Covid-19 fake news sentiment analysis[☆]

Celestine Iwendi^{a,b,*}, Senthilkumar Mohan^c, Suleman Khan^d, Ebuka Ibeke^e,
Ali Ahmadian^{f,g,*}, Tiziana Ciano^h

^a School of Creative Technologies, University of Bolton, Bolton, A676 Deane Rd, Bolton BL3 5AB, United Kingdom

^b Department of Mathematics and Computer Science, Coal City University Enugu, 400231 Enugu, Nigeria

^c School of Information Technology and Engineering, Vellore Institute of Technology, Vellore 632014, India

^d National Centre for Cyber Security (NCCS), Air University, Islamabad 44000, Pakistan

^e School of Creative and Cultural Business, Robert Gordon University, AB10 7AQ, United Kingdom

^f School of Mathematical Sciences, College of Science and Technology, Wenzhou-Kean University, Wenzhou, China

^g Department of Mathematics, Near East University, Nicosia, TRNC, Mersin 10, Turkey

^h Faculty of Business and Law, University of Portsmouth, Richmond Building, Portland Street, Portsmouth PO1 3DE, UK

ARTICLE INFO

Keywords:

Fake news
Social media
Deep learning
NLP
Mining
Emotions

ABSTRACT

'Fake news' refers to the misinformation presented about issues or events, such as COVID-19. Meanwhile, social media giants claimed to take COVID-19 related misinformation seriously, however, they have been ineffectual. This research uses Information Fusion to obtain real news data from News Broadcasting, Health, and Government websites, while fake news data are collected from social media sites. 39 features were created from multimedia texts and used to detect fake news regarding COVID-19 using state-of-the-art deep learning models. Our model's fake news feature extraction improved accuracy from 59.20% to 86.12%. Overall high precision is 85% using the Recurrent Neural Network (RNN) model; our best recall and F1-Measure for fake news were 83% using the Gated Recurrent Units (GRU) model. Similarly, precision, recall, and F1-Measure for real news are 88%, 90%, and 88% using the GRU, RNN, and Long short-term memory (LSTM) model, respectively. Our model outperformed standard machine learning algorithms.

1. Introduction

"Fake News" contents are not a new occurrence. However, they have been on a rapid spread with the aid of social media. The COVID-19 pandemic has shown the negative part of digital technology and the Internet of Things (IoT) that promotes all kinds of conspiracy theories and people who took actions such as applying fake coronavirus cures or making false claims. As a result of these detrimental health instructions, some people were exposed to more dreaded sicknesses than the virus itself. We have witnessed confusion, fear, and distrust from politicians, stakeholders, and so on. Enemies have been created, and others have revolted against compliance with state orders and policy to stay at home. Reports gathered from different multimedia services, including Facebook, show approximately 50 million contents regarding COVID-19 disinformation and misinformation were removed in April 2020. Twitter, on the other hand, questioned about 1.5 million of their users for dissemination of fake news and displaying what

[☆] This paper is for CAEE special section VSI-hci2. Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. Carlos Montenegro.

* Corresponding authors.

E-mail addresses: celestine.iwendi@ieee.org (C. Iwendi), senthilkumar.mohan@vit.ac.in (S. Mohan), suleman.khan@liu.se (S. Khan), e.ibeke@rgu.ac.uk (E. Ibeke), ahmadian.hosseini@gmail.com (A. Ahmadian), tiziana.ciano@unirc.it (T. Ciano).

