2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH) | 978-1-6654-4084-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/IHSH51661.2021.9378748

Fake News detection Using Machine Learning

Nihel Fatima Baarir
Computer science department
Mohamed Khider University of Biskra
nihelbaarir04@gmail.com

Abdelhamid Djeffal
LESIA Laboratory
Mohamed Khider University of Biskra
a.djeffal@univ-biskra.dz

***Abstract*—The phenomenon of Fake news is experiencing a rapid and growing progress with the evolution of the means of communication and Social media. Fake news detection is an emerging research area which is gaining big interest. It faces however some challenges due to the limited resources such as datasets and processing and analysing techniques.**

In this work, we propose a system for Fake news detection that uses machine learning techniques. We used term frequency-inverse document frequency (TF-IDF) of bag of words and n-grams as feature extraction technique, and Support Vector Machine (SVM) as a classifier. We propose also a dataset of fake and true news to train the proposed system. Obtained results show the efficiency of the system.

***Index Terms*—Fake news, Social media, Web Mining, Machine Learning, Support Vector Machine, TF-IDF.**

I. Introduction

In the last decade, Fake News phenomenon has experienced a very significant spread, favored by social networks. This fake news can be broadcasted for different purposes. Some are made only to increase the number of clicks and visitors on a site. Others, to influence public opinion on political decisions or on financial markets. For example, by impacting the reputation of companies and institutions on the Web. Fake news concerning health on social media represents a risk to global health. The WHO warned in February 2020 that the COVID-19 outbreak had been accompanied by a massive 'infodemic', or an overabundance of information—some of which was accurate and some of which was not—which made it difficult for people to find reliable sources and trustworthy information when they needed it. The consequences of disinformation overload are the spread of uncertainty, fear, anxiety and racism on a scale not seen in previous epidemics [11].

In this paper, we present a novel method and tool for detecting fake news that uses:

- **Text preprocessing:** consisting of steaming and analyzing the text by removing stop words and special characters.
- **Encoding of the text:** using bag of words and N-gram then TF-IDF.
- **Extraction of the characteristics:** this allows a precise identification of false information. We use the source of a news, its author, the date and the feeling given by the text as features of a news.
- **Support vector machine:** a supervised machine learning algorithm that allows the classification of new information.

This paper is structured as follows: Section 2 presents some existing proposals for fake news detection. Section 3 details our proposal and its different components. Section 4 presents the implementation of our proposal as well as some of the obtained results. Section 5 concludes the paper and presents some perspectives.

II. Related works

In literature, many works are interested to fake new detection.

Authors of [3] propose a typology of several methods of truth assessment emerging from two main categories: linguistic cue approaches with machine learning and network analysis approaches, for detecting fake news.

In [5], authors present a simple approach to fake news detection using a naive Bayesian classifier. This approach is tested on a set of data extracted from Facebook news posts. They claim to be able to achieve an accuracy of 74%. The rate of this model is good but not the best, as many other works have achieved a better rate using other classifiers. We discuss these works in the following.

Authors of [1] propose a fake news detection model that uses n-gram analysis and machine learning techniques by comparing two different feature extraction techniques and six different classification techniques. The experiments carried out show that the best performances are obtained by using the so-called features extraction method (TF-IDF). The used the Linear Support Vector Machine (LSVM) classifier that gives an accuracy of 92%. This model uses LSVM that is limited to treat only the case of two linearly separated classes.

Authors of [14] describe how users of social networks can ensure the truth of information. They also describe the mechanisms that allow their validation and the role of journalists or what to expect from researchers and official institutions. This work helps people see a little bit of the truth behind the news on social media and not believe anything.

Authors of [9] propose several strategies and types of indices relating to different modalities (text, image, social information). They also explore the value of combining and merging these approaches to assess and verify shared information.

In [8], the authors present an overall performance analysis of different approaches on three different datasets. This work focused on the text of the information and the feeling given by it, and ignores some features like the source, the author

978-1-6654-4084-4/21/$31.00 ©2021 IEEE

or the the date of the publication that can have a dramatic impact on the result. Besides, in our work, we will show that the integration of the feeling in the detection process does not bring any valuable information.

Authors of [16] created a new public dataset of valid new articles and proposed a text-processing based machine learning approach for automatic identification of Fake News with 87% accuracy. It appears that this work focuses on the emerging feelings from the text and not on the contain of the text in it self.

