Detecting Fake News Spreaders with Behavioural, Lexical and Psycholinguistic Features

Notebook for PAN at CLEF 2020

Héctor Ricardo Murrieta Bello, Lukas Heilmann, and Esben Ronan

University of Copenhagen
xhd160@alumni.ku.dk, tnm946@alumni.ku.dk, and srh648@alumni.ku.dk

Abstract In this paper, we propose a multilingual approach to identifying fake news spreaders on Twitter data, as part of the PAN 2020 competition for author profiling. We manually engineered domain-specific features covering behavioural, lexical and psycholinguistic aspects and evaluated them using traditional machine learning models. The focus of this paper is exploring the problem domain by testing domain-specific features on different types of classifiers first, and evaluating a pure multilingual approach on combined English and Spanish texts second. Out of the methods tested, the Gradient Boosting classifiers performed best. We extended our experiments to a multilingual design using less preprocessing and feature selection, with a comparable result to the monolingual Gradient Boosting models.

Keywords: fake news detection, gradient boosting, liwc, psycholinguistic features, behavioural features, lexical features, feature engineering.

1 Introduction

Fake news has become a catchphrase present all over the political arena in the recent years, and its presence is rapidly increasing. The effects are wide-ranging. There is, for example, an undeniable interference of fake news in traditional political processes like political campaigns. It is believed that fake news generated to favor either of the two nominees in the 2016 presidential campaign was shared up to 37 million times on Facebook [19], the top 20 of which generating up to 9 million shares. Fake news reporting the injury of Barack Obama in an explosion caused a wiping out of up to $130 billion in stock value [16].

Fake news is not a new problem, but its ability to be easily and rapidly disseminated in large scale across a population is. Fake news as a term thus uniquely locates its domain in the realm of social media, where its new, pressing importance is found. The rapid democratisation of public expression that social media encapsulates operates hand-in-hand with the new role of fake news.

The multifaceted nature of this phenomenon, and it's crucial relevance to the functioning of political processes, public discourse, and the well-being of citizens highlights the importance to develop cutting-edge techniques in detection of fake news spreading, so damaging information can be cut off before it spreads beyond a critical mass.

Our study answers the PAN Shared Task "Profiling Fake News Spreaders on Twitter"[12]. We approach the problem using a classical machine learning approach utilising Gradient-Boosting methods on domain-specific features, such as lexical features, grammatical features, sentimental analysis with LIWC[1], SentiWordnet[2] and EmoLex[3], Twitter-DNA as well as TF-IDF vectorisation.

2 Related Work

Fake news is a complex phenomenon that has elicited a great variety of approaches from different fields of academia. [19] provides a succinct overview over four broad types of approaches hitherto taken: i) "knowledge-based", which focuses upon the truth-value of the knowledge content of the news; ii) "style-based", which concerns the linguistic manner in which news is communicated; iii) "propagation-based", which focuses on how fake news spreads through a network, and iv) "credibility-based", which investigates the credibility of those who create and those who spread news. We will briefly discuss previous research in the category most relevant to our task: Style-based.

2.1 Style-based Fake News Detection

Style-based approaches focus on quantifiable stylistic features of texts, upon which machine learning algorithms can be trained to detect characteristics of the expression of fake news creators and spreader. These tend towards assessing the *intention* of the text, over its content. There is good experimental evidence in forensic studies to suggest statements derived from factual experiences differ from those derived from statements based in fantasy [19]. Not only this, but the current state-of-the-art in style-based approaches varies from 60-90% [19], which is significantly higher than a normal person's ability to differentiate, as mentioned in the introduction.

A variety of stylistic features can be studied, two main streams of which [19] have identified. The first is "attribute-based language features", which derive from psychological deception theories. They include attributes such as: Quantity (counts of characters, words, sentences, etc), sentiment (positivity and negativity), diversity (number of unique words, unique content words, etc), and uncertainty (degree of modal words, quantifiers, generalising terms, etc used). The other category spans "structure-based language features", which describes content style from four different language levels: i) lexicon, ii) syntax, iii) semantic and iv) discourse. The lexicon level assesses frequency of letters and words using n-gram models or TF-IDF. The syntax level uses NLP techniques such as POS-tagging. The semantic level identifies topics that texts cover, using

