studies, while a random forest generated classifier is a close second. On the other hand, the performance of classifiers using support vector machine and Naïve Bayes is not satisfactory. In addition, we find much improved results on personalized emails and mixed results on non-personalized emails.

Keywords Spam classification · Natural language processing · Stylometry · Supervised machine learning · Text classification · Computational linguistics · Text mining · Performance evaluation

1 Introduction

Spam emails are both unsolicited and massively posted commercial and non-commercial emails. The effects of spams include but are not limited to the loss of individual and organizational productivity, chaotic user inboxes [23], Internet speed degradation, and misappropriation of personal information. The onset of spams has grown exponentially. In March 2013, for instance, approximately 100 billion spams were received by users everyday, which is 98 % more than that from the previous quarter [12]. To date, most anti-spam filters are supervised tools that trap spam emails by detecting some mundane patterns learned during their training [6, 17]. Usually, these patterns are generated from attributes collected either from email headers [23, 42] or from email text [32]. The filters generate classifiers from these attributes using algorithms such as Naïve Bayes (NB) [23, 28, 32], random forest (RF) [36], support vector machine (SVM) [48], and neural networks (NN) [30].

Spammers almost always introduce new techniques to bypass header-based anti-spam filters by originating their menacing emails from *white-list* sources. Spoofing text-based filters, on the other hand, require substantial efforts by the spammers. For example, one of the easiest ways to detect spams is to search for *spam terms* in the email text. Spammers then simply change their vocabulary to circumvent these text-based filters, as pointed out by Orăsan and Krishnamurthy [34]. The authors reported that because of the cleverly removed spam terms, text-based filters tend to underperform over time. As a result, organization

✉ Rushdi Shams
rshams@alumni.uwo.ca

[1] Cognitive Engineering Laboratory, Department of Computer Science, The University of Western Ontario, London, ON, Canada

servers and personal inboxes are still overwhelmed by spam emails, and therefore, classifying spam emails remains a challenging machine learning task.

Metsis et al. [32] and Ma et al. [30] show that natural language attributes of email subject and body have substantial ability to discern spams and hams. These attributes are based on the importance of a term in the email (i.e., term frequency or TF), its rarity in the email dataset (i.e., inverse document frequency or IDF), and their normalized form (TF·IDF). The advantage of exploiting these attributes is that they perform well on personalized filters (see [32, 50]). However, they are found to underperform on non-personalized emails [28]. Furthermore, the calculation of TF·IDF is done on the *word-count space*, so for each newly arrived email, this attribute needs to be re-calculated. The re-calculation, if not done incrementally, can introduce latency. The time sensitivity of the data described earlier is also a reason not to rely on these *term-based* attributes. Therefore, along with these, we need to exploit other natural language attributes that are more pervasive in nature. Among such pervasive attributes are those that are related to the writer's writing style. Afroz et al. [3] and Iqbal et al. [24] showed that using writer stylometry and choice of words, authorship of documents can be identified. This finding can be aligned with the problem in our hand nicely because often, spammers write bulk emails to people by impersonating as a bank manager, business partner, etc., which they are not. These attributes present in the stylometry of emails are still unexplored in our problem domain. Among the stylometry attributes are text readability, grammar and spelling mistakes, and the use of function and content words.

In this paper, we report the development and evaluation of an anti-spam filter named SENTINEL on both non-personalized and personalized emails. The bulk of the attributes used by SENTINEL are natural language attributes related to writer stylometry. Three standard, non-personalized email datasets CSDMC2010, SpamAssassin, and LingSpam, and six personalized datasets in the Enron-Spam collection are used to train and test the filter. SENTINEL generates classifiers using random forest (RF), support vector machine (SVM), and Naïve Bayes (NB), and two meta-algorithms called ADABOOSTM1 and bootstrap aggregating (BAGGING) using RF as the base classifier. Results show that ADABOOSTM1 and BAGGING perform almost equally the best, while the performance of RF is a close second. Interestingly, the performance of SVM depends on the quantity of spams in the training set. NB has the poorest results of all—which is understandable as most of the attributes are not independent. Comparisons show that the performance of SENTINEL surpasses that of a number of filters proposed in previous studies. We also find that the results on personalized emails are better to some extent compared to those on non-personalized emails. Because writer stylometry is language independent, SENTINEL may be an excellent means to classify spam emails in any language.

The paper is organized as follows: In Sect. 2, we discuss some related work in the domain. Section 3 outlines the attributes, learning algorithms, evaluation measures, and experimental procedure and describes the datasets. Results are discussed in Sect. 4. Finally, Sect. 5 concludes the paper with directions to possible future work.

2 Related work

As we are interested to see how our filter performs on both non-personalized and personalized emails, in this section we describe work aimed at classifying emails from non-personalized and personalized email datasets.

2.1 Non-personalized filters

Non-personalized filters are developed to safeguard an email server from spam emails. These filters are so called because their development and evaluation are done using randomly collected spam and ham emails. The spam emails are generally collected from multiple spam traps, while the ham emails are contributed by different individuals and thus do not characterize any particular user. Among the popular non-personalized datasets are CSDMC2010, SpamAssassin, and LingSpam.

Qaroush et al. [36] investigated the performance of their anti-spam filter on the CSDMC2010 dataset. Their filter is capable of generating multiple spam classifiers, but unlike our approach, their predictive models are based on traditional attributes found in email headers. Among the classifiers, they found that the one generated using RF outperforms the rest namely NB, Bayesian Network, and SVM. The authors, like many, conceived the spam classification performance as a cost-sensitive analysis and used both conventional and cost-sensitive evaluation measures. Likewise, Yang et al. [47] developed two filters—one of which is based on SVM, while the other is generated using NB. The filters extract attributes from email content using a novel attribute selection method based on a binomial distribution hypothesis testing called *Bi-Test*. When they compared the filter performances on the CSDMC2010 and LingSpam datasets, they found that the filters perform better on the former.

Ma et al. [30] developed an NN-based filter that uses 328 text attributes. On the SpamAssassin dataset, their reported accuracy of 0.920 was reasonably good. The filter developed by Sirisanyalak and Sornil [43] also uses an NN classifier. The attributes chosen for their experiment are

related to artificial immune systems (AIS). The filter has been reported to be accurate about 92 % of the time on the SpamAssassin dataset. Prior to that, Bratko et al. [7] measured their filter performance on the SpamAssassin dataset with a cost-sensitive evaluation. They found much improved results using data compression techniques.

From our literature survey, we found that the reported performances of filters are relatively low on the LingSpam dataset. Prabhakar and Basavaraju [35], for instance, applied K-NNC and a data clustering algorithm called BIRCH on this dataset. Their TF·IDF-based filter, compared to the filters tested on other datasets, achieved relatively low scores. Cormack and Bratko [14] also found similar results on LingSpam when they experimented with four popular content-based anti-spam filters in the domain.

In addition to the supervised methods summarized above, there have been attempts to detect spam email using semi-supervised learning. Our goals here, supervised learning and the introduction of new attributes into the spam detection problem, are somewhat different, so we will concentrate on those works that have some connection to our work. Xu et al. [46] apply a combination of active learning and semi-supervised learning on the TREC07 dataset, a non-personalized email dataset. Improvements were noted compared to a Naïve Bayes classifier. Meng et al. [31] use random forests, an ensemble classifier, and disagreement-based semi-supervised learning on three nonstandard non-personalized email datasets. Their best results were FPR: .02, FNR: .09, and AUC: .979.

2.2 Personalized filters

Unlike non-personalized filters, personalized anti-spam filters are aimed at protecting particular individuals. These filters learn through the choice of idioms and phrases, writing style, variation of sentences, etc., of the legitimate users. Therefore, to train and test these supervised filters, personalized email data like those in the Enron-Spam collection are preferred.

The earliest of the personalized anti-spam filters were simple and computationally efficient as they used a simple NB classifier. These filters exploited the simple Bayesian framework, a set of rules, and both header and content attributes [38]. Likewise, they attained low misclassification rates on several datasets. The initial success of these filters led several others to emerge, and they simply replaced the handcrafted rules with predictive models (see, e.g., [18, 25, 28, 32]). Several variations of these filters include *multivariate Bernoulli*, *bag of words* (*BoW*), *and multinomial boolean*. Metsis et al. [32], for instance, used 3000 multinomial boolean TF attributes—the results reported by this study were highly impressive. Two years later, they achieved even better results by using the *transformed* TF attributes [26]. Nowadays, the performance of an NB filter is considered to be the de facto standard to compare newly developed personalized filters.

SVM filters are efficient for training in much the same way as NB filters; nevertheless, it was found by Guzella and Caminhas [19] that the filters need incremental training to reduce latency. Furthermore, SVM filters can handle large attribute sets and in many cases attribute selection is not necessary [48]. Most of the benchmark SVM filters use the frequency-based *linear kernel*. As well, there are several SVM filters that use updatable supervised clustering algorithms like the one reported by Haider et al. [20]. In contrast, the disadvantages of using SVM-based filters include high misclassification rate, especially for personalized emails.

The advantages of using *meta-learning* anti-spam filters are manifold. Firstly, when a base learner with a sufficient tree depth is used, they achieve improved misclassification rates on many public datasets [9]. Secondly, these filters are resistant to the problem of overfitting, and therefore, they gain more *appropriate* accuracy even on an unbalanced dataset[1] [21]. However, these filters have the weakness of *ensemble* learning—the interpretation of results is difficult. Studies show that meta-learning filters outperform many decision tree, NB, and SVM filters. Surprisingly, the use of meta-learning filters is still not as widespread as NB and SVM filters.

Over the last decade, Artificial Immune System-based anti-spam filters have become a popular choice. These filters use *detectors* on the email for pattern matching. Detectors are in fact regular expressions that are defined a priori. Each detector is given a *weight* that is adjusted as the filters recognize a pattern in a given email. The weights of the matching detectors are then used (usually combined) to determine the email's class label. Notable immune system-based filters are reported by previous studies [1, 2, 50]. These filters seek specific *signatures* in the emails—this is why they are widely used in personalized email classification where similar patterns can be found in the writing style of the person the filter intends to protect. Also, many of these filters are able to deal with *concept drift*—the gradual or abrupt change in thematic context over time such as new advertisement themes in spam emails.