Authors of [17] introduced LIAR, a new dataset for automatic fake news detection. This corpus can also be used for stance classification, argument mining, topic modeling, rumor detection, and political NLP research. Most of the works in this area have used this benchmark. However, it is well-known that this last is restricted to political information, while others have integrated information from various fields.

The overall drawback of these approaches is that the categorical data encoding may not be valid in reality ! Besides, the usual fake news classification is limited to two values (i.e., namely, Real or Fake), while in reality we can not say that the news is real or fake at 100%, but according to a degree of confidence. We consider that this point is very important to classify the news in social media.

III. Proposed system

The system we propose uses a news dataset to build a decision model based on support vector machine method. The model is then used to classify novel news to fake or real.

A. General architecture of the proposed system

The proposed system takes as input a dataset of comments and their related information, such as date, source and author. It then transforms them into a features dataset that can be used in the learning phase. This transformation is called *preprocessing*, it performs a series of operations such as cleansing, filtering and encoding. The preprocessed dataset is divided into two parts: the first for training and the second for testing. The training module uses the training dataset and support vector machine algorithm to build a decision model that can be applied to the test dataset. If the model is accepted (i.e., it is able to achieve an acceptable accuracy rate), it can be kept and used and then training ends. Otherwise, the parameters of the learning algorithm are revised in order to improve the accuracy rate. Figure 1 illustrates the general scheme of the proposed system.

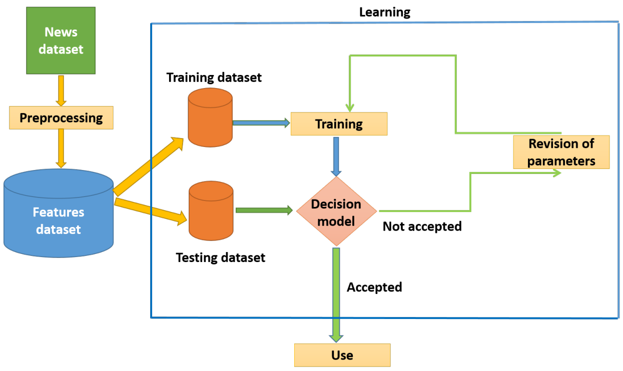

Figure 1. The proposed fake news detection system architecture's

B. Preprocessing

In the news dataset, news characteristics are classified into three categories: textual data, categorical data and numerical data. Each category preprocessing is performed through a set of operations as illustratd in Figure 2:

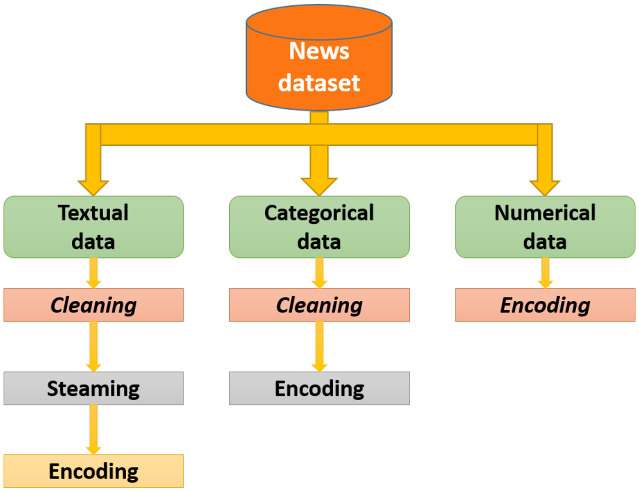

Figure 2. Preprocessing of different categories of news charateristics

- **Textual data.** Represent the text written by the author in a news and pre-processed by the following operations:
  1) Cleaning: eliminating stop words and special characters.
  2) Steaming: transforming the useful words into roots.
  3) Encoding: transforming all the words of the comment into a numerical vector. This needs two steps: the combination of two techniques, namelly, bag of words [13] and N-grams [4], then the application of the TF-IDF method [12] on the result.
  $$TF\text{-}IDF_t = TF_t \times IDF_t = \frac{n}{k} \times \log \frac{D}{\dot{D}_t}$$
  Where:
  - $TF_t = \frac{n}{k}$: the number of appearances of term t in the document n divided by the total number k of

terms in the document, keeping the multiplicity of each term.

