Identifying COVID-19 Fake News on Social Networks Using Deep Learning: You will not know what happens next!

PUSHPENDU KAR[1,2], ZHONGYI WANG[1]
[1]School of Computer Science,
University of Nottingham Ningbo China,
199 Taikang East Road, Ningbo 315100,
CHINA

[2]Next Generation Internet of Everything Laboratory,
University of Nottingham Ningbo China,
199 Taikang East Road, Ningbo 315100,
CHINA

Abstract: Fake News has long been an influential source of information on social media, misleading people and causing unpredictable consequences, especially in the COVID-19 era when the spread of Fake News has been amplified. There is a need to develop a platform that can detect Fake News on social platforms. With the development of Natural Language Processing (NLP) and Deep Learning, the detection of misleading information is becoming a reality. In this paper, we propose a feasible optimization scheme that combines and optimizes NLP and Deep Learning to improve the accuracy of Fake News detection. Our model achieves a hit rate of up to 95.3% compared to state-of-the-art techniques. In the proposed system, a GUI-based interface is also designed and developed to facilitate news detection.

Key-Words: Covid-19; Fake News; NLP; Deep Learning

Received: March 11, 2024. Revised: August 6, 2024. Accepted: September 9, 2024. Published: October 10, 2024.

1 Introduction

The issue of Fake News has gained significant attention due to its potential impact on political decision-making, as exemplified by its suspected involvement in the outcome of the Brexit referendum and the 2016 U.S. presidential election. Consequently, the academic community, particularly within computer science and linguistics, has shown an increasing interest in exploring this phenomenon, [1], [2], [3].

The rapid rise of Fake News can be attributed to several factors. Firstly, the widespread use of the Internet, with its ease of access, low cost, and fast dissemination of information, has led people to consume news on social media. Additionally, social media platforms have become a convenient vehicle for the widespread and swift propagation of Fake News, which often receives more attention and spreads faster than mainstream news, [4].

Moreover, the proliferation of Fake News poses various risks, including social, economic, industrial, and health risks. For instance, the infamous incident where an armed man entered a Washington pizza restaurant in December 2016 was fueled by Fake News. Right-wing blogs and social media stories falsely claimed that the restaurant was the center of an underground child pornography ring run by Hillary Clinton and John Podesta. While no one was hurt in the incident, the negative impact of such stories spreading across social media platforms like Facebook, Reddit, and Twitter is unpredictable, [5]. This highlights one of the many reasons why Fake News has become a pressing concern for the public.

However, before the advent of computers and the internet, Fake News (also known as deceptive journalism) was shared orally in the form of oral rumors (face-to-face) or yellow/sensationalist stories, or naively talking about the lives of others or intentionally damaging the reputation of others or competitors, [6]. Since people are more inclined to believe misleading information and spread false content that lacks control, detecting Fake News is considered a complex task, even though it is not a new problem. Even if a person has an extensive knowledge of the topics covered by Fake News, it can be argued that it is very difficult for a person to accurately detect Fake News and successfully identify the truth or falsity of the information in an article. From this, it can be deduced that teaching an automated system to automatically detect Fake News is a daunting task, [7].

Although academics and experts have done much research on the destructive and deceptive nature of "Fake News", e.g., that Fake News creates fear in the real world, [4], [8], [9], little attention has been paid to the analysis and definition of "Fake News", [10], [11], [12]. Before addressing the classification and description of Fake News, we should address its components --- "Fake" and "News." News is generally considered accurate and is an accurate description of real events, [13], while the word "fake" is often used in conjunction with the words "copy," "fake," and "untrue," regardless of order, [14]. According to the definition of each term, the news must be accurate, which makes the term "Fake News" an oxymoron, [15]. Therefore, the previous method of combining "fake" and "news" separately to form the definition seems to be biased and not strict enough, so a new description should be adopted in this area.

Looking back at the previous studies, we find that the Fake News is defined as follows: 1. rumor; 2. imitation; 3. manipulation; 4. biased; 5. advertisement; 6. propaganda; 7. prejudice, [15]. In this research, we would like to deny two definitions before. First, the satirical news, which has been defined as Fake News in previous studies due to its formal falsehood. However, satire is usually entertaining and reveals its deception to readers, [16], [17], [18]. Basically, they are synonymous with notification and entertainment, because actual events are the basis of the core content of satire; it is only false in form and not in content. Another form is biased news, which is biased or misleading, but not completely false, [19]. Biased articles often have a strong emotional tinge from the author that can make readers feel their position, for example political bias, but the content is often true.

The remaining five are what we consider the definitions of Fake News. What they have in common is that they have some form of credibility in that they try to pass themselves off as real news, but their content is consistently untrue, [20], [21]. Imitation and rumored news form fictional stories in a broad social context, while the author of fabrication and manipulation intentionally misleads from the start without making a disclaimer. Moreover, the ultimate goal is to fabricate and manipulate, either to mislead people or to gain clicks for advertising money, to fool people into believing that the news they have seen is real. By this definition, Fake News has a long history, from the telegraph in the 19th century to modern social media, which opens up new opportunities for deception and fabrication, [10], [22], [23].

1.1 Motivation

At the same time we are now in a dangerous period. The COVID-19 outbreak that began in Wuhan, China in December 2019 is still ongoing. Although COVID-19 is now under control, misinformation related to COVID-19 is rapidly increasing, causing severe social disruption. The fake drug of COVID-19 poses a serious threat to people's lives, and the widespread misinformation is disrupting social order. We do not know how long COVID-19 will exist. Therefore, the development of software to detect Fake News about COVID -19 is especially important at this time.

An overview of previous research can be divided into the following categories: linguistic methods, data representation, systemic functional grammar, semantic analysis, rhetorical analysis and discourse analysis, classifiers, linked data and social networks, [24]. However, as for the database, the number of articles they used for machine learning is not very large and it is a static software, which may be because there is not much Fake News can be used in the real world. Also, they focus on a specific area, such as the Mistext database, [1]. If you enter a phrase that refers to population, the answer to it will be false.

1.2 Contributions

Since the in-depth use and study of traditional NLP practices, it has become apparent that traditional NLP is flawed. For example, all decimal numbers such as 3.89 are split into two different words: 3 and 89. Traditional NLP removes all symbols, while special symbols have their own meaning, such as "$", "%", "°C", an approach that is clearly undesirable.

We outline the general contributions of this work as follows:

- Identify undesirable aspects of traditional NLP.
- We propose two possible NLP optimization solutions.
- Compare the performance of traditional NLP and optimized NLP

1.3 Organization

The rest of the paper is organized as follows. Section II describes the work that has already been done in this area. Section III describes the technology used in this paper. Section IV describes how the NLP and Deep Learning word is made. Section V describes how the model is designed and shows the evaluation of the model. Finally, we conclude our work in Section VI with some advice for future work.

