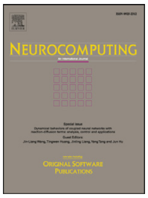
Neurocomputing 000 (2018) 1–11

Contents lists available at ScienceDirect

Neurocomputing

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



Towards automatic filtering of fake reviews

Emerson F. Cardoso, Renato M. Silva, Tiago A. Almeida*

Department of Computer Science, Federal University of São Carlos (UFSCar), Sorocaba, São Paulo, Brazil

ARTICLE INFO

Article history:
Received 24 November 2017
Revised 21 March 2018
Accepted 30 April 2018
Available online xxx

Communicated by Dr Fenza Giuseppe

Keywords:
Fake review
Spam review
Natural language processing
Text categorization
Machine learning

ABSTRACT

Online opinions significantly influence consumer purchase decisions. Unfortunately, this has led to a dramatic increase of fake (or spam) reviews that can damage the reputation of brands and artificially manipulate users' perceptions about products and companies. Despite the efforts of several studies on fake review detection, important questions still remain open. For instance, there is no consensus if the performance of the classification methods is affected when they are used in real-world scenarios that require online learning. Moreover, it is also not known if the performance of the methods decreases due to the time-ordered nature of the reviews. To answer these and other important open questions, this work presents a comprehensive analysis of content-based classification methods for fake review detection. The experiments were performed in multiple settings, employing different types of learning and datasets. A careful analysis of the results provided sufficient evidence to respond appropriately to the open questions, which can be used as a baseline for future studies.

1. Introduction

Expressing opinions is an inherent characteristic of humans. Nowadays, with the popularization of the Internet, opinions are increasingly easier to reach more and more people in different locations around the world. In fact, websites such as TripAdvisor and Yelp are used to share reviews mainly about services, places, and establishments. In the same way, e-commerce platforms, such as Amazon, also allow their users to post reviews about products and services. These online systems have popularized the sharing of reviews and increased people's trust on online opinions. They have also raised competitiveness and, as a consequence, some companies have unfortunately hired people to write fake reviews promoting their products and services or defaming their competitors.

Fake reviews are also known as deceptive opinions, spam opinions, or spam reviews, while their authors are called spammers. They can cause financial loss for product manufacturers and service providers because their brand reputation can be damaged by negative fake reviews. Moreover, companies can also lose customers when fake reviews promote their competitors.

Some examples of spam reviews are shown in Fig. 1. These opinions were written by a group of people in order to elevate the popularity of three specific products. They published 5-star reviews to the same products but on different dates.

Some recent cases of fraudulent use of reviews became public in the news. For example, a chef published fake negative reviews about rival restaurants on TripAdvisor, getting fired after his boss discovered the fraud on social media[1]. In another case, Samsung was fined for hiring spammers to post negative fake reviews about HTC smartphones[2].

The spread of spam reviews is a serious problem on the Internet and they already represent a considerable volume of existing online reviews. Recently, Luca and Zervas [2] estimated that 16% of Yelp restaurant reviews are spam.

In order to reduce this problem, some social networks allow users to report suspicious reviews that might be spam. However, humans are rarely able to detect spam review accurately, since they are written to look authentic [3,4]. To illustrate the difficulty in discerning spam reviews, Ott et al. [5] offered the two real examples presented below. Just one is legitimate.

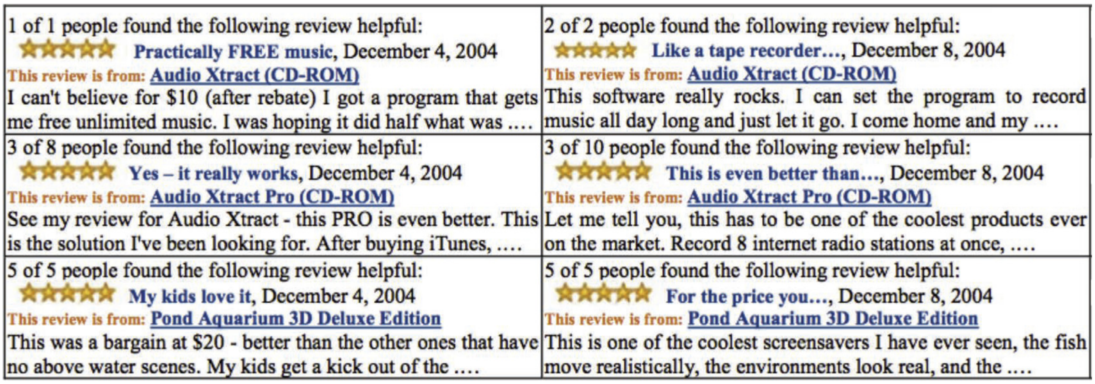
> *I have stayed at many hotels traveling for both business and pleasure and I can honestly stay that The James is tops. The service at the hotel is first class. The rooms are modern and very comfortable. The location is perfect within walking dis-*

* Corresponding author.
E-mail addresses: mersu898@gmail.com (E.F. Cardoso), renatoms@dt.fee.unicamp.br (R.M. Silva), talmeida@ufscar.br (T.A. Almeida).

[1] Chef sacked after putting negative reviews about rivals on TripAdvisor. Available at https://goo.gl/QcR3EZ, accessed on Oct 16, 2017.

[2] Samsung Fined For Paying People to Criticize HTC's Products. Available at https://goo.gl/tmFwYk, accessed on Oct 16, 2017.

https://doi.org/10.1016/j.neucom.2018.04.074
0925-2312/© 2018 Elsevier B.V. All rights reserved.



1 of 1 people found the following review helpful: ★★★★★ **Practically FREE music**, December 4, 2004 This review is from: Audio Xtract (CD-ROM) I can't believe for $10 (after rebate) I got a program that gets me free unlimited music. I was hoping it did half what was ….	2 of 2 people found the following review helpful: ★★★★★ **Like a tape recorder…**, December 8, 2004 This review is from: Audio Xtract (CD-ROM) This software really rocks. I can set the program to record music all day long and just let it go. I come home and my ….
3 of 8 people found the following review helpful: ★★★★★ **Yes – it really works**, December 4, 2004 This review is from: Audio Xtract Pro (CD-ROM) See my review for Audio Xtract - this PRO is even better. This is the solution I've been looking for. After buying iTunes, ….	3 of 10 people found the following review helpful: ★★★★★ **This is even better than…**, December 8, 2004 This review is from: Audio Xtract Pro (CD-ROM) Let me tell you, this has to be one of the coolest products ever on the market. Record 8 internet radio stations at once, ….
5 of 5 people found the following review helpful: ★★★★★ **My kids love it**, December 4, 2004 This review is from: Pond Aquarium 3D Deluxe Edition This was a bargain at $20 - better than the other ones that have no above water scenes. My kids get a kick out of the ….	5 of 5 people found the following review helpful: ★★★★★ **For the price you…**, December 8, 2004 This review is from: Pond Aquarium 3D Deluxe Edition This is one of the coolest screensavers I have ever seen, the fish move realistically, the environments look real, and the ….

Fig. 1. Examples of spam reviews collected from Amazon. Source: Mukherjee et al. [1] (adapted).

> *tance to all of the great sights and restaurants. Highly recommend to both business travellers and couples.*

> *My husband and I stayed at the James Chicago Hotel for our anniversary. This place is fantastic! We knew as soon as we arrived we made the right choice! The rooms are BEAUTIFUL and the staff very attentive and wonderful!! The area of the hotel is great, since I love to shop I couldn't ask for more!! We will definitely be back to Chicago and we will for sure be back to the James Chicago.*

Several approaches for spam review detection are found in literature and most are based on supervised learning [6]. Despite the existing studies, a number of important questions still remain open. Specifically, there is no consensus if the text categorization methods are affected by:

(*i*) the changes in the characteristics of reviews over time;
(*ii*) the polarity of the reviews (compliments vs complaints);
(*iii*) the use of real-world vs artificial reviews to train and evaluate the classifiers;
(*iv*) processing reviews of various type of services or products at the same time;
(*v*) scenarios that naturally require online learning.