In addition to the supervised methods summarized above, there have been attempts to detect spam email using semi-supervised learning. Our goals here, supervised learning and the introduction of new features into the spam detection problem, are somewhat different, so we will concentrate on those works that have some connection to our work. Cheng and Li [11] use a semi-supervised classifier ensemble (SVM and Naïve Bayes) on personalized

[1] Most of the public email datasets are imbalanced [19].

Table 1 The brief summary of the attributes used in our study: their categories, quantity, and description

Category	Quantity	Description
Word-level attributes	11	Spam words, alphanumeric words, function words, verbs, TF [26, 32], TF·ISF, TF·IDF
Error attributes	3	Grammar and spelling mistakes
Readability attributes	23	Simple and complex words, and their TF·IDF, Fog index, simple and inverse Fog index, Smog index, Flesch reading ease score, Forcast, Flesch–Kincaid score, email length, word length
HTML attributes	3	Regular and anchor tags

email. The method is tested on the ECML/PKDD 2006 Discovery Challenge dataset [5] and when combined with a rare word distribution demonstrates AUC values between .92 and .95 depending on the personalized dataset. Cheng and Li [10] combine supervised and semi-supervised learning for personalized spam filtering. The method also is tested on the ECML/PKDD 2006 Discovery Challenge dataset and demonstrates AUC values between .861 and .992 depending on the personalized dataset. The work of Wang and Shen [45] is mainly concerned with developing a different type of semi-supervised learning algorithm, but it is included here because one of the test cases is the UCI Machine Learning Repository Spambase Dataset, a personalized email dataset. The features of this dataset are based on the frequency of appearance of words from a word set and orthographic features related to uppercase letters. They show that Transductive SVM [44] does more poorly than SVM on this dataset and that their algorithms perform better than SVM and ψ-learning [41] which they generalize. Mojdeh and Cormack [33] witness a significantly degraded performance by spam filters built with semi-supervised learners on the personal email of TREC 2005 [15] and TREC 2007 [13] versus the non-personalized email of the ECML/PKDD Discovery Challenge. Also, the filters built with semi-supervised learners performed more poorly than those built with supervised learners on the TREC datasets.

3 Methods and materials

3.1 Attributes

Each email in our experiment is represented as $(\mathbf{x}, y)$, where $\mathbf{x} \in \mathbb{R}^n$ is a vector of n attributes and $y \in \{\text{spam}, \text{ham}\}$ is the label of the email. In our study for the LingSpam dataset, we explored 36 attributes with one class attribute, and therefore, $n = 37$. For the CSDMC2010 and SpamAssassin datasets, however, we considered three more attributes related to HTML tags because the datasets contain them since they are not preprocessed (Table 4). Therefore, for these two datasets, $n = 40$. Finally, for the Enron-Spam collection, in addition to the 37 attributes we used for the LingSpam dataset, we further explored a term frequency (TF) attribute previously utilized by Sahami et al. [26] and Metsis et al. [32]. Therefore, in our study for the Enron-Spam collection, $n = 38$. Table 1 summarizes the attributes used in our experiment.

3.1.1 Word-level attributes

We treated each email as a bag of words to calculate the word-level attributes. We curated a dictionary that comprises 381 spam words. The set of spam words is a superset of the dictionary entries for different anti-spam filters. Using this dictionary, we counted the frequency of spam words in the emails. This attribute is inspired by the interesting findings of Graham [18] who showed that merely finding the word *click* in the emails can detect 79.7 % of spam emails in a dataset with only 1.2 % ham misclassification rate. Our other attributes in this category include the frequency of alphanumeric words, verbs, and function words. To identify the verbs in the emails, we used the Stanford POS Tagger.[2] To identify function words, we used a standard English function word dictionary.

In this category of attributes, we also included the term frequency (TF) attribute used by at least two NB filters developed by Sahami et al. [26] and Metsis et al. [32]. For each term which appears in at least four training emails, *information gain* scores are computed according to the study conducted by Androutsopoulos et al. [4]. Of all the terms in the dataset, the 3000 with the highest scores are considered as the TF *vector* for all the emails. Thereafter, should any of these 3000 terms be found in an email, the term's corresponding frequency value in the TF vector is first incremented and then normalized. Finally, the normalized value is used to calculate the probability score of the email according to Bayes' theorem. The overall method is well documented by Sahami et al. [26]. Of note, the TF attributes work well on personalized emails according to Metsis et al. [32] and are less important for non-

[2] Downloadable at http://nlp.stanford.edu/software/tagger.shtml.

personalized emails. Therefore, in this study, this attribute is only extracted from the personalized emails in the Enron-Spam collection.

Furthermore, we computed the TF·ISF of each email. Here, the term frequency (TF) is the commonly used *square root* of the frequency of a term t in a sentence in an email M, while the inverse sentence frequency (ISF) is a relative measure of whether t is common or rare in the other sentences in M. This attribute is meant to reflect how important each t is to an M. In addition, the measure controls for the fact that in an email, some terms are generally more common than others. We also used the TF·IDF of each email as an attribute. It is a numerical statistic that reflects how important a term t is to an email M in the dataset D to which M belongs. The TF·IDF value increases proportionally to the number of times t appears in the email M, but is offset by the frequency of t in the dataset D. The definition of TF is the same above, while inverse document frequency (IDF) measures whether a given term t is common or rare in the remaining emails in the dataset D.

3.1.2 Error attributes

Our next set of attributes is related to the grammar and spelling errors present in the emails. For each email, we simply counted the frequency of grammar and spelling errors using a Java API called LanguageTool.[3] By summing up the values of these two attributes, we introduced a third attribute in this category named *Language Errors*. It is to be noted that these attributes are normalized separately for spams and hams. Traditional attribute normalization without considering the class labels of the instances is not followed because it is expected that the difference between the values of the attributes for spams and hams is large.

3.1.3 Readability attributes

Readability is a measure of the difficulty of reading a sentence, a paragraph, or a document. At the heart of readability lies the notion of simple and complex words. Simple words are those that have at most two syllables, while complex words contain three or more syllables. Since both types of words have a significant contribution for text readability, we have included both in our attributes. Five standard scores use these two types of words to determine the readability of a given text: Fog index, Smog index, Flesch reading ease score, Forcast, and Flesch–Kincaid index. For each email, all five readability scores are computed and used as attributes. Also, among the scores, *Fog index* measures the relative use of complex words in a document. We modified the Fog index formula to measure the relative use of simple words in a document and considered this as an attribute, too. Furthermore, we considered the arithmetic inverse of Fog index as another attribute. The other attributes in this category include email length (i.e., total sentences in an email), average word length (i.e., total syllables over total terms in an email), and TF·IDF of the set of simple and complex words. The details regarding these attributes can be found elsewhere [40].

3.1.4 HTML attributes

Metsis et al. [32] suggested not using the tracking of HTML tags to classify emails because examining a phishing `url` can sometimes lead to unfortunate results, such as the user inbox becoming compromised by spammers. Therefore, instead of tracking the `urls` in the HTML tags, we are interested in counting them. We exploit attributes related to HTML tags only for the CSDMC2010 dataset as out of the four datasets only this one contains these tags. We extracted three HTML-based attributes from the emails of the CSDMC2010 dataset: (1) frequency of anchor tags (i.e., the number of close-ended tags <a> and), (2) frequency of tags that are not anchors (e.g., <p> or
), and (3) total HTML tags in the emails [e.g., sum of (1) and (2)]. Each of these attributes is normalized by the length of the email, N, which is the number of sentences in the message. To identify HTML tags in the emails, we used a Java HTML parser called *jsoup*.[4]

Note that the aforementioned attributes are computed by both including and excluding the function words in the emails—the exceptions being the frequency of function words, TF, email length, and TF·IDF attributes. Analyzing the real-valued attributes, we have found that their distribution is either left-tailed or right-tailed. This skewness of attribute values, which leads to poor classifiers, has been eliminated by using a *logarithmic transformation* of all the attribute values. Once log-transformed, the distribution of the attributes becomes normal. Lastly, for each attribute, except the error attributes described in Sect. 3.1.2, we perform attribute normalization; the resulting normalized values are therefore in [0, 1].

3.1.5 Attribute selection

To measure attribute importance, we have used an *attribute importance* measuring algorithm named *Boruta*[5] that uses a wrapper around random forest. The details of the *Boruta* algorithm are beyond the scope of this paper but can be found in the study by Kursa and Rudnicki [27]. We applied

[3] Available at: http://www.languagetool.org/java-api/.

[4] Downloadable at http://jsoup.org/download.

[5] http://cran.r-project.org/web/packages/Boruta/index.html.

the algorithm on each of the datasets. Several key findings resulted. First, for all datasets, the most important attribute is *Spam Words*. Second, the *importance scores* computed by *Boruta* are similar for all of the *error attributes* (see Sect. 3.1.2). Third, the TF·IDF of *simple words* and the HTML attributes are important attributes for CSDMC2010 and SpamAssassin emails. The relative importance of *alphanumeric words* is similar in all three non-personalized datasets as is the *Verbs* attribute. Interestingly, the *function word* attribute performed reasonably well—to the best of our knowledge, not many consider this as a spam detection attribute. Except TF·IDF of simple and complex words, most of the *readability attributes* are less important. Based on the algorithm's output, we have excluded the following attributes from the datasets: *HTML attributes* for the LingSpam and all Enron datasets, and the *word length* attribute for the LingSpam dataset; all of the other attributes are treated equally for classification.

3.2 Learning algorithms

As found in the recent research [23, 36], random forest (RF) is able to produce highly accurate spam classifiers. Overall, RF classifiers also have the reputation for being fast and efficient with large data [29]. On the other hand, although it generates complex models, SVM is a popular choice for anti-spam filters. The algorithm has notable performances with header features [23, 28, 36, 48]. NB is also a widely used learning algorithm for anti-spam filters. It is simple yet provides powerful spam detectors [32, 36]. In addition, on many occasions, NB using simple attributes like TF·IDF has even outperformed quality learning algorithms, like SVM. Unlike the others, ADABOOSTM1 and BAGGING are meta-learners that improve a given *weak* learning algorithm most of the time [8, 39]. In the following experiments, we have considered RF as the weak learner for these two algorithms. Both ADABOOSTM1 and BAGGING are simple and fast and, above all, are less susceptible to overfitting the training data. And finally, their performances are better than algorithms like NB and *probabilistic* TF·IDF for text categorization tasks.

