Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/41/20230744

Fake news text detection based on convolutional neural network

Baozhi Fang[1,3], **Haotian Zhou**[2]

[1]Overseas Chinese College, Capital University of Economics and Business, Beijing, 100070, China
[2]Computer Science and Engineering College, Anhui University of Science and Technology, Huainan, Anhui Province, 232001, China

[3]32021140141@cueb.edu.cn

Abstracts. The swift evolution of mobile devices and multimedia technology has made the Internet one of the primary means of learning new information today. However, the huge amount of news information is often mixed with erroneous fake news, which can cause bad news events to spread and trigger people's bad emotions, putting the healthy development of society and economy at risk. Addressing the real-world application problem of swiftly and accurately detecting fake news is imperative. To mitigate the aforementioned challenges, we propose a method that uses deep learning to detect fake news and validate it through empirical studies. We begin by collecting a sizeable fake news dataset from domestic social media platforms and use a pre-trained deep learning model to extract textual features. Furthermore, we amalgamate convolutional neural networks and deep learning models to effectively glean and encompass the patterns and attributes of disinformation through an analysis of the text's semantic and structural characteristics. Finally, we experimentally evaluate the effectiveness of the method. The experimental findings demonstrate that the suggested approach exhibits commendable performance in the task of detecting fake news, effectively discerning between authentic and fabricated information. Our deep learning-based approach proves to be both efficient and highly impactful in addressing the issue of fake news within the realm of social media.

Keywords: Fake News Detection, Natural Language Processing, Convolutional Net, Text Categorization.

1. Introduction

Social network news often includes news content, contextual social content and external information. Where topical content relates to textual information in the article as well as multi-modal information such as pictures and videos [1]. Information about the social context refers to the news publisher, the news distribution network, and comments and retweeting of the news by other users. External knowledge is defined as objective factual knowledge, which is typically represented by External Knowledge knowledge graphs. The Internet has been flooded with fake news due to the rapid growth of the network in recent years, which severely affects the network environment, among which fake news represented by fake news texts is the most prevalent. Because fake news is designed to get readers' attention, it tends to spread more quickly and is not easily screened, which causes great damage to the healthy development of society and economy [2]. With this in mind, how to achieve an accurate and fast realisation of fake news detection has attracted increasing research attention.

We define fake news detection as the news content of a news article, social contextual content, and external knowledge in order to determine the authenticity of a news story given the news story. The utilization of social media presents a dual-sided phenomenon. On one end of the spectrum, social media serves as an accessible, cost-effective, and swiftly disseminating news source. However, on the other end, it becomes a potent vehicle for the wide propagation of fake news, characterized by its low-quality content and deliberate dissemination of false information [3]. Detecting fake news within the realm of social media presents unique features and challenges, rendering existing detection algorithms employed in traditional news media ineffective or inapplicable. To begin with, fake news is crafted with the deliberate intention of deceiving readers into accepting false information as truth, rendering content-based detection a complex and non-trivial task. Hence, we must incorporate supplementary information, such as users' social activities on social media, to facilitate decision-making in the detection process. Moreover, the omission of such supplementary information presents an inherent challenge, given that data stemming from users' interactions with fake news is voluminous, often incomplete, lacking in structure, and riddled with noise [4].

Early detection of fake news relied heavily on manual review, but it is often difficult to guarantee the quality and authenticity of massive news content, reduce the amount of screen time and stop the dissemination of fake news stories. Manual methods can be both inefficient and time-consuming when it comes to identifying fake news [5-6]. The powerful feature representation capability of convolutional neural networks on information and the powerful ability to model complex processes offer the potential to enhance the technology of automatic fake news detection. Kim et al., for example, proposed the TextCNN for text classification in 2014, which uses multiple convolutional kernels of various sizes. For example, Ajao et al. proposed a framework for fake news posts from Twitter posts based on a hybrid neural network model. Current detection methods [7-9] often choose to push close observation ("zoom in") in order to make a judgment about the authenticity of a given news story by capturing specific line profiles, verification of content authenticity based on a knowledge base, and consideration of user feedback. Such methods ignore the information contained in the news environment in which fake news is created and disseminated: in order to enhance the influence and the destructiveness of the news, fake news often has a tendency to "rub the hot spot", causing the news environment to reflect the recent focus of the mainstream media and the public's concern, which has become a prominent reference point in the creation of fake news [10].