[1] http://liwc.wpengine.com/

[2] http://sentiwordnet.isti.cnr.it/

[3] https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

packages such as LIWC. Discourse features include features on the level furthest out, such as Rhetorical Structure Theory (RST). Generally, it seems to be the case that with only one language used, lexicon-level features perform better than all others - 11 out of 14 studies studied by [19] report better performance at the lexicon-level. [**?**] achieved a very high level performance (75-99% accuracy) detecting fake news on a selection of Bulgarian news sites. They utilised linguistic features (n-grams), credibility-related features such as sentiment polarity, capitalisation and punctuation, as well as semantic embeddings. More recently, [4] introduced the usage of novel stylometric features such as Twitter DNA, text readability and TF-IDF to assess the author's profile.

2.2 Author Profiling

A related task to a style-based fake news detection is author profiling. Author profiling is an old problem where one attempts to infer characteristics about the author of a text based on stylistic analysis. This could be, for example, gender, age, native language, personality, age, economic class. Author profiling on social media inspires its own set of problems but previous PAN tasks have shown a success in predicting a variety of characteristics from Twitter data: Whether or not the user is a bot [3], personality traits [15], gender [13] and language variety [14].

2.3 Multilingual Approaches

An interesting direction this research is beginning to take is multilingualism. Combining features across different languages was found to outperform those using features from a single language in eight out of twelve studies, according to a survey by [19]. A recent innovative approach is the one shown by [3], which incorporated counting features and psycho-linguistic features in order to assess whether an user is a bot or a human. They proposed an ensemble model utilising a neural network and a fine-tuned BERT model to a high accuracy.

3 Dataset

The provided dataset had various idiosyncrasies that provided various constraints to analysis. All meta data of users and tweets (e.g. profile picture, timestamps) was omitted - only the textual data remained. This prevented any analysis according to suggestive avatars, or posting time patterns. Beyond that, URLs, hashtags and usernames retweeted were replaced with the respective placeholders #URL#, #HASHTAG#, #USER#. This prevented any classification according to the news they were spreading, and instead forced us to focus on stylistic features specifically.

The task itself was crucially focused upon classifying fake news *spreaders*, which meant we had to concentrate upon the stylistic features of a fake news spreader. Knowledge-based approaches did not seem relevant, since tweets did not necessarily make knowledge claims, but often consisted of clickbait-like expression, e.g."#HASHTAG #The One Super Bowl Moment We Can't Stop Watching #URL# #URL#".

An interesting subtlety that makes this task more difficult, is there may well be a large presence of "naive" fake news spreaders, who only intermittently spread fake news. Thus, some users may only partially have tweets with fake news as content, but they are classified as a whole as fake news.

4 Method

4.1 Preprocessing

To preprocess the raw data, we implemented separate steps that specifically tackled the idiosyncrasies of the dataset. In the first step, we removed all tags which we labelled as user "behavior": Retweets, URLs, hashtags and mentions of users. Further steps provided functionality for: lowercasing words, removing extra whitespaces, expanding contractions, removing punctuation, removing stop words, removing numbers, lemmatizing and removing ellipses. Lemmatisation and lowercasing were reserved for higher level feature extraction like TF-IDF.

4.2 Feature Extraction

We chose to extract domain-specific features from the respectively preprocessed texts in English and Spanish. We did this through multiple extraction steps, which we shall illustrate in the following.

Behavioral Features The first features to be extracted were the URL, hashtag, retweet and user counts per user. We did this because initial data explorations indicated that fake news spreaders often used more hashtags and retweets than normal users.

To investigate the structural distribution of these elements we decided to explore a novel tweet history DNA algorithm designed by [5]. This algorithm was used in the context of bot detection, but it provided a novel way to analyse the behaviour of a Twitter user. We implemented it with some modifications for our purposes. As proposed in [5], specific tweet behaviors are represented by the values illustrated in . A weighted linear combination of these behavioural values in each tweet is transformed into a letter using the ASCII index.

$$A_n = \begin{cases} 0, & \text{plain} \\ 8, & \text{retweet} \\ 16, & \text{reply} \\ 1, & \text{has hashtags} \\ 2, & \text{has mentions} \\ 4, & \text{has URLs} \end{cases} \tag{1}$$

For example, a retweet with hashtags, mentions, but no URLs would be calculated as $0 * 0 + 8 * 1 + 1 * 1 + 2 * 1 + 4 * 0 + 65 = 84 = T$ (each value multiplied by 1 if the feature is present and 0 if not present). Each resultant letter is concatenated to

each other to form a tweet history DNA string 100 letters long. From this string we calculate the frequency of every character in the string so that we finish with a vector of 16 dimensions.