The parameters that we set for the algorithms are described in Table 2.

3.3 Experimental procedure

Treating each dataset independently, SENTINEL extracts the real-valued attributes from each email using its text processing unit. Using a conventional stratified tenfold cross-validation approach, the filter then generates for each dataset five classifiers using the five algorithms described in Sect. 3.2. The classifiers are then evaluated. In a K-fold cross-validation, the original dataset is randomly partitioned into K equal-sized folds or subsets. Then, each classifier is trained on $k-1$ folds and evaluated on the

Table 2 Parameter setup for the learning algorithms used in this experiment to generate classifiers

Learning algorithms	Parameters
Random forest	Maximum depth: unlimited
	Random seed: 1
	Number of trees to be generated: 10
Boosted random forest	Number of iterations: 10
	Resampling: false
	Random seed: 1
	Weight threshold: 100
Bagged random forest	Size of bag (%): 100
	Number of iterations: 10
	Out of bag error: False
	Random seed: 1
Support vector machine	SVM Type: C-SVC
	Degree of kernel: 3
	Gamma: 0.0
	Epsilon: 0.1
	Shrinking heuristics: true
	Cost: 1.0
	EPS: 0.0010
	Kernel type: radial basis
	Probability estimates: false
Naïve Bayes	Use of kernel estimator: false

Table 3 Confusion matrix for the spam classification problem

	Actual	
	Spam	Ham
Prediction		
Spam	$n_{s\to s}$	$n_{h\to s}$
Ham	$n_{s\to h}$	$n_{h\to h}$

remaining fold. Stratification means that the class (i.e., ham or spam) in each fold is represented in approximately the same proportion as in the full dataset. The cross-validation process is then repeated until each of the K folds is used exactly once as the validation data. The final evaluation measures of the classifiers are the averages of the K evaluation measures from the folds. These evaluation measures are described next.

3.4 Evaluation measures

To evaluate anti-spam filters, a significant number of previous works rely on measures such as precision, recall, *F*-score, and accuracy [19]. The reporting of these measures is done according to the confusion matrix given in Table 3. The measures are explained below.

Precision is the fraction of spam predictions that are correct and can be written as follows:

$$\text{Precision} = \frac{n_{s\to s}}{n_{s\to s} + n_{h\to s}}.$$

On the other hand, recall—also known as *spam recall*—examines the fraction of spam emails being retrieved:

$$\text{Recall} = \frac{n_{s\to s}}{n_{s\to s} + n_{s\to h}}.$$

F-score (FM), simply, is the harmonic mean of precision and recall and can be calculated as follows:

$$F\text{-score} = \frac{2 \times precision \times recall}{precision + recall}.$$

Accuracy, on the other hand, is the percentage of correctly identified spams and hams:

$$\text{Accuracy} = \frac{n_{h\to h} + n_{s\to s}}{n_{h\to h} + n_{h\to s} + n_{s\to h} + n_{s\to s}}.$$

Some of the aforementioned measures, however, can be seriously flawed when working with datasets that have class imbalance problems. For instance, accuracy does not distribute weights rationally to the majority and minority classes; rather, it places more weight on the majority class than on minority class. This makes it difficult for a classifier to perform well on the minority class. Moreover, email misclassification can be cost-sensitive, considering that the users might accept some spams to enter into their inbox but that they prefer their hams not end up in the spam traps. To overcome the problem with measures that cannot deal with the class imbalance problem (e.g., accuracy), ham misclassification rate (FPR) and spam misclassification rate (FNR) are also being used.

The ham misclassification (false-positive) rate denotes the fraction of ham emails classified as spams:

$$fpr = \frac{n_{h\to s}}{n_{h\to s} + n_{h\to h}}.$$

In contrast, the spam misclassification (false-negative) rate is the fraction of spams delivered to the user inbox:

$$fnr = \frac{n_{s\to h}}{n_{s\to h} + n_{s\to s}}.$$

The viable alternative to FPR and FNR is to report the *Area Under the Curve* (hereinafter, AUC), measured using the *Receiver Operating Characteristic* (*ROC*) curves. The ROC curve is a 2D graph whose *Y-axis* represents $1 - fnr$ and whose *X-axis* represents FPR, thereby depicting the compromises between the costs of $n_{s\to h}$ and $n_{h\to s}$.

Besides the aforementioned evaluation measures, we also used *cost curves* to report the classification performance. The cost curves, indeed, are the projection of the slopes in the ROC curves on the *X-axis*, while the *Y-axis* is the expected misclassification cost. These curves provide visualizations of 2-class classifiers over the full range of possible class distributions and misclassification costs. That is, for skewed datasets, using these curves, we can even say how good a given classifier would perform if the class distributions were equal—50 % of the emails are spams, and 50 % of the emails are hams. The details about the cost curves are described by Drummond and Holte [16] and Holte and Drummond [22].

3.5 Datasets

For decades, several email datasets, viz. SpamAssassin,[6] CSDMC2010,[7] LingSpam,[8], and Enron-Spam[9] [32], have been used to gauge the performance of anti-spam filters. The first three datasets are composed of randomly collected spam and ham emails over a given time period and therefore are suitable for developing and testing non-personalized anti-spam filters. Enron-Spam, on the other hand, is a collection of emails composed of six datasets each containing ham emails from a single person.

[6] Downloadable at http://spamassassin.apache.org/publiccorpus/.

[7] Downloadable at http://csmining.org/index.php/spam-email-datasets-.html.

[8] Downloadable at http://csmining.org/index.php/ling-spam-datasets.html.

[9] Downloadable at https://labs-repos.iit.demokritos.gr/skel/i-config/downloads/enron-spam.

Table 4 Brief description of the non-personalized email datasets: ham:spam ratio, whether the texts are preprocessed, and the year of curation

Dataset	Ham:Spam	Text preprocessed?	Year of curation
CSDMC2010	2949:1338	No	2010
SpamAssassin	4149:1884	No	2002
LingSpam	2414:481	Yes	2000

CSDMC2010, among the four, is the latest collection of emails and has not been used much. The spam rate of this dataset is reasonable—about 32 %. In contrast, SpamAssassin is one of the most popular datasets. The spam rate of this dataset is almost equal to that of the CSDMC2010 dataset. The LingSpam dataset is both the smallest and oldest datasets that we consider. Its spam rate is also smaller than the others—only about 17 %. It is the odd one out of the three non-personalized datasets because the hams in this dataset are collected from the discussions of a linguistics forum; the spams, on the other hand, are collected randomly. Table 4 summarizes these datasets.

The ham collections of Enron-Spam are named: `farmer-d`, `kaminski-v`, `kitchen-l`, `williams-w3`, `beck-s`, and `lokay-m`. The spams are collected from three different sources. First, a mix of spams that are collected from the SpamAssassin corpus and spam traps of the Honeypot project[10] are put together; these spams are dubbed as SH. Second, BG spams are collected from the spam traps of Bruce Guenter.[11] Third, spams are collected randomly from the mailbox of Georgios Paliouras (GP) [32]. The foregoing six ham email collections are each paired with one of these three spam collections (SH, BG, and GP). Thereafter, the six collections are dubbed as Enron 1–6. Of the six collections, Enron 1–3 are ham-skewed (ham:spam is 3:1), while Enron 4–6 are spam-skewed (ham:spam is 1:3). The summary of the characteristics of the Enron-Spam collection can be found in Table 5.

During the development of the Enron-Spam dataset [32], it is noticed that spam emails have some attributes that can too easily distinguish them from hams, and therefore, they should be preprocessed. To remove these attributes, the following preprocessing steps are considered. First, the SUBJECT field of spams can contain symbols such as $ or !. As well, they contain spam words such as *porn*, *webcam*, or *lottery*. Therefore, such entries are excluded from the SUBJECT fields of the emails. Second, many emails in the datasets can contain an ATTACHMENT field. If it exists, this extraneous field is removed from an email. Third, non-ASCII characters in the email text are removed since the values of our natural language attributes are affected by their presence.

The reasons for choosing these datasets are manifold. Firstly, emails contained in them are sent out with a span of 10 years—between 2000 and 2010. This provides an interesting test bed that characterizes the change in language of both spams and hams spanning across a decade. Secondly, we include spam-skewed datasets in our experiment (i.e., Enron 4–6) because previously, many works reported that although an anti-spam filter can do well on ham classification using ham-skewed datasets such as CSDMC2010, SpamAssassin, LingSpam, and Enron 1–3, its performance on spam classification can be seriously flawed (see, e.g., [7]). Thirdly, we include LingSpam in our dataset because not only are its hams domain-specific but the hams are also excerpts of scholarly discussions on linguistics. We believe that the results of our stylometric approach with this dataset will be interesting to the anti-spam community. Fourthly, we include datasets that are popular and reported by many (SpamAssassin and LingSpam) as well as those that are not explored by as many (CSDMC2010 and Enron-Spam). Last but not least, it is one of our goals to investigate the performance of SENTINEL on both non-personalized and personalized emails.

4 Results and discussions

4.1 Performance on non-personalized emails

In this section, we report the traditional as well as cost-sensitive evaluation of SENTINEL on the non-personalized email data. The most remarkable result to emerge from the data is that for all datasets, BAGGING generates classifiers that have the lowest ham misclassification rates (FPR), while the classifiers generated by ADABOOSTM1 have the lowest spam misclassification rates (FNR). These results can be found in Table 6. Given the results, it is difficult to decide which classifier is better. A good way to report the best-performing classifier is to refer to a balanced measure of the two misclassification rates: the AUC. As can be seen in Figs. 1 and 3, of the two, the BAGGED RF classifiers have the better AUC. Table 6 also summarizes that of all the datasets, SENTINEL misses hams the least (FPR $= 1$ %) on the LingSpam dataset. On the other hand, the lowest FNR achieved by the filter is about 7 % on the SpamAssassin dataset. Besides, the results with SVM and NB are below expectations. The reasons to underperform for a standard algorithm like SVM are further investigated in Sect. 4.2. In our analysis, we found that the attributes are highly correlated which contradicts the naive assumption of the NB algorithm. This apparent dependency among the attributes can be attributed to the algorithm's poor performance.