The purpose of this paper is to address the issue of detecting fake news and suggests a method that utilizes deep learning. In particular, we first collect a sizeable fake news dataset from national social media platforms and use a pre-trained deep learning model to extract textual features. Secondly, we combine convolutional neural networks and deep learning models in order to learn and capture the patterns and characteristics of disinformation from the semantic and structural characteristics of the text.

2. Method

2.1. Outline of proposed method

We primarily delineate three key modules: data preprocessing, model construction, and news detection. The goal of data preprocessing is to find the dataset's storage path, read the dataset, define the feature and category attributes, split the training set and test set, using jieba to perform word segmentation, constructing a dictionary, setting the maximum word length to take as 200, intercepting those that exceed the length of the text, fill ing in zeros for those that are not sufficient in the length of the text, matrixing the text, and converting the text into words via the training set data generator and the test set data generator. When constructing the model, we add the self-attention mechanism for weight adjustment. In order to strengthen the training again, the parameters will be updated and the trained model will be saved to the same directory. In addition, we add a confusion matrix to evaluate the model, forward propagation to process the data, and compute the output precision and loss curve. Training the model, updating the parameters to re-strengthen the training, and saving the trained model to the same repertoire. Finally, we call the model to detect the user's news entry and give the output, adding a loop to allow multiple inputs to be called at a time for discrimination.

2.2. Implementation Process

2.2.1. Data Preprocessing. This code segment is defined as the 'handle_csv' function, whose main function is to open the CSV file, read the data, and process it. This function extracts the data from the CSV file and replaces the line breaks and invisible characters in it with empty strings. After processing, the data in the CSV file is shuffled and stored in separate lists of Query1, Query2, and labels. Finally, the function returns three lists, Query1, Query2, and label, as tuples for subsequent processing and analysis. This function has a wide range of applications in the field of data processing and analysis, and provides practical help for students and teachers. The goal of the paper is to improve the data processing and analysis capabilities of individuals and teams, so it is crucial to ensure that the code is optimized and correct. When writing code, avoid errors and unnecessary complexity as much as possible.

CSV file processing: CSV files are a commonly used format for data storage and transfer. The system stores tabular data in plain text format, with each row representing a record and each field separated by commas. In order to work with CSV files, libraries or associated functions must be used to open, read, and process the data within them. If you work with CSV files, you may run into situations where you have to replace certain special characters or invisible characters. One of the most common ways to do this is to replace these characters with empty strings for later processing and analysis.

Data shuffling: Data shuffling refers to the random ordering of data to disrupt the original order of data. Which is a common operation for tasks such as cross validation and training set construction in machine learning and data analytics.

2.2.2. Construction of training model. This code defines a CNN class, which inherits from the torch.nn.Module class. In the initialization function of the class, it creates several layers of the neural network, including the Embedding layer, Conv1d, MaxPool1d, and Linear layers. Specifically, within the embed layer, it creates an embed layer with vocabulary size (vocab_size) and embed dimension (embedding_size) as inputs. In the convolution layer, it uses a 1D convolution layer with 100 output channels, 3 convolution kernel sizes, 1 fill, and 1 step length. This convolutional layer will detect the words at each time step to extract features. MaxPool1d is used to reduce the number of features in the pooling layer by pooling the output of the convolutional layer. Finally, it defines a Self-Attention layer for extracting contextual information from a sentence. All of these layers will be convolved into an end-to-end model for performing the classification task.

CNNS (Convolutional Neural networks): CNNS is a deep learning model that is widely used, particularly suitable for tasks involving structured data such as pictures and audio. Feature extraction is performed using convolutional layers and pooling layers, as well as classifiers or regression through fully connected layers.

Torch.nn.Module: torch.nn.Module is the basic class that defines the model of neural networks in PyTorch. The inheritance of this class makes it easy to define one's own neural network model and to use the various tools and capabilities provided by PyTorch for training and prediction.

The Embedding layer: It is a method for mapping discrete symbols or terms to continuous vector spaces. This approach can transform high dimensional discrete features into low dimensional dense vector representations for better processing by neural network models.

Convolutional layers (Conv1d): The use of convolutional layers in CNNs is a common method for extracting local features from input data. In 1D convolution, the convolution kernel glides along the temporal dimension, performs convolution operations on the input data, and outputs a series of feature maps from the convolution kernel.