- $IDF_t = \log \dfrac{D}{D_t}$: the total number of documents D divided by the number D_t of documents citing this term.

- **Categorical data.** Represent the source of the news such as TV channel, newspaper or magazine, and its author. The pre-treatment of these data is performed through two steps:
  - Cleaning: eliminating special characters and transforming letters into lowercase.
  - Encoding: for sources we used a label encoding. For authors, we created our own encoding to convert the author's names into digital numbers, so that authors from the same source are close to each other compared to authors from other sources.

    We created a list containing two fields, the first for the source and the second for its authors, then we replaced each author by its index number by adding the sum of the sizes of the previous sources plus one. Figure 3 shows an example where:

    * T is the number of authors of the source (size).
    * i_k is the author index number k.

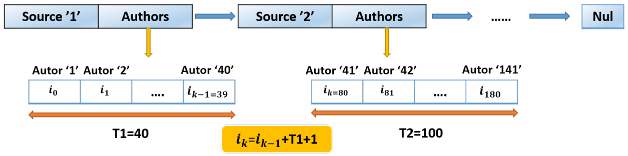

Figure 3. Calculation of authors indices

- **Numerical data.** Represent the date of posting the comment and the sentiment given by the text. Since the date is already represented by a numerical value, we only split it into three unique values: day, month and year. For the sentiment given by the text, we calculate the sum of the sentiment degrees of the words.

  According to the experts, each word has a degree of sentiment which allows it to be classified into three classes:
  - If the sum is less than 0, the feeling is negative.
  - If the sum is greater than 0, the sentiment is positive.
  - If the sum is 0, the feeling is neutral.

C. Learning

It brings together two modules, namely, training and validation.

1) Training: to train our model, we have chosen the support vector machine algorithm [15]. This allows to use the value of the decision function given for a news as a confidence level of its classification: a positive value for the decision function designates, at the same time, a true news as well as its degree of truth and vice-versa, a negative value of the decision function designates a Fake news as well as its degree of fakeness. Figure 4 illustrates this idea.

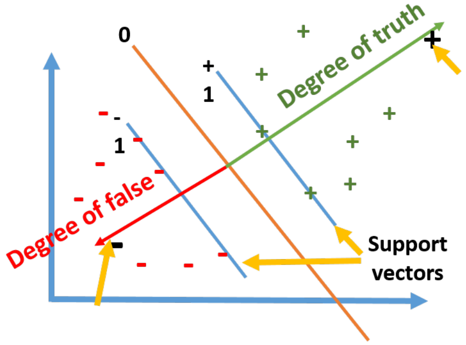

Figure 4. Degree of Confidence for news classification using support vector machine decision function

The maximum and minimum of the decision function are therefore calculated during the training phase and used to compute the degree of truth or fallacy by the following function:

$$p = \begin{cases} \dfrac{Dec}{Max_{dec}} \times 100 & if\ Dec > O \\ \\ \dfrac{Dec}{Min_{dec}} * 100 & else \end{cases}$$

Where:

- Dec is the decision function value;
- Max_{dec} and Min_{dec} are the maximum and minimum values of the decision function;
- p is the percentage of truth or fake.

2) Validation: to measure the capacity of the model to recognize new examples, we set aside some of the examples to be used as test models. The features dataset is then subdivided into two parts, a training part and a test part. Its usefulness consists in avoiding over-fitting, i.e., testing the model on the same training dataset. The subdivision is not done at random but according to a particular sample using the method of cross validation [10].

D. Revision of parameters

This operation aims to improve the model's accuracy by tuning or setting the parameters of the support vector machine algorithm, namely, Cost,γ, ϵ and change the cross-validation variant [2].

E. Use

This is the last and most important phase in our system. After reaching the best recognition rate, i.e., after building the best model, we can now use it on new unlabeled news, and the model allows us to predict their classes: wrong or true, with a confidence degree.

IV. Experiments and Results

The performances of the proposed system was tested using a dataset that we build by merging a true news datset with a fake news one.