2 Related Work

Fake News has been around for a long time, and although it has only received widespread attention since 2016, many methods for detecting Fake News already exist on the Internet. In this section, we briefly introduce the existing methods and analyze their respective strengths and weaknesses. Based on previous research, Fake News detection methods can be broadly divided into the following three categories: Approaches based on content, language, and the Internet.

2.1 Approaches based on content

In the 'Writing Style' approach, [25], the same author may have a different writing style when describing real data and false data, e.g., frequency of adjectives, deterministic words, etc.. To analyse writing style, the Wmatrix corpus analysis method should be used, which is commonly used for corpus comparability, [26]. Wmatrix is a tool that provides standard corpus linguistic analysis, including word frequency lists and an analysis of major grammatical categories and semantic domains. The drawback of this approach is that it requires the use of a third-party tool to analyse and compare a large number of writing books, and without a comprehensive analysis, it is not possible to accurately determine whether the article or comment is true or false.

For the 'Emotional Vocabular' approach, [27], liars would like to use more words of negative emotions, few self-references and other references, they have less cognitive complexity compared to those who tell the truth. They may exhibit more negative emotions due to unconscious anxiety outbursts.

For the 'Emotional Vocabulary' approach, [27], liars would like to use more words of negative emotions, less self-references and other references, they have less cognitive complexity compared to those who tell the truth. They may show more negative emotions due to unconscious anxiety outbursts.

In the 'Manually produced features' approach, [28], the existing works are based on manually created features such as features based on the message, user, topic and distribution. These features developed by machine learning algorithms are tedious and require arduous manual work. These extracted features are then put into conventional supervised classification models such as Bayesian, decision tree, logistic regression, k-nearest neighbor (KNN) and Support Vector Machine (SVM), where the best classifier is selected as a result of the experiment.

Jing proposes a model based on recurrent neural networks (RNNs), [29]. Experimental results show that the RNN-based approach is more efficient than existing techniques, including leading Internet noise reduction services. The disadvantages are obvious: it is more time-dependent on the number of generated real texts, and the longer the microblogging event generation time, the higher the accuracy rate.

In the Capture, Scoring, Integration (CSI) model approach, [30], the model consists of three modules: capture, scoring, and integration. The first module uses a recurrent neural network to capture temporal patterns of user activity on a given article, based on responses and text. The second module learns sourcing features based on user behavior. The previous two modules are then integrated with a third module to separate goods into fakes and real. The model CSI is more accurately based on experimental analysis of real-world data than existing models.

For the 'Posture detection' approach, [31], position detection means that the machine automatically decides based on the content of the text, whether the author of the text is for, against or neutral to the written statement based on the text content. Targets can be individuals, organizations, government policies, movements, products, etc. For example, a speech by Barack Obama can be used to infer that he favors stricter gun laws in the United States.

2.2 Approaches based on Language

The 'Word Bag' method is the simplest way of representing text, where each word has the same status. In this method, the frequencies of single words and multiple words are aggregated and analyzed to determine the difference between truth and falsehood, [32]. The simplicity of this representation of "n-grams", which are analyzed to obtain features that are then used to figure out which classifier works best, leads to its biggest drawback: these n-grams are always out of sync with useful contextual information, [24]. The simplicity of this representation also leads to the biggest drawback of the "Word Bag" approach: it is completely language-dependent and relies on isolated n-grams, which is always out of sync with useful contextual information. There is no solution to the ambiguity of the meaning of words. However, this method is often shared with other analysis methods and can effectively improve accuracy.

'Deep Grammars', the analysis of word usage, is usually not sufficient to predict deception. Deep grammar analysis is achieved by probabilistic context-free grammar (PCFG). These sentences are often borrowed from third-party tools such as Stanford Parser, [33], AutoSlog- TS, [34], Grammatical Analyzer etc. to automatically split the grammatical structure and replace each word with its lexical counterpart, e.g. noun, verb etc. The method has an accuracy of 85%-91% when used to distinguish between rule categories, depending on the rule classification used. Syntactic analysis alone is not sufficient to identify spoofing. Therefore, this method is often combined with network analysis techniques, [35].

In the 'Semantic analysis' approach, as an alternative to deception tips, authenticity signals are analysed and obtained by characterising the degree of compatibility between personal experiences and content profiles obtained from similar datasets. The method extends the n-gram plus grammar model by combining profile compatibility features and shows that the additional functionality significantly improves classification performance. The content extracted from the keyword consists of a pair of attribute:descriptor. Then by adjusting the profile and the description of the author's personal experience, the authenticity assessment is a function of the compatibility score: 1. Compatibility with certain unique aspects (e.g., an art museum near the hotel). 2. compatibility with descriptions of certain general aspects, such as location or services. The accuracy of predicting errors using this method is about 91%, [36]. The approach has been limited to the application domain. There are two potential limitations with this approach: first, the ability to determine the match between attributes and descriptors depends on having enough summary files to mine for content. Another one is the challenge of correctly linking the descriptors to the extracted attributes.

2.3 Approaches based on the Internet

'Social networking behaviour', the willingness to use social networking services based on one's needs, social influences, and social networking technology, and the resulting sum of various usage activities, such as the number of tweets retweeted, etc, [24].

For the 'source reliability and trustworthiness' approach, generally news from fact-checking websites, Such as at a time when fake images were flooding the internet, it was estimated that 86% of tweets were retweeted instead of being original. One experiment showed that out of 10,215 users, the top 30 users (0.3%) resulted in 90% of fake images being retweeted, [37].

Another approach is called 'crowdsourcing approach': Crowdsourcing-oriented approaches use 'popular intelligence' to enable ordinary people to comment on news content, [38]. Facebook introduced tools in 2017 that allow users to flag Fake News, [39].

2.4 Synthesis

These are most of the existing methods. In our opinion, Fake News detection must be based on textual content. Otherwise, the accuracy of machine detection can be easily compromised if there are issues such as fact bias, subject substitution, and reason confusion.

3 Description of the work methodology

3.1 Three ways to represent text by numbers

3.1.1 One-hot encoding

The simplest way is to give each word in the sentence a "unique heat". For example, suppose we have a sentence: "The cat sat on the mat". Following the principle of deduplication of sets, the only words in this sentence are (the, cat, sat, on, mat). Then, for each word, a null vector is created whose length is equal to the set. The index corresponding to this word is given the value 1 and the remaining words are given the value 0. The Fig. 1 shows this approach.

One-hot encoding

	cat	mat	on	sat	the
the =>	0	0	0	0	1
cat =>	1	0	0	0	0
sat =>	0	0	0	1	0

Figure 1: One-hot encoding example

To create a vector that contains the sentence encoding, we can concatenate a unique heat vector for each word. The drawback of this approach is that it is inefficient. A unique heat encoding vector is very sparse (meaning that most indices are zero). For example, if we have a vocabulary of 10,000 words, a vector is created where 99.99% of the elements are zero to uniquely encode each word.