Pooled layer (MaxPool1d): The model's complexity and computation are reduced by the pooled layer, which reduces the number of features. MaxPool1d is a common pooling operation that chooses the maximum value in the input feature map as the output value.

Self-Attention: Self-Attention is a method used to model the contextual information of a sequence of events. It can better capture important information in the sequence by computing the correlation between the different positions in the sequence.

2.2.3. Model evaluation module. Binary classification task: Binary classification task is one of the common kinds of task, which aims to partition the input data into two different classes or labels. We focus here on metrics used to assess performance on binary classification tasks, including precision, recall, precision and F1.

Gradient Descent and Backpropagation algorithms: Gradient descent is an optimization algorithm used to update the parameters. Backpropagation algorithms are methods for computing the gradient, which pass the error from the output layer to the input layer, and update the parameters of the model based on the gradient.

Optimizer: An optimiser is an algorithm used to update the parameters of the model. The optimizer used in this paper is passed to the training function as a parameter, in order to update the model parameters with the optimizer at each iteration.

Loss function: The loss function is a function that measures how different the predicted output of the model is from the true label. The objective of this paper is to pass the loss function into the training function as a parameter to compute the loss at each iteration and update the model parameters using the gradient descent algorithm.

Iterator: An iterator is an object used for iterating over the training dataset. Iterators are passed through the training function as arguments in this paper to obtain the next input data at each iteration.

2.2.4. Textual processing. Jieba segmentation is a Chinese word segmentation tool that is capable of menting Chinese texts based on word units. jieba segmentation employs a rule-based and statistical method with high accuracy and speed of segmentation.

Vocabulary: A vocabulary is a dictionary that arranges all words according to some set of rules. A program uses a vocabulary to store all the words that appear in the textual data and maps them to a single integer value

Padding: Padding refers to the length of the sequence data is unified, with the addition of a specific padding symbol at the end of the sequence such that all sequence lengths are fixed values. The program allows the author to populate the textual data so that the length of all text is 200. Thereby providing a better foundation for subsequent model formation and prediction processes.

2.3. Model Training

In this paper, we describe the kernel part of the training function (train()), which aims to train the model, track the loss, precision, recall rate, F1 value, and other indicators in order to improve the performance of the model. In each iteration, the function first blanks the gradient of the model, then uses backpropagation to compute the loss between the model output and the true label, and updates the model weights based on the computed results. Once the model has been updated, the function computes the values of loss, precision, recall, and F1 and returns the results as train_loss, train_acc, train_recall, and train_F1. The function also outputs the training results and the learning rate (obtained from optimizer.param_groups[0][' lr ']) after each round of training to facilitate monitoring of the model state. The function's ultimate objective is to minimize loss and improve model performance.

The code defines the function evaluate() to assess the performance of training the model on the validation data. The function evaluates the given model at each iteration using validation data. The function initially feeds the data into the model, calculating the loss by comparing the predicted value with the actual label.

3. Experiments

3.1. Evaluative indicators

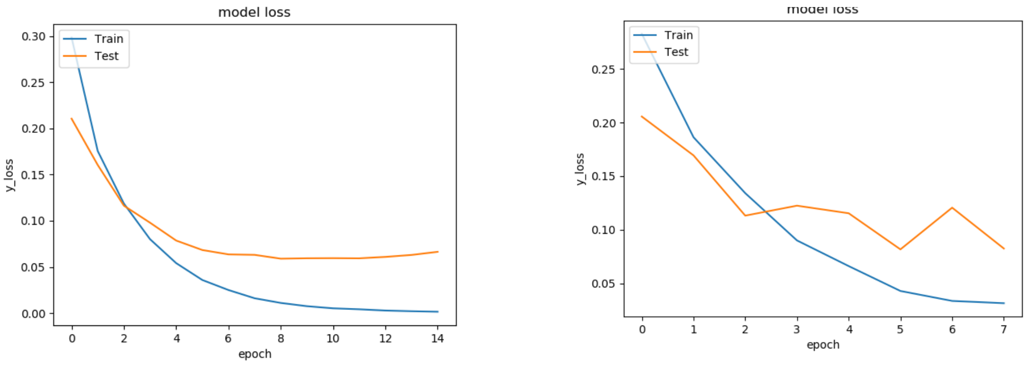
For the sake of fairness and comparability of the experiment, for both the fake news dataset categories 0 (true news) and 1 (fake news) into four types based on the combination of their true category and predicted category. Based on this, we select accuracy and F1-score as indices of model evaluation. Accuracy, in this context, represents the proportion of correctly predicted samples out of the total number of samples. The F1-score, a statistical measure for assessing the accuracy of binary classification models, takes into account both precision and recall. This metric ranges from a minimum value of zero to a maximum value of one and can be viewed as a harmonic mean that balances the model's precision and recall. Ultimately, the F1-score serves as an indicator of how effectively a classification model performs.

