DEVELOPPEZ UNE PREUVE DE CONCEPT

Detecting fake news using sci-kit learn and deep learning

FEBRUARY 19, 2021

OPENCLASSROOMS

Sommaire

I. Introduction

Observant les dommages que peut causer la propagation rapide de fausses nouvelles (fake news), le journal « ***Paalga*** » nous demande de développer un modèle robuste de détection automatique de fake news afin d'améliorer le modèle « ***Binomial Logistic Regression*** » qui est en production depuis plus de cinq (5) ans et qui réalise un score ROC-AUC de ***95%.***

1. Objectif

L'objectif de notre projet est d'utiliser les modèles récents d'apprentissage supervisés afin d'améliorer la détection automatique de fausses nouvelles. Notre méthode baseline sera le modèle ***Binomial Logistic Regression*** en production.

2. État de l'art des algorithmes de détection des fake news

La résolution de la problématique liée à la détection des fausses nouvelles fait appel à l'apprentissage supervisé du machine learning. En effet, les données utilisées pour l'apprentissage comportent les étiquettes ou labels qui permettent d'identifier si un article donné est une fausse ou vraie nouvelle. C'est plus précisément un problème de classification. Ce problème se décompose en trois parties :

- Les ***données*** en entrée du système.
- Le ***classifieur*** qui apprend à reconnaître les fausses informations ;
- Les ***classes*** sont les différentes réponses possibles du classifieur : ***vrai*** ou ***faux***.

a) Aux origines : La détection de spam

La détection de courriels de spam fut une des premières applications des algorithmes automatiques de détection de contenu frauduleux. Ces algorithmes utilisent des techniques de machine learning permettant de classifier du texte comme étant du spam ou du contenu légitime.

Ils impliquent le pré-traitement du texte, l'extraction de caractéristiques (souvent appelées features) via des méthodes comme les sacs de mots, et la sélection de ces features basée sur ceux menant aux meilleures performances dans la performance de l'algorithme sur un jeu d'entraînement. Ces features sont ensuite classifiés en utilisant

divers classifieurs comme les ***K Nearest Neighbours*** (***KNN***), les ***Support Vector Machines*** (***SVM***) ou les ***Naive Bayes*** Classifiers.

Comme pour la détection de fake news, le but de ces algorithmes est de séparer des exemples de textes véridiques d'exemples de textes fallacieux. Parmi les différents algorithmes, le classifieur ***naïf de Bayes*** obtient de bons résultats (Androutsopoulos et al., 2000), bien que sa difficulté soit de s'adapter à de nouveaux types de spams (Sasaki & Shinnou, 2005), pour lesquels l'utilisation des ***KNNs*** permettait une adaptation rapide du modèle en obtenant des résultats similaires à ceux obtenus avec des algorithmes robustes en traitement du langage comme les ***SVMs***. Toutefois, plusieurs des études publiées rapportant l'efficacité des algorithmes de détection de spam ne rapportaient que les résultats sans donner d'indications sur le contenu permettant aux algorithmes de classifier un mail comme étant un spam ou non (eg. Androutsopoulos et al., 2000 ; Debarr & Wechsler, 2009 ; Sasaki & Shinnou, 2005).

b) Détecter les fake-news à l'ère du deep learning

L'année 2018 a été décisive pour le Traitement de Langage Naturel (NLP). L'apprentissage par transfert, en particulier des modèles tels que Allen AI's ELMO, OpenAI's Open-GPT et BERT de Google, a permis aux chercheurs de résoudre les problèmes de classification de texte avec une précision absolument remarquable[1].

