# Automatic Online Fake News Detection Combining Content and Social Signals

Marco L. Della Vedova
Università Cattolica
Brescia, Italy
marco.dellavedova@unicatt.it

Eugenio Tacchini
Università Cattolica
Piacenza, Italy
eugenio.tacchini@unicatt.it

Stefano Moret
École Polytechnique Fédérale
Lausanne, Switzerland
moret.stefano@gmail.com

Gabriele Ballarin
Independent researcher
Padova, Italy
gabriele.ballarin@gmail.com

Massimo DiPierro
DePaul University
Chicago, USA
mdipierro@cs.depaul.edu

Luca de Alfaro
Department of Computer Science
UC Santa Cruz, CA, USA
luca@ucsc.edu

***Abstract*—The proliferation and rapid diffusion of fake news on the Internet highlight the need of automatic hoax detection systems. In the context of social networks, machine learning (ML) methods can be used for this purpose. Fake news detection strategies are traditionally either based on content analysis (i.e. analyzing the content of the news) or - more recently - on social context models, such as mapping the news' diffusion pattern.**

**In this paper, we first propose a novel ML fake news detection method which, by combining news content and social context features, outperforms existing methods in the literature, increasing their already high accuracy by up to 4.8%. Second, we implement our method within a Facebook Messenger chatbot and validate it with a real-world application, obtaining a fake news detection accuracy of 81.7%.**

## I. Introduction

The reliability of information diffused on the World Wide Web (WWW) is a central issue of modern society. In particular, in the recent years the spreading of misinformation [1] and *fake news* on the Internet has drawn increasing attention, and has reached the point of dramatically influencing political and social realities. As an example, [2] showed the significant impact of fake news in the context of the 2016 US presidential elections; [3] analyzed the most viral *tweets* related to the Boston Marathon blasts in 2013, finding that the share of rumors and fake content was higher than the share of true information. Note that, in this work, we define *fake news* as "*news articles that are intentionally and verifiably false*" [2], [4]. We use the terms *fake news* and *hoax* interchangeably, as in [5].

The traditional way of verifying online content, i.e. via "manual" knowledge-based fact-checking [4], is made difficult – or, practically, impossible – by the "*enormous volume of information that is now generated online*" [6] and the rapidity of its diffusion. This is particularly true in the case of social network sites (SNSs), online platforms where users can freely share content which can go viral in a few hours [7]. Thus, various authors agree on the need of automatic, computational hoax detection systems [8], [9], [6]. As argued by [6], this "*may significantly enhance our ability to evaluate the veracity of dubious information*".

Fake news detection methods have been recently classified into two categories - news content models and social context models - based on their main input sources [4]. Methods belonging in the first category focus on the content of the news, i.e. the body-text, the title, and few additional metadata (when available); here, we refer to these methods as *content-based* methods. Methods belonging in the second category focus on social features and signals, such as the engagement and interaction of users with a given news on social media (e.g. "liking" a news on Facebook, "retweeting" it on Twitter, etc.); here, we refer to these methods as *social-based* methods. A similar classification of methods was previously proposed by [10], who additionally pointed out that "*both [types of methods] typically incorporate machine learning (ML) techniques*". Thus, we focus our review solely on ML methods, although other approaches have been proposed, e.g. based on statistical analyses [11] or knowledge graphs [9], [6], [12].

Content-based methods are the traditional approach, as they find application in conventional news media and - more in general - in all cases in which no social information is available. Historically, these methods have been used for spam detection in email messages [13] and webpages [14]. In the last years, they have also been applied for fake news detection: [15] exploited syntactic and semantic features for classifying between real articles and articles generated by sampling a trigram language model, obtaining a 91.5% accuracy; [16] used convolutional neural networks with text and additional metadata on a large political fake news dataset; in the context of a recent fake news challenge, [17] showed that using a relatively simple approach based on term frequency (TF) and term frequency-inverse document frequency (TF-IDF) could already offer a good baseline accuracy of 88.5%; [18] also recently used TF-IDF and six different ML classifiers on a 2000 news dataset, obtaining a 92% accuracy.
The main difficulty in applying content-based methods for real-world fake news detection is that these news are "*in-*

ISSN 2305-7254

*tentionally written to mislead consumers, which makes it nontrivial to detect simply based on news content*" [19]. Additionally, [20] recently showed that they could use the news' writing style to effectively discriminate hyperpartisan news and satire from mainstream news, but they could not claim "*to have solved fake news detection via style analysis alone*". These difficulties are probably the reason behind the rather limited use of content-based methods *alone* for fake news detection on social media. In fact, on platforms such as SNSs, additional information about social context features is available, which can help identifying fake news with higher accuracies compared to content-based approaches, as shown by [21] for the case of Twitter.

