An Innovative Framework for Supporting Social Network Polluting-content Detection and Analysis

Alfredo Cuzzocrea[1], Fabio Martinelli[2] and Francesco Mercaldo[2]
[1]*University of Trieste, Trieste, Italy*
[2]*IIT-CNR, Pisa, Italy*

Keywords: Social Networks, Social Network Security, Social Network Analysis, Machine Learning, Word Embedding, Text Classification.

Abstract: In last years we are witnessing a growing interest in tools for analyzing big data gathered from social networks in order to find common opinions. In this context, content polluters on social networks make the opinion mining process difficult to browse valuable contents. In this paper we propose a method aimed to discriminate between pollute and real information from a semantic point of view. We exploit a combination of word embedding and deep learning techniques to categorize semantic similarities between (pollute and real) linguistic sentences. We experiment the proposed method on a data set of real-world sentences obtaining interesting results in terms of precision and recall.

1 INTRODUCTION

The Reuters Institute recently released the 2017 Digital News Report, analyzing surveys from 70,000 people across 36 countries and providing a comprehensive comparative analysis of modern news consumption (Dormann et al., 2018).

The real impact of the growing interest in fake news has been the realization that the public might not be well-equipped to separate quality information from false information (Peters et al., 2018). In fact, a majority of Americans are confident that they can spot fake news. When Buzzfeed (BuzzFeed, 2018) surveyed American high scholars (the BuzzFeed Quiz and Skills!, 2018), they too were confident they could spot, and ignore, fake news online: the reality, however, is that it might be more difficult than people think.

The report reveals several important media trends, including rising polarization in the United States. It is interesting to observe that while 51% of left-leaning Americans trust the news, only 20% of conservatives say the same: this is symptomatic that right-leaning Americans are far more likely to say they avoid the news because they do not rely on news to be true.

In order to generate pollute content, as fake news or spam on social network, usually social bot are employed (Wu and Liu, 2018; Wang et al., 2018).

A social bot is a software able to automatically generate messages (for instance post in social networks like Twitter of Facebook) or in general advocate certain ideas, support campaigns, and public relations either by acting as a ”follower” or even as a fake account that gathers followers itself (socialbot, 2018).

Social bots demonstrated to have played a significant role in the 2016 United States presidential election (Friends and Profit, 2018; Shao et al., 2017), and their history appears to go back at least to the 2010 United States midterm elections (Ratkiewicz et al., 2011).

It is estimated that 9-15% of active Twitter accounts may be social bots (Varol et al., 2017) and that 15% of the total Twitter population active in the US Presidential election discussion were bots. At least 400,000 thousand bots were responsible for about 3.8 million tweets, roughly 19% of the total volume (Ferrara et al., 2016).

Social bots, besides being able to produce messages autonomously, also share many traits with spam-bots with respect to their tendency to infiltrate large user groups.

Automated accounts or bots play a pivotal role in the spread of fake news or misinformation. The spread of such misinformation can have polarizing effects on a society and prevent any consensus from evolving, leading to a grid-lock, and bringing down welfare levels in the society (Azzimonti and Fernan-

Cuzzocrea, A., Martinelli, F. and Mercaldo, F.
An Innovative Framework for Supporting Social Network Polluting-content Detection and Analysis.
DOI: 10.5220/0007737403030311
In *Proceedings of the 21st International Conference on Enterprise Information Systems (ICEIS 2019)*, pages 303-311
ISBN: 978-989-758-372-8

des, 2018): researchers show that even if only a tenth of users in a social network fall for fake news, it can lead to significant misinformation and polarization because of network effects, i.e., the reliance of other users on them for news and views.

Researchers also demonstrate that a society that is successful in eliminating a source of fake news promoting one extreme of the political spectrum may end up worse off due to the unintended consequences of making the other extreme relatively more powerful. This, in the end, would generate greater misinformation and lower welfare, despite effectively reducing polarization (LiveMint, 2018). On the other hand, these topics are extremely important in the emerging big data setting (e.g., (Li et al., 2015; Braun et al., 2014; Yang et al., 2014)), as also highlighted by recent studies (e.g., (Shafahi et al., 2016)).