A. *Used Dataset*

We have merged two existing datasets "Getting Real about Fake News" [6] containing fake news and "All the news" [7] containing real news. These datasets were obtained from the Kaggle site, the first contains text and metadata extracted from 244 websites marked as false by Daniel Sieradski's BS Detector Chrome detector, extracted using the API webhose.io. This dataset contains approximately 12,999 social media posts, divided into 20 columns of different types; categorical, numeric and textual. The second dataset contains texts and metadata taken from New York Times, Breitbart, CNN, Business Insider, Atlantic, Fox News, Talking Points Memo, Buzzfeed News, National Review, New York Post, The Guardian, NPR, Reuters, Vox and the Washington Post, retrieved using BeautifulSoup and stored in Sqlite, split into three separate CSV files. This last dataset contains texts and metadata subdivided into 10 columns of different types; categorical, numeric and textual.

After pre-processing the two datasets and testing the features one by one until reaching the best accuracy rate. We have obtained a dataset which contains the following features:

- 5 words obtained by the bag of words method,
- 3 compound words obtained by the N-gram method,
- date: day, month and year,
- feeling,
- source,
- author,
- class: fake or real.

B. *Results and discussion*

To get the best decision model with highest accuracy, we tuned many parameters. First we tried to get the best parameters from both bag of words and n-gram techniques which give the best recognition rate on our dataset.

For bag of words' technique we have directed the number of most frequent words taken from each comment. This operation is repeated several times until the best rate is reached. At each time we increased the number of the most frequent words. On the Weka software and using the SM0 library, and with the cross validation for 10 parts, we obtained the following results:

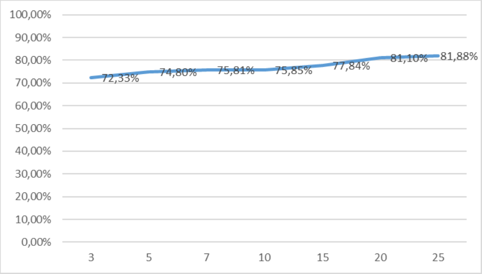

Figure 5. Evolution of the rate according to the number of frequent words for the word bag

As shown in Figure 5 the recognition rate increases with the number of most frequent words, up to 25 words, then begins to decrease, which we explain by the phenomenon of over-fitting.

For the n-gram's technique, we stepped the number of grams. This operation was also repeated several times increasing the value of n each time. We got these results:

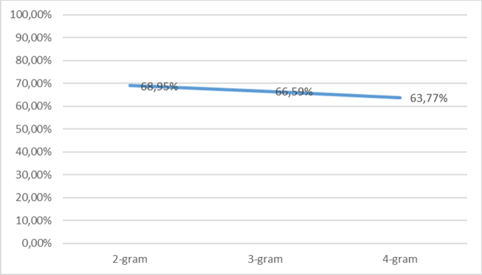

Figure 6. Evolution of the rate according to the n-gram

In Figure 6, we observe that after 2-grams the recognition rate started to decrease, which is very logical, due to the small size of the text of the news; a block of words of more than 2 will not be repeated several times in the same piece of news which will not exceed 5 lines at most.

We stopped at 2-grams and proceeded to switch the number of most frequent word blocks k. We obtained the following results:

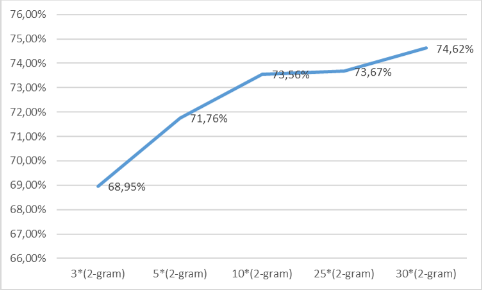

Figure 7. Evolution of the rate according to the k * (2-grams)

In Figure 7 we observe that the rate continued to increase without exceeding the rate obtained by the word bag technique. This is due, in our opinion, to two causes: either the small size of the information text, or the incompatibility of n-grams with the TF-IDF method. We then thought to combine the two techniques. We started by combining 5 frequent words with the frequent 3 * (2-gram) which gave us a rate of 52.30 %. Then, we added the other characteristics to measure the influence of each one on the recognition rate. The following figure 8 represents the evolution of accuracy rate depending on different features: by testing on the training data and using the RBF kernel of the LIBSVM [2] method in WEKA [18].