3.1.2 Encode each word with a unique number

this assigns a unique numerical value to each keywords and forms a numerical vector. Continuing with the example above, we can assign 1 to "the", a 2 to "the cat", and so on. The sentence "The cat sat on the mat" can be encoded as a dense vector, e.g. [1,2,3,4,1,5]. Now we have a dense vector where all elements are full, instead of a sparse vector. So we can say that this approach is efficient. However, there are two drawbacks to this approach. The first one is that any relationship between words cannot be captured. The other disadvantage is that the integer encoding is quite difficult for the model to interpret.

3.1.3 Word Embedding Vectors

The word embedding vector approach provides us with a way to use efficient, dense representations in which similar words have similar encodings. If we can visualize the vector, we can see that the similar words are very close to each other. Most importantly, we do not have to manually set this encoding, but it can more accurately represent the connections between words. The embedding vectors are dense vectors with floating point numbers. The length of the vector is a parameter to be set by the user. These are trainable parameters and it is not necessary to manually set the values of the embedding vector. For small datasets, word embedding vectors with 8 dimensions are common, and when working with large datasets, word embedding vectors with up to 1024 dimensions are possible. Assuming we have enough data to train the vectors, we can have a higher dimensional embedding vector that can capture the granular relationships between words. The Fig. 2 shows an example of a word embedding vector with 4 dimensions.

A 4-dimensional embedding

cat =>	1.2	-0.1	4.3	3.2
mat =>	0.4	2.5	-0.9	0.5
on =>	2.1	0.3	0.1	0.4

Figure 2: a 4-dimensional embedding example

The third method requires too much hardware, as it is limited by hardware. Therefore, we chose the second method as the basis for our Deep Learning vocabulary.

3.2 Mathematical Model of Deep Learning

Deep Learning uses neural network principles with a large number of parameters called layers in various architectures. Our model consists of a set of fully linked layers, and the main method for changing the weights is backpropagation, as shown in Fig. 3. First, the weights in the model are initialized randomly, then the outputs are computed for each input training example.

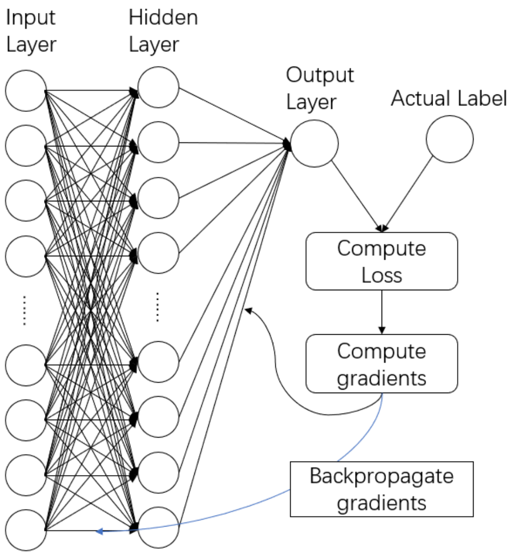


Figure 3: Backpropagate Gradients

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial W_{kj}} = -\eta \delta_k y_i \qquad (1)$$

where $\delta_k = \frac{\partial E}{\partial o_k} \frac{\partial \sigma(net_k)}{\partial net_k}$

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial W_{ji}} = -\eta \delta_j x_i \qquad (2)$$

where $\delta_j = \sum_{k=1}^{k} (\delta_k W_{kj}) \frac{\partial \sigma(net_j)}{\partial net_j}$

Next, compute the errors of the output and hidden neurons and the update rule for the output and hidden layer weights. See equation 1 and 2. Then compute the total weights for all training examples. Once all training examples have been presented to the network, update the weights. See equation 3.

$$W_i = W_i + \Delta W_i \qquad (3)$$

Repeat the above processes until all training sets have been trained and the number of epochs has been reached. This completes the entire process of training the model.

4 System Model

This section describes the process which includes data acquisition, NLP preprocessing of data and using Deep Learning. All the steps are shown in Fig. 4.

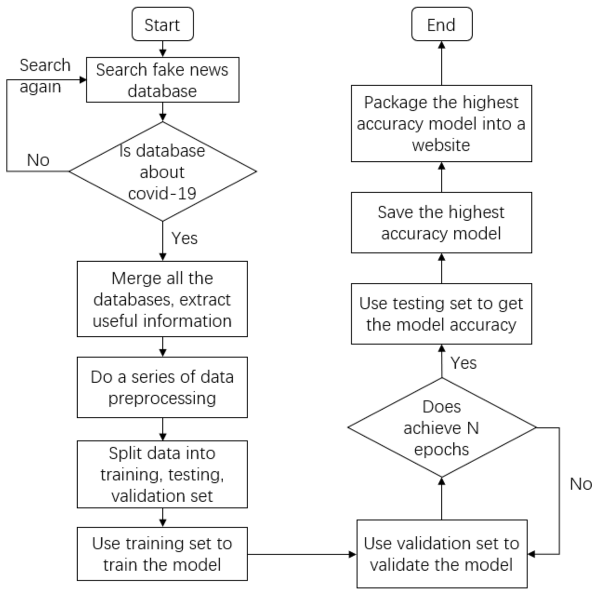


Figure 4: Flowchart

4.1 Data collection

A structured dataset is easier to manage than unstructured (or semi-structured) (e.g. text). If we know the linguistic domain, we can make better predictions about the nature and use of deception. This article is mainly concerned with Fake News in the period COVID -19. After comparing different databases available on the Internet, we have chosen the following three databases:

4.1.1 Susan Li's dataset

. First of all, this dataset comes from https://towardsdatascience.com/automatically-detect-COVID-19-misinformation-f7ceca1dc1c7. It is a dataset about COVID-19 published in a blog by towards data science and it comes from a variety of sources. Most of the real news came from the Centers for Disease Control and Prevention of America, the World Health Organization, the New York Times, Harvard Health Publishing, Bloomberg School of Public Health, Johns Hopkins, and so on. The fake ones were collected from a far-right website called Natural News, an alternative medicine website called orthomolecular.org, and Facebook posts. The ratio of true to Fake News in this dataset was perfect: 1,164 articles, of which 586 were true and 578 were fake. Although the authors themselves said they did not know if selection bias had occurred in the collection of the data, the articles in which this dataset was found were not authoritative beyond that. When we received the dataset, we randomly selected 30 articles, manually searched the web, and compared the tags. Finally, we determined that the tags of the randomly selected articles were correct, so we used this dataset.