Le principal avantage de l'utilisation des modèles du deep learning par rapport aux approches classiques du machine learning est qu'il ne nécessite pas d'extraction de features fait à la main ; au contraire, il permet d'identifier le meilleur ensemble de features caractérisant le jeu de données. La puissante capacité d'apprentissage du modèle deep learning est principalement due à l'utilisation de plusieurs étapes d'extraction de caractéristiques qui peuvent automatiquement à partir de l'ensemble de données. Dans les approches existantes (Jwa H, *Appl Sci.* 2019;9(19):4062.)**,** plusieurs idées inspirantes ont été discutées pour faire progresser les réseaux neuronaux convolutifs (CNN). L'idée d'utiliser un bloc de couches comme unité structurelle gagne également en popularité parmi les chercheurs (Kaliyar RK, *Cognitive Systems Research.* 2020;61:32–44).

[1] https://medium.com/@aniruddha.choudhury94/part-2-bert-fine-tuning-tutorial-with-pytorch-for-text-classification-on-the-corpus-of-linguistic-18057ce330e1

En ce qui concerne l'utilisation des algorithmes modernes du deep learning, la société Fabula.ai, récemment acquise par Twitter, utilise une méthode qui prend en compte à la fois le contenu des informations et les caractéristiques extraites des réseaux sociaux, permettant résultats de 0,93 de score ROC-AUC (Federico Monti, 2019).

Dans Junaed Younus Khan (2019), les performances de plusieurs algorithmes (classique et de deep learning) sont comparées en classant les nouvelles vrais et de faux, avec des résultats de 95% de précision.

Enfin, en utilisant uniquement le contenu des informations (Yang Yang, 2018), une technique basée sur un réseau neuronal convolutif (CNN) est proposée pour détecter les fausses nouvelles en utilisant les titres et l'image de la rubrique, obtenant des résultats de 92 % de précision.

3. Méthodologie

Pour notre travail, nous proposons une approche utilisant le deep learning basée sur les représentations d'encodeurs bidirectionnels des transformateurs (BERT). Nous utilisons BERT comme un encodeur de phrases, qui peut obtenir avec précision la représentation du contexte d'une phrase. Ce travail est en contraste avec les travaux de recherche précédents (De S, Sohan FY, 2018) où les chercheurs regardaient une séquence de texte de manière unidirectionnelle. De nombreuses méthodes existantes et utiles ont été présentées (De S, Sohan FY, 2018 ; Malik S, 1991) avec des réseaux neuronaux séquentiels pour coder les informations pertinentes. Cependant, un réseau neuronal profond avec une approche d'entraînement bidirectionnelle peut être une solution optimale et précise pour la détection de fausses nouvelles. La méthode que nous proposons améliore les performances de détection des fausses nouvelles grâce à la puissante capacité de capturer les dépendances sémantiques et à longue distance dans les phrases.

Nous allons nous appuyer principalement sur le travail de Ria Gandhi[2] qui présente un cas pratique de l'implémentation de BERT. Les avantages que présentent l'utilisation du modèle BERT de Google sont : un développement rapide, une bonne performance sur petit jeu de données et de meilleurs résultats.

[2] https://towardsdatascience.com/getting-real-with-fake-news-d4bc033eb38a

II. Présentation des données

Dans cette partie, nous présenterons les données utilisées dans notre projet.

1. Origine des données

Les données utilisées dans notre projet sont obtenues de deux façons différentes :

- Le web-scraping de 9 sites web comportant des articles d'actualité.
- Téléchargement des données hébergées par Kaggle qui contiennent des articles d'actualité faux et vrais de 2015 à 2018.

a) Web scraping

Dans le but d'obtenir des articles d'actualité variés et récents, nous parcourrons 9 sites web afin de télécharger des articles. Ces articles obtenus seront classés faux ou vrai suivant la notoriété du site. Ainsi, nous obtenons le classement suivant :

[Vraie nouvelle]: CNN, BBC, The Guardian, Fox News, NBC News, Washington Post

[Fausse nouvelle] : BreitBart, The Onion, InfoWars

Après avoir installé les librairies ***feedparser***[3]***, newspaper***[4] et utiliser l'algorithme initialement développé par ***Joachim Holwech***[5] **,** nous avons pu télécharger au total **1 526** articles d'actualité caractérisés par le ***titre***, le ***texte***, le ***lien web***, la ***date de publication*** et le ***site***.

b) Kaggle datasets.