Furthermore, there is no consensus on which of the evaluated content-based classification approaches is the best available choice to be used as a baseline for further comparisons.

To fill these important gaps, we conducted a comprehensive comparison of benchmark machine learning methods applied for content-based spam review detection. The experiments were performed in multiple settings and using different types of learning and datasets in order to answer the open questions and offer results for future baseline comparisons.

The remainder of this paper is organized as follows. Section 2 presents related work and open questions in spam review filtering. The experimental settings are detailed in Section 3. Section 4 presents all the results and analysis. Finally, we give our main conclusions and suggestions for future research in Section 5.

2. Related work and open questions

In this section, we briefly introduce several related work on spam detection and spam review detection. We also present open questions in spam review filtering.

2.1. Spam detection

In the past decades, machine learning methods have been applied to a wide range of problems, such as facial recognition [7], font recognition [8], speech recognition [9], diagnosis of diseases [10], and fraud detection [11]. In the last years, machine learning has also been explored to combat spam, a problem that is spreading to various online applications.

Spam detection has been extensively studied in several types of media, such as email [12], webpage [13–15], blogs [16], microblogs [17,18], SMS [19,20], and YouTube [20]. Many traditional machine learning-based methods have been employed, such as support vector machines (SVM) [14,16,17,19–21], naïve Bayes [14,16,20,22], decision trees (DT) [14,16,20,22–24], and k-nearest neighbors (KNN) [14,16,20]. In general, spam filtering approaches are based on the textual content and spam detection is seen as a binary text categorization problem where the categories are spam or ham (non-spam) [19,20,23,25].

Spam review is a different kind of spam. While spam on email, Twitter, blogs, and other media can be easily identified by an experienced user, spam on review imposes extra challenges, since even an experienced user is rarely able to detect it. As a consequence, few labeled datasets are available to train the classification methods [5]. Moreover, there is no consensus if the methods currently used to identify spam are effective to filter spam review.

2.2. Spam review detection

The problem of spam review detection has been studied more exhaustively over the last years. In general, the existing researches can be divided based on the data and features they analyze, as presented below [6,26].

1. *Content-based spam filtering*: studies that have proposed approaches based on textual content of the reviews [3,5,23,25,27–32].
2. *Behavior-based spam filtering*: studies that used information to improve identifying atypical behaviors of reviewers, such as the posting time and geolocation [25,30,33–35].
3. *Spam detection based on information about the product*: studies that have analyzed the information about each product, such as the sales volume, price, product description, and star rating of reviews [30,36].
4. *Spammer groups detection*: studies that focused on detecting groups of spammers [1,37].

The first studies analyzed millions of reviews about electronics collected from the Amazon website [27–29]. Jindal and Liu [27] found the following categories of spam: (*i*) reviews that analyze only specific brands, (*ii*) non-opinion reviews with unrelated

Please cite this article as: E.F. Cardoso et al., Towards automatic filtering of fake reviews, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.04.074

content (usually advertising) and (*iii*) deceptive reviews created to promote or defame a product. The first two categories of spam are easily detectable by traditional machine learning approaches [27]. However, the last one is much more complex and hard to identify even by humans and, as a consequence, it is very difficult to create a labeled dataset [3].

Early studies proposed to create datasets by labeling the duplicates or near duplicates reviews as spam and the remainder as legitimate [28,29]. However, Jindal and Liu [29] found out that many fake reviews were carefully written to look authentic and many of them were not duplicates. Therefore, they concluded that using duplication factor to distinguish spam reviews is not appropriate to create labeled datasets.

Due to the difficulty to label real-world reviews, Ott et al. [5] created a dataset with spam reviews written by a group of people recruited from Amazon Mechanical Turk[3], while the legitimate reviews were collected from the TripAdvisor website. All reviews of this dataset have a positive sentiment polarity (compliments). Ott et al. [31] used a similar approach to create another dataset with only negative sentiment reviews (complaints). In the same way, the dataset created by Harris [3] comprised spam reviews with both positive and negative sentiment polarities. All these datasets can be considered artificial because the spam reviews were not written by real spammers.

Many further researches also employed artificial spam reviews (*e.g.*, Li et al. [4], Ott et al. [31], Yoo and Gretzel [38], Hernández Fusilier et al. [39], Etaiwi and Naymat [40]). The problem is that there is still no consensus if artificial reviews have the same characteristics of reviews written by real spammers. Therefore, it is necessary to examine if the results presented by these studies are not overestimated. In this way, Mukherjee et al. [32] performed experiments using SVM trained with a dataset comprised of real-world spam reviews identified by the Yelp anti-spam filter. According to the authors, the accuracy of the same approach used by Ott et al. [5] (trained with artificial spam reviews) decreased from 89% to less than 70% when real-world spam reviews were employed. So, the authors concluded that methods trained with artificial reviews are not recommended to be used to classify real-world examples. Nevertheless, other classification approaches and learning scenarios need to be considered for more robust conclusions.

In addition, most of the existing studies have ignored the chronological order of reviews and evaluated the methods using an offline (or batch) learning scenario [4,5,23,28,31,32]. However, spam filtering is a typical adversarial classification problem, since the spammers constantly attempt to evade filtering [20,21]. Consequently, static models created by offline learning methods may not be appropriated for spam review detection in real-world scenarios, because the characteristics of the reviews may change over time and the time-ordered nature of the reviews can be very important. In this context, it is mandatory the use of online learning methods, since the examples can be presented one at a time and there is no need to store all the examples in memory during the learning process. Therefore, online learning classifiers are appropriate for dynamic scenarios and, moreover, they are also indicated to deal with large-scale problems [20,41–43].

As mentioned, Ott et al. [5] have studied the problem of spam review filtering using only positive sentiment polarity reviews, Ott et al. [31] used only negative ones, and Harris [3] and Chan et al. [19] analyzed the performance of the methods for each of the polarities and both together. However, as far as we know, there is no study in the literature that analyzes if the sentiment polarity of the reviews can affect the performance of the online and offline classification methods. Hence, it is desirable to find out whether or not specialized models trained with each type of polarity are better than generic models trained with both positive and negative polarities. Hernández Fusilier et al. [39] did this partially by comparing the results of their proposed approach with the results of naïve Bayes and SVM using datasets of positive reviews, negative, and both polarities. The authors concluded that negative reviews are more difficult to classify and a model trained with both types of reviews is better than two specialized models. However, the authors evaluated only two batch learning methods using a single dataset.

Many studies used reviews from only one domain. For example, Ott et al. [5], Chan et al. [19], Al Najada and Zhu [23], Ott et al. [31], Yoo and Gretzel [38], and Hernández Fusilier et al. [39] used datasets with reviews from hotels, Li et al. [25] used only reviews from restaurants, and Harris [3] used only reviews about popular bodybuilding supplements. However, to the best of our knowledge, there is no study that compares the performance of specialized models for each type of service/product with generic models trained with reviews from different domains. Recently, Li et al. [4] presented a sentence weighted neural network (SWNN) model to learn the representation of reviews. They evaluated their approach using reviews from three domains (hotel, restaurant, and doctor). The experiments were performed with each domain individually and with all together. Although they did not make a direct comparison between the results, we can see the performance of their approach was better in the experiments that used only reviews from hotels or restaurants. However, as the authors evaluated only one approach and considered only the offline learning scenario, these findings do not provide enough evidence to conclude if there is a difference between the performance of specialized and generic models.

Based on the issues mentioned above, in this study we raised and analyzed the following open research questions:

Q1: Is the performance of the methods affected in scenarios where it is mandatory the learning process be carried out in an online fashion?