[10] Consult with http://www.projecthoneypot.org.

[11] Overview at http://untroubled.org/spam.

Table 5 Brief descriptions of the Enron-Spam collection as described by Metsis et al. [32]: composition of the datasets, ham:spam ratio, and timestamp

Ham + Spam	Ham:Spam	Ham and Spam Timestamp
Enron 1 (farmer-d + GP)	3672:1500	[12/99, 1/02] and [12/03, 9/05]
Enron 2 (kaminski-v + SH)	4361:1496	[12/99, 5/01] and [5/01, 7/05]
Enron 3 (kitchen-l + BG)	4012:1500	[2/01, 2/02] and [8/04, 7/05]
Enron 4 (williams-w3 + GP)	1500:4500	[4/01, 2/02] and [12/03, 9/05]
Enron 5 (beck-s + SH)	1500:3675	[1/00, 5/01] and [5/01, 7/05]
Enron 6 (lokay-m + BG)	1500:4500	[6/00, 3/02] and [8/04, 7/05]

Table 6 Ham misclassification rates (FPR) and spam misclassification rates (FNR) of SENTINEL

Dataset	Classifiers	FPR	FNR
CSDMC2010	RF	0.040	0.092
	ADABOOSTM1	0.030	0.089
	BAGGING	0.021	0.107
	SVM	0.028	0.390
	NB	0.101	0.396
SpamAssassin	RF	0.035	0.093
	ADABOOSTM1	0.027	0.079
	BAGGING	0.023	0.099
	SVM	0.052	0.292
	NB	0.104	0.558
LingSpam	RF	0.018	0.162
	ADABOOSTM1	0.017	0.162
	BAGGING	0.010	0.193
	SVM	0.014	0.341
	NB	0.219	0.277

To understand the performance of the ensemble methods like ADABOOSTM1 and BAGGING better, we also have applied them by changing their base learner on the non-personalized datasets. This time as base learners, we have used NB and SVM. Overall for the non-personalized datasets, the performance of the filter did not improve by using these base learners for ADABOOSTM1 and BAGGING. The results are detailed in Table 7.

The *F*-score and AUC of the five classifiers of SENTINEL on CSDMC2010 dataset are compared to those found by three recently proposed, cutting-edge filters: RF-HEADER [36], NB-BITEST, and SVM-BITEST [47]. Unexpectedly, SENTINEL is outperformed by all three filters in terms of *F*-score as shown in Fig. 1a. Also, a *paired t test* with $\alpha = 0.05$ confirms that the differences are statistically significant. A possible explanation for this can be found in Table 6: The spam recall ($1 -$ FNR) of SENTINEL's classifiers is much higher than that achieved by the filters. However, when it comes to cost-sensitive analysis, SENTINEL performs more ideally—its BAGGED RF classifier's AUC outperforms RF-HEADER and NB-BITEST; SVM-BITEST with a reported AUC of 0.995 is the only exception. These comparisons are summarized in Fig. 1b.

Similarly, the accuracies of the classifiers of SENTINEL, and two standard filters named NN-TEXT [30] and NN-AIS [43] are compared. Figure 2 shows that all of our ensemble-classifier accuracies are better than NN-TEXT and NN-AIS. The differences found in the accuracies are significant at $\alpha = 0.05$.

Finally, for the LingSpam dataset, the *F*-scores of SENTINEL are compared with two more filters called BIRCH [35] and NB-BITEST [47]. We found that all the *F*-scores of SENTINEL's classifiers, except that induced by NB, are better than the BIRCH filter. On the other hand, the *F*-score of NB-BITEST, however, is better than SENTINEL but only at $\alpha = 0.10$. The comparison is found in Fig. 3a. In a similar way, the AUC of SENTINEL is compared with that reported by PPM [14] and NB-BITEST [47]. As shown in Fig. 3b, PPM performs the best in terms of AUC and the ensemble classifiers of SENTINEL are close seconds. Again, the differences are statistically significant at $\alpha = 0.05$.

The cost curves of the classifiers generated by SENTINEL for the three non-personalized datasets are shown in Figs. 4, 5, and 6. The cost curves are a good means not only to explore the expected cost associated with the classifiers but also their performance for different ham:-spam ratios in the datasets. Before we discuss further, please note that the *X-axis* of the curves denotes the *probability cost*, i.e., $x = 0.5$ illustrates the situation where the numbers of spams and hams in the dataset are equal and *trivial classifiers* are those that identify all emails as either hams or spams.

As anticipated, the poor performances of the SVM and NB classifiers are captured by the cost curves, too. For any ratio of ham and spam emails, the normalized expected costs associated with these two classifiers are very high (Figs. 4, 5, 6). According to the cost curves, the remaining classifiers perform much better for all ratios of ham and spam emails. However, we observe differences in their performances for different datasets. For instance, the ADABOOSTM1 classifier is expected to perform better than the BAGGED RF and RF classifiers if the majority of the emails are spams in the CSDMC2010 and SpamAssassin datasets (i.e., when probability cost > 0.5 in Figs. 4, 5). Interestingly, had these two datasets contained more spams, we could have expected that RF would have outperformed BAGGING.

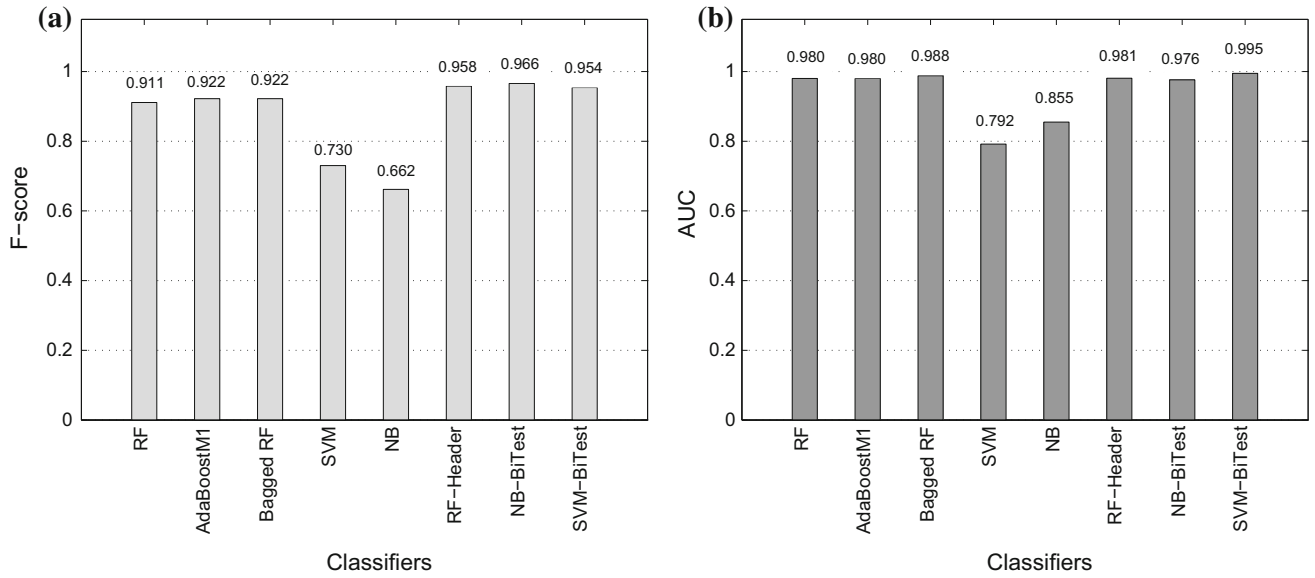

Fig. 1 Comparison of **a** *F*-score and **b** AUC of the classifiers generated by SENTINEL and classifiers from previous studies on the CSDMC2010 dataset

Table 7 Ham misclassification rates (FPR) and spam misclassification rates (FNR) of SENTINEL using SVM and NB as base learners for BAGGING and ADABOOSTM1

Dataset	Classifiers	FPR	FNR
CSDMC2010	BAGGING-NB	0.101	0.400
	BAGGING-SVM	0.047	0.245
	ADABOOSTM1-NB	0.081	0.263
	ADABOOSTM1-SVM	0.046	0.239
SpamAssassin	BAGGING-NB	0.103	0.559
	BAGGING-SVM	0.060	0.223
	ADABOOSTM1-NB	0.094	0.223
	ADABOOSTM1-SVM	0.061	0.223
LingSpam	BAGGING-NB	0.285	0.400
	BAGGING-SVM	0.017	0.158
	ADABOOSTM1-NB	0.033	0.265
	ADABOOSTM1-SVM	0.019	0.162

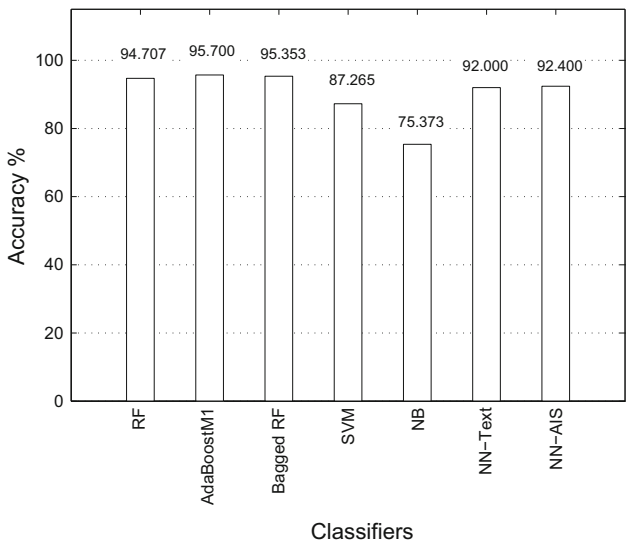

Fig. 2 Comparison of accuracy of the classifiers generated by SENTINEL and classifiers from previous studies on the SpamAssassin dataset

This observation is confirmed according to the Figs. 4 and 5, where RF starts to perform better than BAGGING at some point after probability cost > 0.5. On the other hand, for the LingSpam dataset, the RF classifier can be expected, in most of the cases, to outperform both the BAGGED RF and ADABOOSTM1 classifiers regardless of the skewness in the dataset (Fig. 6).