Starting from these considerations in this paper we propose a method to discriminate between real and pollute contents. We exploit word embedding, a technique able to encode text into numerical vectors and machine learning techniques, aimed to build a model able to label a sentence under analysis as pollute or real.

The remaining part of this paper is organized as follows. In Section 2, we focus the attention of state-of-the-art proposals that are relevant to our research. Section 3 reports a case study that better illustrates how our proposed framework works in practice. In Section 4, we provide the main social network polluting-content detection techniques embedded in our framework. Section 5 shows the experimental results of our proposed techniques on a real-life Twitter data sets. Finally, in Section 6, we draw conclusions and future work of our research.

2 RELATED WORK

In this section we review state-of-the-art literature focused about the polluting content detection in social networks.

(Smadi et al., 2018) investigate a set of features to detect unsolicited bulk emails and select the set of best features to detect spam emails. The feature representation and feature selection techniques have also been used to enhance the SVM-based spam detector (Diale et al., 2016).

In addition to detect spam emails, feature selection techniques play an important role in spam detection in online social networks. An optimal Random Forest-based spam detection model has been proposed in (Lee et al., 2010b), which optimizes two parameters of RF and determines importance of variable to select the most important features and to eliminate irrelevant ones.

(Sohrabi and Karimi, 2018) propose a method aimed to filter spam message on Facebook social network, as one of the largest and most popular social networks, exploiting a Particle Swarm Optimization feature selection method and combining supervised and unsupervised classification techniques.

(Liu et al., 2018) investigate crowdretweeting spam in Sina Weibo, the counterpart of Twitter in China. They find that although spammers are likely to connect more closely than legitimate users, the underlying social connections of crowdretweeting campaigns are different from those of other existing spam campaigns because of the unique features of retweets that are spread in a cascade; from these considerations they consider several algorithms in order to find more suspicious spamming account obtaining, in the best scenario, a precision equal to 0.915 and a recall equal to 0.683.

(Li and Shen, 2011) propose a social network spam filter (SOAP) aimed to exploit the social relationship among email correspondents to detect the spam adaptively and automatically. SOAP integrates three components into the basic Bayesian filter: social closeness-based spam filtering, social interest-based spam filtering, and adaptive trust management. They evaluate performance of the proposed method based on the trace data from Facebook social network: the main outcome is that SOAP improves the performance of Bayesian networks in term of spam detection accuracy and training time.

Machine-learning techniques were also diffused to build model able to predict whether a sentence under analysis is spam or legal. For instance, (Bilge et al., 2009) show that after an attacker has entered the network of trust of a victim, the victim will likely click on any link contained in the messages posted, irrespective of whether she knows the attacker in real life or not. Another interesting finding by researchers (Jagatic et al., 2007) is that phishing attempts are more likely to succeed if the attacker uses stolen information from victims' friends in social networks to craft their phishing emails. For example, phishing emails from shoppybag were often sent from a user's friendlist and hence a user is often tricked into believing that such emails come from trusted friends and hence willingly provides login information of his/her personal email account.

In (Yardi et al., 2009), the authors created a popular hashtag on Twitter and observed how spammers started to use it in their messages. They discuss some features that might distinguish spammers from legitimate users e.g. node degree and frequency of messages. However, merely using simple features like node degree and frequency of messages may not be enough since there are some young Twitter users or TV anchors that post many messages. A larger spam study is reported in (Stringhini et al., 2010).

(Stringhini et al., 2010) generate honey profiles to lure spammers into interacting with them. They create 300 profiles each on popular social networking sites like Facebook, Twitter and MySpace. Their 900 honey profiles attract 4250 friends request (mostly on Facebook) but 361 out of 397 friend requests on Twitter were from spammers. They later suggest using features like the percentage of tweets with URLs, message similarity, total messages sent, number of friends for spam detection. Their detection scheme based on the Random Forest classifier can produce a false positive rate of 2.5% and a false negative rate of 3% on their Twitter data set.