Social-based methods make use of this additional information, and thus constitute a more recent and promising strategy for fake news detection on social media [4]. Example of features which have been used for this purpose are the characteristics of users (e.g. registration age, number of followers, etc.) - as proposed by [22] and [23] for the case of Twitter - or their opinions and viewpoints, exploited by [24] to assess credibility of content in the same SNS. An alternative social-based strategy for fake news detection on social media is based on mapping the diffusion pattern of information. The rationale behind this strategy lies in the dynamics of social media sharing and interaction; in fact, according to [7], "*users tend to aggregate in communities of interest, which causes reinforcement and fosters confirmation bias, segregation, and polarization*", and "*users mostly tend to select and share content according to a specific narrative and to ignore the rest*". The idea has been first proposed and implemented by [5], who showed that Facebook posts can be classified with high accuracy as hoaxes or non-hoaxes on the basis of the users who "like" them. They applied two ML algorithms (logistic regression and harmonic boolean label crowdsourcing) using the IDs of users as features to classify posts, and obtained accuracies exceeding 99% even with very small training sets.
Although the method proposed by [5] offers difficult-to-beat performance, its application is inherently limited to cases in which information about the propagation of news in the network is available. In other words, as the method uses social interactions (i.e. "likes") as signals to help classifying Facebook posts, it cannot be used when a post has no likes, and it will presumably perform worse when a post collects only few social interactions. Having little information about social interactions is a rather typical situation on SNSs, e.g. in the case of just-posted content (*early detection*), or when information is shared through copy-pasting across the network.

The key idea behind our work, which constitutes its first - methodological - novelty, is that in these cases, in which social-based methods perform poorly, content-based methods can complement them. Thus, building on the work by [5], we present a novel fake news detection approach which, by combining content-based and social-based methods, outperforms existing approaches in the literature. In particular, we obtain higher accuracies than [5] and [19], using their respective Facebook and Twitter datasets for the comparison. Combining content-based and social-based approaches for prediction and classification tasks is a solution that has been successfully applied in other fields: in the recommender systems field, for example, the so-called *hybrid recommender systems* are used to overcome the limitation that collaborative filtering (i.e. social-based) methods face when an item has zero ratings [25], a situation also known as *cold-start*: in those cases, an additional technique based on the analysis of the item's content is combined with the collaborative filtering approach to mitigate the cold-start problem. As previously highlighted, the problem we face with fake news detection is quite similar, hence the idea of combining context-based and social-based methods to provide automatic detection tools that can work without (or with limited) social signals. This can make easier the task of early detection of fake news, that, in turns, can limit the spread of fake news as a whole.

Second, we implement our method within a Facebook Messenger *chatbot* and validate it with an external and independent set of fake and real news shared on the SNS to simulate a real-world application, obtaining a detection accuracy of 81.7%. To the best of our knowledge, this is one of the very first automatic fake news detection bots.

## II. Methodology

Our goal is to classify a news item as reliable or fake; in this section, we first describe the datasets we used for our tests, then we present the content-based approach we implemented and the method we propose to combine it with a social-based approach available in the literature.

### A. Datasets

We validated our approach using three different datasets. The first one is the same used in [5]: this allows to easily compare the accuracy of our method with the accuracy of a purely social-based method. The dataset consists of the public posts and posts' likes of a list of Facebook pages (selection based on [1]) belonging in two categories: scientific news sources vs. conspiracy news sources. The resulting dataset is composed of 15,500 posts, coming from 32 pages (14 conspiracy pages, 18 scientific pages), with more than 2,300,00 likes by 900,000+ users. 8,923 (57.6%) posts are hoaxes and 6,577 (42.4%) are non-hoaxes. Additional details about the dataset are provided by [5].

The second and third datasets come from the FakeNewsNet dataset, recently published by [4]; we used both the PolitiFact and BuzzFeed news sets they provide: the former contains a ground truth of 240 news (half labeled as fake, half labeled as real by the well recognized fact-checking website PolitiFact – http://www.politifact.com/subjects/), the latter a ground truth of 182 news (half labeled as fake, half labeled as real by expert opinion of journalists from BuzzFeed – https://www.buzzfeed.com). Both datasets provide, for each news, the text content of the news and the anonymized IDs of the users who posted/spread the news on Twitter (among other information).

In the remaining of this article, we will use the short terms FacebookData, PolitiFactData, BuzzFeedData to denote, respectively, the three aforementioned datasets.

### B. Content-based method

For the FacebookData dataset, we produced, for each Facebook post, a text corpus joining the actual text content of the post (retrieved using the Facebook Graph APIs - https://developers.facebook.com/docs/graph-api) and, if the post shared a link, the title and text preview of the link (as provided by the Facebook Graph APIs) together with the actual content of the shared webpage.

To retrieve the content of a webpage, we applied some simple heuristics: we removed the CSS and Javascript content from the page, then we extracted the text contained in the remaining HTML tags and, in order to discard useless content (such as menu items), we kept only the lines having more than $n$ words. In this work, we fixed $n = 7$. Each word of the corpus has then been stemmed and each post has been represented as a vector of TF-IDF frequencies on the stems vocabulary. Note that we used Python snowballstemmer (https://pypi.python.org/pypi/snowballstemmer), setting the language to Italian since all the text content of the pages was in Italian Finally, we performed the post classification using a logistic regression model.