4.2 Performance on personalized emails

Traditionally, the studies on Enron-Spam report the performances of their filters by averaging the scores described in Sect. 3.4 across the six datasets [32]. In doing so, they almost always have used the *arithmetic mean*. However, our tests reveal that like other cutting-edge filters [26, 32], our proposed filters perform completely opposite for Enron 1–3 (hams are missed less often because more hams are in the training data) and Enron 4–6 (spams are missed less often because more spams are in the training data). These extreme trends in the performances are exhibited in Fig. 7 where the reported values for the six datasets largely vary. Such extreme points affect the overall average if it is calculated using the arithmetic mean. The viable alternative

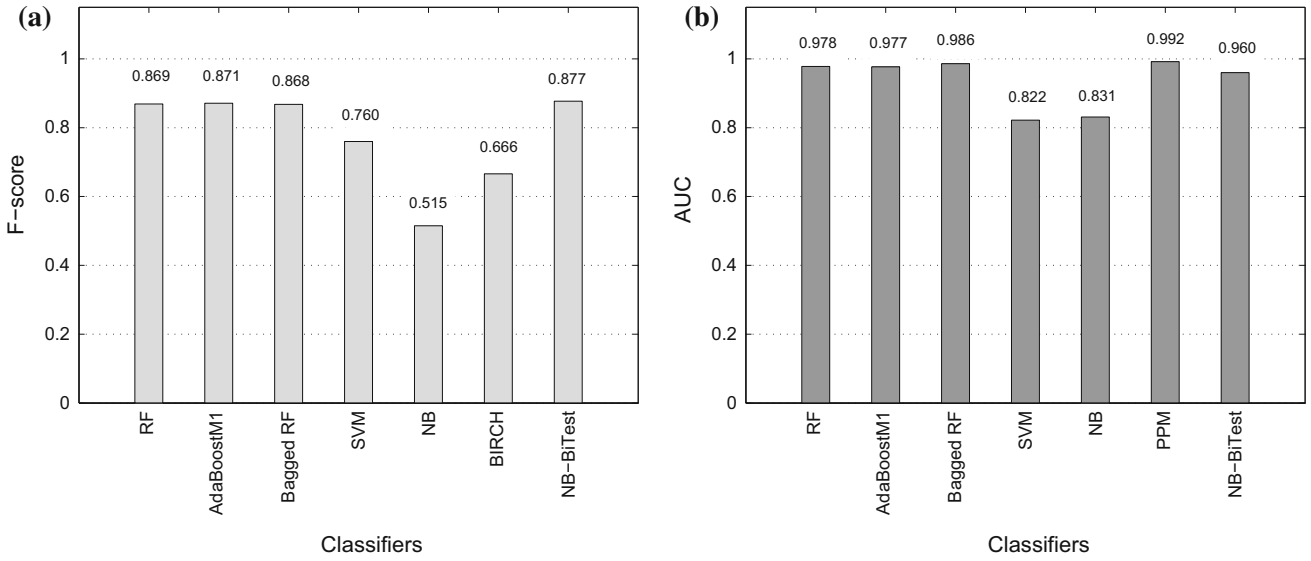

Fig. 3 Comparison of **a** *F*-score and **b** AUC of the classifiers generated by SENTINEL and classifiers from previous studies on the LingSpam dataset

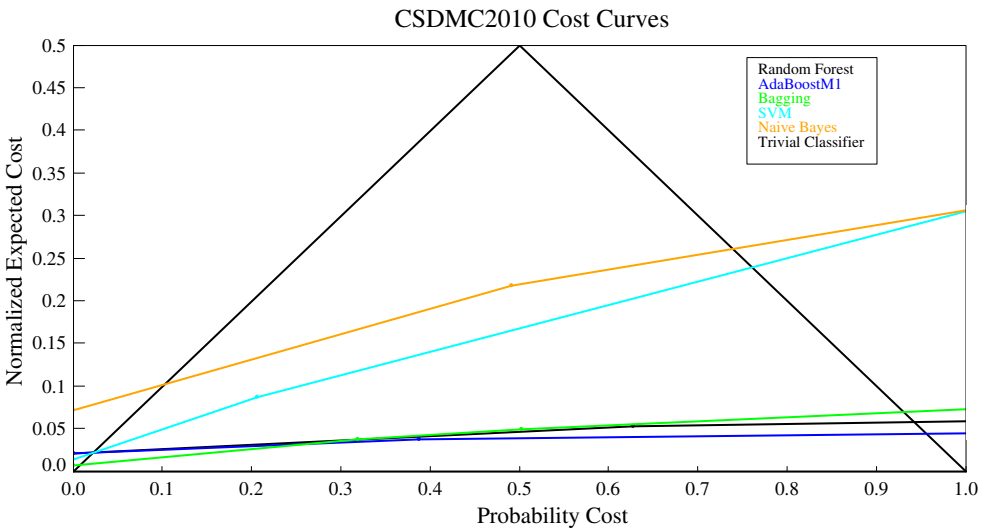

Fig. 4 Cost curves of the five classifiers generated by SENTINEL on the CSDMC2010 dataset

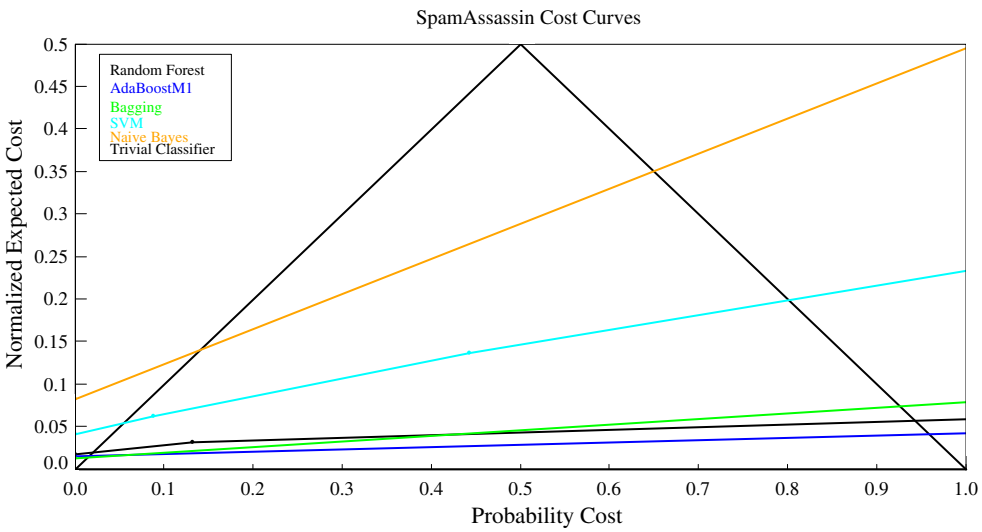

Fig. 5 Cost curves of the five classifiers generated by SENTINEL on the SpamAssassin dataset

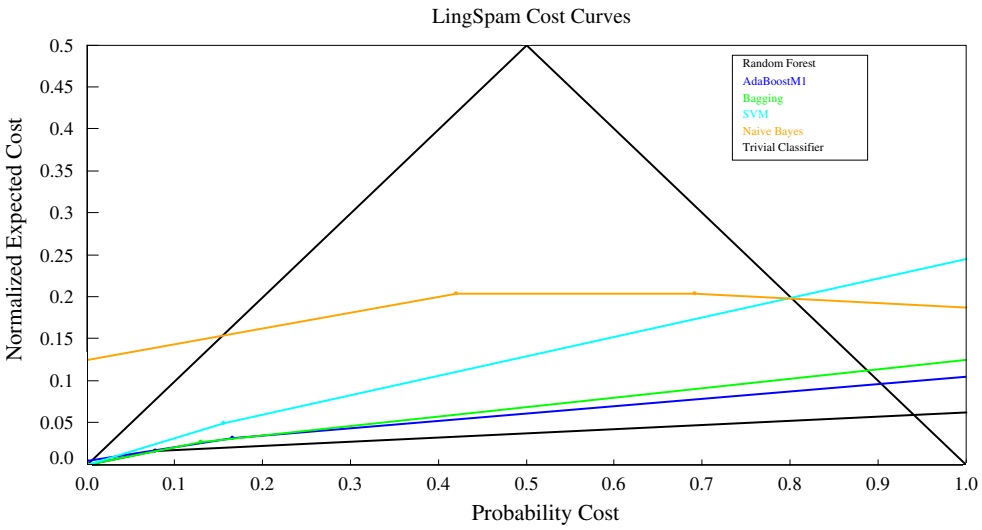

Fig. 6 Cost curves of the five classifiers generated by SENTINEL on the LingSpam dataset

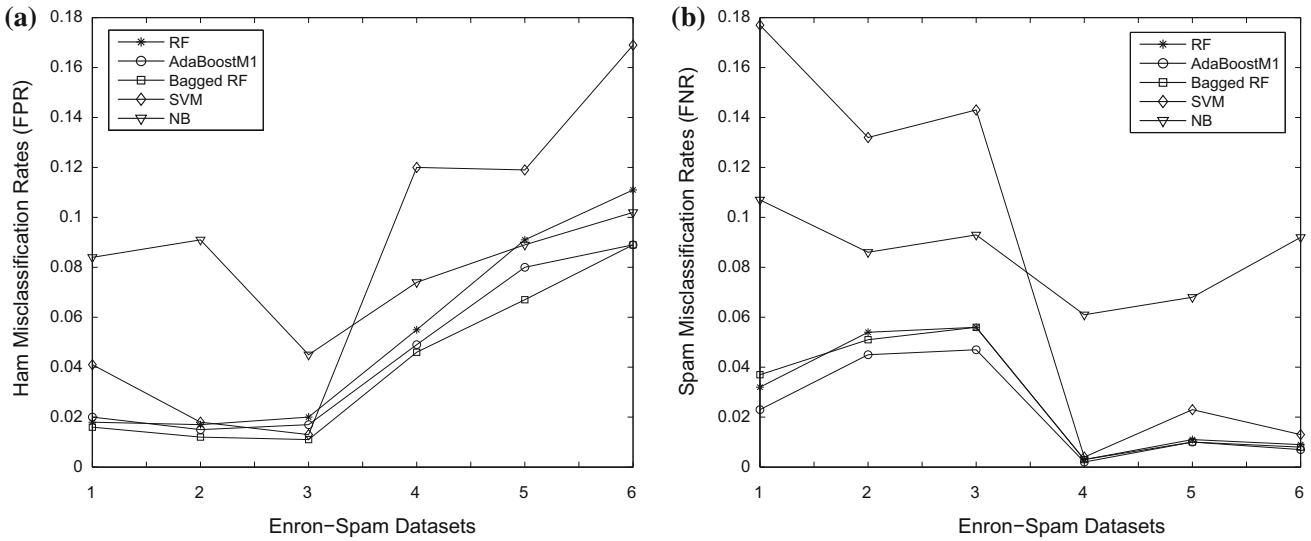

Fig. 7 **a** Ham misclassification rates and **b** spam misclassification rates of the five SENTINEL-generated classifiers on the Enron-Spam collection

for the averaging is then the *harmonic mean* which reduces the effect of outliers on the average. That said, in this section, we report the *harmonic mean* of the results found from the six datasets.