The idea behind the POISED framework (Nilizadeh et al., 2017) is that there are differences in propagation between benign and malicious messages on social networks, and from this assumption authors designed POISED to identify spam and other unwanted content. They defined three entropy-based metrics: completeness, homogeneity, and V-measure of topics detected in communities. Authors experiments the proposed framework on 1.3M tweets collected from 64K users, demonstrating that their approach is able to reach 91% precision and 93% recall in malicious messages detection.

(Zhang et al., 2012) propose a three steps framework to detect spam on Twitter: firstly linking accounts who post URLs for similar purposes, secondly extracting candidate campaigns which may exist for spam or promoting purpose and finally distinguishing their intents. Their method is able to obtain 0.903 of precision and 0.849 of recall.

(Noll et al., 2009) offer a graph-based algorithm to rank experts based on the user's ability of identifying interesting or useful resources before others in a collaborative tagging system. (Yamaguchi et al., 2010) address the problem of ranking authoritative users based on the actual information flow in Twitter. (Das Sarma et al., 2010) adopt comparison-based scoring mechanism to achieve approximate rankings with the feedback comparison results from users.

(Lee et al., 2010a) deploy honeypots to harvest deceptive spam profiles from social networking communities and train classifiers to detect existing and new coming spammers. (Bogers and Van den Bosch, 2009) detect malicious users on the basis of a similarity function that adopts language modeling.

(Benevenuto et al., 2010) approach the problem of detecting trending-topic spammers users who include unrelated URLs with trending words in tweets, in order to make the tweets appear in the results of searches or meme-tracking tools. They manually label a collection of users as spammers or non-spammers and identified properties that are able to distinguish between the two classes of users through a machine learning approach.

3 CASE STUDY: SOCIAL NETWORK POLLUTING-CONTENT SPREAD

In this section we discuss a real-world scenario aimed to better understand the context of the proposed method to detect polluting contents in social networks.

Figure 1 shows a real-world scenario: in the left side there are the malicious users (social network bot, fake news, aggressive advertisement) and in the right one the legitimate users (i.e., Bob and Alice). In the center of the scenario there are social networks, and all users (malicious and legitimate) are able to publish contents.

Figure 1: The scenario for the proposed case study: in the left side there are the malicious users (social network bot, fake news, aggressive advertisement) and in the right one the legitimate users (i.e., Bob and Alice).

As depicted in Figure 1 in the real-world malicious users and/or social network bot start a campaign of contents. Basically the type of polluting contents that are diffused are fake news and aggressive advertisement.

This is the typical flow in which polluting contents are diffused on social networks:

- a social network start to publish messages about a specified topic (usually aggressive advertisement) or a malicious user publish a fake news;
- once the polluting content is published for the first time, different actions can be performed at this

point: for instance the polluting contents can be shared using the '@' character and the name of the user to cite in order to spread the contents. Whether the user cited is an important company, all the follower of the company will be able to visualize the polluting contents;

- Alice is an user of social networks, she visualizes everyday the social profile of his favorite perfume company. Today, the company published several contents related to an interesting offer for a men perfume: "Save up to 75% with our men fragrance offers on branded perfumes!".
- Alice shares this content on the Bob profile (from the Alice point of view the content is from the perfume company and it is considered legitimate), in this way he will receive a notification from Alice about this offer (social networks offer a one-click option to share content in an easy way): from the Bob point of view the content is from Alice and it is considered legitimate;
- in this way the polluting content is shared between users that are considered legitimate and not malicious and the polluting content will propagate itself using this mechanism.

This mechanism is based on the trust of users in the profile of a well-known company and in trust between users: this is the reason why polluting content are able to spread their self on social networks.

In a typical scenario like the one we discussed, the proposed method will be able to intercept the polluting content and to block it in order to avoid its diffusion.

4 AN EFFECTIVE AND EFFICIENT SOCIAL NETWORK POLLUTING-CONTENT DETECTION TECHNIQUE

The method we propose to distinguish between polluting and real sentences is described in following section.

In Figure 2 the flowchart of the proposed framework is depicted.