Q2: Is the performance of the methods affected when the reviews are presented in chronological order?

Q3: Since many studies in the literature used artificial datasets, is the performance of the methods preserved in the classification of real-world examples?

Q4: Is the performance of the methods affected by the sentiment polarity of the reviews (positive × negative)?

Q5: Is the performance of the methods affected by the diversity of services/products (training with reviews of various types of services/products vs training with reviews of only one type of service/product)?

Q6: Is there a statistically significant difference between the performance of the established content-based learning methods?

3. Experimental settings

To answer the open research questions, we performed a comprehensive analysis considering the following experimental scenarios:

- *Scenario 1* – offline learning using reviews not ordered chronologically;
- *Scenario 2* – offline learning using reviews sorted by posting time;
- *Scenario 3* – online learning using reviews not ordered chronologically;
- *Scenario 4* – online learning using reviews sorted by posting time.

[3] *Amazon Mechanical Turk*. Available at https://mturk.com, accessed on Oct 16, 2017.

Please cite this article as: E.F. Cardoso et al., Towards automatic filtering of fake reviews, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.04.074

In Scenarios 1 and 2, we evaluated offline learning approaches and, therefore, all the training reviews are presented simultaneously to the classification method. On the other hand, in Scenarios 3 and 4, the methods process and learn with one example at a time. The results obtained in these scenarios can be used to analyze whether the performance of the methods is preserved when they are used in real-world applications that require online learning (Question **Q1**).

We also compared the results of Scenario 1 with Scenario 2 and the results of Scenario 3 with Scenario 4 to analyze whether studies that ignore the chronological order of reviews reflect the reality of spam review filtering (Question **Q2**).

In these four scenarios, we compared the results obtained by methods using artificial reviews and real-world reviews in order to answer the Question **Q3**. Furthermore, we also performed experiments using only negative polarity reviews or positive ones to answer the Question **Q4**. To answer the Question **Q5**, we trained and tested the methods using reviews about only one type of product/service and also with reviews about more than one domain.

In order to answer the Question **Q6**, we performed a comparative analysis of methods widely used as baseline approaches for text classification and with different learning strategies, such as ensemble, optimization, decision tree, probability, compression, and neighborhood. For the evaluation, we employed two large, well-known, and public data collections: TripAdvisor [5] and Yelp [32].

The TripAdvisor collection is the combination of the datasets created by Ott et al. [5] and Ott et al. [31]. The legitimate (ham) reviews were obtained from the 20 most popular Chicago hotels posted in TripAdvisor website. On the other hand, the spam reviews were written by a group of people recruited from Amazon Mechanical Turk and, therefore, these messages are artificial (not created by spammers). In this collection, just the original text messages and classes are available. Other interesting information, such as the IP address, user details, and posting time, is not known.

The Yelp collection [32] comprises real-world reviews about hotels and restaurants extracted from Yelp website. Messages labeled as spam are real and they were identified by the Yelp anti-spam filter. Moreover, in addition to textual content and class, the posting time of the reviews is also available.

For both collections, we first selected the reviews written in English. Then, with the goal of answering the open research questions, we generated datasets with distinct characteristics to conduct the experiments: (*i*) dataset composed only of positive sentiment polarity reviews (compliments), (*ii*) dataset with only negative sentiment polarity reviews (complaints), (*iii*) dataset with both negative and positive polarity reviews, (*iv*) dataset with reviews in a chronological order (posting time-ordered), (*v*) dataset with reviews without a posting order, (*vi*) dataset with reviews about only one product/service, and (*vii*) dataset with reviews about multiple products/services.

In all experiments, we applied lowercase conversion and then we used non-alphanumeric characters as delimiters in the tokenization process. For representing the text, we used the vector space model using n-grams, where the value of n was selected using grid search over the following range of values: $n = \{1, 2, 3\}$. As commonly used in spam filtering, we did not remove stopwords and we did not apply stemming or lemmatization because these processing techniques could remove features that are important for the classification [44].

Tables 1 and 2 summarize the main statistics for each of the datasets composed by messages not ordered chronologically and time-ordered, respectively. The column ID presents the dataset identifier used throughout the text, $|\mathcal{D}|$ corresponds to the number of reviews, $|\mathcal{V}|$ is the number of features (vocabulary size), *#Spam* is the number of reviews labeled as spam, and *#Ham* is the number of reviews labeled as ham (legitimate). Moreover, $\mathcal{M}$ and IQR are the median and the interquartile range of the number of tokens per review, respectively. The TripAdvisor datasets are not presented in Table 2 because the posting time of the reviews is not known.

We can observe the average number of tokens in each review is much smaller than the vocabulary size, which means that each sample is computationally represented by a highly dimensional and sparse vector. Furthermore, the number of terms in negative polarity reviews is on average higher than in positive ones, which indicates that users tend to write more when they make complaints than compliments. These characteristics can also influence the performance of the learning approaches.

3.1. Classification methods

We evaluated established text categorization methods for offline and online learning scenarios as detailed in the following.

3.1.1. Offline learning tasks

The following approaches were evaluated in the offline learning experiments: naïve Bayes multinomial (M.NB) [45], naïve Bayes Bernoulli (B.NB) [45], k-nearest neighbors (KNN) [46], decision trees (DT) [47], random forest (RF) [48], Rocchio [49], and support vector machines (SVM) [50].

In addition to the established methods above, we also performed experiments with MDLText which was recently proposed by Silva et al. [51]. It is an online and multinomial text classification method based on the minimum description length principle. Silva et al. [51] claim that the advantages of this method are its incremental learning capability, low computational cost, and robustness to prevent overfitting which are desirable characteristics for real-world, online, and large-scale text classification problems. Silva et al. [51] evaluated this method using 45 text datasets from various domains in online and offline learning scenarios. According to the authors, MDLText outperformed state-of-the-art methods in both scenarios. Another research by Silva et al. [20] concluded that MDLText also outperformed established online learning methods when it is applied to detect spam in short text messages. More details about the method, such as the mathematical basis, training and classification procedures, and computational complexity are presented in Silva et al. [51] and Silva et al. [20].

We used the implementations of M.NB, B.NB, KNN, DT, RF, Rocchio, and SVM from the `scikit-learn` Library[4], available in `Python`[5]. The experiments with the MDLText were conducted using the original implementation in C++ provided by Silva et al. [51][6].

As the performance of SVM, KNN, RF, and MDLText can be highly affected by the choice of parameters, we performed a grid search using 5-fold cross-validation to find the best values for their main parameters. For the other methods, we used their default values. Table 3 presents the parameters and the range of values tested in the grid-search, where C is the parameter cost for SVM, k corresponds to the number of neighbors for KNN, $|T|$ is the number of trees used in RF, and Ω is a parameter of the MDLText used to preserve a portion of the description length for tokens that have never been seen in training stage.

The experiments with SVM were conducted using a linear kernel because this kernel function usually performs better in high-dimensional problems [52]. For all methods, we used the TF-IDF term weight scheme, except for B.NB, in which we used the binary text representation because it is intrinsic to this method.

[4] *Scikit learn*. Available at http://scikit-learn.org/stable/, accessed on Oct 16, 2017.
[5] *Python*. Available at http://www.python.org, accessed on Oct 16, 2017.
[6] *MDLText*. Available at https://github.com/renatoms88/MDLText, accessed on Oct 16, 2017.

Table 1
Datasets with reviews not ordered chronologically.