The accuracy and F-score of the classifiers can be found in Fig. 8a, b, respectively. It can be noted that the values reported here for personalized emails are significantly better than the values reported in Sect. 4.1 for non-personalized emails. We also compared the accuracy and F-score to that found by four benchmark personalized email filters: LC [50], NB-BOW [37], SVM-BOW [37], and ICRM [1]. The two ensemble classifiers of SENTINEL perform about equally well and surpass the performances of the rest. The filter that can claim to be a reasonably close second is the LC filter inspired by artificial immune systems [50]. The differences between SENTINEL's optimal classifier (BAGGED RF) and LC are significant at $\alpha = 0.05$.

The average email misclassification rates of SENTINEL can be found in Fig. 8c, d, respectively. BAGGING and ADABOOSTM1 perform the best—BAGGING misclassifies hams the least (FPR = 2.1 %, see Fig. 8c), while ADABOOSTM1 misclassifies spams the least (FNR = 0.7 %, see Fig. 8d). In addition, SENTINEL is compared to personalized filters that are used as yardsticks in the domain: NB-TF [32], BAYESIAN [25], and NB-SLWE [49]. Except for the NB-TF filter, the two ensemble classifiers of SENTINEL outperform the rest. Four of SENTINEL's classifiers (except NB) surpassed the FNR of NB-TF ,but the FPR of our second best classifier—ADABOOSTM1—ties with NB-TF; the FPR of BAGGING is higher than NB-TF—the difference, however, is statistically significant only at $\alpha = 0.10$.

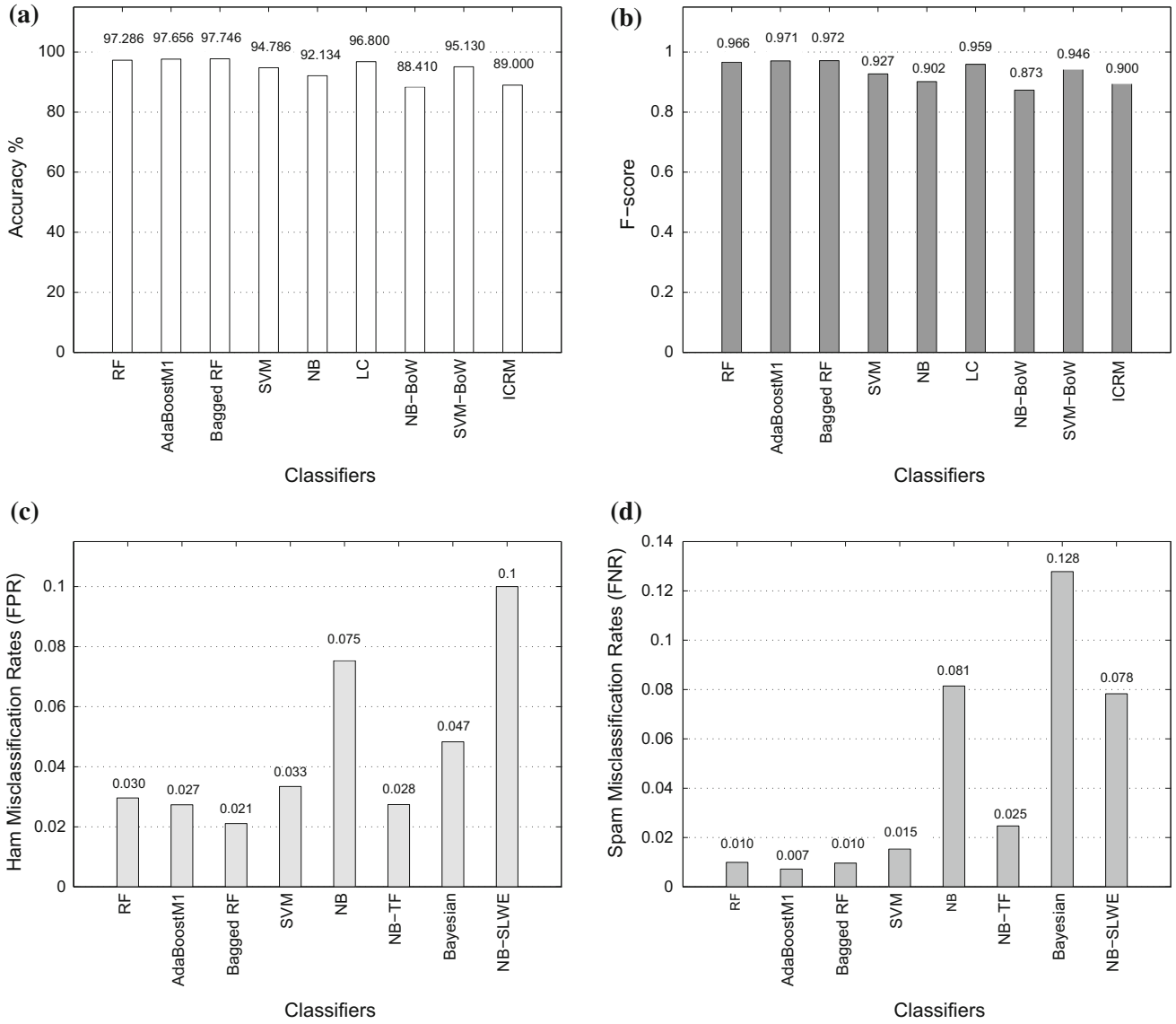

Fig. 8 Comparison of the performances on the Enron-Spam collection of the five classifiers generated by SENTINEL and classifiers from previous studies. **a** Comparison of accuracies, **b** comparison of *F*-scores, **c** comparison of FPRs (ham misclassification rates), **d** comparison of FNRs (spam misclassification rates)

Just like what we did for the non-personalized datasets, we have changed the base learners BAGGING and ADABOOSTM1 from RF to NB and SVM. Again, the performance of the filter did not improve by using these base learners. The results are detailed in Table 8.

Further tests on each of the six datasets reveal that the skewness of data has a detrimental effect on the training of anti-spam filters. For instance, except for the aberrant trend displayed by the NB classifier, the remaining classifiers misclassify fewer hams in Enron 1–3 (ham-skewed) and fewer spams in Enron 4–6 (spam-skewed). This experiment also suggests that RF exhibits a similar ability to the meta-learners for spam classification (see Fig. 7b). However, its ability to identify hams diminishes when more spams are included in its training data (see Fig. 7a)—even so that for Enron 5 and 6, it is outperformed by NB. Overall, the results indicate that an SVM classifier is more sensitive to the skewness of the training data; hence, the training set should be carefully selected. With spam-skewed training data, an SVM classifier's ham misclassification rate is as high as 17 %. Similarly, with ham-skewed training data, the spam misclassification rate nears 18 %.

We have investigated the expected performance of the classifiers on each dataset. Although we have produced cost curves for all the classifiers for the six datasets in the Enron-Spam collection, we present only the cost curves of

Table 8 Ham misclassification rates (FPR) and spam misclassification rates (FNR) of SENTINEL on the Enron Datasets using SVM and NB as base learners for BAGGING and ADABOOSTM1

Dataset	Classifiers	FPR	FNR
Enron 1	BAGGING-NB	0.089	0.106
	BAGGING-SVM	0.041	0.155
	ADABOOSTM1-NB	0.059	0.097
	ADABOOSTM1-SVM	0.040	0.158
Enron 2	BAGGING-NB	0.091	0.086
	BAGGING-SVM	0.017	0.085
	ADABOOSTM1-NB	0.040	0.106
	ADABOOSTM1-SVM	0.016	0.088
Enron 3	BAGGING-NB	0.044	0.092
	BAGGING-SVM	0.015	0.089
	ADABOOSTM1-NB	0.040	0.095
	ADABOOSTM1-SVM	0.016	0.090
Enron 4	BAGGING-NB	0.075	0.060
	BAGGING-SVM	0.071	0.007
	ADABOOSTM1-NB	0.068	0.038
	ADABOOSTM1-SVM	0.073	0.008
Enron 5	BAGGING-NB	0.089	0.067
	BAGGING-SVM	0.105	0.021
	ADABOOSTM1-NB	0.090	0.056
	ADABOOSTM1-SVM	0.105	0.021
Enron 6	BAGGING-NB	0.104	0.092
	BAGGING-SVM	0.145	0.018
	ADABOOSTM1-NB	0.119	0.069
	ADABOOSTM1-SVM	0.144	0.018

the two best classifiers: ADABOOSTM1 and BAGGING. These cost curves can be found in Figs. 9 and 10. From the curves, it is evident that we can expect that our classifier performances will vary for different ratios of hams and spams. What is interesting in these curves is that not only is the ratio of ham to spam playing a role in expected performances, but also the spams themselves. Note that the spam-skewed datasets display similar curves for both classifiers and show similar spreads among themselves. Stylometric differences among the spams in these three datasets may be the cause for these observations. The ham-skewed datasets produce similar curves, but they are more tightly bundled. Even though the hams are from personal mailboxes, they are more likely to be stylometrically similar because they are business-related. Some influence, albeit diminished because of the underrepresentation of spams, comes from the spams. Four of the curves are in a narrow band when probability cost = 0.5.