Lexical Features To extract all lexical, grammatical and miscellaneous features we implemented an additional vectorisation step. We counted all character-level features (digits, alpha characters, capital characters, punctuation, etc), all token-level features (word lengths, syllable counts, emoji/emoticon counts, etc), as well as POS tag counts, NER tag counts and others (see table below for more details).

To extract the emoticons and emojis, we utilised the `emot` library [4]. To extract NER tags and POS tags we used the open-source `SpaCy` library. To count misspellings, we utilised the open-source `pyspellchecker` library [5]. Due to the limited range of the library, it was crucial to subtract any identified named entities from the total count.

We furthermore implemented some measures of vocabulary diversity. The first was hapax legomena frequency, which calculates the amount of words that appear only once [10]. Hapax legomena frequency is commonly used in author profiling [6], as often their frequency correlates with the size of the author's vocabulary. Another way to measure vocabulary diversity is by the type-token ratio - the ratio of unique word count to total word count.

We also implemented a universal quantifier count, since we hypothesised that fake news spreaders often utilise more absolute manners of expression, e.g. "why liberals are always wrong".

The length of the final vector outputted per user in regard to lexical measures was 166. The exact composition is depicted in Table 4.2.

[4] https://pypi.org/project/emot/
[5] https://pypi.org/project/pyspellchecker/

Description	Number of features
Stop word count	1
Digit character count	1
Alpha character count	1
Capital character count	1
Capital character ratio	1
Capitalised word count	1
Capitalised word ratio	1
Punctuation count	1
Punctuation sequence count	1
Character count	1
Word count	1
Average word length	1
Syllable count	1
Average syllable count	1
Emoji count	1
Emoji ratio	1
Emoticon count	1
Emoticon ratio	1
Misspell count	1
Type-token ratio	1
Universal quantifier count	1
Hapax count	1
Individual alphabet count	26
Individual punctuation count	32
Individual POS tag counts	68
Individual NER tag counts	18

Table 1. Index of lexical features implemented in `LexicalVectorizer`

Psycholinguistic Features Inspired by the work of Joo et. al. [3], we considered psycholinguistic lexica as a valuable resources for extracting features. While humans are trained to perceive such underlying mental mechanisms, machines require the help of annotated lexical resources, which we implemented in a variety of manners.

We started off by incorporating sentiment polarity scores that have proven useful as facilitators for automatic authorship profiling in earlier competitions. [9] SentiWordNet 3.0[6], a publicly available lexical resource for opinion mining in English, holds three sentiment scores for each synonym set present in the popular WordNet[7]: positivity, negativity and objectivity. As an example, the English adjective "amazing" is represented by the values 0.875, 0.125 and 0 for the aforementioned measures. The noun "secret", however, has the scores 0.125, 0.375 and 0.5. Hence, "amazing" can be interpreted as a more positive word, while "secret" reveals a rather negative and objective nature. The objectivity score of each word is calculated with the formula $1 - (positivity + negativity)$, which consequently serves as a general measure of affectivity. To tackle words with a

[6] http://sentiwordnet.isti.cnr.it/

[7] https://wordnet.princeton.edu/

plurality of meanings, the part-of-speech tag of the given word is integrated. Accordingly, we computed part-of-speech tags per document as a preparatory step. We then queried SentiWordNet through the Natural Language Toolkit[8] (NLTK) and transformed the obtained scores into features.

Linguistic Inquiry and Word Count[9] (LIWC) is a lexical resource for psycholinguistic measures. By integrating the tool, we intended to capture the author's social and psychological states, which have proven to be effective in author profiling tasks. [3] To our advantage, the lexicon was available in both English and Spanish. With the purpose of summarizing language variables and linguistic dimensions, the first category LIWC captures includes word frequencies, POS-frequencies, etc. The second category aims at associating a word with certain psychological processes, such as affective processes, social processes (e.g. family), personal concerns, etc. [3] As we already covered a significant amount of the stylistic features in the aforementioned lexical approach, we directed our attention at the second category.