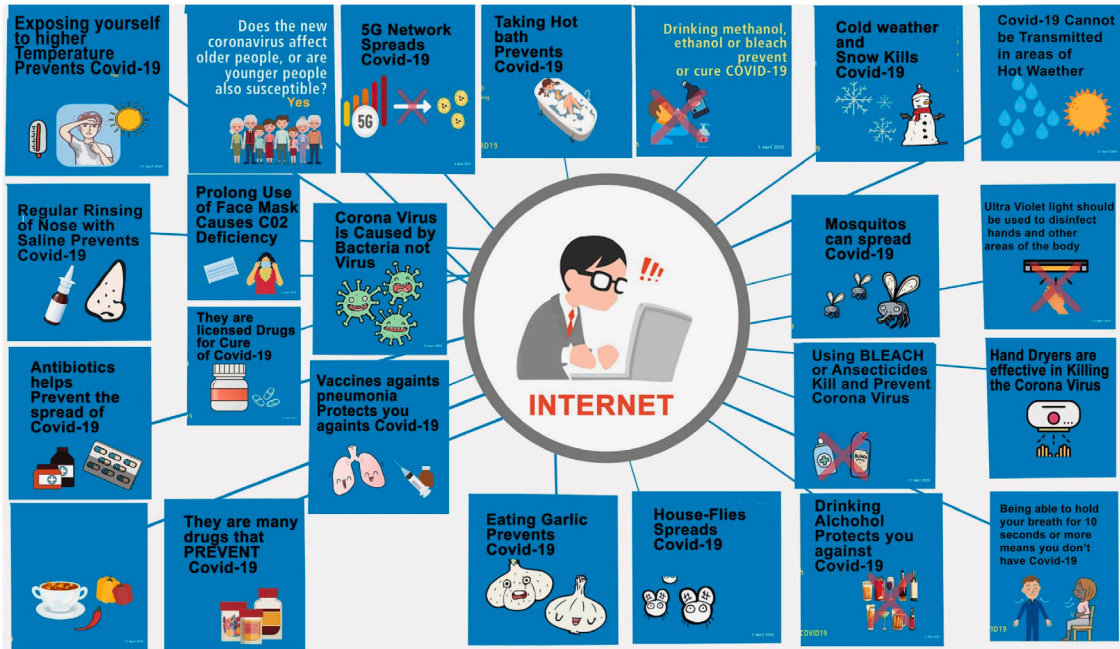


Fig. 1. Fake news realities.

they called false “manipulative behaviors” in the same period. Google is not left as well as YouTube and Gmail, where about 18 million scam emails were flagged and blocked while many uploaded videos on YouTube with misinformation about coronavirus were brought down [1]. The authors in [2] concur with the idea that multimedia services have become a citadel of rumor-mongers spreading fake news, an avenue of unfiltered news with the motive of defrauding, confusing, and misleading other users. Fig. 1 shows what is currently obtainable regarding fake news on the internet and the mind-boggling scenarios created for internet users.

No one will dispute the fact that idleness and stay-at-home policy increased in social media usage and thus endorsed the explosion of fake news argued by the author in [3], who proposed a framework to solve it. This framework combines the domains of frequency and pixels, and engages them as visual information to detect fake news in the network. The authors used a CNN-based network in their design and the captured image features are extracted following the use of the multi-branch convolutional neural network–recurrent neural network (CNN–RNN) model. Further analysis on a real-world dataset demonstrated that their model performs better than other existing state-of-the-art models with 9.2% in accuracy and performance detection increased by over 5.2%.

The contributions of this paper are such:

- The research uses the process of Information Fusion to obtain real news data from various multimedia services: the New York Times, Health Harvard, Centers for Disease Control and Prevention (CDC), World Health Organization (WHO), and Global Health Now. While Fake News data is collected from Facebook, YouTube, and other social media sites.
- The dataset used consists of only text and class labels, and its performance on deep learning classifiers like Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM), and Recurrent Neural Network (RNN) is very low. To improve the accuracy, precision, recall, and F1-Measure, we have proposed 39 novel features for text, which has not been used before for fake news to the best of our knowledge.
- We proposed sentiment features, linguistic features, and named entity-based features. After extracting features from the text, our new features detect COVID-19 fake news with an accuracy of 86.12%. Thus, the accuracy is increased by 20% with novel features.

The rest of this paper is arranged in an organized manner as follows: Section 2 explains in detail the related work and general view of the detection of fake news with their control measures. Section 3 explains the methodology used in this research, Section 4 gives the Experimental setup, Section 5 describes the evaluation work, results, and performance analysis, Section 6 gives the conclusion drawn from the work.

2. Related works

A quantitative analysis of the themes that emerged from the Twitter accounts of Prime Minister Trudeau’s of Canada and President Trump of USA during the climax of the COVID-19 pandemic shows how different their sentiments and perspectives of the virus are. Politics is the emerging theme of one, while public health and policy has been the focus of the other [4]. Bonato and Nazareth [4]

use Network Science which considers the interactions between systems in the formation called “co-occurrence networks” to link the keywords that are present in the two tweets. For example, keywords like “COVID-19” and “pandemic” were linked in the same tweet. The top 100 keywords from @realDonaldTrump and @JustinTrudeau were extracted based on how many times they were used. The analysis conducted on the network proposes that politicians’ social media messages play a role in molding the views of the public concerning the pandemic and attempts to regulate it based on the expressed sentiments.