3.2. Model-convergence analysis

To assess the convergence of the model, we visualize the evolution of the loss value throughout the model training process. The results of this visualization are depicted in Fig 1. The training loss is able to smoothly decrease from 0.3 to approximately 0.05 as the number of training rounds continues to increase. The same pattern of decreasing loss is also apparent in the test set. This shows that the current model has been adequately trained, and also indicates that the many attention strategies introduced in this paper can operate efficiently.

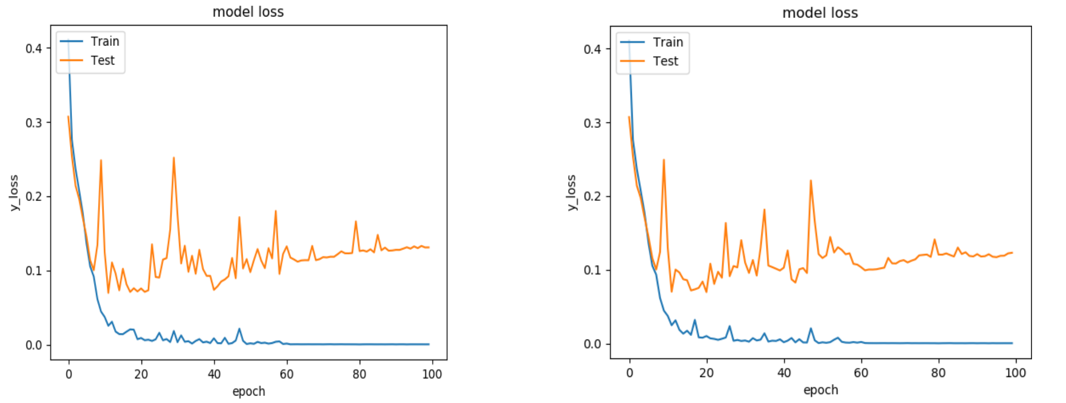
3.3. Model performance analysis

In Fig. 2 and 3, we present the model's accuracy, recall, and F1 score, which are displayed respectively. The original model achieves up to 99% training accuracy and 97% test accuracy as the elapsed training time increases, as can be seen in Fig. 2. When a variety of attention mechanisms are introduced, the training accuracy of the model eventually converges to 100% and the test accuracy is further improved up to 98%. The results above show the need for and effectiveness of introducing attention strategies, which can strengthen the semantic and discriminative properties of the features. In Fig. 3, the results of Recall rate and F1 score also prove the efficacy of this paper's method. Furthermore, when the amount of data is small, the accuracy of the model on the test set is superior to that on the training set, but the rate at which the model loses accuracy is exactly opposite to the change in accuracy, and the loss rate in the test set remains at a certain level as the amount of data is increased up to a certain value.