To deepen the study of the emotional patterns of fake news spreaders, we investigated the NRC Word-Emotion Association Lexicon[10] (EmoLex) developed by Mohammad et. al.. This vectorizer captures eight basic emotions from Plutchik's wheel of emotions: anger, fear, anticipation, trust, surprise, sadness, joy, and disgust. [8,**?**] The lexicon has been used for sentiment and emotion analysis, abusive language detection, personality trait identification, etc. on word-, sentence-, and tweet-level and has proven useful in the SemEval shared task competition 2018. [7]

Token Count Vectorizers An important set of features to employ alongside the aforementioned feature extraction methods is TF-IDF. TF-IDF is important in measuring not just the relevance of a word to a document (using frequency as a heuristic), but it's *relative* relevance, as compared to the compositions of other documents.

TF-IDF is composed of: The *term frequency* within a document, the frequency a term appears within a given document; and the *inverse document frequency*, which takes the logarithm of the inverse of the frequency of documents in which the term appears. These two are calculated for each term in each document and compiled into a vector. The vector is restricted to a length specified by the hyperparameter "max_features". After experimentation the performance was found to be optimum when max_features was equal to 500. This restriction includes only the most frequent terms.

4.3 Models

We investigated a variety of different experimental designs which we will now present. We used as input the aforementioned reduced set of features for English and Spanish texts, as well as all features, in order to allow for later comparisons.

To tune hyperparameters of each algorithm, we utilised a grid search automation. Grid search is an exhaustive brute-force algorithm that takes a list of lists specifying a range of values for each hyperparameter, derives all possible permutations and tests for

[8] https://www.nltk.org/

[9] http://liwc.wpengine.com/

[10] https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

their accuracy. Each test was achieved by averaging the accuracies of a repeated stratified 10-fold (*scikit-learn*'s implementation). Since we used *scikit-learn*'s implementation, we were unable to perform TF-IDF within each fold, since the module implements cross-validation automatically. We reasoned, however, that since it was only for tuning parameters, it did not matter so much.

Multinomial Naive Bayes Multinomial Naive Bayes is a supervised machine learning algorithm often used for text classification. It relies upon Bayesian conditional reasoning and generating data from which to classify. One manner of generating the data is by creating Gaussian distributions from the mean and standard deviation of each class, another is by creating simple multinomial distributions. Since "the multinomial distribution describes the probability of observing counts among a number of categories" [18], this algorithm seems to be most appropriate for our vectors composed of count features.

We tested hyperparameters using grid search. The hyperparameters available to tune were: "Alpha", which we tested values between 0.5 and 1.5; and "fit_prior", for which there were values of True and False.

Random Forest Random Forest is a flexible and quick-to-train ensemble machine learning algorithm based upon decision trees [17]. Decision trees iteratively split the dataset into subsets according to a quantitative threshold along an axis, the resultant class for the subset derived from a majority vote. Random forest prevents the overfitting tendency of decision trees by averaging the results of decision trees trained upon randomly split training data (bagging).

Hyperparameters to adjust included number of estimators (decision trees), for which we selected logarithmically interpolated values: 10, 100, 1000, and max_features (for node-splitting), for which we chose: 'sqrt', 'log2'.

Support Vector Machine (SVM) SVM is a supervised learning algorithm which learns by finding the most optimum hyperplane to linearly separate data into distinct areas, corresponding to a positive and negative classification. The hyperplane is calculated by the utilisation of support vectors, data points close to the hyperplane, the distance (margin) from which is to be maximised. Though in its basic form it is designed for linear classification, it can be mapped to non-linear sets by using different "kernels" [2].

To tune it we utilised *scikit-learn*'s `GridsearchCV` with a stratified 5-fold cross validation procedure. Kernel parameters we tested were: 'poly' (polynomial), 'rbf' (radial base function), 'sigmoid' and 'linear'. The regularization parameter C values we tested were: 50, 10, 1.0, 0.1, and 0.01. We used "scale" as the gamma hyperparameter value and a linear kernel.