The first step in the flowchart in Figure 2 accepts as input the word embeddings (*word embeddings* step) and the text sentences (*text sentences* step) to analyze (i.e., the data set): basically word embeddings are exploited to map to vector of real numbers words or phrases from the vocabulary. In order to use word embeddings, we need to load pre-

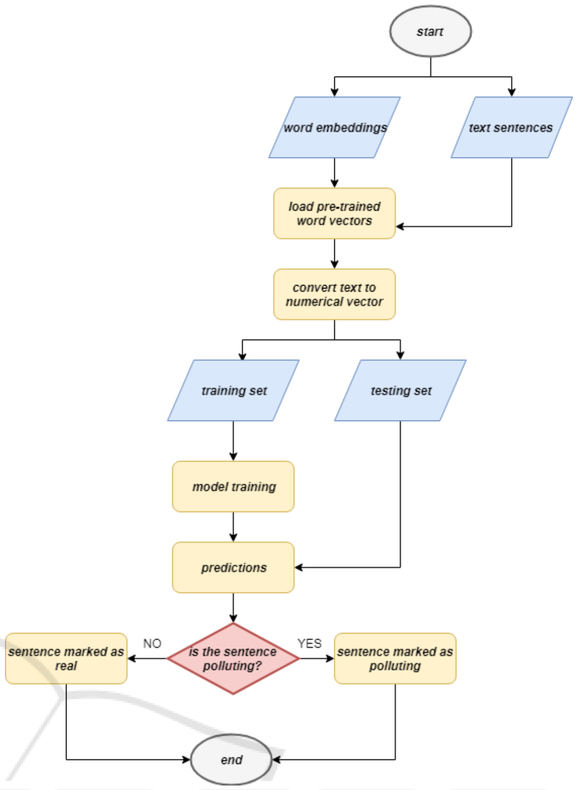

Figure 2: Flowchart of the proposed approach.

trained word embeddings (*load pre-trained word vectors* step). Once loaded the pre-trained word vectors, through the *convert text to numerical vector* step we are able to perform several computation with the data set text sentences, for instance we can get most similar words to some word or load a word into the vocabulary to see embeddings: these operations are possible via the numeric representation of the words.

Once obtained the word embeddings for text classification of sentences, we split the sentences data set in two sub data set: the training, containing the 80% of the sentences (*training set*) and the testing, containing the remaining 20% of the sentences (*testing set*).

Considering that the proposed method applies supervised machine-learning, the sentences belong to two classes (polluting and real), the labels for classes will be assigned later as 0 (i.e., polluting) or 1 (i.e., real).

In order to discriminate between polluting and real sentences we need to build the model (the *model training* step) with the sentences belonging to the training set. Once built the model we evaluate the model with the sentences belonging to the testing set (*predictions* step). Training and testing set contain both polluting and real sentences.

The sentence under analysis from the *predictions* step is labeled as real (*sentence marked as real* step) or polluting (*sentence marked as polluting* step).

Listing 1 shows the pseudo-code of the algorithm we developed.

It is basically divided into three main phases: the first one is related to obtain word embeddings, the second one to divide the sentence data set into training and testing set with the consequent label attachment (we consider supervisioned learning) and the last phase is related to model building and evaluation using the MLP algorithm.

We designed an experiment in order to evaluate the effectiveness of the word embedding feature vector we propose.

More specifically, the experiment is aimed at verifying whether the word embedding feature set is able to discriminate between real and polluting sentences.

The classification is carried out by using the MLP classifier with different configurations.

The evaluation consists of a classification analysis aimed at assessing whether the features are able to correctly classify between real and polluting sentences.

We adopt the supervised learning approach, considering that the features evaluated in this work are labeled.