Sur le site de Kaggle, nous avons trouvé une base de données d'articles d'actualité constituée de deux fichiers regroupant d'une part les fausses nouvelles et de l'autre, les vraies nouvelles. Nous avons **23 481** fausses nouvelles et **21 417** vraies nouvelles. Ces nouvelles sont caractérisées par le ***titre***, le ***texte***, le s***ujet***, la ***date de publication*** et le ***label***.

[3] https://pythonhosted.org/feedparser/
[4] https://buildmedia.readthedocs.org/media/pdf/newspaper/latest/newspaper.pdf
[5] https://holwech.github.io/blog/Automatic-news-scraper/

2. Fusion des datasets

Nous avons pu obtenir au total ***46 424*** articles d'actualité téléchargés. Après avoir supprimé les colonnes non pertinentes, reformater et ajouter des étiquettes, nous obtenons **24 095** fausses nouvelles et **22 310** vraies nouvelles caractérisées par le ***titre***, le ***texte***, la ***date de publication*** et le ***label***

3. Analyse exploratoire des données

a) Analyse temporelle des publications

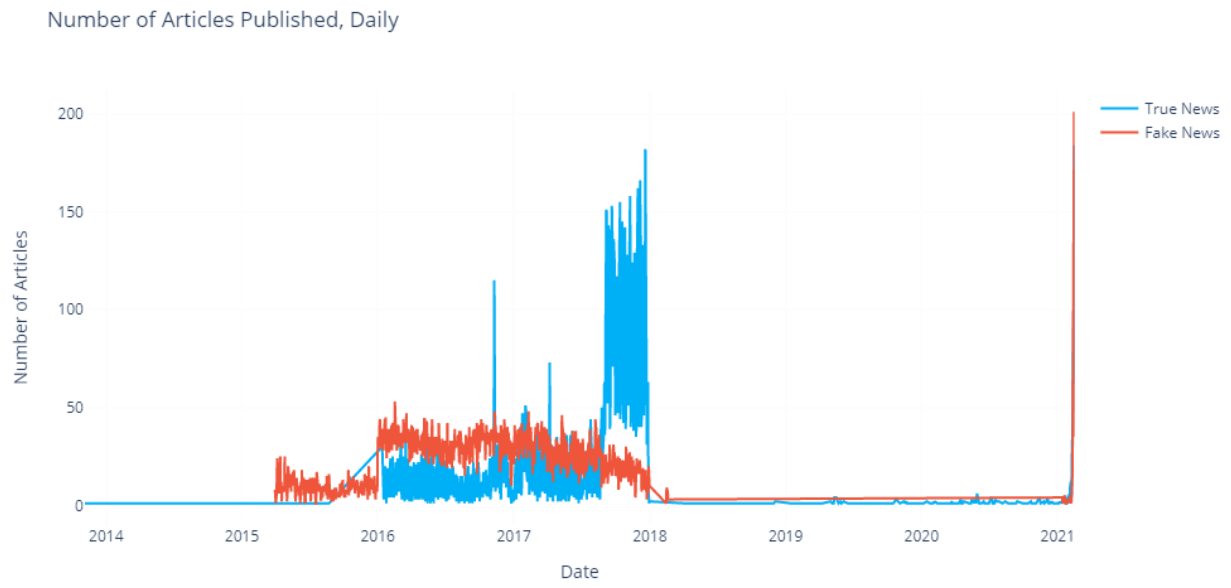

Figure 1 : Nombre d'articles publiés par jour

D'après la figure 1, la date de publication des articles va du 01 novembre 2013 au 06 février 2021. La hausse de fausses nouvelles observées entre 2016 et 2018 est principalement dû aux élections présidentielles américaine de 2016. Celle observée entre fin 2020 et 2021 est dû au COVID19.

b) Analyse des articles

Il est difficile de différencier une fausse nouvelle d'une vraie. En effet selon la figure 2 la distribution du nombre de mots contenu dans l'article est quasiment la même dans les deux cas.