As for the PolitiFactData and BuzzFeedData datasets the content was already available, we used only the *text* value as provided in [4] and we applied the same classification method, only changing the stemmer, since the text content of all the news was in English. We used the Porter Stemmer (available at http://www.nltk.org/) in this case.

### C. Combining social and content signals

Our intuition, as discussed in the Introduction, is that social-based methods - and in particular the boolean crowdsourcing algorithms presented in [5] - work extremely well (even with very limited training sets) when they have to classify a post whose number of social interactions is above a certain threshold, while their performance might get worse when only little information about social interactions is available. In these cases, content-based methods can complement them.

We therefore defined a threshold $\lambda$ and classified the posts combining content-based and social-based approaches. In particular, we combined each of the two social-based methods proposed in [5] with the content-based method introduced in the previous section using a simple rule:

$$\begin{cases} likes < \lambda : \text{use the content-based classifier} \\ likes \geq \lambda : \text{use the social-based classifier} \end{cases} \quad (1)$$

Where *likes* is the number of users who like a post or, more generally, the number of social interactions collected by the post. The model is intentionally simple, yet it captures the different contributions of the two (alternative) approaches, it guarantees a simple implementation and, as we will show in the results section, its accuracy is higher than the one provided by more sophisticated models.

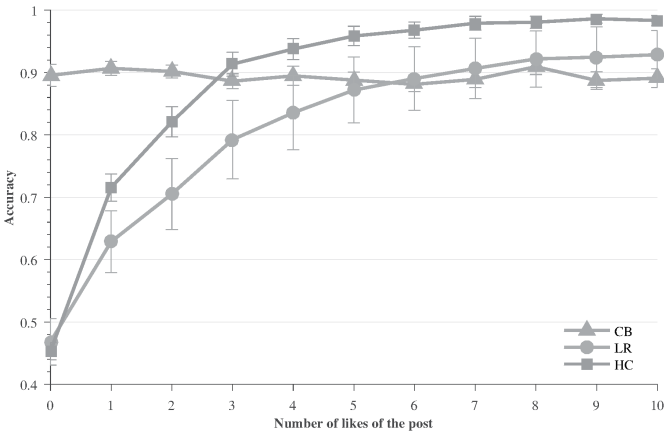


Fig. 1. Accuracy of baseline methods on classifying news items with different number of likes (FacebookData – shuffle split cross validation with 50 iterations – training set size 0.1)

We then evaluated the performances of the combined method using *accuracy* (the same metric used in [5]) plus some additional metrics that make easier the comparison with other methods in the literature: *F1 score*, *precision*, *recall*. We also carried out a sensitivity analysis to study how the accuracy of our classifier is affected by changes in the threshold $\lambda$.

## III. Results

### A. Baseline methods evaluation

First, we analyze how the content-based only and social-based only methods perform when varying the volumes of social interactions (i.e. number of likes on Facebook, shares of Twitter, etc.). As stated in the previous section, we consider the following methods:

- Content-based (CB)
- Logistic regression (LR) on social signals
- Harmonic boolean label crowdsourcing (HC) on social signals

where CB is the method based on the content of the news item described in Section II-B; while LR and HC are the methods based on social signals only, as proposed in [5]. The results shown in Fig. 1 are obtained with a shuffle split cross validation with 50 iterations and a training set size equals to the 10% of the entire FacebookData dataset. On the one hand, it can be noted that accuracy of CB does not vary significantly with the number of likes on the Facebook post. On the other hand, the accuracy of LR and HC increases with the number of likes, as expected. This confirms our intuition: the accuracy of the social-based methods is lower than CB when the volume of social interactions is low, and higher when this volume is high.

### B. Sensitivity analysis for the proposed methods

The last observation suggests to explore the combination of the content-based with the social-based methods for taking the best of both approaches and thus increase the overall accuracy.

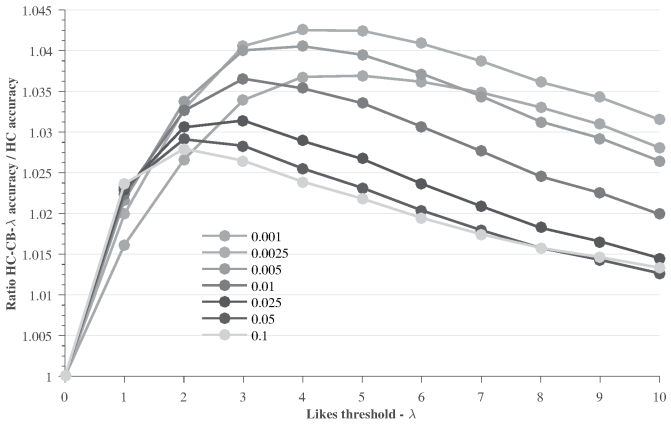


Fig. 2. Accuracy improvement of HC-CB-$\lambda$ w.r.t HC varying the $\lambda$ parameter and the training set size (FacebookData – shuffle split cross validation, avg. of 50 iterations). The legend indicates the size of the training set.