Lastly, we evaluated the performance of SENTINEL on the *stacks* of the six datasets. We generated two sets of stacks, and following is the process of generating the first set of stacks: starting with the stack composed of Enron 1, one

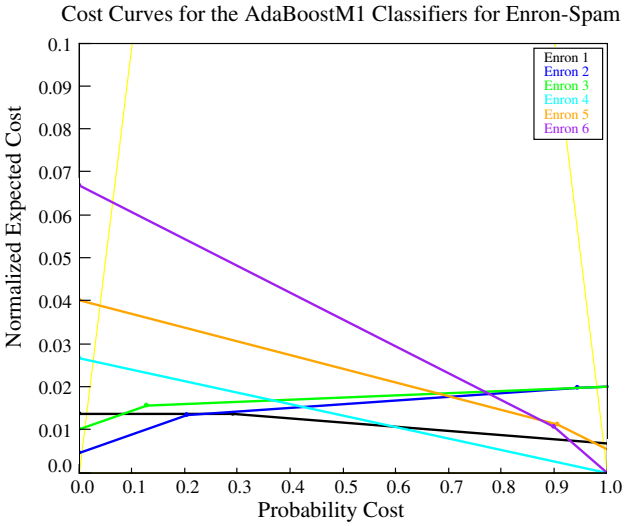

Fig. 9 Cost curves for the ADABOOSTM1 classifiers generated for the six datasets in the Enron-Spam collection

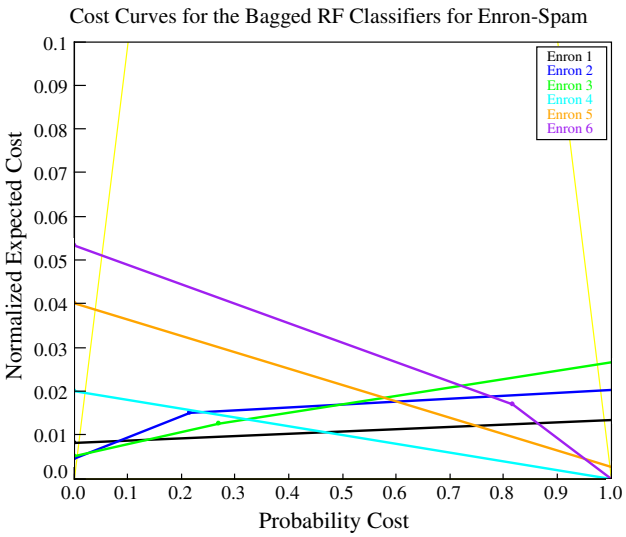

Fig. 10 Cost curves for the BAGGED RF classifiers generated for the six datasets in the Enron-Spam collection

dataset (in numerical order) is added to the previous stack. This process of creating the dataset stacks continues until Enron 6 is combined with Enron 1–5. Thus, the number of hams dominates up to the third stack, and the ratio of spams and hams becomes closer to 1 after adding Enron 6 to the stacks of Enron 1–5. The second set of stacks is generated as follows: starting with the stack composed of Enron 6, one dataset (in reverse numerical order) is added to the previous stack. This process of creating the dataset stacks continues until Enron 1 is combined with Enron 6–2. Thus, the number of spams dominates up to the third stack, and the ratio of spams and hams becomes closer to 1 after adding Enron 1 to the stacks of Enron 6–2.

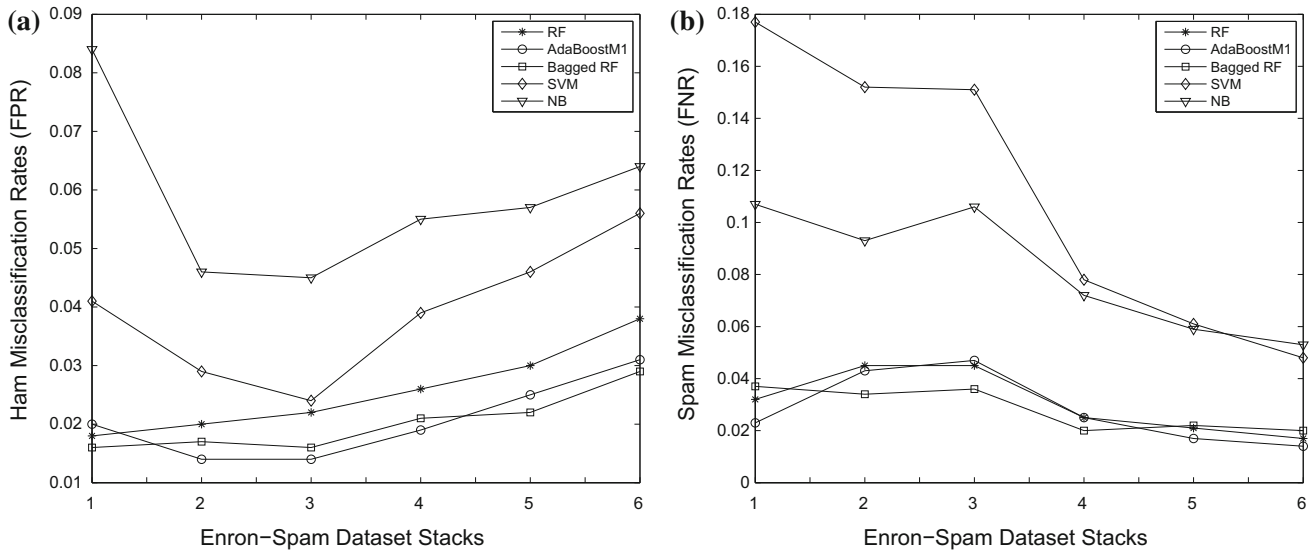

Fig. 11 The incremental **a** ham misclassification rates and **b** spam misclassification rates of the SENTINEL classifiers on the Enron-Spam collection

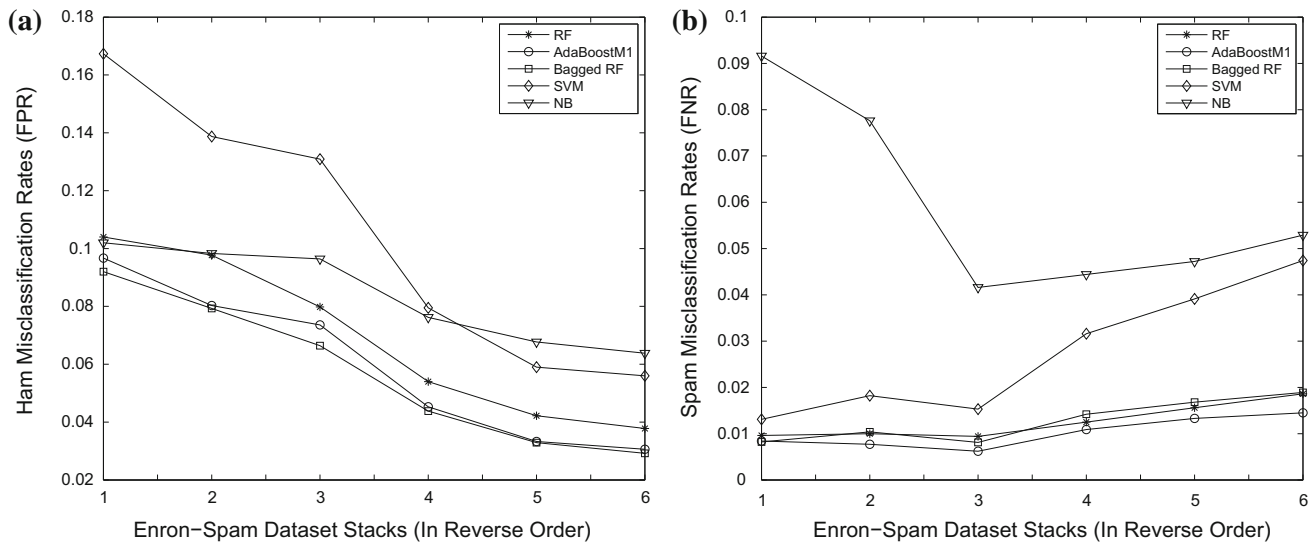

Fig. 12 The reverse incremental **a** ham misclassification rates and **b** spam misclassification rates of the SENTINEL classifiers on the Enron-Spam collection

As we evaluated SENTINEL on the stacks, it is evident that on a *numerically* balanced dataset, the best ham misclassification rate is achieved by BAGGED RF followed by ADABOOSTM1, RF, SVM, and NB, while the best spam misclassification rate is achieved by ADABOOSTM1 followed by RF, BAGGED RF, SVM, and NB (see Stack 6 in Figs. 11, 12). Interestingly, we found that in addition to class imbalance in the datasets, the performance of SENTINEL depends on two other factors: the sources of the emails and the number of training emails for each class. A case in point, a notable change in ham misclassification rates can be observed for the first three reverse stacks (Fig. 12a), even though the ratio of spams to hams in the three stacks is the same (i.e., 3:1). There are three differences among the stacks: the number of spam emails, the number of ham emails, and the email sources. This clearly suggests that besides the class imbalance problem, the email source and the number of training emails may have an influence on our filter's performance.

5 Conclusions

In this paper, we describe the development and evaluation of an anti-spam filter named SENTINEL. The filter uses natural language attributes, the majority being connected to stylometric aspects of writing. The real-valued, natural language attributes

extracted from the email texts are used to generate binary classifiers. The classifiers explored in this study are induced by five state-of-the-art learning algorithms. We evaluate the filter with benchmark non-personalized email datasets such as CSDMC2010, SpamAssassin, and LingSpam as well as standard personalized emails like those in the six datasets of the Enron-Spam collection. The evidence from extensive experiments implies that the classifiers that perform the best are of two ensemble methods: ADABOOSTM1 and BAGGING. In general, the performance of SENTINEL is mixed on non-personalized email data. This result is not unexpected because our findings demonstrate that the filter has limitations for non-personalized email data—mainly due to the absence of unique writing patterns in the randomly collected emails. Contrary to this, on personalized email data, SENTINEL surpasses the performances of a number of state-of-the-art personalized anti-spam filters. These outcomes imply that the attributes related to writer stylometry can better capture the imprinted patterns in personalized hams. One limitation of the filter is that its performance is affected by the extreme proportions of spams and hams in non-personalized datasets. On a good note, the filter is not affected at all by this factor on personalized datasets.