Dataset	Polarity	ID	$\vert\mathcal{D}\vert$	$\vert\mathcal{V}\vert$	*#Spam*	*#Ham*	$\mathcal{M}$	IQR
TripAdvisor - Hotels	Neg	T-H-N	800	7596	400	400	99	53
TripAdvisor - Hotels	Pos	T-H-P	800	5548	400	400	71	40
TripAdvisor - Hotels	Pos+Neg	T-H-PN	1600	9571	800	800	84	51
Yelp - Hotels	Neg	Y-H-N	684	7708	400	284	98	82
Yelp - Hotels	Pos	Y-H-P	916	7204	500	416	71	70
Yelp - Hotels	Pos+Neg	Y-H-PN	1600	10,673	900	700	80	78
Yelp - Restaurants	Neg	Y-R-N	1200	10,146	600	600	83	73
Yelp - Restaurants	Pos	Y-R-P	1200	10,358	600	600	69	86
Yelp - Restaurants	Pos+Neg	Y-R-PN	2400	14,923	1200	1200	77	80
Yelp - Hotels+Rest.	Neg	Y-HR-N	1884	13,380	1000	884	87	76
Yelp - Hotels+Rest.	Pos	Y-HR-P	2116	13,401	1100	1016	70	78
Yelp - Hotels+Rest.	Pos+Neg	Y-HR-PN	4000	19,043	2100	1900	78	80

Table 2
Datasets with time-ordered reviews.

Dataset	Polarity	ID	$\vert\mathcal{D}\vert$	$\vert\mathcal{V}\vert$	*#Spam*	*#Ham*	$\mathcal{M}$	IQR
Yelp - Hotels	Neg	Y-H-N-Ord	500	6631	250	250	96	84
Yelp - Hotels	Pos	Y-H-P-Ord	500	5308	250	250	72	75
Yelp - Hotels	Pos+Neg	Y-H-PN-Ord	1000	8676	500	500	82	79
Yelp - Restaurants	Neg	Y-R-N-Ord	1000	9588	500	500	79	70
Yelp - Restaurants	Pos	Y-R-P-Ord	1000	9351	500	500	67	71
Yelp - Restaurants	Pos+Neg	Y-R-PN-Ord	2000	13,773	1000	1.000	74	71
Yelp - Hotels+Rest.	Neg	Y-HR-N-Ord	1500	12,254	750	750	85	75
Yelp - Hotels+Rest.	Pos	Y-HR-P-Ord	1500	11,384	750	750	69	73
Yelp - Hotels+Rest.	Pos+Neg	Y-HR-PN-Ord	3000	16,926	1500	1500	77	76

Table 3
Parameters and range of values used in the grid-search.

Method	Parameter	Range
SVM	C	2^{-15}–2^{15}
KNN	k	5–50
RF	$\vert T\vert$	10–150
MDLText	Ω	2–2^{25}

3.1.2. *Online learning tasks*

The following online learning methods were compared: naïve Bayes multinomial (M.NB) [45], naïve Bayes Bernoulli (B.NB) [45], stochastic gradient descent (SGD) [53], perceptron [54], and MDLText [51].

We performed the experiments with M.NB, B.NB, perceptron, and SGD using the functions of the `scikit-learn` Library, available in `Python`[7]. The experiments with the MDLText were performed using the implementation in C++ provided by Silva et al. [51].

As the performance of MDLText can be affected by the choice of the value for the parameter Ω, we performed a grid search using 5-fold cross-validation to find the best value for it. The range tested in the grid-search was the same presented in Table 3. For the other methods, we used their default value of parameters.

The term weighting scheme we used for all methods was the TF-IDF, except for B.NB, for which we used the binary text representation because it is intrinsic to this method.

4. Results

The experiments were divided into two different sets according to the kind of learning task: *offline* and *online*. In each one, we performed experiments considering different scenarios to find answers for each question that remains open in this context.

4.1. *Offline learning task*

This section presents the results obtained in *Scenarios 1* and *2*. In both, the classification methods were trained using offline learning.

4.1.1. *Scenario 1: results obtained using datasets not ordered chronologically*

In this scenario, the experiments do not take into account the posting time of the reviews. A stratified 5-fold cross-validation was employed to evaluate the performance of the methods.

Table 4 shows the F-measure obtained by each evaluated method. To facilitate comparing the results, the scores are presented as a grayscale heat map in which the better the score of a method for a given dataset, the darker the cell color. The bold values indicate the best scores.

SVM obtained the best performance in the TripAdvisor datasets, while MDLText obtained the best scores for most of the Yelp datasets. We can observe the scores obtained by SVM in the classification of real-world examples of spam reviews (Yelp datasets) were much lower than the scores related to the artificial spam reviews (TripAdvisor datasets). This is in line with the findings reported by Mukherjee et al. [32]. The performance of the other methods was also negatively affected in the experiments with real-world datasets. Therefore, it is clear that using artificial spam reviews can lead to an overestimated performance.

The results obtained in the experiments using only positive polarity reviews were, in general, higher than the ones obtained when only negative polarity reviews were used. Moreover, combining positive and negative reviews did not improve the performance. Therefore, we can conclude that a spam filtering system composed of specialized models for each one of the polarities is more recommended than a system composed of only one generic model.

Another important point is that the combination of reviews of different types of service/products, in general, did not improve the performance of the methods. For example, the scores obtained for the Y-HR-P dataset were, on average, 15.1% lower than the scores

[7] *Python*. Available at http://www.python.org, accessed on Oct 16, 2017.

Table 4
F-measure obtained by the methods in the offline learning scenario for the datasets not ordered chronologically.

ID	B.NB	DT	KNN	M.NB	MDLText	RF	Rocchio	SVM
T-H-N	0.835	0.662	0.684	0.760	0.848	0.848	0.785	**0.873**
T-H-P	0.870	0.686	0.849	0.889	0.870	0.861	0.891	**0.899**
T-H-PN	0.846	0.681	0.761	0.846	0.866	0.848	0.843	**0.878**
Y-H-N	0.613	0.529	0.599	0.588	**0.701**	0.604	0.610	0.607
Y-H-P	0.626	0.564	0.614	0.568	**0.764**	0.650	0.638	0.648
Y-H-PN	0.616	0.557	0.609	0.566	**0.750**	0.637	0.615	0.633
Y-R-N	0.595	0.556	0.542	0.551	**0.722**	0.600	0.569	0.583
Y-R-P	0.778	0.792	0.802	0.762	**0.894**	0.843	0.828	0.869
Y-R-PN	0.680	0.632	0.623	0.566	**0.735**	0.703	0.677	0.687
Y-HR-N	0.564	0.535	0.523	0.535	**0.708**	0.605	0.569	0.577
Y-HR-P	0.681	0.649	0.692	0.637	0.719	0.721	0.719	**0.750**
Y-HR-PN	0.633	0.588	0.583	0.548	**0.717**	0.651	0.632	0.639

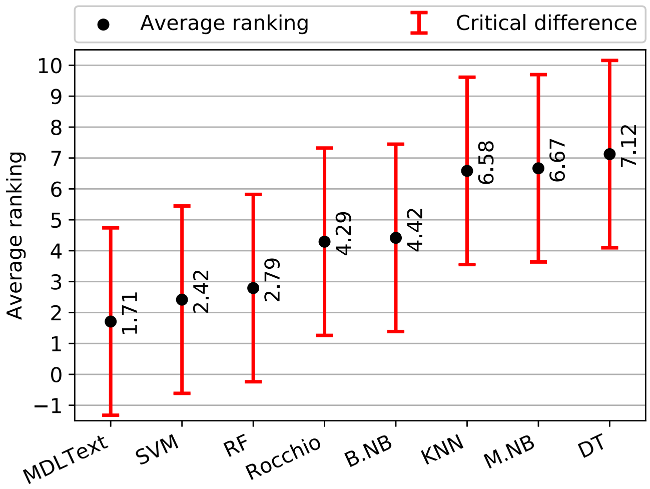


Fig. 2. Average rankings and critical difference calculated using the Nemenyi post hoc test for the Scenario 1.

obtained for the Y-R-P dataset. Therefore, these results indicate that specialized models for a specific product or service can be more efficient.