The supervised learning approach is composed of two different steps:

1. **Learning Step:** starting from the labeled data set (i.e., where each feature is related to a class. In our case, the class is represented by the real or polluting sentences), we filter the data in order to obtain a feature vector gathered by word embedding. The feature vectors, belonging to all the sentences involved in the experiment with the associated labels (i.e., real or polluting), represent the input for the machine learning algorithm that is able to build a model from the analyzed data. The output of this step is the model obtained by the labeled data set.
2. **Prediction Step:** the output of this step is the classification of a feature vector belonging to the real or polluting sentence. Using the model built in the previous phase, we input this model using a feature vector without the label: the classifier will output with their label prediction (i.e., real or polluting).

5 EXPERIMENTAL ASSESSMENT AND ANALYSIS

In this section we describe the real-world data set used in this paper and the results of the experiment.

5.1 The Data Set

The evaluated data set contains 2,353,473 polluting tweets and 3,259,693 legitimate tweets (Lee et al., 2011) collected from December 30, 2009 to August 2, 2010 on the Twitter social network.

As described in (Lee et al., 2011) the data set was obtained through 60 social honeypot accounts on Twitter whose purpose is to pose as Twitter users, and report back what accounts follow or otherwise interact with them.

The Twitter-based social honeypots can post four types of tweets: (1) a normal textual tweet; (2) an "@" reply to one of the other social honeypots; (3) a tweet containing a link; (4) a tweet containing one of current top 10 trending topics of Twitter, which are popular n-grams.

To seed the pool of tweets that the social honeypot accounts would post we crawled the Twitter public timeline and collected 120,000 sample tweets (30,000 for each of our four types). The social honeypot accounts are intentionally designed to avoid interfering with the activities of legitimate users. They only send @ reply messages to each other, and they will only follow other social honeypot accounts.

Once a Twitter user makes contact with one of the social honeypots via following or messaging the honeypot, the information is passed to the Observation system. The Observation system keeps track of all the users discovered by the system. Initially, all information about each user's account and all the user's past tweets are collected. Every hour the Observation system checks each user's status to determine if more tweets have been posted, the number of other accounts that the user is following, the number of other Twitter accounts following the user and if the account is still active.

The system ran from December 30, 2009 to August 2, 2010. During that time the social honeypots tempted 36,043 Twitter users, 5,773 (24%) of which followed more than one honeypot.

5.2 The Evaluation

In order to evaluate the proposed method we consider four metrics: Precision, Recall, F-Measure and RocArea.

Listing 1: Algorithm for Polluting Sentence Detection.

```
#got word embeddings
def sent_vectorizer(sent, model):
    sent_vec =[]
    numw = 0
    for w in sent:
        try:
            if numw == 0:
                sent_vec = model[w]
            else:
                sent_vec = np.add(sent_vec, model[w])
            numw+=1
        except:
            pass

    return np.asarray(sent_vec) / numw

V=[]
for sentence in sentences:
    V.append(sent_vectorizer(sentence, model))

#divide data intro training and testing set
X_train = V[0:6]
X_test = V[6:9]

#attach class labels
Y_train = [0, 0, 1, 1, 0, 1]
Y_test =  [0, 1, 1]

#load data to MLP classifier to perform text classification
classifier = MLPClassifier(alpha=0.7, max_iter=400)
classifier.fit(X_train, Y_train)

df_results = pd.DataFrame(data=np.zeros(shape=(1,3)),
columns = ['classifier', 'train_score', 'test_score'] )
train_score = classifier.score(X_train, Y_train)
test_score = classifier.score(X_test, Y_test)
```

The precision has been computed as the proportion of the examples that truly belong to class X among all those which were assigned to the class. It is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved:

$$Precision = \frac{tp}{tp+fp}$$

where *tp* indicates the number of true positives and *fp* indicates the number of false positives.

The recall has been computed as the proportion of examples that were assigned to class X, among all the examples that truly belong to the class, i.e., how much part of the class was captured. It is the ratio of the number of relevant records retrieved to the total number of relevant records:

$$Recall = \frac{tp}{tp+fn}$$

where *tp* indicates the number of true positives and *fn* indicates the number of false negatives.

The F-Measure is a measure of a test's accuracy. This score can be interpreted as a weighted average of the precision and recall:

$$F\text{-}Measure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

The RocArea is defined as the probability that a positive instance randomly chosen is classified above a negative randomly chosen.