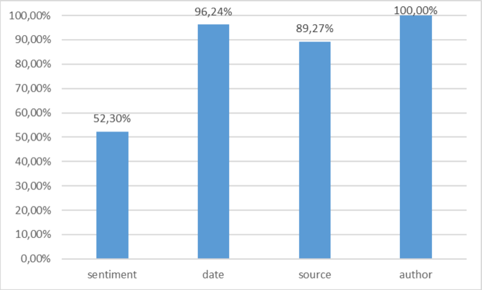

Figure 8. Influence of different features on accuracy

In Figure 8, we notice that the influence of the feature "Sentiment" on the accuracy is almost negligible, which seems to be very logical: if a feeling released by a comment was negative it does not mean that it is fake. However, the characteristic "source" increased it up to 89.27%, and "date" up to 96%. While the author feature pushed it to 100%, which shows the effectiveness of the encoding we have proposed.

Figure 9 shows the results obtained by the different kernels: LIBSVM and their tunning on WEKA.

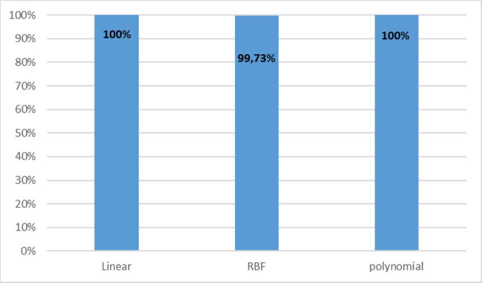

Figure 9. Accuracy according to the kernel type

It is clear that linear and polynomial kernels give the best results. The linear kernel is parameterless and faster, however, theoretically it cannot model the cases of complicated overlap of the two classes. On the other hand the Gaussian kernel makes it possible to model any type of overlap but its accuracy depends on the parameters C (Cost), ϵ and γ. We have studied the influence of these parameters on the precision of the model.

- Influence of Cost C: the following Figure 10 represents the evolution of accuracy according to the Cost C: by testing on the training dataset and using the RBF kernel of the LIBSVM method in WEKA:

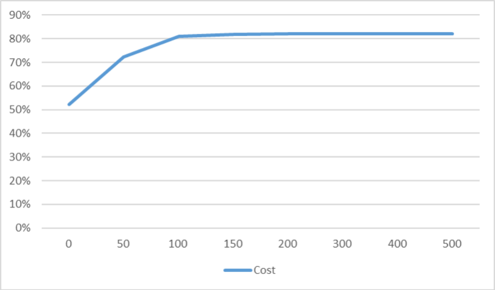

Figure 10. Evolution of the accuracy according to the Cost C

at the start the cost is equal to 0 and the value of the rate is 52% then by increasing the cost value we observe a rapid increase in the rate up to the value 150, then a stabilization of the rate all around the value 82% despite the fact that we continued to increase the cost with high values.
This is due to our opinion for the following reason: it is known that for high values of C, the optimization will choose a hyper-plane with a smaller margin, conversely, a very small value of C will cause the optimization to seek a separation hyper-plane with a larger margin. So in this case the two classes are very close to each other, then the separation margin is small and that is found with the value 150, after this value there is no data in the margin..

- Influence of ϵ: Figure 11 represents the evolution of accuracy depending on ϵ: by testing on the training data and using the RBF kernel of the LIBSVM method in WEKA:

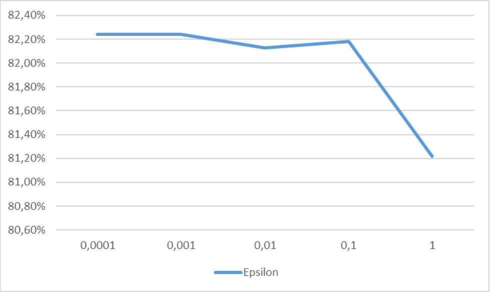

Figure 11. Evolution of the accuracy rate according to ϵ

we observe a stabilization of the rate around 82% up to the value 0.1 then a slight drop in the rate which can be neglected up to the value 1. This shows that the ϵ parameter does not have a great influence on the rate of recognition. Which is very logical because this parameter determines the tolerance of the termination criterion. That's the allowed error rate that's all.