To ensure that the results were not obtained by chance, we performed a statistical analysis using the non-parametric Friedman test, carefully following the methodology described in Demšar [55]. Friedman test uses information about the average ranking to check if the null hypothesis, which states that all methods presented equivalent performance, can be rejected. Fig. 2 shows the average ranking of the methods based on their F-measure. The smaller the ranking, the better the performance.

For a confidence interval $\alpha = 0.05$, the Friedman test indicated a statistically significant difference between the scores of the methods. Therefore, we performed a pairwise comparison using the Nemenyi post hoc test, which states that the performance of two methods differs significantly if the difference between their ranks is greater than or equal to a critical difference (CD) [55].

The CD calculated using the Nemenyi test for a confidence interval $\alpha = 0.05$ was 3.031 (error bar of Fig. 2). Therefore, sufficient statistical evidence was found to conclude that the performance obtained by MDLText (the method with the lowest average ranking) was significantly better than that of DT, KNN, and M.NB. However, there was not enough statistical evidence to state that the performance obtained by MDLText was significantly better than that of SVM, RF, B.NB, and Rocchio. Furthermore, DT obtained the highest average ranking and, according to the Nemenyi test, its performance was statistically worse than that of MDLText, SVM, and RF.

4.1.2. Scenario 2: results obtained using time-ordered datasets

In this scenario, the reviews were presented to the classification methods in chronological order. The first 80% of reviews were used for training the models and the remainder was used to evaluate their performances.

Table 5 shows the F-measure obtained by each evaluated method. The bold values indicate the best scores.

In general, all methods found more difficult to classify data that vary over time (presented in chronological order). The results obtained in this scenario were lower than the ones obtained in Scenario 1 (Table 4). It indicates that studies which considered a static environment may have reported overestimated results. It is important to consider the chronological order of the reviews because spammers constantly attempt to evade the filter and, therefore, in real-world applications, most recent reviews can contain characteristics not reflected by models trained with past samples.

In this scenario, M.NB was the best method for most of the datasets. The MDLText demonstrated a large drop in performance, which suggests that it has been greatly affected by the time-ordered nature of the experiments. Moreover, as in the previous scenario, the specialized models for each one of the polarities and specialized models for a specific product or service, in general, obtained higher scores than generic models.

We also performed a statistical analysis based on the average ranking of each classification methods (Fig. 3). For a confidence interval $\alpha = 0.05$, the Friedman test indicated that the performance of the methods differs significantly.

We used the Nemenyi test to perform a pairwise comparison between the methods. For a confidence interval $\alpha = 0.05$, CD was equal to 3.500. Consequently, there was enough statistical evidence

Table 5
F-measure obtained by the methods in the offline learning scenario for the time-ordered datasets.

ID	B.NB	DT	KNN	M.NB	MDLText	RF	Rocchio	SVM
Y-H-N-Ord	0.593	0.505	0.495	**0.679**	0.611	0.632	0.673	0.557
Y-H-P-Ord	0.674	0.551	0.692	0.721	0.660	0.702	**0.743**	0.667
Y-H-PN-Ord	0.649	0.599	0.644	**0.721**	0.691	0.636	0.686	0.622
Y-R-N-Ord	0.515	0.537	0.696	**0.711**	0.693	0.639	0.611	0.667
Y-R-P-Ord	**0.786**	0.678	0.700	0.680	0.661	0.728	0.717	0.688
Y-R-PN-Ord	0.689	0.611	0.677	0.704	**0.707**	0.686	0.674	0.697
Y-HR-N-Ord	0.549	0.592	0.614	**0.720**	0.671	0.572	0.647	0.673
Y-HR-P-Ord	**0.697**	0.620	0.693	0.688	0.657	0.681	0.694	0.693
Y-HR-PN-Ord	0.641	0.571	0.621	**0.724**	0.672	0.650	0.672	0.707

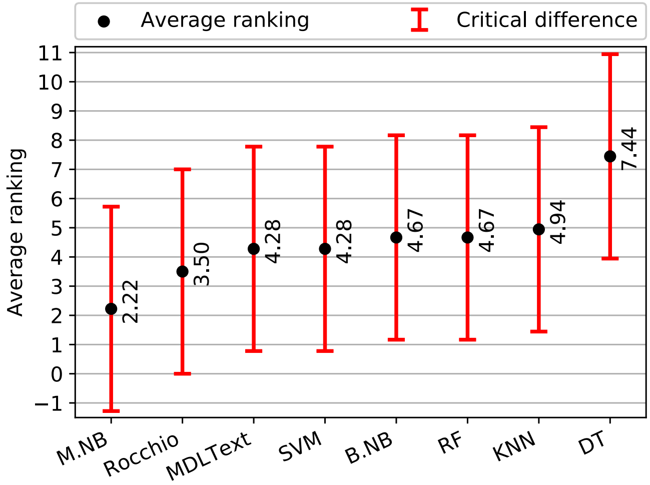


Fig. 3. Average rankings and critical difference calculated using the Nemenyi post hoc test for the Scenario 2.

to state that the performance obtained by the method with the lowest average ranking (M.NB) was significantly better than that of DT. On the other hand, there was not enough statistical evidence to conclude that the performance obtained by M.NB was significantly better than that of the other evaluated methods.

4.2. Online learning task

This section presents the results obtained in *Scenarios 3* and *4*. In both, the classification methods were trained using online learning.

4.2.1. Scenario 3: results obtained using datasets not ordered chronologically

To simulate a real-world online system, we used an evaluation procedure similar to the one recommended by Silva et al. [51]. We randomly partitioned the dataset into k subsets, with $k = 5$, where each subset kept the same proportion of examples in each class. The evaluation procedure was then repeated k times. For each one of the repetitions, only one subset was used for training, while the remainder was used for testing the model. In the testing stage, one message was presented at a time to the classifier, which made its prediction. Then, the classifier received a feedback and if its prediction was wrong, the training model was updated with the true label. We limited the number of updates to 40% of the number of examples in order to simulate a real-world scenario where the user is likely to stop giving feedback to a model that makes many prediction errors. Table 6 shows the average F-measure obtained from this evaluation procedure.

In general, the scores obtained by B.NB, M.NB, and MDLText in this scenario, were lower than the ones obtained in Scenario 1 (offline learning). The differences between the results obtained in these two scenarios indicate that using offline learning to analyze the spam review problem may offer overestimated results. The online learning scenarios are more challenging because the methods are tested after being trained with a small number of examples, which simulates real-world conditions where obtaining labeled examples is very expensive and time-consuming. With feedback being received from the users, the performance of the models tends to improve over time.

In this scenario, M.NB obtained the best results for most of the datasets. On the other hand, perceptron and SGD obtained the lowest scores in most of the experiments. Moreover, as in Scenario 1, the scores obtained using the TripAdvisor datasets are higher than the ones obtained using the Yelp datasets, which indicates that artificial spam reviews are easier to classify than real-world ones.

In online learning, as well as in offline learning, the results obtained for the positive polarity datasets, in general, were better than the results obtained for the negative ones or combined. Furthermore, the results obtained by specialized models (trained with reviews from only one domain) were higher than the results obtained by generic models (trained with reviews from two domains – hotel and restaurant reviews).

In this scenario, we also used the Friedman test to determine whether there were significant differences between the scores of the methods. Based on the average ranks shown in Fig. 4, for a confidence interval $\alpha = 0.05$, the Friedman test indicated that the performance of the methods differs significantly. Then, in the pairwise comparison between the methods using the Nemenyi test, for $\alpha = 0.05$, CD was 1.761. Therefore, M.NB was statistically superior to perceptron and SGD. On the other hand, SGD was statistically inferior to M.NB, MDLText, and B.NB.

4.2.2. Scenario 4: results obtained using time-ordered datasets

This is the most real-world and challenging scenario because it evaluates online learning methods presenting the reviews in chronological order.