4.1.2 ReCOVery dataset, [40]

There are two files in this dataset, one containing all the news and the other containing all social media records. However, due to Twitter's privacy policy, all data must be approved by Twitter before it can be downloaded, so we decided to use only the file containing all messages. Both true and Fake News are taken from some well-known American newspapers and media outlets: usatoday, theverge, nytimes, cbsnews, etc. This file contains 13 columns, namely news id - the unique identifier of each news article; url - the URL of each article on the website where each news item is published; publisher - the publisher of the news article; publication date - the date the news article was published; author - the author of the news article title - the title of the news article. image - the header image of the news article. body - the complete content of the news article. newsguard score - the news source's rating by newsguard. mbfc level - the media influence/fact check level of each news source. Political bias - the political bias of each news source. Country - the country of the news source. Reliability - the reliability label of the news article (1 for reliable, 0 for unreliable). It contains all the information we need, so this dataset is used.

4.1.3 coAID dataset, [41]

CoAID (Covid-19 heAlthcare mIsinformation Dataset) is a diverse dataset on COVID-19 healthcare misinformation, including Fake News on websites and social platforms and users' social engagement with these news. The dataset includes 5,216 news items, 296,752 associated user interactions, 958 posts on social platforms about COVID-19, and ground truth tags. The genuine articles are largely from WHO and a website with daily updates on global research processes called Sciencedaily, a web-based source of medical news aimed at both doctors and the general public called Medical News Today. The fake information comes from two fact-checking websites, Politifact and Leadstories, and a health-related website called Healthline. The dataset consists of 4 datasets from different months and in each dataset we only downloaded all the news due to Twitter's privacy policy.

4.1.4 Processing three datasets

When we collected the three datasets, there were inevitably some duplicate articles in these datasets. Therefore, we primarily extracted the titles, contents, and tags of all articles and combined them into a data frame structure, which was de-duplicated using the data frame's own de-duplication method. The total number of real messages was 4611, and the total number of fake messages was 1491. Then we obtained a database with a ratio of real to fake data of 3:1, which is not consistent with the data requirements of the "relu" activator. So we limited the length of the real messages. Messages with character length from 490 to 5000 are accepted. Now we can get 2492 real messages and 1491 fake messages. So we have two options: one is to randomly sample the real messages and the other is to replicate the fake messages, i.e. replicate 1000 of these messages to change the ratio of real to fake messages to 1:1. Considering the amount of messages, we chose the second option. At the same time, we had to make sure that the data in the test set was outside the replication. After the computation, we first extract 491 true messages and 491 fake messages for the test set before the replication process. The copy operation is performed only with data that is not used as a test set.

4.2 Text Processing with NLP

Pre-processing of data is the most important aspect before using machine learning and has the great advantage of improving the generality of the model. The data for any machine learning application is collected through various "sensors" which can be software programs (e.g., web crawlers, manual surveys, etc.), physical devices, and instruments. Noise and error messages due to hardware failures, software failures, instrument failures, human errors, etc. can creep in and seriously affect the performance of the model. In addition, there is also redundant information that needs to be removed. For example, a person's age is irrelevant to predicting whether or not it will rain tomorrow. Our project is similar to a text classifier in that there are several stop words that may be redundant and need to be analyzed. Also, since the publicly available database is obtained directly from the Internet, there may be some outliers in the data that need to be removed to improve the performance of the classifier. These are some of the reasons why it is important to process the data before machine learning.

In our system, we use the "NLTK" library, which is a collection of open-source Python modules that allow programs to process human speech data. It deals with tagging, parsing, and named entity identification, as well as many other features. In addition to some traditional NLP operations, we proposed two possible operations that can improve the accuracy of machine learning. First, the improvement of NLP in this paper mainly focuses on handling the decimal system - in normal NLP, all decimal numbers such as 3.89 are split into two different words: 3 and 89, so our improvement is to keep the original meaning of decimal numbers in NLP and Deep Learning. Another improvement is that some special characters such as "$", "%", "°C", etc. are converted to English and are not lost in the NLP phase, which changes the meaning of the sentences.

We also included headline of the articles into the content. The reason for this is that we believe that the headline is the most important part of the article. It is a short text that summarizes or evaluates the content of the news and serves to organize, order, reveal and evaluate the content of the news. Also, some articles include keywords in the title, so combining the title with the content of the article makes it easier to see the importance of the article. Of course, one must be aware that some article titles are missing from the database - in other words, one must be aware of the operation of the missing values. Also, a space should be inserted in the combination of title and content to ensure that the words are not mixed together. The concrete steps to process stored Fake News are as follows.

4.3 Check the correctness of the words

Since each word has to be used as a variable in the vector in the Deep Learning phase, it is necessary to ensure the correctness of each word. The presence of Fake News in the database, as well as the fact that Fake News does not assume the correctness of words, often leads to the appearance of words that are incorrect but whose meaning is known to people, such as 'telll', which we know at first glance means 'tell', but which is itself incorrect. A corpus was downloaded from the internet. The corpus itself contains a variety of English words, and we counted the number of each word in the corpus and added the words 'covid' and 'wuhan', which were necessary because the corpus we downloaded had been edited so early that these two words did not yet occur in English. Then, the best match for the word is returned according to the rules we defined: 1. The priority is based on the edit distance of the word: 0 distance is first, 1 is second, and 2 is last. 2. if there are several matching words with the same edit distance, the word with the highest frequency is returned, according to the frequency of the matching words in the corpus. The edit distance here means: 0 distance stands for the word itself, stands for a collection of words formed by the following operations: Deleting a character, e.g. cat→at; exchanging the position of two characters, e.g. cat→tac; replacing a character with one of 26 characters, e.g. cat→caw; inserting one of 26 characters, e.g. cat→cats. 3. then represents the set of words with an edit distance of 1 to the word edit distance of 1, i.e., in the set of edit distance of 1, and again for each word in the set to compute the set of words with an edit distance of 1, and finally return a large set of words. The partition equation is particularly important when computing edit distances, for example cat can be split into these four pairs: ('','cat'), ('c','at'), ('ca','t'), ('cat',''), and it saves a lot of time to calculate edit distances on this basis.

4.3.1 Preserve the decimal point

Since in English the decimal point and the symbol for the end of a sentence are identical, decimal numbers are often split into two integers in regular NLP operations. To retain the decimal point, we first find all decimal places in a sentence by determining whether the decimal point is a number to the left or right of it. If it is, we keep it, and if it is not, we replace where the full stop is with a space. The reason you do not just delete the periods, but replace them with spaces, is so that the words at the end of the previous sentence and the words at the beginning of the next sentence do not combine.

4.3.2 Convert acronyms to complete words

For example: what's → what is. Since 'is' is a stop word here, "s' can also be omitted as meaning 'is'. All the cases we can think of are {' 'll':' will', ' 's':' is', ' 'm':' am', ' 've':' have', 'n't':' not'}, enter them in a dictionary and substitute all the abbreviated cases in the sentence. There are some special cases where abbreviations have rounded and semicircular corners, so all cases should be included in this dictionary.