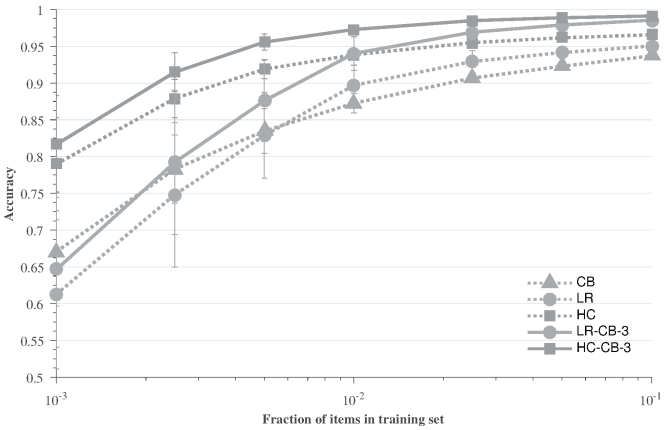


Fig. 3. Accuracy varying the training size (FacebookData – shuffle split cross validation with 50 iterations)

In particular, we consider two combined methods as stated in Section II-C, namely

- HC-CB-$\lambda_H$, which combines HC with CB, and
- LR-CB-$\lambda_L$, which combines LR with CB.

In our notation, $\lambda_H$ and $\lambda_L$ represent the thresholds on social signals that decide whether to use the CB method or the social-based method (see Eqn. 1) with HC and LR, respectively. As an example, we denote with HC-CB-3 the classification method in which CB is used for news items with 2 or less social interactions and HC is used for news items with 3 or more social interactions.

The optimal value of these threshold-parameters can depend on many factors, such as the composition of the dataset and the ratio between the training and the test set sizes. In Table I we show the accuracy of HC-CB-$\lambda$ and LR-CB-$\lambda$ varying the threshold $\lambda$ and the training set size. Results correspond to the average value of the accuracy obtained with a shuffle split cross validation with 50 iterations. Note that the column with $\lambda = 0$ corresponds to the accuracy of the social-based methods without the CB contribution: in this case, the accuracy of the combined methods increases for each $\lambda$ listed in the table. This fact can be better observed in Fig. 2, where the values of HC-CB-$\lambda$'s accuracy are plotted normalized to the corresponding HC accuracies, varying $\lambda$. The optimal value for each training set size corresponds to the $\lambda$ for which the curve is maximized, which ranges from 2 to 5 depending on the training set size. Therefore, even if it is difficult to establish a general rule to derive the optimal $\lambda$, the combined methods are better than the purely social-based methods in terms of accuracy for any (small) value of the threshold.

### C. Comparison with the literature

In this section we compare the proposed methods with methods recently proposed in the literature, in particular in the works by [5] and by [19]. Notably, in what follows we analyze the performances of our methods against theirs, using the same datasets presented in their original papers: FacebookData for [5], BuzzFeedData and PolitiFactData for [19]. For the proposed methods, we set the threshold parameters to $\lambda_L = \lambda_H = 3$, so we consider the so-called LR-CB-3 and HC-CB-3.

For the methods presented in [5], namely LR and HC, we have partly shown the comparison in the previous section, since we integrate those methods in our approach. In addition, Fig. 3 shows the improvement in the accuracy of the proposed LR-CB-3 and HC-CB-3 w.r.t the corresponding original methods. Note that the experiment setup and the plot are exactly the same as in the original paper, except that the posts having 0 likes were discarded in [5]. So, the accuracy of the social-based methods is here slightly lower than in the original figure. The comparison in terms of the other performance metrics, i.e. precision, recall, and F1 score, are reported in Table II for different values of the training set size. All the results are obtained using a shuffle split cross validation with 50 iterations on the FacebookData dataset. As expected, since in the original dataset there are many Facebook posts with few likes, the proposed methods outperform the corresponding original methods for each training set size in each metric, except the recall, where the performance are slightly worse. Considering the average among the training set size, the accuracies are increased by 4.8% for LR-CB-3 w.r.t. LR and by 3.4% for HC-CB-3 w.r.t. HC.

Last, we show the comparison with the work by [19], in which the TriFN method is presented. TriFN combines content and social signals by modeling the tri-relationship of publisher-news and news-user. This method is compared in the original paper with other methods in the literature, in particular the so-called:

- RST [26] – which is content-based only and leverages a SVM classifier,
- LIWC [27] – which is content-based only and extracts psycholinguistic categories,
- Castillo [21] – which is social-based only and considers users' profiles information and friendship network,
- RST+Castillo – which combines content and social signals,