Gradient Boosting Gradient boosting is a powerful supervised machine learning algorithm that is born out of the theoretical assumption that "weak learners" can be turned into better learners [1]. Weak learners are observations who as hypotheses perform only

slightly better than chance. An ensemble of these can sequentially handle their respective observation and iteratively tackle difficult problems. Though it is comparatively computationally expensive algorithm, it is powerful at grasping subtleties in data, and since our dataset is small, computational expenditure is not as big an issue.

The hyperparameters we selected to tune are: number of estimators with values of 10, 100, 500, 800, 1000; Learning rate with logarithmic values of: 0.001, 0.01, 0.1, subsample with values of 0.5, 0.7, 1.0. and max depth with values of 3, 7, 9. Since the `GradientBoostingClassifier` turned out to be our most promising algorithm, we decided to implement a manual grid search that allowed implementation of TF-IDF within each fold, so that we could be assured to discover the best parameters.

5 Results

We subsequently present our results for the aforementioned methods employed. As we aimed for a stabilized comparison of performances among the different experimental setups, we consistently utilized a stratified 10-fold cross validation with 3 repeats, each with a training size of 80% of the features.

5.1 Experiment 1

For the first experiment we tested each dataset individually using a variety of machine learning algorithms previously described. First we tested the algorithms on just the domain-specific features as described in the method section. After that we tested them with a combination vector of domain-specific features and the TF-IDF vector. We then tuned the hyperparameters for each one using the `SKLearn`'s `GridSearchCV`.

As can be seen in Table 2, the Gradient Boosting algorithm performed significantly better on both the English and Spanish datasets than any of the other algorithms. Random Forest performed next best, with the Support Vector Machine and Multinomial Naive Bayes coming last. Interestingly, in every single case we tested, there was a higher accuracy on the Spanish dataset than on the English dataset.

5.2 Experiment 2

Our multilingual approach consisted of individual preprocessing of the English and the Spanish dataset, followed by extraction of the manually engineered features for each dataset. We deleted language dependent features such as sentimental analysis with SentiNet (only English) and the additional ten LIWC categories for Spanish. We then concatenated the two vectors into one. Subsequently, we split the datasets into training and validation sets (20%) in each fold. Finally, as a further experiment, we merged English and Spanish tweets and fitted a TF-IDF vectorizer. In this setup we did not explore the application of any feature extraction algorithm or hyper-parameter optimization. Since the best algorithm in the monolingual setup was gradient boosting, we decided to use that algorithm for this section.

Language	Model	Features	Accuracy	Std
English	Multinomial Naive Bayes	ALL	0.651	0.089
Spanish	Multinomial Naive Bayes	ALL	0.703	0.073
English	Multinomial Naive Bayes	ALL + TFIDF	0.651	0.089
Spanish	Multinomial Naive Bayes	ALL +TFIDF	0.703	0.073
English	Random Forest	ALL	0.703	0.079
Spanish	Random Forest	ALL	0.72	0.070
English	Random Forest	ALL + TFIDF	0.71	0.078
Spanish	Random Forest	ALL +TFIDF	0.73	0.076
English	Support Vector Machine	ALL	0.652	0.075
Spanish	Support Vector Machine	ALL	0.717	0.085
English	Support Vector Machine	ALL + TFIDF	0.651	0.085
Spanish	Support Vector Machine	ALL +TFIDF	0.711	0.084
English	Gradient Boosting	ALL	0.669	0.075
Spanish	Gradient Boosting	ALL	0.73	0.068
English	Gradient Boosting	ALL + TFIDF	**0.726**	0.087
Spanish	Gradient Boosting	ALL +TFIDF	**0.747**	0.064

Table 2. Results for English and Spanish

As can be seen in the results table, the gradient boosting algorithm used upon the vector combining all features and TF-IDF functioned best, though not by a particularly large margin. There was however no increase in accuracy compared to the monolingual setups, and it performed significantly worse than upon just the spanish dataset.

Model	Features	Accuracy	Std
Gradient Boosting	All	0.71	0.075
Gradient Boosting	All + TF-IDF	**0.7218**	0.0460

Table 3. Accuracies and standard deviations for experiments in the multilingual setup

5.3 Results on test data

Our final submission was made using the methodology of the second experiment. However during for the test phase, we predicted our answers separately for the Spanish tweet feed and for the English tweet feed. In order to test the efficiency of our algorithm we made use of the software submission platform [11]
To our surprise, the accuracy on the Spanish dataset vastly outperformed the accuracy on the English dataset, with an accuracy of 0.7450 to 0.65550. This was in contrast to our first experiments, where the difference was markedly less.