4.3.3 Deleting Stop Words from Sentences

A stop word may have various definitions. We can consider a stop word as a word that has a high frequency in the corpus. Alternatively, we can consider a stop word as a word that has no real meaning in the given contexts. These words can be removed without negatively affecting the final model we train. The meaning of a word can vary depending on the dataset and also depends on what you are trying to achieve.

Sentiment analysis is more sensitive to word deactivation than document classification. Suppose we have a sentence "I told you she was unhappy". Let us use the Aruana library, [42], to remove the deactivated words: the result is ['told', 'happy']. For the purposes of sentiment analysis, the overall meaning of the resulting sentence is 'happy', which is not correct at all. Another example is the famous Shakespearean sentence "To be, or not to be", which no longer exists after the deactivation operation. Nevertheless, removing deactivated words offers several advantages. The size of the dataset is reduced and the time to train the model is reduced after removing the deactivation words. Removing stop words can improve performance by leaving fewer meaningful tokens. Therefore, it can improve the classification accuracy. However, since our dataset focuses on covid-19 and it is a text classification project, the above drawbacks are largely absent for us and removing stop words is a wise decision since the benefits outweigh the drawbacks. It's worth noting that when detecting whether a word is disabled or not, by the way, all words are lowercased, which saves a lot of time even though they are all constant.

4.3.4 Translate some important characters

Translate special characters like '%', '$' into English words to make sure the meaning of the sentence is not lost. A typical example is that we have 9% of people supporting us. After you delete the '%', only 9 people support us, which would obviously change the meaning of the sentence, so it is a wise decision to keep the meaning of these characters before deleting them.

4.3.5 Delete extra punctuation and unnecessary spaces

In some cases, more than one space becomes one. These steps are to set the stage for the next Deep Learning, separate each word in each sentence with a space, and get the data in Deep Learning, which can be done in just one step to get good data.

4.4 Delete the suffix of a word

Among the benefits, of course, is the shortening of the lexical space and thus a significant improvement in the size of the index (or feature space). Stemming based solely on dictionaries or rules (e.g. Porter Stemmer) is very fast. Stemming involves converting the words in a sentence into their invariant parts. In the above examples of amusing, amused, and amused, the stemmer is amus. The Algorithms 1 and 2 show the preprocess technique of words and sentences in the proposed method.

At this point, all the operations of NLP are complete. Let us now turn to Deep Learning.

Algorithm 1 correctword

Input: a word
Output: set of processing words

Begin
$deletes \leftarrow set(delete\_one\_character)$
$transposes \leftarrow set(change\_two\_positions)$
$replaces \leftarrow set(replace\_one\_character)$
$inserts \leftarrow set(insert\_one\_character)$
return $deletes + transposes + replaces + inserts$
end

Algorithm 2 preprocess

Input: a sentence
Output: a sentence after preprocessing

Begin
for all word in sentence **do**
  if word not in corpus **then**
    $distance_1 \leftarrow$ CORRECTWORD($word$)
    for word in $distance_1$ **do**
      $distance_1 \leftarrow$ CORRECTWORD($word$)
      $distance_2 \leftarrow distance_2 + distance_1$
    end for
  end if
  if one_word in $distance_1$ in corpus **then**
    $word \leftarrow one\_word$
  else
    if one_word in $distance_2$ in corpus **then**
      $word \leftarrow one\_word$
    else
      $word \leftarrow empty$
    end if
  end if
  if $wordinstopwords$ **then**
    $word \leftarrow empty$
  end if
  if $wordisspecialsymbols$ **then**
    $replacesymbolbyEnglish$
  end if
  $Delete\_all\_punctuation\_and\_whitespace$
  $word \leftarrow stem\_of\_word$
  $after\_process \leftarrow after\_process + word$
end for
return $after\_process$
end

4.5 Deep Learning

4.5.1 Create an exclusive vocabulary

This vocabulary does not have to contain all English words, because a vocabulary with all English words would be too large and would reduce the efficiency of machine learning. Also, the order of the words in the vocabulary does not affect the efficiency of machine learning, so at this point it is sufficient to create a vocabulary that includes all the data from training, testing, and validation.

First, an empty set is predefined, then each sentence is separated into multiple words by spaces, and the vocabulary is updated using the update() method in the set. Since this is a small change from the traditional NLP and ML process, this step needs to determine if the words are floating point numbers, and if so, they are stored in an array called "reserved_tokens" for the next step.

4.5.2 Encode text to numbers

First, set a Tokenizer that does not remove the decimal point from the floating point number stored in the previous step, and pass reserved_tokens as a parameter to get it. Second, an encoder is set up to recognise only the words in the exclusive vocabulary and passed to the Tokenizer as the basis for segmenting the string. All data should be encoded by this encoder.

```
In [3]: sentence = "You just need to add water,
and the drugs and vaccines are ready to be
administered. There are two parts to the kit: one
holds pellets containing the chemical machinery
that synthesises the end product, and the other
holds pellets containing instructions that telll
the drug which compound to create. Mix two parts
together in a chosen combination, add water, and
the treatment is ready."

In [4]: sentence = all_steps(sentence)
   ...: print(sentence)
you need add water drug vaccines ready administer
there two part kit one hold pellet contain
chemical machinery synthesis end product hold
pellet contain instruction tell drug compound
create mix two part together choose combination
add water treatment ready

In [5]: sentence = encoder.encode(sentence)
   ...: print(sentence)
[23599, 15291, 25367, 24352, 20356, 6907, 10120,
29016, 12791, 12086, 23003, 9890, 8435, 28131,
1677, 2695, 5480, 20894, 28325, 21381, 7001,
28131, 1677, 2695, 10545, 21856, 20356, 12797,
10384, 29353, 12086, 23003, 14682, 6313, 14662,
25367, 24352, 21209, 10120]
```

Figure 5: Encoding text into integers

Suppose we have a set: "You just need to add water, and the drugs and vaccines are ready to be administered." The set has two parts: one part contains the chemical machinery that synthesises the final product, and the other part contains the instructions that tell the drug what compound to make. Mix the two parts together in the desired combination, add water, and the treatment is ready. After the above NLP operations and coding, the result is shown in the Fig. 5.

Table 1: Parameters of Model

Parameter	Value
Number of records in the training set	2946
Number of records in the testing and validation set	982
Number of LSTM layers	2
Number of Dense layers	4, one for output layer
optimizer	Adam
loss	binary_crossentropy
epoches	10

4.5.3 Split into training, testing, and validation sets

Before splitting the training, testing and validation set, we first disorganised all the encrypted data. The reason for this is that we do not have a large amount of data for machine learning, no more than 10,000 data, so the training, testing and validation sets are split in a ratio of 3:1:1. That is, the first 60% of the coded data is the training set, the next 20% is the test set, and the rest is the validation set. There is no guarantee that there are large, contiguous chunks of identically labelled data in our data. And if there are, without the step of randomly interrupting the data after splitting it into training, test, and validation sets, it might look like the ratio of true to false labels in the training set is 2:1 rather than 1:1. This would give the model a better chance of guessing true rather than false when it comes to the final test. In other words, if the ratio of labels in the training set is unbalanced, the model is more likely to guess the one with more labels, which greatly affects the accuracy of the model. Fig. 6 shows the structure of the model.