With the emergence of various deep neural network models, sentiment analysis tasks have once again made significant progress. However, these neural network models could not accurately capture sentiment information on sentiment analysis tasks, which leads to their instability. Aspect-category or aspect-terms have been combined to form a final sentiment. The emotions and sentiments of software developers largely influence software productivity and quality. However, existing studies on emotion mining and sentiment analysis are still in the early stage in software engineering in terms of accuracy, the size of datasets, and the specificity of the analysis [5].

False information littering about the COVID-19 pandemic is an essential aspect that could greatly influence the response to the virus [1]. People are encouraged to verify information about the pandemic by checking various multimedia sources such as the official site of the WHO, reading the full article instead of just the headline, as well as checking out the credentials and qualifications of authors. It has been shown recently that visual information such as images and videos almost always accompanies fake news and is quite useful in deciphering which is fake news.

There have been several studies on fake news identification. The Multi-domain Visual Neural Network (MVNN) proposed in [3], ascertains fake news by aligning the semantic information in the pixel domains and their frequency, i.e. it studies the visual content accompanying a news post to determine if the news is fake or not. It is a well-known fact that news channels sponsored by governments are fond of distributing fake news around social media to divert people’s attention from crucial matters such as healthcare, education and the state of countries’ economy. India was used as a case study to prove that mainstream news channels are deliberately spreading fake news to incite unhealthy nationalism, hatred, and division between citizens of various communities with variant religious and political beliefs to distract them from the real issues at hand [6]. In concurrence with this, Qi et al. [7] describe how some writers produce fake news articles to influence the results of an election, as well as cause harm by triggering the negative emotions of people. Most of such articles are quite a resemblance of real articles but include information that are usually exaggerated. Moreover, these articles possess misleading headlines, known as clickbait to lure people to view them. Often, people do not care to read the articles; they absorb the misleading headlines and distribute them.

In recent times, society has become very dependent on social media for the provision of information. And while it has proven useful with its quick and cheap access to information, it has its disadvantages. This includes the rapid dissemination of fake news without people verifying the authenticity of the news. More can be done in the detection of fake news. In [8], an idea was proposed to detect fake news by maintaining properties that explain the structure of various types of news. Machine learning approaches as well as other detection methods such as linguistics, clustering, and predictive modeling, have proven to be effective in detecting fake news from real news. Another machine learning system, Bayesian, was developed using specific toolkits such as SciPy, to examine the probability that a news post was fake. This fresh technique is called influence mining and was propounded in [9]. It is important to note that time is of essence in differentiating between real and fake news. Detection of fake news, while still in the early stages of its dissemination, will go a long way in avoiding massive misinformation and misleading contents. Notwithstanding recent advancements in fake news detectors, no work has intricately proposed early detection of fake news.

Bhoir [10] also used various machine learning models such as Gated Recurrent Unit Network (GRU), Naïve Bayes, Decision Tree to determine whether a news is fake based on previously witnessed fake or real news. The solution to the problem of fake news is simply a very effective and efficient way of identifying the authenticity of news. In [11], an automatic inference model that could detect the credibility of fake news was introduced. This ‘FAKEDETECTOR’ creates a unit model called the GDU, which accepts and studies various sources such as articles and creators simultaneously and fuses them to create an output. In 2019, a survey was conducted to obtain the public opinion on whether false information being disseminated was being manipulated by government officials of the Thailand government [12]. 291 respondents participated in the survey, and most of them reported that they saw fake news several times a month. 29% reported that they chanced upon fake news every day. A majority of the interviewees claimed to not put trust in the media and blamed the government for tampering with the dissemination of authentic information. An ensemble technique which uses GRU and LSTM (neural networks) as well as Android software to determine the veracity of the news article has been proposed [13]. The app developed for this study connects to the Java Web Server and verifies from various sources on the Internet whether the news is “fake” before displaying the result. The experimental results of the proposed model look propitious and may further be experimented with.