As the reviews are sorted by posting time, the first 20% of them is used to train the models, while the remainder is used for testing. The testing stage of this scenario is similar to Scenario 3 (Section 4.2.1), except that the examples are presented in chronological order. Table 7 shows the F-measure obtained by the evaluated learning methods.

Please cite this article as: E.F. Cardoso et al., Towards automatic filtering of fake reviews, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.04.074

Table 6
Average F-measure obtained by the methods in the online learning scenario for the datasets not ordered chronologically.

ID	B.NB	M.NB	MDLText	Perceptron	SGD
T-H-N	**0.823**	0.819	0.806	0.659	0.709
T-H-P	0.848	**0.856**	0.835	0.733	0.740
T-H-PN	**0.852**	**0.852**	0.825	0.724	0.742
Y-H-N	0.644	**0.662**	0.652	0.608	0.607
Y-H-P	0.699	0.703	**0.718**	0.619	0.601
Y-H-PN	0.674	0.690	**0.692**	0.620	0.616
Y-R-N	0.669	**0.676**	0.657	0.576	0.571
Y-R-P	0.799	**0.870**	0.836	0.801	0.797
Y-R-PN	0.712	**0.739**	0.719	0.665	0.664
Y-HR-N	0.655	**0.668**	0.659	0.570	0.600
Y-HR-P	0.748	**0.785**	0.762	0.695	0.711
Y-HR-PN	0.680	**0.716**	0.706	0.637	0.638

Table 7
F-measure obtained by the methods in the online learning scenario for the time-ordered datasets.

ID	B.NB	M.NB	MDLText	Perceptron	SGD
Y-H-N-Ord	0.578	0.577	**0.654**	**0.654**	0.647
Y-H-P-Ord	0.673	0.641	**0.772**	0.655	0.689
Y-H-PN-Ord	0.628	0.555	**0.744**	0.658	0.665
Y-R-N-Ord	0.634	0.618	**0.723**	0.677	0.653
Y-R-P-Ord	0.784	0.760	**0.788**	0.765	0.767
Y-R-PN-Ord	0.697	0.629	**0.752**	0.698	0.710
Y-HR-N-Ord	0.595	0.600	**0.653**	0.578	0.614
Y-HR-P-Ord	0.742	0.721	**0.757**	0.701	0.686
Y-HR-PN-Ord	0.654	0.662	**0.665**	0.643	0.645

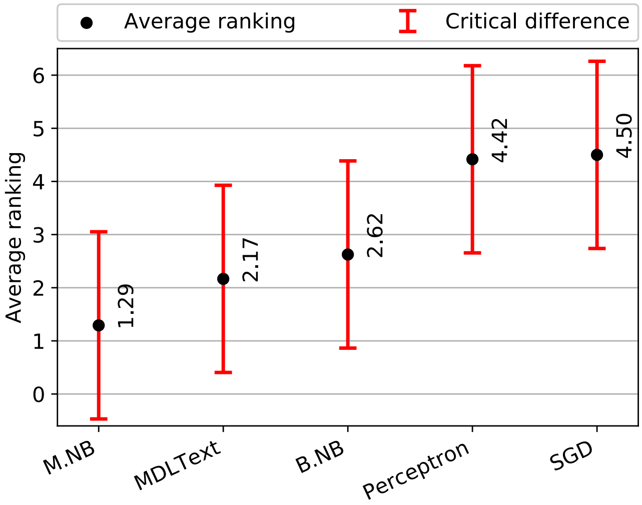


Fig. 4. Average rankings and critical difference calculated using the Nemenyi post hoc test for the Scenario 3.

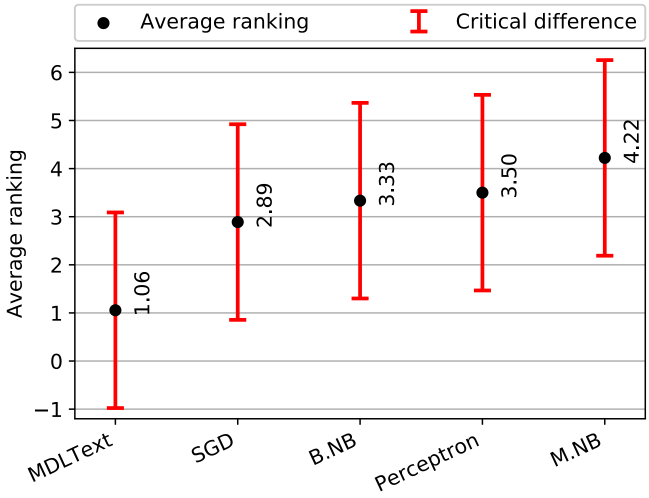


Fig. 5. Average rankings and critical difference calculated using the Nemenyi post hoc test for the Scenario 4.

In this scenario, MDLText achieved the highest scores for all datasets. On the other hand, M.NB obtained the worst performance for most of the datasets. Moreover, as in other scenarios, specialized models for a specific domain (*e.g.*, hotel or restaurant reviews), in general, obtained better performance than generic models. Furthermore, specialized models for only one sentiment polarity also outperformed the models trained and tested with both positive and negative polarity reviews (Fig. 5).

Please cite this article as: E.F. Cardoso et al., Towards automatic filtering of fake reviews, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.04.074

If we compare the results obtained in this scenario with Scenario 3, we can see that the performance of the methods, in general, decreased. However, presenting the reviews in chronological order in the online learning scenario did not impact the performance as negatively as the offline one. The frequent updating of the models made them more robust to the changes in the characteristics of reviews over time.

In this scenario, the Friedman test using $\alpha = 0.05$ also indicated that the performance of the methods differs significantly. Then, in the pairwise comparison, the CD for the Nemenyi test, for $\alpha = 0.05$, was 1.862. Therefore, the performance of MDLText was statistically superior to perceptron, M.NB, and B.NB. However, there is not enough statistical evidence to conclude that it is better than that of SGD.

5. Conclusions

In this study, we raised and analyzed open questions related to spam review filtering which required a careful evaluation to be properly answered. To provide adequate answers, we conducted a comprehensive comparative analysis of different established content-based classification methods. We performed experiments using artificial and real-world datasets. From these, the reviews were divided into two categories: (*i*) reviews sorted by posting time and (*ii*) reviews not ordered chronologically. We also analyzed two different learning strategies: offline learning and online learning. Then, the experiments were divided into four different scenarios:

- *Scenario 1* – offline learning using reviews not ordered chronologically;
- *Scenario 2* – offline learning using reviews sorted by posting time;
- *Scenario 3* – online learning using reviews not ordered chronologically;
- *Scenario 4* – online learning using reviews sorted by posting time.

The results obtained in these four scenarios provided evidence to answer the following research questions:

Q1: *Is the performance of the methods affected in scenarios where it is mandatory the learning process be carried out in an online fashion?*
Yes. The F-scores in Scenarios 3 and 4 were slightly lower than in the offline learning scenarios, probably because the online learning methods are usually trained with a small number of reviews. Therefore, initially, they tend to make more prediction errors. However, since the number of errors tends to decrease with the feedback received over time, we are confident that if there were more reviews in the datasets the results would have been better. Furthermore, we recommend using online methods because they are suitable for scenarios demanding scalable solutions and more adaptable to the continuously evolving spamming techniques.

Q2: *Is the performance of the methods affected when the reviews are presented in time order?*
Yes. In Scenarios 2 and 4, the overall performance of the methods decreased because of the time-ordered nature of the reviews. Therefore, in real-world applications, we recommend the prediction model be updated frequently since we found evidence that there is a degradation of its predictive power over time.

Q3: *Since many studies in the literature used artificial datasets, is the performance of the methods preserved in the classification of real-world examples?*
No. As can be seen in Scenarios 1 and 3, all methods obtained higher scores in the experiments with datasets of artificial reviews. This indicates the studies that used these types of samples presented overestimated results.