5 Performance evaluation

5.1 Model Design

In this section, we present the evaluation of the model. In this model, we choose a 'Sequential' model as the base model and start with the embedding vector layer. The embedding vector layer creates and stores a vector for each word. When called, it converts the sequence of word indexes into a sequence of vectors. These vectors are trainable. After training with enough data, words with similar meanings

```
In [5]: reconstructed_model.summary()
Model: "sequential_9"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_9 (Embedding)      (None, None, 64)          1978880
_________________________________________________________________
bidirectional_18 (Bidirectio (None, None, 128)         66048
_________________________________________________________________
bidirectional_19 (Bidirectio (None, 64)                41216
_________________________________________________________________
dense_36 (Dense)             (None, 64)                4160
_________________________________________________________________
dense_37 (Dense)             (None, 64)                4160
_________________________________________________________________
dense_38 (Dense)             (None, 64)                4160
_________________________________________________________________
dense_39 (Dense)             (None, 2)                 130
=================================================================
Total params: 2,098,754
Trainable params: 2,098,754
Non-trainable params: 0
_________________________________________________________________
```

Figure 6: Model structure

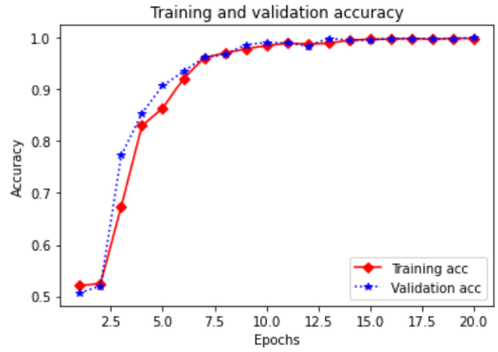
tend to have similar vectors. This index-finding approach is much more efficient than the equivalent operation of passing a unique hot-coded vector through a "dense" layer.

Since all layers of the model have only one input and produce one output, here we chose the sequential model of Keras, [43]. Then we stacked two layers of LSTMs, here two LSTM layers. The main reason for this is the greater complexity of the model and the smaller volume of data. Also, stacking LSTMs allows for better accuracy.

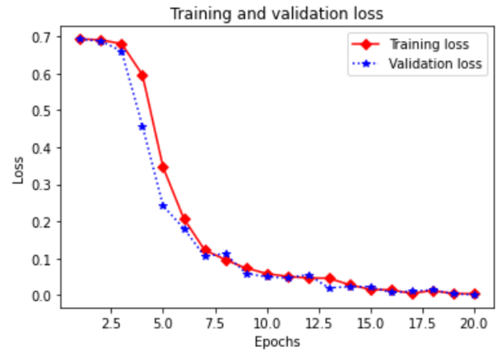
The fixed-length output vector is transferred through three fully connected (Dense) layers with 64 cells of the hidden layer using "relu" activation. The last Dense layer is connected to a single output node using a sigmoid activation function that has a floating value between 0 and 1 indicating the probability or confidence level. In addition, the model requires a loss function and an optimizer after each epoch of training. Since the model outputs probability values and it is a binary classification problem, we decide to use the binary cross entropy loss function. The binary cross entropy measures the "distance" between probability distributions and is therefore more suitable for dealing with probabilities. The Table 1 shows the parameters of this model.

5.2 Model Evaluation

At the beginning of each epoch, we use the training set to train the model. After that, we use the validation set to validate the trained model and get better accuracy. At the end of all epochs, we use the test set to get the final result. The reason we use the validation set and not the test set is because we use the training data only to develop and tune the model and use the test data only once to evaluate the accuracy.



(a) Change of Training and Validation accuracy over 20 epochs

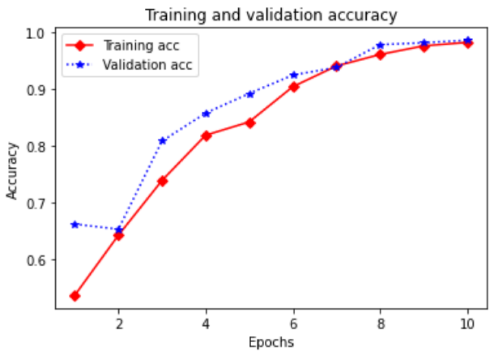


(b) Change of Training and Validation loss over 20 epochs

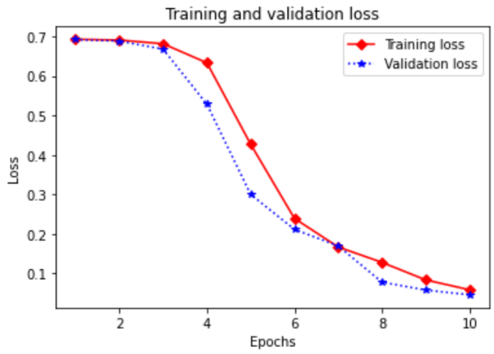
Figure 7: 20 epochs training accuracy and loss overtime

Set the training iterations to 20 epochs at the beginning. The accuracy and loss of training and validation period are shown in Fig. 7(a) and Fig. 7(b). In Fig. 7, the red solid line with diamonds represents the validation loss values and accuracy, while the blue dotted line with stars represents the training loss values and accuracy. We can see that the training loss value decreases while the training accuracy increases in each epoch. This is reasonable because we use gradient descent optimization for our deep learning model and it is logical to minimise the expected value in each iteration. The model performs better on training data than on previously unseen data. After that, the model is over-optimized and learns a representation specific to the training data and is not able to generalise to the test data. However, this is not the case for the loss values and the accuracy of the validation process - they seem to peak after 10 epochs. In this particular case, we can stop training after about 10 epochs to avoid an overfitting problem. Fig. 8(a) and Fig. 8(b). show the variation of accuracy and loss over time for the best case when training with accuracy up to 95.3%.