Language	accuracy
English	0.6550
Spanish	0.7450

Table 4. Results for English and Spanish on the test data

6 Conclusion

Our study has demonstrated the importance of carefully selected domain-specific features in the domain of fake news identification. These features are integral in classifying monolingual texts as well as multilingual text. We can conclude that it was possible to classify fake news spreaders on a limited dataset consisting of 300 Twitter users, by applying gradient boosting to a set of lexical, behavioural and psycholinguistic features and TFIDF to represent the text. Furthermore, these features have been shown to provide useful information in multilingual environments and are not limited to monolingual contexts.

References

1. Friedman, J.H.: Stochastic gradient boosting. Computational statistics & data analysis **38**(4), 367–378 (2002)
2. Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intelligent Systems and their applications **13**(4), 18–28 (1998)
3. Joo, Y., Hwang, I.: Author Profiling on Social Media: An Ensemble Learning Model using Various Features. Notebook for PAN at CLEF 2019 (2019)
4. Kosmajac, D., Keselj, V.: Twitter bot detection using diversity measures. In: Proceedings of the 3rd International Conference on Natural Language and Speech Processing. pp. 1–8 (2019)
5. Kosmajac, D., Keselj, V.: Twitter user profiling: Bot and gender identification. In: CLEF (Working Notes) (2019)
6. Martinc, M., Skrjanec, I., Zupan, K., Pollak, S.: Pan 2017: Author profiling-gender and language variety prediction. In: CLEF (Working Notes) (2017)
7. Mohammad, S.M., Bravo-Marquez, F., Salameh, M., Kiritchenko, S.: Semeval-2018 Task 1: Affect in tweets. In: Proceedings of International Workshop on Semantic Evaluation (SemEval-2018). New Orleans, LA, USA (2018)
8. Mohammad, S.M., Turney, P.D.: Crowdsourcing a word-emotion association lexicon **29**(3), 436–465 (2013)
9. Patra, B.G., Banerjee, S., Das, D., Saikh, T., Bandyopadhyay, S.: Automatic author profiling based on linguistic and stylistic features. Notebook for PAN at CLEF **1179** (2013)
10. Popescu, I.I., Altmann, G.: Hapax legomena and language typology. Journal of Quantitative Linguistics **15**(4), 370–378 (2008)
11. Potthast, M., Gollub, T., Wiegmann, M., Stein, B.: TIRA Integrated Research Architecture. In: Ferro, N., Peters, C. (eds.) Information Retrieval Evaluation in a Changing World. Springer (Sep 2019)
12. Rangel, F., Giachanou, A., Ghanem, B., Rosso, P.: Overview of the 8th Author Profiling Task at PAN 2020: Profiling Fake News Spreaders on Twitter. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (eds.) CLEF 2020 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2020)

13. Rangel, F., Rosso, P.: Overview of the 7th author profiling task at pan 2019: Bots and gender profiling in twitter. In: Proceedings of the CEUR Workshop, Lugano, Switzerland. pp. 1–36 (2019)
14. Rangel, F., Rosso, P., Potthast, M., Stein, B.: Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter. Working notes papers of the CLEF pp. 1613–0073 (2017)
15. Rangel Pardo, F.M., Celli, F., Rosso, P., Potthast, M., Stein, B., Daelemans, W.: Overview of the 3rd author profiling task at pan 2015. In: CLEF 2015 Evaluation Labs and Workshop Working Notes Papers. pp. 1–8 (2015)
16. Rapoza, K.: Can 'fake news' impact the stock market? by Forbes (2017)
17. Svetnik, V., Liaw, A., Tong, C., Culberson, J.C., Sheridan, R.P., Feuston, B.P.: Random forest: a classification and regression tool for compound classification and qsar modeling. Journal of chemical information and computer sciences **43**(6), 1947–1958 (2003)
18. VanderPlas, J.: Python data science handbook: Essential tools for working with data. " O'Reilly Media, Inc." (2016)
19. Zhou, X., Zafarani, R.: Fake news: A survey of research, detection methods, and opportunities. CoRR **abs/1812.00315** (2018), http://arxiv.org/abs/1812.00315