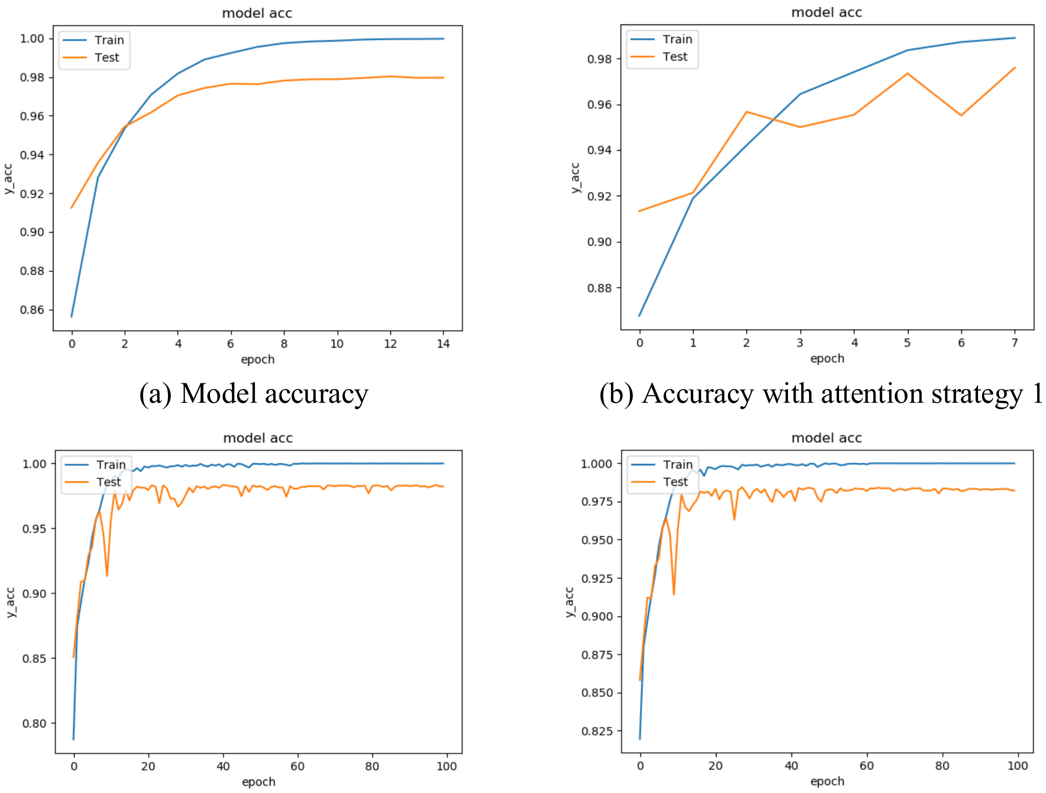
(a) Original loss

(b) Training loss with attention strategy 1



(c) Training loss with attention strategy 2 (d) Training loss with attention strategy 3

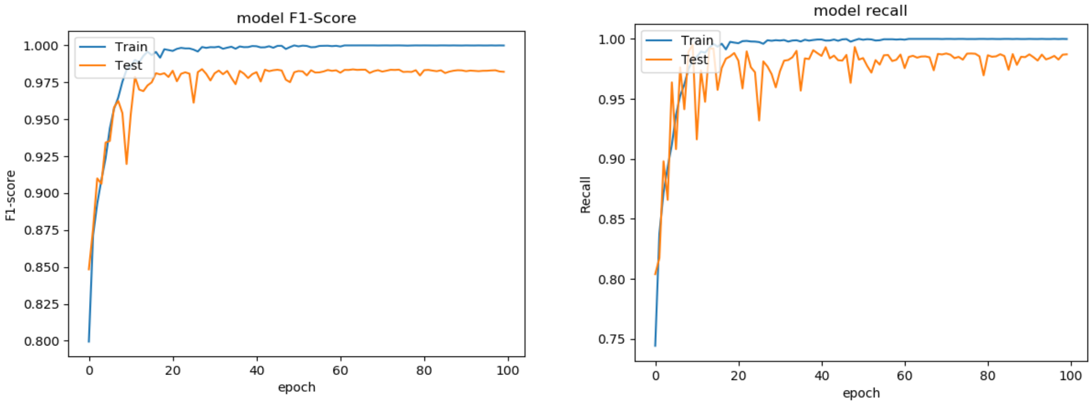
Figure 1. Training loss of proposed method.



(a) Model accuracy (b) Accuracy with attention strategy 1

(c) Accuracy with attention strategy 2 (d) Accuracy with attention strategy 3

Figure 2. Accuracy of proposed method.



(a) F1-score of proposed method (b) Recall of proposed method

Figure 3. F1-score and recall of proposed method.

4. Discussion

Here, we develop a convolutional neural network-based approach for detecting bogus news. To improve the accuracy of the system we must first prepare an appropriate dataset. We used publicly available fake news datasets in this design and preprocessed the text through techniques such as word segmentation, disallowance elimination, and word embeddings. The preprocessed data is stored on the disk. We then used the forward propagation and confusion matrix to analyse and process the data in order to further improve the accuracy of the model. To train the deep learning model we must define the model and the loss function and optimise the model. We used a convolutional neural network and a loss function in this design and used a back-propagation algorithm to enhance the training of the model. We tuned the model parameters and hyperparameters during the training process by monitoring changes in the loss function and evaluation metrics, which led to high performance and high precision. Once the model is trained, we must evaluate and deploy the trained model. As part of the evaluation process, we choose a test set to assess the model prediction accuracy and plot and compare the performance between different models across the acc curves and loss values. As part of the deployment process, we use the previously trained models to test the classification of new samples and then assess and output the results.