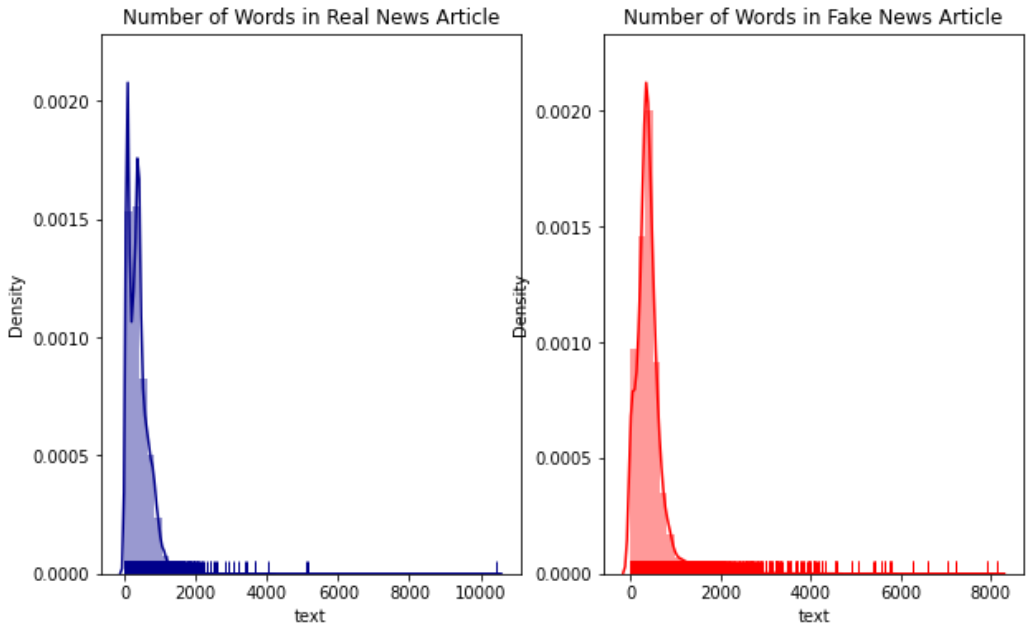

Figure 2:Nombre de mots par article

De même, les fausses et vraies nouvelles ont pratiquement la même distribution de la polarité (***fig.3***). Ce qui rend difficile à un être humain de différentier les deux types de nouvelles.