TABLE I. ACCURACY FOR COMBINED METHODS VARYING THRESHOLD $\lambda$ AND TRAINING SET SIZE (FACEBOOKDATA– SHUFFLE SPLIT CROSS VALIDATION, AVG. OF 50 ITERATIONS)

| Method | $\lambda$ (likes) Train. size | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| HC-CB-$\lambda$ | 0.0010 | 0.8000 | 0.8117 | 0.8197 | 0.8248 | 0.8269 | **0.8271** | 0.8267 | 0.8256 | 0.8239 | 0.8223 |
| | 0.0025 | 0.8755 | 0.8932 | 0.9051 | 0.9128 | 0.9156 | **0.9163** | 0.9159 | 0.9145 | 0.9128 | 0.9114 |
| | 0.0050 | 0.9175 | 0.9365 | 0.9475 | 0.9532 | **0.9536** | 0.9527 | 0.9506 | 0.9481 | 0.9452 | 0.9433 |
| | 0.0100 | 0.9383 | 0.9590 | 0.9690 | **0.9729** | 0.9718 | 0.9699 | 0.9671 | 0.9643 | 0.9613 | 0.9594 |
| | 0.0250 | 0.9550 | 0.9770 | 0.9843 | **0.9850** | 0.9825 | 0.9803 | 0.9774 | 0.9746 | 0.9720 | 0.9704 |
| | 0.0500 | 0.9613 | 0.9838 | **0.9896** | 0.9890 | 0.9863 | 0.9840 | 0.9814 | 0.9791 | 0.9770 | 0.9756 |
| | 0.1000 | 0.9656 | 0.9885 | **0.9927** | 0.9914 | 0.9888 | 0.9867 | 0.9844 | 0.9825 | 0.9808 | 0.9797 |
| LR-CB-$\lambda$ | 0.0010 | 0.6242 | 0.6350 | 0.6461 | 0.6552 | 0.6630 | 0.6697 | 0.6761 | 0.6811 | 0.6861 | **0.6901** |
| | 0.0025 | 0.7251 | 0.7420 | 0.7578 | 0.7710 | 0.7810 | 0.7897 | 0.7971 | 0.8029 | 0.8086 | **0.8132** |
| | 0.0050 | 0.8175 | 0.8357 | 0.8516 | 0.8637 | 0.8714 | 0.8775 | 0.8817 | 0.8846 | 0.8871 | **0.8893** |
| | 0.0100 | 0.8837 | 0.9037 | 0.9188 | 0.9289 | 0.9339 | 0.9372 | 0.9386 | 0.9390 | 0.9391 | **0.9394** |
| | 0.0250 | 0.9267 | 0.9485 | 0.9610 | 0.9678 | 0.9696 | 0.9705 | **0.9698** | 0.9685 | 0.9671 | 0.9662 |
| | 0.0500 | 0.9400 | 0.9625 | 0.9736 | 0.9787 | **0.9793** | 0.9792 | 0.9779 | 0.9764 | 0.9749 | 0.9738 |
| | 0.1000 | 0.9504 | 0.9732 | 0.9826 | **0.9857** | 0.9852 | 0.9845 | 0.9829 | 0.9813 | 0.9798 | 0.9789 |

TABLE II. PERFORMANCE OF VARYING TRAINING SET SIZE (FACEBOOKDATA – SHUFFLE SPLIT CROSS VALIDATION WITH 50 ITERATIONS)

| Train. size | Metric | CB | LR | LR-CB-3 | HC | HC-CB-3 |
|---|---|---|---|---|---|---|
| 0.001 | accuracy | 0.671 ± 0.073 | 0.613 ± 0.100 | 0.646 ± 0.105 | 0.790 ± 0.063 | **0.817 ± 0.066** |
| | precision | 0.778 ± 0.110 | 0.822 ± 0.182 | **0.826 ± 0.152** | 0.741 ± 0.060 | 0.784 ± 0.072 |
| | recall | 0.661 ± 0.232 | 0.601 ± 0.408 | 0.632 ± 0.360 | **0.987 ± 0.019** | 0.959 ± 0.035 |
| | f1 | 0.673 ± 0.138 | 0.551 ± 0.262 | 0.598 ± 0.234 | 0.844 ± 0.036 | **0.852 ± 0.039** |
| 0.005 | accuracy | 0.835 ± 0.030 | 0.829 ± 0.058 | 0.876 ± 0.055 | 0.919 ± 0.013 | **0.956 ± 0.011** |
| | precision | 0.856 ± 0.038 | 0.858 ± 0.103 | 0.894 ± 0.073 | 0.878 ± 0.021 | **0.936 ± 0.019** |
| | recall | 0.859 ± 0.080 | 0.881 ± 0.185 | 0.908 ± 0.137 | **0.996 ± 0.010** | 0.988 ± 0.006 |
| | f1 | 0.854 ± 0.034 | 0.845 ± 0.080 | 0.886 ± 0.067 | 0.933 ± 0.010 | **0.961 ± 0.008** |
| 0.010 | accuracy | 0.873 ± 0.013 | 0.897 ± 0.027 | 0.940 ± 0.023 | 0.939 ± 0.006 | **0.973 ± 0.005** |
| | precision | 0.882 ± 0.023 | 0.877 ± 0.062 | 0.928 ± 0.035 | 0.904 ± 0.011 | **0.961 ± 0.010** |
| | recall | 0.898 ± 0.037 | 0.965 ± 0.081 | 0.973 ± 0.046 | **0.998 ± 0.005** | 0.990 ± 0.003 |
| | f1 | 0.889 ± 0.013 | 0.913 ± 0.027 | 0.947 ± 0.022 | 0.949 ± 0.005 | **0.975 ± 0.004** |
| 0.050 | accuracy | 0.923 ± 0.004 | 0.941 ± 0.008 | 0.979 ± 0.004 | 0.962 ± 0.001 | **0.989 ± 0.001** |
| | precision | 0.917 ± 0.008 | 0.914 ± 0.028 | 0.970 ± 0.009 | 0.937 ± 0.002 | **0.985 ± 0.002** |
| | recall | 0.950 ± 0.009 | 0.993 ± 0.024 | 0.993 ± 0.004 | **1.000 ± 0.000** | 0.994 ± 0.001 |
| | f1 | 0.933 ± 0.003 | 0.951 ± 0.006 | 0.981 ± 0.004 | 0.967 ± 0.001 | **0.989 ± 0.001** |
| 0.100 | accuracy | 0.937 ± 0.002 | 0.950 ± 0.005 | 0.985 ± 0.002 | 0.966 ± 0.001 | **0.991 ± 0.001** |
| | precision | 0.930 ± 0.005 | 0.920 ± 0.008 | 0.978 ± 0.004 | 0.943 ± 0.001 | **0.988 ± 0.001** |
| | recall | 0.962 ± 0.005 | **1.000 ± 0.000** | 0.995 ± 0.001 | **1.000 ± 0.000** | 0.995 ± 0.001 |
| | f1 | 0.946 ± 0.002 | 0.958 ± 0.004 | 0.986 ± 0.002 | 0.971 ± 0.001 | **0.991 ± 0.001** |