The classification analysis consisted of building deep learning classifiers with the aim to evaluate the feature vector accuracy to distinguish between pollut-

Table 1: Classification results: Precision, Recall, F-Measure, RocArea computed with the MLP classification algorithms. We considered two different deep learning networks, the first one with 0 hidden states (i.e., MLP 0), while the second one with 1 hidden state (i.e., MLP 1).

Algorithm	Precision	Recall	F-Measure	RocArea
MLP 0	0.753	0.400	0.563	0.775
MLP 1	0.791	0.756	0.773	0.791

ing and real sentences.

For training the classifier, we defined T as a set of labeled messages (M, l), where each M is associated to a label $l \in \{IM, NM\}$. For each M we built a feature vector F $\in R_y$, where y is the number of the features used in training phase.

For the learning phase, we consider a k-fold cross-validation: the data set is randomly partitioned into k subsets. A single subset is retained as the validation data set for testing the model, while the remaining $k-1$ subsets of the original data set are used as training data. We repeated this process for $k = 10$ times; each one of the k subsets has been used once as the validation data set. To obtain a single estimate, we computed the average of the k results from the folds.

We evaluated the effectiveness of the classification method with the following procedure:

1. build a training set $T \subset D$;
2. build a testing set $T' = D \div$T;
3. run the training phase on T;
4. apply the learned classifier to each element of T'.

Each classification was performed using 90% of the data set as training data set and 10% as testing data set employing the full feature set.

To build models able to classify the sentences we consider Multilayer Perceptron (MLP) (Pal and Mitra, 1992), a class of feedforward artificial neural network.

We consider two neural network MLP-based, whit a number of hidden layers equal to 0 and to 1.

Table 1 shows the results of the evaluation we performed.

From the results of the classification analysis, we observe that the MLP network with 1 hidden state (i.e., MLP 1 in Table 1) obtains better performance that the one with 0 hidden states (i.e., MLP 0 in Table 1).

We obtain a precision equal to 0.753 and a recall equal to 0.4 with the MLP network with 0 hidden states, while with regards to the MLP network with 1 hidden state the precision obtained is equal to 0.791 and the recall is equal to 0.756.

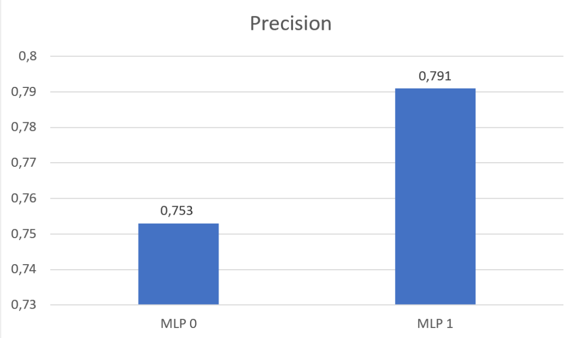

Figure 3: Bar chart related to precision metric.

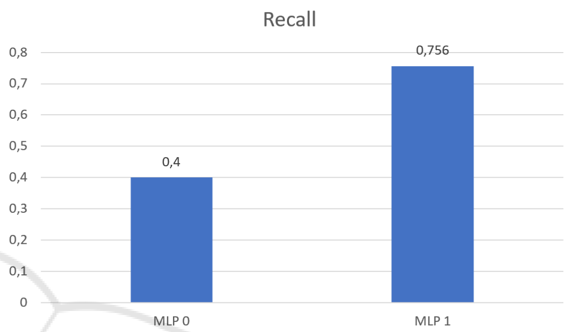

Figure 4: Bar chart related to recall metric.

The bar charts related to precision (in Figure 3) and recall (in Figure 4) metrics depicted show a comparison between the two algorithms.

6 CONCLUSIONS AND FUTURE WORK

A method aimed to discriminate between polluting and real sentences is proposed in this paper. We exploit word embeddings and deep learning techniques and we experiment the proposed solution using a real world data set gathered by the Twitter social network.