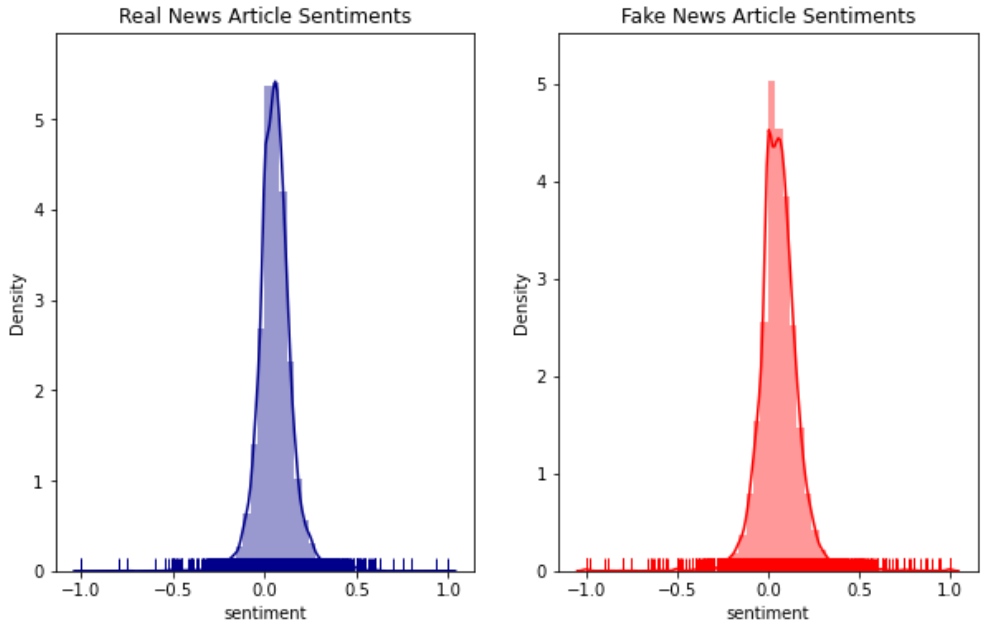

Figure 3: Distribution de la polarité des articles

III. Traitement des données

Afin d'utiliser les algorithmes de classification de scikit-learn, le traitement des données est une phase impérative avant la modélisation prédictive. En effet, les données mises à notre disposition comportent des irrégularités et doivent être nettoyées avant de pouvoir être utilisées par les algorithmes de prédiction.

Dans le cas du transfert learning (BERT), le niveau de traitement est beaucoup moins élevé que dans l'approche utilisant les algorithmes de classification de scikit-learn car BERT a été entraîné avec des phrases complètes.

1. Nettoyage des articles

Le nettoyage des articles consiste d'une part à supprimer les stops-words et les caractères non alpha-numérique et d'autre part, à réduire chaque mot à sa forme élémentaire en le lemmatisant.

2. N-grams

Afin de prendre en compte les expressions, c'est-à-dire les groupes de mots qui n'ont de sens que une fois employés ensemble, nous allons procéder à la construction de unigrams, bigrams et trigrams.

Nous obtenons au total 216 tokens à partir du dataset.

3. Séparation du jeu de données

Après la phase précédente, nous allons scinder notre dataset en jeux de données d'apprentissage (training set) et de test (testing set). Le training set contitue 80% du jeu de données total et les testing set 20%.

Nous utiliserons le training set pour l'analyse du corpus et l'entraînement des modèles de prédiction et le testing set pour évaluer le modèle.

4. Features engineering pour les modèles de scikit-learn

Nous utiliserons la fonction ***TfidfVectorizer*** du module ***feature_extraction.text*** de ***sklearn*** pour transformer les articles en un ***ndarray*** de dimension ***(m, t)*** où ***m*** est le nombre d'articles et ***t*** le nombre de tokens obtenu au total à partir des articles. La valeur se trouvant à la i^{ème} ligne et la j^{ème} colonne du ***ndarray*** est le nombre X définit par:

$$X = \frac{bow}{T} \times \log\left(\frac{m}{m'}\right)$$

Où :

- ***bow*** = nombre de fois le token ***j*** est présent dans l'article ***i ;***
- ***T =*** nombre de tokens de l'article ***i ;***
- ***m =*** nombre d'articles
- $\boldsymbol{m'}$ = nombre d'articles comportant le token ***j.***

Ce traitement permet de donner un poids plus important aux tokens les moins présents dans le corpus et donc considérés comme plus porteurs d'information.

IV. Modélisation

1. Modèle Baseline

Notre modèle baseline sera le modèle ***Binomial Logistic Regression*** qui est en production au niveau de l'entreprise. Nous allons reprendre le modèle baseline sur notre jeu de données afin de comparer ses performances d'une part avec d'autres modèles de scikit-learn et d'autre part avec BERT.