- LIWC+Castillo – which combines content and social signals.

These last two methods have been crafted by the authors of [19] combining the first three methods in the list.

Results in Table III are obtained with a repeated 5-fold cross validation with 10 iterations. We added our HC-CB-3 method in the last column using the same experiment setup as in the original paper. TriFN and HC-CB-3 methods have higher performance than the other methods in all the considered metrics and in both datasets. Among these two methods, the proposed HC-CB-3 performs slightly worse than the TriFN method on the BuzzFeedData dataset in terms of accuracy and F1 score, although HC-CB-3 performs better in terms of recall. In the PolitiFactData dataset, HC-CB-3 outperforms TriFN in all the considered metrics. Overall, considering the weighted average, with weights equal to the number of posts in each dataset, HC-CB-3 improves the accuracy of TriFN by 3.5% (0.872 vs. 0.903).

## IV. REAL-WORLD APPLICATION

Having demonstrated the validity of our combined method on three datasets, we decided to implement a real-world fake news detection application that uses the classifier trained on the FacebookData dataset. We updated the posts retrieving the content from Jan 1, 2017 to Oct 31, 2017, obtaining 11,461 posts (4,983 hoaxes). We decided to build a *chatbot*, because it is a form of user interface that is becoming more and more popular; in particular, we built a Facebook Messenger chatbot, i.e. a chatbot that can be queried using Facebook Messenger.

TABLE III. PERFORMANCE EVALUATION COMPARISON WITH LITERATURE (10 TIMES REPEATED 5-FOLD CROSS VALIDATION)

| Dataset | Metric | RST [26] | LIWC [27] | Castillo [21] | RST+Cast. [19] | LIWC+Cast. [19] | TriFN [19] | **HC-CB-3** |
|---|---|---|---|---|---|---|---|---|
| BuzzFeed | accuracy | .610 ± .023 | .655 ± .075 | .747 ± .061 | .758 ± .030 | .791 ± .036 | **.864 ± .026** | .856 ± .052 |
| | precision | .602 ± .066 | .683 ± .065 | .735 ± .080 | .795 ± .060 | .825 ± .061 | **.849 ± .040** | .791 ± .076 |
| | recall | .561 ± .057 | .628 ± .021 | .783 ± .048 | .784 ± .074 | .834 ± .094 | .893 ± .013 | **.966 ± .045** |
| | f1 | .555 ± .057 | .623 ± .066 | .756 ± .051 | .789 ± .056 | .802 ± .023 | **.870 ± .019** | .867 ± .050 |
| PolitiFact | accuracy | .571 ± .039 | .637 ± .021 | .779 ± .025 | .812 ± .026 | .821 ± .052 | .878 ± .020 | **.938 ± .029** |
| | precision | .595 ± .032 | .621 ± .025 | .777 ± .051 | .823 ± .040 | .856 ± .071 | .867 ± .034 | **.899 ± .057** |
| | recall | .533 ± .031 | .667 ± .091 | .791 ± .026 | .792 ± .026 | .767 ± .120 | .893 ± .023 | **.948 ± .046** |
| | f1 | .544 ± .042 | .615 ± .044 | .783 ± .015 | .793 ± .032 | .813 ± .070 | .880 ± .017 | **.921 ± .041** |

Being Facebook the largest SNS in term of active users, it also constitutes a typical channel for the spreading of misinformation: providing a Facebook Messenger chatbot is therefore a very convenient solution since users can query the bot to check the reliability of a Facebook post while reading it, without having to leave the platform and using an interface they are already familiar with.