Our work clearly has some limitations. Firstly, several aspects of the filter, viz. its real-time training and response latency are not considered. It will require extensive tests to confirm SENTINEL's usability as an online filter. Secondly, personalized datasets share an interesting phenomenon called *concept drift* which is yet to be investigated. The reaction of the proposed filter with respect to this phenomenon can be tested by substituting the spams of the Enron-Spam collection with more recent data.

Because our results suggest that ensemble methods perform the best, further tests should be carried out to see the performance of the filter by stacking several algorithms to generate its classifiers. Future studies can extend the work by replacing the supervised algorithms used in this study with semi-supervised learning algorithms.

Acknowledgments Support for this work was provided through a Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant to Robert E. Mercer (Grant No. 36853–2010 RGPIN). We are indebted to Vangelis Metsis, Aris Kosmopoulos, and Robert Holte for their correspondences regarding the use of their TERM FREQUENCY attribute and Cost Curve Tool.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Abi-Haidar A, Rocha LM (2008a) Adaptive spam detection inspired by a cross-regulation model of immune dynamics: a study of concept drift. In: Artificial immune systems. Springer, Berlin, pp 36–47
2. Abi-Haidar A, Rocha LM (2008b) Adaptive spam detection inspired by the immune system. In: ALIFE, pp 1–8
3. Afroz S, Brennan M, Greenstadt R (2012) Detecting hoaxes, frauds, and deception in writing style online. In: 2012 IEEE symposium on security and privacy (SP), pp 461–475
4. Androutsopoulos I, Koutsias J, Chandrinos KV, Spyropoulos CD (2000) An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. In: 23rd Annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 160–167
5. Bickel S (2006) Ecml-pkdd discovery challenge 2006 overview. In: Proceedings of the ECML/PKDD discovery challenge workshop, pp 1–9
6. Blanzieri E, Bryl A (2008) A survey of learning-based techniques of email spam filtering. Artif Intell 29(1):63–92
7. Bratko A, Cormack GV, R D, Filipic B, Chan P, Lynam TR (2006) Spam filtering using statistical data compression models. J Mach Learn Res 7:2673–2698
8. Breiman L (1996) Bagging predictors. Mach Learn 24:123–140
9. Carreras X, Màrquez L (2001) Boosting trees for anti-spam email filtering. In: RANLP-2001, 4th International conference on recent advances in natural language processing, pp 58–64
10. Cheng V, Li C (2007) Combining supervised and semi-supervised classifier for personalized spam filtering. In: Proceedings of the 11th Pacific-Asia conference on knowledge discovery and data mining (PAKDD 2007), pp 449–456. doi:10.1007/978-3-540-71701-0_45
11. Cheng V, Li CH (2006) Personalized spam filtering with semi-supervised classifier ensemble. In: 2006 IEEE/WIC/ACM international conference on web intelligence (WI 2006), pp 195–201. doi:10.1109/WI.2006.132
12. Commtouch (2013) Internet threats trend report. Technical report, Commtouch, USA. http://www.commtouch.com/uploads/2013/04/Commtouch-Internet-Threats-Trend-Report-2013-April.pdf
13. Cormack GV (2007) TREC 2007 spam track overview. In: Proceedings of the sixteenth text retrieval conference, TREC 2007. http://trec.nist.gov/pubs/trec16/papers/SPAM.OVERVIEW16.pdf
14. Cormack GV, Bratko A (2006) Batch and online spam filter comparison. In: Conference on email and anti-spam, CEAS 2006, Mountain View, CA
15. Cormack GV, Lynam TR (2005) TREC 2005 spam track overview. In: Proceedings of the fourteenth text retrieval conference, TREC 2005. http://trec.nist.gov/pubs/trec14/papers/SPAM.OVERVIEW.pdf
16. Drummond C, Holte R (2006) Cost curves: an improved method for visualizing classifier performance. Mach Learn 65(1):95–130
17. Goodman J, Cormack GV, Heckerman D (2007) Spam and the ongoing battle for the inbox. Commun ACM 50(2):24–33
18. Graham P (2003) A plan for spam. http://paulgraham.com/spam.html
19. Guzella TS, Caminhas WM (2009) A review of machine learning approaches to spam filtering. Expert Syst Appl 36(7):10,206–10,222
20. Haider P, Brefeld U, Scheffer T (2007) Supervised clustering of streaming data for email batch detection. In: 24th International conference on machine learning. ACM, pp 345–352
21. Hastie T, Tibshirani R, Friedman J (2001) The elements of statistical learning. Springer Series in Statistics. Springer, Berlin
22. Holte RC, Drummond C (2008) Cost-sensitive classifier evaluation using cost curves. Lecture Notes in Computer Science. In: Washio T, Suzuki E, Ting KM, Inokuchi A (eds) Pacific-Asia conference on knowledge discovery and data mining (PAKDD), vol 5012. Springer, Berlin, pp 26–29

23. Hu Y, Guo C, Ngai EWT, Liu M, Chen S (2010) A scalable intelligent non-content-based spam-filtering framework. Expert Syst Appl 37(12):8557–8565
24. Iqbal F, Khan LA, Fung BCM, Debbabi M (2010) E-mail authorship verification for forensic investigation. In: Proceedings of the 2010 ACM symposium on applied computing, ACM, New York, NY, SAC '10, pp 1591–1598
25. Issac B, Jap WJ, Sutanto JH (2009) Improved Bayesian anti-spam filter implementation and analysis on independent spam corpuses. In: 2009 International conference on computer engineering and technology, vol 02. IEEE Computer Society, pp 326–330
26. Kosmopoulos A, Paliouras G, Androutsopoulos A (2008) Adaptive spam filtering using only naive Bayes text classifiers. In: Fifth conference on email and anti-spam (CEAS 2008)
27. Kursa MB, Rudnicki WR (2010) Feature selection with the Boruta package. J Stat Softw 36(11):1–13. http://www.jstatsoft.org/v36/i11/
28. Lai CC, Tsai MC (2004) An empirical performance comparison of machine learning methods for spam e-mail categorization. In: Fourth international conference on hybrid intelligent systems. IEEE Computer Society, HIS '04, pp 44–48
29. Liaw A, Wiener M (2002) Classification and regression by randomForest. R News 2(3):18–22. http://CRAN.R-project.org/doc/Rnews/
30. Ma Q, Qin Z, Zhang F, Liu Q (2010) Text spam neural network classification algorithm. In: 2010 International conference on communications. Circuits and systems (ICCCAS), pp 466–469
31. Meng Y, Li W, Kwok L (2014) Enhancing email classification using data reduction and disagreement-based semi-supervised learning. In: IEEE international conference on communications, ICC 2014, Sydney, Australia, pp 622–627. doi:10.1109/ICC.2014.6883388
32. Metsis V, Androutsopoulos I, Paliouras G (2006) Spam filtering with naive Bayes—Which naive Bayes? In: Third conference on email and anti-spam (CEAS)
33. Mojdeh M, Cormack GV (2008) Semi-supervised spam filtering: does it work? In: Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval, SIGIR 2008, pp 745–746. doi:10.1145/1390334.1390482
34. Orăsan C, Krishnamurthy R (2002) A corpus-based investigation of junk emails. In: Third international conference on language resources and evaluation (LREC-2002), Spain, pp 1773–1780
35. Prabhakar R, Basavaraju M (2010) A novel method of spam mail detection using text based clustering approach. Int J Comput Appl 5(4):15–25. published By Foundation of Computer Science
36. Qaroush A, Khater IM, Washaha M (2012) Identifying spam e-mail based-on statistical header features and sender behavior. In: CUBE international information technology conference. ACM, pp 771–778
37. Razmara M, Razmara A, Narouei M (2012) Textual spam detection: an iterative pattern mining approach. World Appl Sci J 20(2):198–204
38. Sahami M, Dumais S, Heckerman D, Horvitz E (1998) A Bayesian approach to filtering junk e-mail. In: Learning for text categorization: papers from the 1998 workshop, AAAI Technical Report WS-98-05, pp 55–62
39. Schapire RE (1999) A brief introduction to boosting. In: 16th international joint conference on Artificial intelligence, vol 2, Morgan Kaufmann Publishers Inc., Los Altos, CA, IJCAI'99, pp 1401–1406
40. Shams R, Mercer RE (2013) Classifying spam emails using text and readability features. In: 2013 IEEE 13th international conference on data mining, pp 657–666. doi:10.1109/ICDM.2013.131
41. Shen X, Tseng GC, Zhang X, Wong WH (2003) On psi-learning. J Am Stat Assoc 98:724–734. http://EconPapers.repec.org/RePEc:bes:jnlasa:v:98:y:2003:p:724-734
42. Sheu JJ (2009) An efficient two-phase spam filtering method based on e-mails categorization. Int J Netw Secur 9(1):34–43
43. Sirisanyalak B, Sornil O (2007) Artificial immunity-based feature extraction for spam detection. In: Software engineering, artificial intelligence, networking, and parallel/distributed computing. SNPD 2007. Eighth ACIS international conference on, vol 3, pp 359–364
44. Vapnik V (1998) Statistical learning theory. Wiley, New York
45. Wang J, Shen X (2007) Large margin semi-supervised learning. J Mach Learn Res 8:1867–1891. http://dl.acm.org/citation.cfm?id=1314561
46. Xu JM, Fumera G, Roli F, Zhou ZH (2009) Training spamassassin with active semi-supervised learning. In: Sixth conference on email and anti-spam
47. Yang J, Liu Y, Liu Z, Zhu X, Zhang X (2011) A new feature selection algorithm based on binomial hypothesis testing for spam filtering. Knowl Based Syst 24(6):904–914
48. Ye M, Tao T, Mai FJ, Cheng XH (2008) A spam discrimination based on mail header feature and SVM. In: Fourth international conference on wireless communications, networking and mobile computing (WiCom08), pp 1–4
49. Zhan J, Oommen BJ, Crisostomo J (2011) Anomaly detection in dynamic systems using weak estimators. ACM Trans Internet Technol 11(1):3:1–3:16
50. Zhu Y, Tan Y (2011) A local-concentration-based feature extraction approach for spam filtering. IEEE Trans Inf Forensics Secur 6(2):486–497