For comparison, we proposed two other methods for the same metadata and model: one that does nothing to the data and trains it directly, and another that

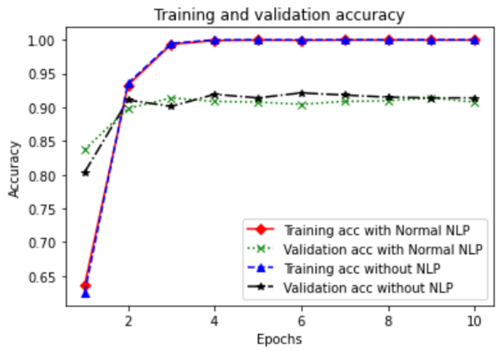


(a) Change of Training and Validation accuracy in the best model

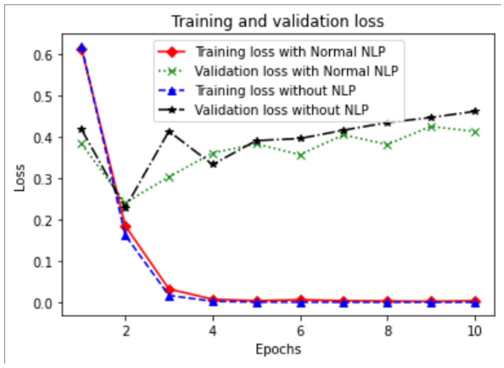


(b) Change of Training and Validation loss in the best model

Figure 8: 10 epochs training accuracy and loss overtime



(a) Change of Training and Validation accuracy in other two approaches



(b) Change of Training and Validation loss in other two approaches

Figure 9: 10 epochs training accuracy and loss overtime in other two approaches

performs traditional NLP processing on the data and trains it. For these two approaches, set the training iterations to 10 epochs at the beginning. In Fig. 9, the red solid line with diamonds represents the validation loss values and accuracy with Normal NLP, while the blue dotted line with triangles represents the validation loss values and accuracy without NLP; the green dotted line with forks represents the training loss values and accuracy with Normal NLP, while the black dotted line with stars represents the training loss values and accuracy without NLP. They seem to peak after 5 epochs, which is mainly due to the overfitting problem.

Then we change the number of epochs to 5 and present the results in Fig. 10. We can see that for the same text, there is little difference in training loss and validation accuracy between normal NLP processing and unprocessed training for this text.

Next, we take 10 epochs for our model and 5 epochs for the other two approaches. We run each of these approaches 10 times and present the results in Fig. 11. It is easy to see that the accuracy rates after traditional NLP operations are on average 1% higher than those without NLP. At the same time, the average increase in accuracy after better NLP is 1.5% - 2% and can reach 95.3%. Therefore, we can conclude that the more fine-tuned the NLP, the better the accuracy.

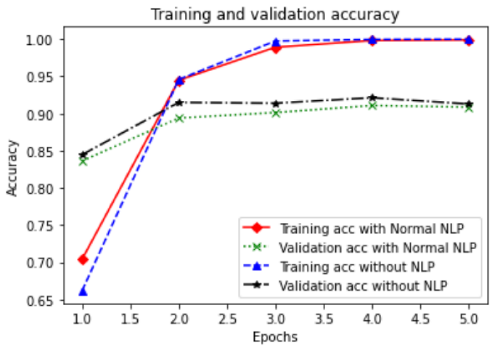
5.3 GUI

This GUI is a user-interactive interface where the user can enter the article they want to test and the model returns a true or false label. The whole GUI interface is shown in the Fig. 12. The specific working principles are as follows: enter the article in the text box which user wants to test, click on test and the user gets True, False, or Ambiguous answer and the confidence. If the difference of true and false confidence is less than 0.1, then this should be considered as ambiguous.

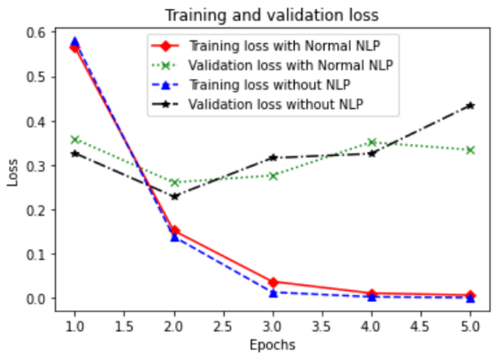
If the user feels there is a problem with the output, they can click on the 'Seems wrong' button and enter 'Name', 'email address', 'Select one label of what he/she think of the entered article is', 'Article he/she tested', and 'Comment'. Thereafter, user can submit the feedback to administrator for further investigation.

6 Conclusion

In this paper, we proposed two possible improvements for the NLP phase, worked on the NLP in more



(a) Change of Training and Validation accuracy in other two approaches



(b) Change of Training and Validation loss in other two approaches

Figure 10: 5 epochs training accuracy and loss overtime in other two approaches

detail, and worked on the Deep Learning phase so as not to destroy the data from the NLP phase. Finally, we achieve an accuracy rate of 95%. A web interface was developed for users to interact with the system and validate news. Although this accuracy is high enough, machine recognition is not yet a substitute for human recognition and can only provide users with a specific reference point, which may not be the whole picture.

In the future, we would like to do more subtle manipulations of NLP, such as recognising URLs in articles, years, months, and other words with special meaning. We also hope to provide a plugin for web browsers that will allow users to detect Fake News at any time while surfing the web.

7 Declaration

The authors have no competing interests to declare that are relevant to the content of this article.

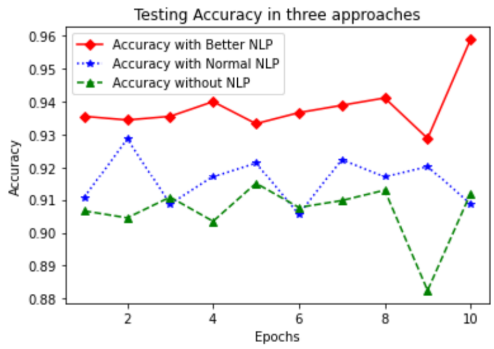


Figure 11: Testing accuracy in three approaches

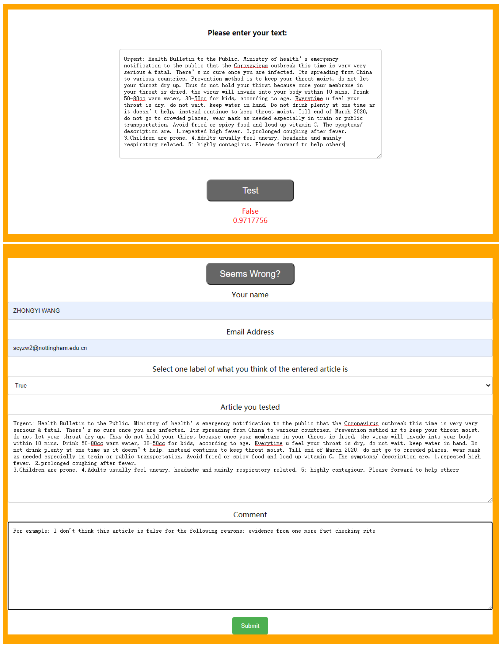


Figure 12: Sample of result and feedback

References:

[1] Fatemeh Torabi Asr and Maite Taboada. Big data and quality data for fake news and misinformation detection. *Big Data & Society*, 6(1):1--14, 2019.

[2] Dongfang Gaozhao. Flagging fake news on social media: An experimental study of media consumers' identification of fake news. *Government Information Quarterly*, 38(3):101591, 2021.