Creating a Facebook Messenger chatbot requires the creation of a Facebook page and a Facebook app; then, through a mechanism based on Webhooks, it is possible to call a custom URL that receives in input the text submitted by a user to the chatbot via Messenger, and produces as output the answer the chatbot should give to the user. The custom URL, in our case, is the endpoint of a REST API we developed using Python and the Flask (see http://flask.pocoo.org/) framework. Such API is the interface of an application that handles both the messaging with the users and the actual classification of the post, that is processed on-line.

### A. How the chatbot works

The chatbot receives in input a string, checks if the string is a valid Facebook post URL (we support several Facebook post URL syntaxes) and then retrieves, through the Facebook Graph API (see https://developers.facebook.com/docs/graph-api/reference/v2.7/object/likes), the set of users $UL$ who liked the post $p$. In particular, for each user $u \in UL$, we get the Facebook ID of the user.

Note that, due to privacy reasons, Facebook allows to retrieve the set $UL$ only for posts published by public Facebook Pages. It is not possible to retrieve the set $UL$ for posts published on personal profiles, even if the audience of the post is set to "public".

The online classification of the post works as follows. On the one hand, If the cardinality of $UL$ is $\geq \lambda$ (i.e. enough users liked the posts, and thus the social-based methods are supposed to work well) we classify the post applying the (previously fitted) boolean crowdsourcing classifier model to the retrieved set of user IDs. It is important to highlight that we consider only the IDs of the users that previously appeared in our fitted model, i.e. users who liked at least one of the posts in the FacebookData dataset (the like of an "unknown" user, in fact, would not provide any valuable information to the classifier). On the other hand, If the cardinality of $UL$ is $< \lambda$ (i.e. only a few users liked the post, and thus we probably do not have enough social signals) we apply the content-based method as explained in Section II-C: we produced a text corpus joining the actual text content of the post (retrieved using the Facebook Graph APIs) and, if the post shared a link, the title and text preview of the link (as provided by the Facebook Graph APIs) together with the actual content of the shared webpage. The corpus is then stemmed and then the post is classified applying the (previously fitted) content-based classifier model to the corpus.

### B. Results

We then tested the chatbot with a completely independent set of news, whose content was not part of the three previously mentioned datasets. It is important to highlight that the chatbot classifier, therefore, was not trained on this (fourth) set of news.

This real-world dataset is composed by 230 Facebook posts, sharing 180 fake and 50 real news articles. The 180 fake news items were collected by searching posts of Facebook pages which shared a list of verified hoaxes; the list was provided to us by an independent Italian fact-checker and debunker (http://www.bufale.net/). The 50 real news items were collected by taking the most recent posts on the Facebook page of the Italian news press agency – ANSA, http://www.ansa.it/. Collected on January 3rd, 2018., and discarding the posts which did not contain actual news.

The results are reported in Table IV. The HC-CB-4 method, combining the harmonic boolean label crowdsourcing (HC) and the content-based (CB) methods for $\lambda = 4$ (which is the optimal value of the threshold in this case), offers higher performances than HC and CB alone, reaching an accuracy of 81.7%. Additionally, the results obtained on this fourth - independent - set of news provides a strong evidence that the model is not overfitted.

TABLE IV. REAL-WORLD APPLICATION RESULTS (HC-CB-4 METHOD)

| **Metric** | **[%]** |
|---|---|
| Accuracy | 81.7 |
| Precision | 91.0 |
| Recall | 85.0 |
| F1 | 87.9 |

## V. Conclusions and future work

We presented a novel automatic fake news detection method that combines social and content signals. We built on the work by [5], combining their social-based method (that uses, as the only source of information, the ID of the users who socially interacted with a news item) with a content-based method (that analyzes the actual text of the news item).

The results confirm our hypothesis: although social-based methods offer good performances, these performances get worse for news items that collect only few social interactions, and therefore the combination with content-based approaches can increase the overall detection accuracy. In this work, we combined content-based and social-based methods based on a threshold rule which, despite its simplicity, is able to capture well the different contributions of these two approaches and to outperform other - more sophisticated - methods presented in the literature.

Our HC-CB-3 method offers higher accuracies than [5] and [19], using their respective datasets for the comparison; this is the first - methodological - contribution of this work, which also suggests that our method can work across different SNSs (namely Facebook and Twitter). The addition of a content-based component not only allows to improve the classification for pieces of news that did not spread massively (i.e. news that collected limited social interactions), but it also makes easier the task of early detection of fake news (*cold-start* situation), that, in turns, can limit the spread of fake news as a whole.

As an additional contribution, we implemented our method within a chatbot that works in the Facebook Messenger environment. This chatbot accepts Facebook post URLs in input from users and answers providing a classification for a given post, i.e. real or fake. We trained the classifier used by the chatbot with an updated version of the dataset used in [5] and then tested the chatbot using an independent and fourth set of fake and real news, obtaining a detection accuracy of 81.7%.

Our approach currently works well in distinguishing a fake news from a real news; our future work includes focusing on cases harder to classify, in particular when the content of the news is true but the title or the comment to the content is misleading or clickbait.
On the chatbot implementation side, since the classifier has been trained on the dataset provided in [5], the text corpus for the train was in Italian and most of the users interacting with the posts were also from the same country; this makes both the content-based component and the social-based component of the chatbot currently suitable only for classifying Italian news. Thus, we plan to train the bot classifier with ground truth in other languages in order to extend it to other countries and language communities.