As future work, we plan to extend the proposed method by: (i) evaluating the effectiveness of more deep neural network, (ii) classifying the sentences by different type of polluting content (for instance fake news and spam category), and (iii) considering the interesting case represented by inter-social-networks, like in similar studies (e.g., (Cucchiarelli et al., 2012; Wu et al., 2013)).

ACKNOWLEDGEMENTS

This work was partially supported by the H2020 EU funded project *NeCS* [GA #675320], by the H2020 EU funded project *C3ISP* [GA #700294].

REFERENCES

Azzimonti, M. and Fernandes, M. (2018). Social media networks, fake news, and polarization. Technical report, National Bureau of Economic Research.

Benevenuto, F., Magno, G., Rodrigues, T., and Almeida, V. (2010). Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, volume 6, page 12.

Bilge, L., Strufe, T., Balzarotti, D., and Kirda, E. (2009). All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web*, pages 551–560. ACM.

Bogers, T. and Van den Bosch, A. (2009). Using language modeling for spam detection in social reference manager websites. In *Proceedings of the 9th Belgian-Dutch Information Retrieval Workshop (DIR 2009)*, pages 87–94.

Braun, P., Cameron, J. J., Cuzzocrea, A., Jiang, F., and Leung, C. K. (2014). Effectively and efficiently mining frequent patterns from dense graph streams on disk. In *KES*, volume 35 of *Procedia Computer Science*, pages 338–347. Elsevier.

BuzzFeed (2018). https://www.buzzfeed.com/. [Online; accessed 21-August-2018].

Cucchiarelli, A., D'Antonio, F., and Velardi, P. (2012). Semantically interconnected social networks. *Social Netw. Analys. Mining*, 2(1):69–95.

Das Sarma, A., Das Sarma, A., Gollapudi, S., and Panigrahy, R. (2010). Ranking mechanisms in twitter-like forums. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 21–30. ACM.

Diale, M., Walt, C. V. D., Celik, T., and Modupe, A. (2016). Feature selection and support vector machine hyper-parameter optimisation for spam detection. In *2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*, pages 1–7.

Dormann, C., Demerouti, E., and Bakker, A. (2018). A model of positive and negative learning: Learning demands and resources, learning engagement, critical thinking, and face news detection. In *Positive learning in the age of information (PLATO): A blessing or a curse*. Springer International.

Ferrara, E., Varol, O., Davis, C., Menczer, F., and Flammini, A. (2016). The rise of social bots. *Communications of the ACM*, 59(7):96–104.

Friends, S. M. B. O. P. and Profit, R. (2018). https://www.nytimes.com/2014/11/20/fashion/social-media-bots-offer-phony-friends-and-real-profit.html. [Online; accessed 21-August-2018].

Jagatic, T. N., Johnson, N. A., Jakobsson, M., and Menczer, F. (2007). Social phishing. *Communications of the ACM*, 50(10):94–100.

Lee, K., Caverlee, J., and Webb, S. (2010a). Uncovering social spammers: social honeypots+ machine learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 435–442. ACM.

Lee, K., Eoff, B. D., and Caverlee, J. (2011). Seven months with the devils: A long-term study of content polluters on twitter. In *ICWSM*, pages 185–192.

Lee, S. M., Kim, D. S., Kim, J. H., and Park, J. S. (2010b). Spam detection using feature selection and parameters optimization. In *2010 International Conference on Complex, Intelligent and Software Intensive Systems*, pages 883–888.

Li, K., Jiang, H., Yang, L. T., and Cuzzocrea, A., editors (2015). *Big Data - Algorithms, Analytics, and Applications*. Chapman and Hall/CRC.

Li, Z. and Shen, H. (2011). Soap: A social network aided personalized and effective spam filter to clean your e-mail box. In *INFOCOM, 2011 Proceedings IEEE*, pages 1835–1843. IEEE.

Liu, B., Ni, Z., Luo, J., Cao, J., Ni, X., Liu, B., and Fu, X. (2018). Analysis of and defense against crowd-retweeting based spam in social networks. *World Wide Web*, pages 1–23.