[3] Rahul R Mandical, N Mamatha, N Shivakumar, R Monica, and A N Krishna. Identification of fake news using machine learning. In *IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pages 1--6, 2020.

[4] Fantahun Bogale Gereme and William Zhu. Early detection of fake news" before it flies high". In *Proceedings of the 2nd International Conference on Big Data Technologies*, pages 142--148, Jinan, China, August 2019.

[5] German Lopez. Pizzagate, the fake news conspiracy theory that led a gunman to dc's comet ping pong, explained. *Vox*, 2016.

[6] Rafael A Monteiro, Roney LS Santos, Thiago AS Pardo, Tiago A De Almeida, Evandro ES Ruiz, and Oto A Vale. Contributions to the study of fake news in portuguese: New corpus and automatic detection results. In *International Conference on Computational Processing of the Portuguese Language*, pages 324--334, Canela, Brazil, September 2018.

[7] N Lemann. Solving the problem of fake news. the new yorker (2017).

[8] Oluwaseun Ajao, Deepayan Bhowmik, and Shahrzad Zargari. Fake news identification on twitter with hybrid cnn and rnn models. In *Proceedings of the* 9^{th} *International Conference on Social Media and Society*, page 226–230, 2018.

[9] Deepak S and Bhadrachalam Chitturi. Deep neural approach to fake-news identification. *Procedia Computer Science*, 167:2236--2243, 2020.

[10] Axel Gelfert. Fake news: A definition. *Informal Logic*, 38(1):84--117, 2018.

[11] Gaurav Bhatt, Aman Sharma, Shivam Sharma, Ankush Nagpal, Balasubramanian Raman, and Ankush Mittal. Combining neural, statistical and external features for fake news stance identification. In *Companion Proceedings of the The Web Conference*, page 1353–1357, 2018.

[12] Ethar Qawasmeh, Mais Tawalbeh, and Malak Abdullah. Automatic identification of fake news using deep learning. In *Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 383--388, 2019.

[13] James W Kershner. *Elements of News Writing*. Pearson Higher Ed, 2011.

[14] Andrea Mecacci. Aesthetics of fake. an overview. *Aisthesis. Pratiche, linguaggi e saperi dell'estetico*, 9(2):59--69, 2016.

[15] Edson C Tandoc Jr, Zheng Wei Lim, and Richard Ling. Defining "fake news" a typology of scholarly definitions. *Digital journalism*, 6(2):137--153, 2018.

[16] Meital Balmas. When fake news becomes real: Combined exposure to multiple news sources and political attitudes of inefficacy, alienation, and cynicism. *Communication research*, 41(3):430--454, 2014.

[17] Paul R Brewer, Dannagal Goldthwaite Young, and Michelle Morreale. The impact of real news about `fake news': Intertextual processes and political satire. *International Journal of Public Opinion Research*, 25(3):323--343, 2013.

[18] Zhiwei Jin, Juan Cao, Yongdong Zhang, and Jiebo Luo. News verification by exploiting conflicting social viewpoints in microblogs. In *Proceedings of the Thirtieth AAAI conference on artificial intelligence*, February 2016.

[19] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211--36, 2017.

[20] Shrutika S. Jadhav and Sudeep D. Thepade. Fake news identification and classification using dssm and improved recurrent neural network classifier. *Applied Artificial Intelligence*, 33(12):1058--1068, 2019.

[21] C. Batailler, Skylar M. Brannon, Paul E. Teas, and Bertram Gawronski. A signal detection approach to understanding the identification of fake news. *Perspectives on Psychological Science*, 2021.

[22] Seyedmehdi Hosseinimotlagh and Evangelos Papalexakis. Unsupervised content-based identification of fake news articles with tensor decomposition ensembles. 02 2018.

[23] Karishma Sharma, Feng Qian, He Jiang, Natali Ruchansky, Ming Zhang, and Yan Liu. Combating fake news: A survey on identification and mitigation techniques. *ACM Trans. Intell. Syst. Technol.*, 10(3), apr 2019.

[24] Nadia K Conroy, Victoria L Rubin, and Yimin Chen. Automatic deception detection: Methods for finding fake news. *Association for Information Science and Technology*, 52(1):1--4, 2015.

[25] David M Markowitz and Jeffrey T Hancock. Linguistic traces of a scientific fraud: The case of diederik stapel. *PloS one*, 9(8):e105937, 2014.

[26] Paul Rayson. From key words to key semantic domains. *International journal of corpus linguistics*, 13(4):519--549, 2008.

[27] Matthew L Newman, James W Pennebaker, Diane S Berry, and Jane M Richards. Lying words: Predicting deception from linguistic styles. *Personality and social psychology bulletin*, 29(5):665--675, 2003.

[28] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In *Proceedings of the 20^{th} international conference on World wide web*, pages 675--684, Hyderabad, India, March 2011.

[29] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 3818--3824, 2016.

[30] Natali Ruchansky, Sungyong Seo, and Yan Liu. Csi: A hybrid deep model for fake news detection. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, pages 797--806, Singapore, November 2017.

[31] Saif M Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. Stance and sentiment in tweets. *ACM Transactions on Internet Technology (TOIT)*, 17(3):1--23, 2017.

[32] 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, Vancouver, BC, Canada, October 2017.

[33] Ashwini Rahangdale and AJ Agrawal. Information extraction using discourse analysis from newswires. *International Journal of Information Technology Convergence and Services*, 4(3):21, 2014.

[34] Shereen Oraby, Lena Reed, Ryan Compton, Ellen Riloff, Marilyn Walker, and Steve Whittaker. And that's a fact: Distinguishing factual and emotional argumentation in online dialogue. *arXiv preprint arXiv:1709.05295*, 2017.

[35] Song Feng, Ritwik Banerjee, and Yejin Choi. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 171--175, 2012.

[36] Vanessa Wei Feng and Graeme Hirst. Detecting deceptive opinions with profile compatibility. In *Proceedings of the sixth international joint conference on natural language processing*, pages 338--346, Nagoya, Japan, October 2013.

[37] Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, and Anupam Joshi. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In *Proceedings of the 22^{nd} international conference on World Wide Web*, pages 729--736, Rio de Janeiro, Brazil, May 2013.

[38] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22--36, 2017.

[39] David Mandell Freeman. Can you spot the fakes? on the limitations of user feedback in online social networks. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1093--1102, Perth, Australia, April 2017.

[40] Xinyi Zhou, Apurva Mulay, Emilio Ferrara, and Reza Zafarani. Recovery: A multimodal repository for covid-19 news credibility research. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*, Virtual Event, Ireland, October 2020.

[41] Limeng Cui and Dongwon Lee. Coaid: Covid-19 healthcare misinformation dataset, 2020.

[42] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. 2009.

[43] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825--2830, 2011.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.