2. Modèles de Scikit-learn

Nous entraînerons les modèles ***naive bayes***, le ***SVM*** et le ***random forest*** sur notre jeu de données afin de comparer leurs performances avec le modèle. Quelques hyper-paramètres ont été optimisés par cross-validation sur le train set. Le module ***scikit learn*** de python nous fournira les algorithmes nécessaires pour implémenter la modélisation.

3. Transfert Learning (BERT)

Selon les travaux de Ria Gandhi, le modèle BERT pourrait nous fournir de très bonnes prédictions. Nous entrainerons le modèle BERT sur notre jeu de données afin de comparer les performances de ce modèle à notre modèle baseline.

V. Présentation des résultats

1. Métriques

Les métriques nous permettent de mesurer la performance des modèles et de comparer ainsi des modèles entre eux.

La métrique que nous choisirons d'utiliser afin de mesurer la performance de nos modèles sera la ROC-AUC score. Elle est un indice synthétique calculé à partir de la courbe **ROC**.

2. Synthèse

Nous avons réalisé au total 5 modèles de prédiction dont les performances ont été représentées dans la figure ci-dessous (***fig4***).

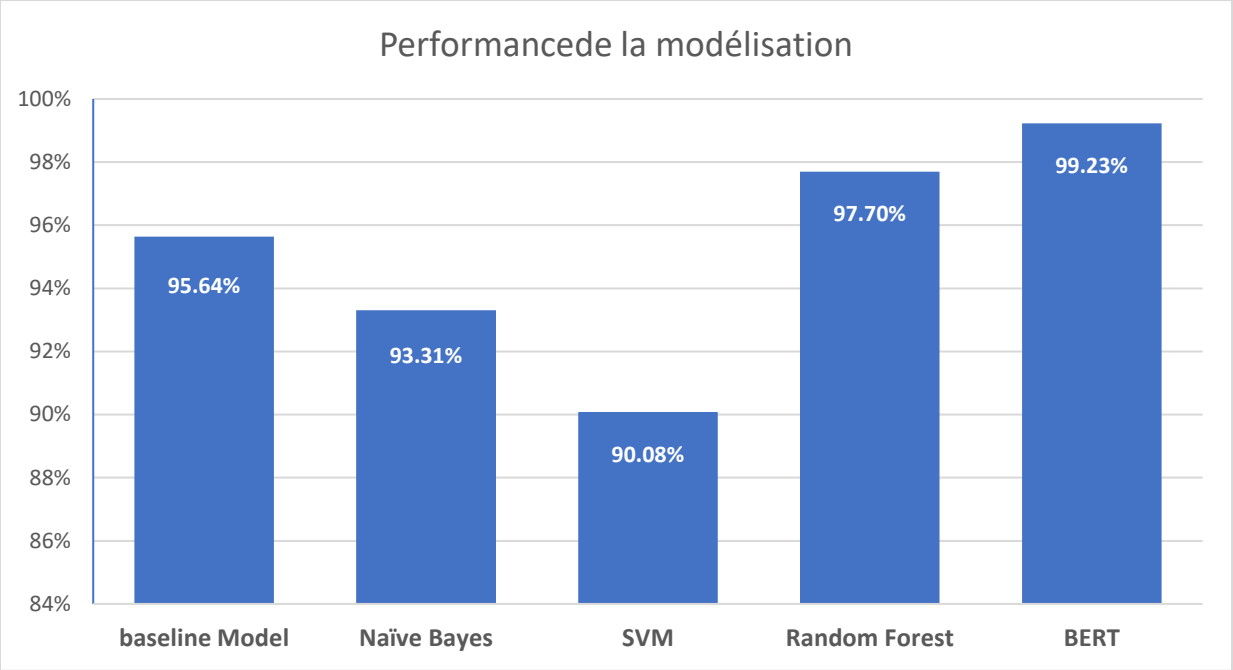

Figure 4:Comparaison des performances de la modélisation

Le modèle baseline a un score de 95,64% et performe nettement mieux que le modèle naive bayes (93.31%) et le support vecteur machine (90%).