LiveMint (2018). Bots amplify the spread of fake news, and harm economies, new research shows. https://www.livemint.com/Politics/H6usHoKAZUTtUpM4ZLxHUK/Bots-amplify-the-spread-of-fake-news-and-harm-economies-ne.html [Online; accessed 21-August-2018].

Nilizadeh, S., Labrèche, F., Sedighian, A., Zand, A., Fernandez, J., Kruegel, C., Stringhini, G., and Vigna, G. (2017). Poised: Spotting twitter spam off the beaten paths. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1159–1174. ACM.

Noll, M. G., Au Yeung, C.-m., Gibbins, N., Meinel, C., and Shadbolt, N. (2009). Telling experts from spammers: expertise ranking in folksonomies. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 612–619. ACM.

Pal, S. K. and Mitra, S. (1992). Multilayer perceptron, fuzzy sets, classifiaction.

Peters, M. A., Rider, S., Hyvönen, M., and Besley, T. (2018). *Post-truth, fake news: Viral modernity & higher education*. Springer.

Ratkiewicz, J., Conover, M., Meiss, M. R., Gonçalves, B., Flammini, A., and Menczer, F. (2011). Detecting and tracking political abuse in social media. *ICWSM*, 11:297–304.

Shafahi, M., Kempers, L., and Afsarmanesh, H. (2016). Phishing through social bots on twitter. In *BigData*, pages 3703–3712. IEEE Computer Society.

Shao, C., Ciampaglia, G. L., Varol, O., Yang, K., Flammini, A., and Menczer, F. (2017). The spread of low-credibility content by social bots. *arXiv preprint arXiv:1707.07592*.

Smadi, S., Aslam, N., and Zhang, L. (2018). Detection of online phishing email using dynamic evolving neural network based on reinforcement learning. *Decision Support Systems*, 107:88–102.

socialbot (2018). https://whatis.techtarget.com/definition/socialbot. [Online; accessed 21-August-2018].

Sohrabi, M. K. and Karimi, F. (2018). A feature selection approach to detect spam in the facebook social network. *Arabian Journal for Science and Engineering*, 43(2):949–958.

Stringhini, G., Kruegel, C., and Vigna, G. (2010). Detecting spammers on social networks. In *Proceedings of the 26th annual computer security applications conference*, pages 1–9. ACM.

the BuzzFeed Quiz, T. and Skills!, T. Y. F. N. D. (2018). https://www.psafe.com/en/blog/take-buzzfeed-quiz-test-your-fake-news-knowledge-skills/. [Online; accessed 21-August-2018].

Varol, O., Ferrara, E., Davis, C. A., Menczer, F., and Flammini, A. (2017). Online human-bot interactions: Detection, estimation, and characterization. *arXiv preprint arXiv:1703.03107*.

Wang, P., Angarita, R., and Renna, I. (2018). Is this the era of misinformation yet? combining social bots and fake news to deceive the masses. In *The 2018 Web Conference Companion*.

Wu, L. and Liu, H. (2018). Tracing fake-news footprints: Characterizing social media messages by how they propagate. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 637–645. ACM.

Wu, Z., Yin, W., Cao, J., Xu, G., and Cuzzocrea, A. (2013). Community detection in multi-relational social networks. In *WISE (2)*, volume 8181 of *Lecture Notes in Computer Science*, pages 43–56. Springer.

Yamaguchi, Y., Takahashi, T., Amagasa, T., and Kitagawa, H. (2010). Turank: Twitter user ranking based on user-tweet graph analysis. In *International Conference on Web Information Systems Engineering*, pages 240–253. Springer.

Yang, C., Liu, J., Hsu, C., and Chou, W. (2014). On improvement of cloud virtual machine availability with virtualization fault tolerance mechanism. *The Journal of Supercomputing*, 69(3):1103–1122.

Yardi, S., Romero, D., Schoenebeck, G., et al. (2009). Detecting spam in a twitter network. *First Monday*, 15(1).

Zhang, X., Zhu, S., and Liang, W. (2012). Detecting spam and promoting campaigns in the twitter social network. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 1194–1199. IEEE.