## Acknowledgments

The authors would like to thank Nilva Scarton for the precious support and the time dedicated to the review of the work.

## References

[1] A. Bessi, M. Coletto, G. A. Davidescu, A. Scala, G. Caldarelli, and W. Quattrociocchi, "Science vs Conspiracy: Collective Narratives in the Age of Misinformation," *PLOS ONE*, vol. 10, no. 2, p. e0118093, Feb. 2015.

[2] H. Allcott and M. Gentzkow, "Social Media and Fake News in the 2016 Election," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–236, May 2017.

[3] A. Gupta, H. Lamba, and P. Kumaraguru, "$1.00 per RT #BostonMarathon #PrayForBoston: Analyzing fake content on Twitter," in *2013 APWG eCrime Researchers Summit*, Sep. 2013, pp. 1–12.

[4] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, 2017.

[5] E. Tacchini, G. Ballarin, M. L. Della Vedova, S. Moret, and L. de Alfaro, "Some Like it Hoax: Automated Fake News Detection in Social Networks," in *Proceedings of the Second Workshop on Data Science for Social Good*, vol. 1960. Skopje, Macedonia: CEUR-WS, 2017.

[6] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini, "Computational Fact Checking from Knowledge Networks," *PLOS ONE*, vol. 10, no. 6, p. e0128193, Jun. 2015.

[7] M. Del Vicario, A. Bessi, F. Zollo, F. Petroni, A. Scala, G. Caldarelli, H. E. Stanley, and W. Quattrociocchi, "The spreading of misinformation online," *Proceedings of the National Academy of Sciences*, vol. 113, no. 3, pp. 554–559, Jan. 2016.

[8] J. C. Hernandez, C. J. Hernandez, J. M. Sierra, and A. Ribagorda, "A first step towards automatic hoax detection," in *Proceedings. 36th Annual 2002 International Carnahan Conference on Security Technology*, 2002, pp. 102–114.

[9] Y. Wu, P. K. Agarwal, C. Li, J. Yang, and C. Yu, "Toward computational fact-checking," *Proceedings of the VLDB Endowment*, vol. 7, no. 7, pp. 589–600, 2014.

[10] N. J. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015.

[11] A. Magdy and N. Wanas, "Web-based statistical fact checking of textual documents," in *Proceedings of the 2nd International Workshop on Search and Mining User-Generated Contents*. ACM, 2010, pp. 103–110.

[12] B. Shi and T. Weninger, "Fact checking in heterogeneous information networks," in *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 101–102.

[13] M. Vuković, K. Pripužić, and H. Belani, "An Intelligent Automatic Hoax Detection System," in *Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, Berlin, Heidelberg, Sep. 2009, pp. 318–325.

[14] M. Sharifi, E. Fink, and J. G. Carbonell, "Detection of Internet scam using logistic regression," in *2011 IEEE International Conference on Systems, Man, and Cybernetics*, Oct. 2011, pp. 2168–2172.

[15] S. Badaskar, S. Agarwal, and S. Arora, "Identifying Real or Fake Articles: Towards better Language Modeling." in *IJCNLP*, 2008, pp. 817–822.

[16] W. Y. Wang, ""Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*. Vancouver, BC, Canada: ACL, Jul. 2017.

[17] B. Riedel, I. Augenstein, G. P. Spithourakis, and S. Riedel, "A simple but tough-to-beat baseline for the Fake News Challenge stance detection task," *arXiv:1707.03264 [cs]*, Jul. 2017.

[18] H. Ahmed, I. Traore, and S. Saad, "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques," in *Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, ser. Lecture Notes in Computer Science. Springer, Cham, Oct. 2017, pp. 127–138.

[19] K. Shu, S. Wang, and H. Liu, "Exploiting Tri-Relationship for Fake News Detection," *arXiv:1712.07709 [cs]*, Dec. 2017.

[20] M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein, "A Stylometric Inquiry into Hyperpartisan and Fake News," *arXiv preprint arXiv:1702.05638*, 2017.

[21] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th International Conference on World Wide Web*. ACM, 2011, pp. 675–684.

[22] ——, "Predicting information credibility in time-sensitive social media," *Internet Research*, vol. 23, no. 5, pp. 560–588, 2013.

[23] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier, "Tweetcred: Real-time credibility assessment of content on twitter," in *International Conference on Social Informatics*. Springer, 2014, pp. 228–243.

[24] Z. Jin, J. Cao, Y. Zhang, and J. Luo, "News Verification by Exploiting Conflicting Social Viewpoints in Microblogs." in *AAAI*, 2016, pp. 2972–2978.

[25] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, *Recommender Systems Handbook*. Springer, 2015.

[26] V. L. Rubin, N. J. Conroy, and Yimin Chen, "Towards News Verification: Deception Detection Methods for News Discourse," 2015.

[27] J. W. Pennebaker, R. L. Boyd, K. Jordan, and K. Blackburn, "The Development and Psychometric Properties of LIWC2015," 2015.