Cependant le modèle Random Forest performe mieux que notre modèle baseline avec un score de 97.70%.

Le modèle BERT réalise un score ROC-AUC de 99,23% et performe nettement mieux que tous les autres modèles. Nous déploierons le modèle BERT à la place du modèle baseline et améliorer ainsi le modèle prédictif de l'entreprise « Paalga ».

VI. Conclusion

Le journal « ***Paalga*** » utilise le modèle ***Binomial Logistic Regression*** pour la prédiction des fake news. Bien que ce modèle réalise un score ROC-AUC de plus de 95%, les conséquences que présentent les fake news pour un journal nous amènent à concevoir un modèle qui performerait mieux que le modèle baseline. Le transfert learning nous offre de très bonnes perspectives. L'implémentation du modèle BERT nous donne un score de plus de 99% . Le deployement du modèle BERT à la place du modèle baseline nous permettra de répondre aux attentes de l'entreprise « ***Paalga*** »

Sources bibliographiques :

1. Tanik Saikh, Arkadipta De, Asif Ekbal, Pushpak Bhattacharyya (11 May 2020): A Deep Learning Approach for Automatic Detection of Fake News
2. Ria Gandhi (May 3, 2020): Detecting fake news using sci-kit learn and deep learning: https://towardsdatascience.com/getting-real-with-fake-news-d4bc033eb38a
3. https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset?select=True.csv
4. https://planspace.org/20150607-textblob_sentiment/
5. Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980.
6. https://medium.com/@aniruddha.choudhury94/part-2-bert-fine-tuning-tutorial-with-pytorch-for-text-classification-on-the-corpus-of-linguistic-18057ce330e1
7. https://github.com/aniruddhachoudhury/BERT-Tutorials/blob/master/Blog%202/BERT_Fine_Tuning_Sentence_Classification.ipynb
8. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
9. Veronica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic Detection of Fake News. In Proceedings of the 27th International Conference on Computational Linguistics, pages 3391–3401, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
10. Gottfried Jeffrey and Shearer Elisa. 2016. News use Across Social Media Platforms 2016. In Pew Research Center Reports. 1: Long Papers), pages 231–240, Melbourne, Australia. Association for Computational Linguistics.
11. Jwa H, Oh D, Park K, Kang JM, Lim H. exBAKE: Automatic fake news detection model based on bidirectional encoder representations from transformers (BERT) Appl Sci. 2019;9(19):4062.[Google Scholar]

12. Kaliyar RK, Goswami A, Narang P, Sinha S. FNDNetA deep convolutional neural network for fake news detection. Cognitive Systems Research. 2020;61:32–44.[Google Scholar]

13. Munandar D, Arisal A, Riswantini D, Rozie AF (2018) Text classification for sentiment prediction of social media dataset using multichannel convolution neural network. In: 2018 International conference on computer, control, informatics and its applications (IC3INA). IEEE, pp 104–109

14. Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M. Bronstein. Fake news detection on social media using geometric deep learning, 2019.

15. Junaed Younus Khan, Md. Tawkat Islam Khondaker, Anindya Iqbal, and Sadia Afroz. A benchmark study on machine learning methods for fake news detection, 2019.

16. Yang Yang, Lei Zheng, Jiawei Zhang, Qingcai Cui, Zhoujun Li, and Philip S. Yu. Ti-cnn: Convolutional neural networks for fake news detection, 2018.

17. De S, Sohan FY, Mukherjee A (2018) Attending sentences to detect satirical fake news. In: Proceedings of the 27th international conference on computational linguistics, pp 3371–3380

18. Malik S, Sentovich EM, Brayton RK, Sangiovanni-Vincentelli A. Retiming and resynthesis: Optimizing sequential networks with combinational techniques. IEEE Trans Comput-Aided Design Integr Circuits Syst. 1991;10(1):74–84 [Google Scholar].