Q4: *Is the performance of the methods affected by the sentiment polarity of the reviews? (compliments × complaints)?*
Yes. In general, the scores obtained in the experiments with only negative reviews were worse. Furthermore, in the experiments when the methods were trained and tested with reviews with both positive and negative polarities, their scores were lower than in the experiments that used only reviews with positive polarity. Therefore, we recommend the use of specialized models for each type of polarity.

Q5: *Is the performance of the methods affected by the diversity of services/products (training with reviews of various types of services/products* vs *training with reviews of only one type of service/product)?*
Yes. We observed that the performance of the methods can be affected depending on the kind of product/service. The experiments with reviews on Yelp about hotels presented lower scores than using reviews about restaurants. Furthermore, there was no increase in performance for joined reviews about both services. Therefore, we recommend using specialized models for each type of service/product.

Q6: *Is there a statistically significant difference between the performance of the content-based methods for fake review detection?*
To answer the above question, we compared the performance of the following benchmark machine learning methods considering the offline learning scenarios: M.NB, Rocchio, SVM, MDLText, B.NB, RF, KNN, and DT. Moreover, in the online learning scenarios, we analyzed the following methods: M.NB, MDLText, B.NB, perceptron, and SGD. The statistical analysis of the results found no evidence that there was any method superior to the others in all scenarios. However, in the most real-world scenario (Scenario 4 – the one that used real-world reviews, online learning, and time-ordered samples), the MDLText obtained the best average ranking and was considered statistically superior to the following methods: M.NB, B.NB, and perceptron. Thus, we recommend using this method as a baseline in a future comparison.

In future research, we intend to investigate the potential benefits of combining the content-based features with other, such as the behavior of the reviewer, information about the products/services, and their sales volume. Another issue that can be addressed in future research is the proposition of new labeled datasets which would make it possible to perform more robust analysis. Alternatively, considering the difficulty in obtaining real-world labeled samples, the effectiveness of semi-supervised learning methods can be investigated.

Acknowledgment

The authors are grateful for financial support from the Brazilian agencies FAPESP and Capes (Grant no. 1709642).

References

[1] A. Mukherjee, B. Liu, N. Glance, Spotting fake reviewer groups in consumer reviews, in: Proceedings of the 21st international conference on World Wide Web (WWW), ACM New York, Lyon, France, 2012, pp. 191–200.
[2] M. Luca, G. Zervas, Fake it till you make it: Reputation, competition, and Yelp review fraud, Manag. Sci. 62 (12) (2015) 3412–3427.
[3] C.G. Harris, Detecting deceptive opinion spam using human computation, in: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI), AAAI Press, Toronto, Canada, 2012, pp. 87–93.

[4] L. Li, B. Qin, W. Ren, T. Liu, Document representation and feature combination for deceptive spam review detection, Neurocomputing 254 (Supplement C) (2017) 33–41. Recent Advances in Semantic Computing and Personalization. doi: 10.1016/j.neucom.2016.10.080.

[5] M. Ott, Y. Choi, C. Cardie, J.T. Hancock, Finding deceptive opinion spam by any stretch of the imagination, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT), vol. 1, Association for Computational Linguistics, Portland, OR, USA, 2011, pp. 309–319.

[6] M. Crawford, T.M. Khoshgoftaar, J.D. Prusa, A.N. Richter, H.A. Najada, Survey of review spam detection using machine learning techniques, J. Big Data 42 (7) (2015) 3634–3642.

[7] D. Tao, Y. Guo, Y. Li, X. Gao, Tensor rank preserving discriminant analysis for facial recognition, IEEE Trans. Image Process. 27 (1) (2018) 325–334, doi:10.1109/TIP.2017.2762588.

[8] D. Tao, X. Lin, L. Jin, X. Li, Principal component 2-d long short-term memory for font recognition on single chinese characters, IEEE Trans. Cybern. 46 (3) (2016) 756–765, doi:10.1109/TCYB.2015.2414920.

[9] X. Zhang, X. Liu, Z.J. Wang, Evaluation of a set of new ORF kernel functions of SVM for speech recognition, Eng. Appl. Artif. Intell. 26 (10) (2013) 2574–2580, doi:10.1016/j.engappai.2013.04.008.

[10] L. Nanni, An ensemble of classifiers for the diagnosis of erythemato-squamous diseases, Neurocomputing 69 (7) (2006) 842–845. New Issues in Neurocomputing: 13th European Symposium on Artificial Neural Networks. doi: 10.1016/j.neucom.2005.09.007.

[11] J.Z. Lei, A.A. Ghorbani, Improved competitive learning neural networks for network intrusion and fraud detection, Neurocomputing 75 (1) (2012) 135–145, doi:10.1016/j.neucom.2011.02.021.

[12] C.H. Li, J.X. Huang, Spam filtering using semantic similarity approach and adaptive bpnn, Neurocomputing 92 (2012) 88–97. Data Mining Applications and Case Study. doi: 10.1016/j.neucom.2011.09.036.

[13] T. Largillier, S. Peyronnet, Webspam demotion: Low complexity node aggregation methods, Neurocomputing 76 (1) (2012) 105–113.

[14] R.M. Silva, T. Almeida, A. Yamakami, Machine learning methods for spamdexing detection, Int. J. Inf. Secur. Sci. 2 (3) (2013) 86–107. http://www.ijiss.org/ijiss/index.php/ijiss/article/view/35.

[15] R.M. Silva, T.A. Almeida, A. Yamakami, An analysis of machine learning methods for spam host detection, in: Proceedings of the 11th International Conference on Machine Learning and Applications (ICMLA), IEEE, Boca Raton, Florida, USA, 2012, pp. 227–232. doi: 10.1109/ICMLA.2012.161.

[16] T.C. Alberto, J.V. Lochter, T.A. Almeida, Post or block? advances in automatically filtering undesired comments, J. Intell. Robot. Syst. 80 (1) (2015) 245–259, doi:10.1007/s10846-014-0105-y.

[17] F. Wu, J. Shu, Y. Huang, Z. Yuan, Co-detecting social spammers and spam messages in microblogging via exploiting social contexts, Neurocomputing 201 (2016) 51–65, doi:10.1016/j.neucom.2016.03.036.

[18] H. Shen, F. Ma, X. Zhang, L. Zong, X. Liu, W. Liang, Discovering social spammers from multiple views, Neurocomputing 225 (2017) 49–57, doi:10.1016/j.neucom.2016.11.013.

[19] P.P. Chan, C. Yang, D.S. Yeung, W.W. Ng, Spam filtering for short messages in adversarial environment, Neurocomputing 155 (C) (2015) 167–176, doi:10.1016/j.neucom.2014.12.034.

[20] R.M. Silva, T.C. Alberto, T.A. Almeida, A. Yamakami, Towards filtering undesired short text messages using an online learning approach with semantic indexing, Expert Syst. Appl. 83 (2017) 314–325, doi:10.1016/j.eswa.2017.04.055.

[21] H. Xiao, B. Biggio, B. Nelson, H. Xiao, C. Eckert, F. Roli, Support vector machines under adversarial label contamination, Neurocomputing 160 (2015) 53–62, doi:10.1016/j.neucom.2014.08.081.

[22] X. Zheng, Z. Zeng, Z. Chen, Y. Yu, C. Rong, Detecting spammers on social networks, Neurocomputing 159 (2015) 27–34, doi:10.1016/j.neucom.2015.02.047.

[23] H. Al Najada, X. Zhu, iSRD: Spam review detection with imbalanced data distributions, in: Proceedings of the 15th IEEE International Conference on Information Reuse and Integration (IRI), IEEE Computer Society, Redwood City, CA, USA, 2014, pp. 553–560.

[24] M. Alsaleh, A. Alarifi, F. Al-Quayed, A. Al-Salman, Combating comment spam with machine learning approaches, in: Proceedings of the 14th International Conference on Machine Learning and Applications (ICMLA), IEEE, Miami, FL, USA, 2015, pp. 295–300.