While the aforementioned work greatly improves the accuracy of fake news detection, in the process of model building there are still the following issues that must be solved. (1) Once the model has been successfully built, we must use the model to detect and discriminate the input text, and there is always an error in the calling process and we can only use it successfully after we have changed the path and added text processing and vector transformations. (2) How do we select the network structure and model parameters? The choice of network structure and model parameters may be subject to some selection and tuning issues. Cross-validation techniques can be used to determine the optimal model hyperparameters, which can split training and validation sets on training data, assess model performance, and choose optimal hyperparameters. At the same time, the network structure needs to be tuned to maximize the model's expressiveness and reduce the possibility of overfitting. (3) What can be done to prevent overfitting of the model? There may be a danger of overfitting when training a model. This happens when the model has good performance on the training data, but may not perform well on the new data. One way to prevent overfitting is to use regularization techniques (e.g., L1 and L2 regularisation), add missing layers, use early stopping techniques, and use data augmentation and cross validation to assess model performance. (4) How do we choose an appropriate parameter initialization method? Selecting the correct parameter initialization method can help the model achieve better performance in the initial stages of training. Attempting to use different initialisation methods to find the most appropriate initialisation for the model. At the same time, in order to prevent problems of gradient blow-up and vanishing, techniques such as gradient shear, gradient normalization, etc. can be used.

5. Conclusion

In this research, we aim to offer a natural language processing-based fake news detecting method. We use deep learning techniques and CNN as a model for our implementation of the system. A comprehensive and detailed description of the system design and experimental methodology can be found in, with the overarching objective of furnishing a robust tool for the detection of fake news across various domains, including education, news, politics, and other industries. The system as a whole covers all aspects of data collection, pre-processing, training, the purpose of this course is to help students understand the basics and how deep learning can be used to detect fake news, as well as to investigate the academic frontiers of the field in greater depth. The results all demonstrate the efficacy of our method, which can provide new insights into the research area of fake news detection. We still need to improve and update the system in the future in order to better adapt to the ever-changing and evolving needs and challenges of society. Through continuous effort and innovation, we believe that we can better address the challenges of fake news and add more value to the society.

Author’s contribution

All the authors contributed equally, and their names were listed in alphabetical order.

References

[1] Sahoo R G B B .Multiple features based approach for automatic fake news detection on social networks using deep learning[J].Applied Soft Computing, 2021, 100(1).

[2] Faustini P H A, Coves T F .Fake news detection in multiple platforms and languages - ScienceDirect[J].Expert Systems with Applications, 2020, 158.DOI:10.1016/j.eswa.2020.113503.

[3] Bharadwaj P, Shao Z .Fake News Detection with Semantic Features and Text Mining[J].International Journal on Natural Language Computing, 2019, 8(3):17-22.DOI:10.5121/ijnlc.2019.8302.

[4] Katsaros, Dimitrios, G. Stavropoulos, and D. Papakostas. "Which machine learning paradigm for fake news detection?." 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI) 0.

[5] Sheikhi, Saeid. "An effective fake news detection method using WOA-xgbTree algorithm and content-based features." Applied Soft Computing 109.2(2021):107559.

[6] Gravanis, Georgios, et al. "Beh9ind the cues: A benchmarking study for fake news detection." Expert Systems with Applications 128.AUG.(2019):201-213.

[7] A, Micha Chora, et al. "Advanced Machine Learning techniques for fake news (online disinformation) detection: A systematic mapping study - ScienceDirect." Applied Soft Computing (2020).

[8] Liang Zhaojun et al Rumor Detection Based on BERT Model and Enhanced Hybrid Neural Networks. Computer Applications and Software (2021)

[9] Ruffo, Giancarlo, et al. "Studying fake news spreading, polarisation dynamics, and manipulation by bots: A tale of networks and language." Computer science review (2023).

[10] Sven Grüner, and Felix Krüger. "Infodemics: Do healthcare professionals detect corona-related false news stories better than students?." PLoS ONE 16.3(2021):e0247517.