- Influence of γ: the following Figure 12 represents the evolution of accuracy depending on the γ: by testing on the training data and using the RBF kernel of the LIBSVM method in WEKA:

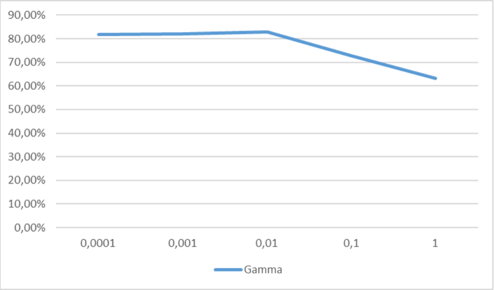

Figure 12. Evolution of the accuracy rate according to Gamma

With a $C = 300$ and a $\epsilon = 0.0001$, the recognition rate increases to the value of $\gamma = 0.001$, then a stabilization around the rate 82 % then we observe a rapid decrease from the value of γ = 0.01.

At the end we obtained the best model accuracy with the following parameters: Cost C = 300, ϵ = 0.0001 and γ = 0.001.

V. Conclusion

This paper presents a method of detecting fake news using support vector machine, trying to determine the best features and techniques to detect fake news. We started by studying the field of fake news, its impact and its detection methods. We then designed and implemented a solution that uses a dataset of news preprocessed using cleaning techniques, steaming, N-gram encoding, bag of words and TF-IDF to extract a set of features allowing to detect fake news. We applied then Support Vector Machine algorithm on our features dataset to build a model allowing the classification of the new information.

Through the research carried out during this study, we obtained the following results:

- the best features to detect fake news are in order: text, author, source, date and sentiment.
- the followed process resulted in a recognition rate of 100%.
- the analysis of the sentiment given by the text is interesting, however it would be more influential in the case of opinion mining.
- the N-gram method gives a better result than the bag of words with bulky datasets and with large texts.
- the support vector machine seems the best algorithm to detect fake news, because it gave a better recognition rate, and allowed to give for each information a degree of confidence for its classification.
- the parameters influencing the support vector machine are in order: Cost C, gamma γ and epsilon ϵ.

The work we have done could be completed and continued in different aspects. It would be relevant to extend this study with a larger dataset, and to evolve its supervised learning by another online for a continuous update and automatic integration of new fake news.

References

[1] Hadeer Ahmed, Issa Traore, and Sherif Saad. Detection of online fake news using n-gram analysis and machine learning techniques. In *International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, pages 127–138. Springer, 2017.

[2] Chih-Chung Chang and Chih-Jen Lin. LIBSVM – A Library for Support Vector Machines, July 15, 2018.

[3] Niall J Conroy, Victoria L Rubin, and Yimin Chen. Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4, 2015.

[4] Chris Faloutsos. Access methods for text. *ACM Computing Surveys (CSUR)*, 17(1):49–74, 1985.

[5] Mykhailo Granik and Volodymyr Mesyura. Fake news detection using naive bayes classifier. In *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, pages 900–903. IEEE, 2017.

[6] Kaggle. Getting Real about Fake News, 2016.

[7] Kaggle. All the news, 2017.

[8] Junaed Younus Khan, Md Khondaker, Tawkat Islam, Anindya Iqbal, and Sadia Afroz. A benchmark study on machine learning methods for fake news detection. *arXiv preprint arXiv:1905.04749*, 2019.

[9] Cédric Maigrot, Ewa Kijak, and Vincent Claveau. Fusion par apprentissage pour la détection de fausses informations dans les réseaux sociaux. *Document numerique*, 21(3):55–80, 2018.

[10] Refaeilzadeh Payam, Tang Lei, and Liu Huan. Cross-validation. *Encyclopedia of database systems*, pages 532–538, 2009.

[11] Cristina M Pulido, Laura Ruiz-Eugenio, Gisela Redondo-Sama, and Beatriz Villarejo-Carballido. A new application of social impact in social media for overcoming fake news in health. *International journal of environmental research and public health*, 17(7):2430, 2020.

[12] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142. New Jersey, USA, 2003.

[13] Gerard Salton and J Michael. Mcgill. 1983. *Introduction to modern information retrieval*, 1983.

[14] Florian Sauvageau. *Les fausses nouvelles, nouveaux visages, nouveaux défis. Comment déterminer la valeur de l'information dans les sociétés démocratiques?* Presses de l'Université Laval, 2018.

[15] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Adaptive Computation and Machine Learning series, 2018.

[16] DSKR Vivek Singh and Rupanjal Dasgupta. Automated fake news detection using linguistic analysis and machine learning.

[17] William Yang Wang. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*, 2017.

[18] Lechevallier Y. *WEKA, un logiciel libre d'apprentissage et de data mining"*. INRIA-Rocquencourt.