[25] H. Li, Z. Chen, A. Mukherjee, B. Liu, J. Shao, Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns., in: Proceedings of the 9th International AAAI Conference on Web and Social Media (ICWSM), AAAI Press, Oxford, England, 2015, pp. 634–637.

[26] A. Heydari, M.A. Tavakoli, N. Salim, Z. Heydari, Detection of review spam: a survey, Expert Syst. Appl. 42 (7) (2015) 3634–3642.

[27] N. Jindal, B. Liu, Review spam detection, in: Proceedings of the 16th International Conference on World Wide Web (poster paper) (WWW.), ACM New York, Banff, Canada, 2007, pp. 1189–1190.

[28] N. Jindal, B. Liu, Analyzing and detecting review spam, in: Proceedings of 7th IEEE International Conference on Data Mining (ICDM), IEEE Computer Society, Omaha, NE, USA, 2007, pp. 547–552.

[29] N. Jindal, B. Liu, Opinion spam and analysis, in: Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM), ACM New York, Palo Alto, CA, USA, 2008, pp. 219–229.

[30] C.L. Lai, K.Q. Xu, R.Y.K. Lau, Y. Li, L. Jing, Toward a language modeling approach for consumer review spam detection, in: Proceedings of the 7th IEEE 7th International Conference on E-Business Engineering (ICEBE), Shanghai, China, 2010, pp. 1–8.

[31] M. Ott, C. Cardie, J.T. Hancock, Negative deceptive opinion spam., in: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT–NAACL), The Association for Computational Linguistics, Atlanta, Georgia, USA, 2013, pp. 497–501.

[32] A. Mukherjee, V. Venkataraman, B. Liu, N. Glance, What Yelp fake review filter might be doing? in: Proceedings of the 7th International Conference on Weblogs and Social Media (ICWSM), AAAI Press, Cambridge, MA, USA, 2013, pp. 409–418.

[33] N. Jindal, B. Liu, E.P. Lim, Finding unusual review patterns using unexpected rules, in: Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM), ACM New York, Toronto, Canada, 2010, pp. 1549–1552.

[34] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, R. Ghosh, Exploiting burstiness in reviews for review spammer detection, in: Proceedings of the 7th International Conference on Weblogs and Social Media (ICWSM), AAAI Press, Cambridge, MA, USA, 2013, pp. 175–184.

[35] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, et al., Spotting opinion spammers using behavioral footprints, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), ACM, Chicago, Illinois, USA, 2013, pp. 632–640, doi:10.1145/2487575.2487580.

[36] S.P. Algur, A.P. Patil, P.S. Hiremath, S. Shivashankar, Conceptual level similarity measure based review spam detection, in: Proceedings of the 2010 International Conference on Signal and Image Processing (ICSIP), Chennai, India, 2010, pp. 416–423. doi: 10.1109/ICSIP.2010.5697509.

[37] H. Li, G. Fei, S. Wang, B. Liu, W. Shao, A. Mukherjee, et al., Bimodal distribution and co-bursting in review spam detection, in: Proceedings of the 26th International Conference on World Wide Web (WWW'17), International World Wide Web Conferences Steering Committee, Perth, Australia, 2017, pp. 1063–1072.

[38] K.H. Yoo, U. Gretzel, Comparison of deceptive and truthful travel reviews, in: Proceedings of the 2009 International Conference on Information and Communication Technologies in Tourism (ENTER), Springer Vienna, Amsterdam, Netherland, 2009, pp. 37–47. ISBN 978-3-211-93971-0. doi: 10.1007/978-3-211-93971-0_4.

[39] D. Hernández Fusilier, M. Montes-y Gómez, P. Rosso, R. Guzmán Cabrera, Detecting positive and negative deceptive opinions using pu-learning, Inf. Process. Manag. 51 (4) (2015) 433–443, doi:10.1016/j.ipm.2014.11.001.

[40] W. Etaiwi, G. Naymat, The impact of applying different preprocessing steps on review spam detection, in: Proceedings of the 8th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN), 2017, pp. 273–279, doi:10.1016/j.procs.2017.08.368.

[41] L. Guo, J. hua Hao, M. Liu, An incremental extreme learning machine for online sequential learning problems, Neurocomputing 128 (2014) 50–58, doi:10.1016/j.neucom.2013.03.055.

[42] ViktorLosing, B. Hamme, H. Wersing, Incremental on-line learning: A review and comparison of state of the art algorithms, Neurocomputing 275 (2018) 1261–1274, doi:10.1016/j.neucom.2017.06.084.

[43] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, F. Herrera, A survey on data preprocessing for data stream mining: Current status and future directions, Neurocomputing 239 (2017) 39–57, doi:10.1016/j.neucom.2017.01.078.

[44] J.R. Méndez, E.L. Iglesias, F. Fdez-Riverola, F. Díaz, J.M. Corchado, Tokenising, stemming and stopword removal on anti-spam filtering domain, in: Proceedings of the 11th Spanish Association Conference on Current Topics in Artificial Intelligence (CAEPIA), Springer-Verlag, Santiago de Compostela, Spain, 2006, pp. 449–458.

[45] A. McCallum, K. Nigam, et al., A comparison of event models for naive bayes text classification, in: Proceedings of the AAAI Workshop on Learning for Text Categorization, Citeseer, 1998. 752, 41–48.

[46] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.

[47] L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen, Classification and Regression Trees, CRC Press, Boca Raton, FL, USA, 1984.

[48] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

[49] J.J. Rocchio, Relevance feedback in information retrieval, in: G. Salton (Ed.), The Smart Retrieval System - Experiments in Automatic Document Processing, Prentice-Hall, Englewood Cliffs, NJ, 1971, pp. 313–323.

[50] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.

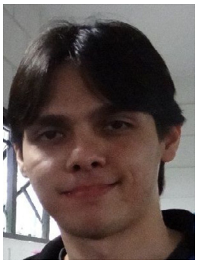
[51] R.M. Silva, T.A. Almeida, A. Yamakami, MDLText: An efficient and lightweight text classifier, Knowl. Based Syst. 118 (2017) 152–164.

[52] C. Hsu, C. Chang, C. Lin, A Practical Guide to Support Vector Classification, Tech. Rep.; National Taiwan University. 2003.

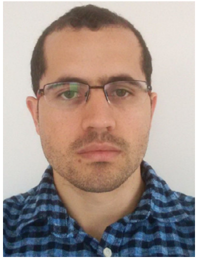
[53] T. Zhang, Solving large scale linear prediction problems using stochastic gradient descent algorithms, in: Proceedings of the twenty-first international conference on Machine learning (ICML), ACM New York, Banff, Alberta, Canada, 2004, p. 116.

[54] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain., Psychol. Rev. 65 (6) (1958) 386.

[55] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (7) (2006) 3634–3642.



Emerson F. Cardoso He received the M.Sc. degree in Computer Science from the Federal University of São Carlos (UFSCar), Sorocaba, São Paulo, Brazil. His current research interests include fake reviews, machine learning, and natural language processing.



Renato M. Silva He received the Ph.D. degree in Electrical Engineering from the School of Electrical and Computer Engineering at University of Campinas (Unicamp), São Paulo, Brazil. He is currently a postdoctoral researcher at the Department of Computer Science at Federal University of São Carlos. His current research interests include artificial intelligence, machine learning, natural language, and pattern recognition.

Tiago A. Almeida He is a professor and researcher from the Department of Computer Science at Federal University of São Carlos (UFSCar), Sorocaba, São Paulo, Brazil. He is Ph.D. in Electrical Engineering from the University of Campinas (Unicamp), São Paulo, Brazil. He has experience in computer science with emphasis on machine learning, artificial intelligence, natural language, and pattern recognition.