While discussing the spread of misinformation, it is important to decipher which is satire and which is fake news. Although fake news is often meant to mislead people, satire holds up human vices to ridicule them. Its purpose is usually more amusing than misleading. A different line of research was carried out in [14] to determine how long fake news stayed on the internet. They posit that fake news typically vanishes from the internet after a certain period of time (probably after it has achieved its goal), while real news stays on the internet forever.

The issue of fake news is more of a menace than we give it credit for and should be seriously examined. As at 2011, it was recorded that 32.8% of the earth’s population used the internet. It is a no-brainer that the number has surely increased by now, and the risk of distributing fake news is greater now more than ever. Bhutani et al. [15] dwelled on this growing risk and noted that while various studies have been conducted to find solutions to the problem, the proposed models are still in the early stages of development and are not ready to be implemented. They also discussed how concepts like freedom of expression, the neutrality of the internet, and archaic digital laws make it challenging to solve the issue of fake news. They, however, proposed solutions

such as creating websites that check information accuracy and merge them with teams that monitor media. The authors of [16] analyzed different text preprocessing techniques and used sentiment to accelerate the accuracy of detecting fake news. They also stated that focus will be made on neural network algorithms to further enhance the accuracy, as well as expanding their dataset to visual content. Sear et al. [17] propounded that the agents of fake news are malevolent software (bots) that socialize with natural users. They described credulous users as those who have a higher number of ‘bots’ as friends than their social friends. The results of the experiment carried out proved that credulous users post a larger amount of fake news than their counterparts. They offered that the study of these credulous users could aid in the understanding of misinformation and could be turned into a weapon to battle fake news.

The COVID-19 pandemic has made room for another rift between groups such as the ‘anti-vax community’; people who are against vaccination and the ‘pro-vax community’; people who are for it. A recent study shows that the anti-vax communities online had more distinct and broader conversations about COVID-19, thereby encouraging new users to join their community [18]. The lack of a vaccine and the hindrance of the creation of one is disturbing because then, there is nothing to stop this pandemic. Their paper tries to measure the contents of COVID-19 information among the various communities’ posts of health guidance concerning vaccinations. It also tries to provide strategies to solve the issues of users going through a large amount of misleading information about their health. It has been discovered that a quick diagnosis of COVID-19 aids in the successful medical treatment of it. In [19], a learning framework for classifying information on COVID-19 and lesion localization was developed. This framework can be applied clinically to get a quick and accurate COVID-19 diagnosis. As a result of the rapid misinformation spreading at such a crucial time, Rustam et al. [20] provided an encyclopedic review of the key aspects of the pandemic. A lot of countries’ economy has been in shambles as a result of COVID-19, so the paper also explored the use of technologies such as Blockchain, 5G, Artificial Intelligence (AI) in alleviating the impact of the outbreak.

Cui, Wang, and Lee [21] employed a machine learning model for fake news detection at a premature stage without depending on external information, which may prevent it from causing public uproar. They used two kinds of models, the bag-of-words model and the neural networks. The evidence presented indicates that fake news causes harm; the 2016 US presidential election was used as a reference. Another study [22] attempts to compare different approaches in alleviating the issue of fake news distribution. Some of the approaches used include the hybrid CNN, RNN, as well as Naïve Bayes. The paper chooses a more effective method from either the machine or deep learning method for providing a solution to the issue of the balance of accuracy and falsity. The authors in [23] strove to identify, review, and compare various methods and tools, current research has dug up as a panacea to the challenge of fake news, specifically taking into consideration the algorithms of machine learning and natural language. It is not unheard of for people’s social media accounts to be hacked, especially those with a large number of followers. These cyber-attacks allow fake news to be distributed to a large amount of people using the hacked account. Recently, the Twitter accounts of some millionaires and celebrities were hacked, with the scammers offering to double the Bitcoin sent to their Blockchain wallet by the followers. A lot of people fell prey to this scam, and thousands of dollars were lost on that fateful night. Iwendi et al. [24] proposed a Zero code based Watermark detection approach called the KeySplitWatermark that provides software against cyber-attacks. When the software is tampered with to a certain extent, the original software code is restored, making it strong enough to withhold further attacks. The phenomenon of fake news is not unfamiliar; relevant authorities should consider this study and take calculated and measured steps to mitigate it. Therefore, we have proposed a model using fake news propaganda around COVID-19 to detect, mitigate, and provide a quick response.

3. Proposed architecture

In this study, 39 features were created from texts and are then used to detect the fake news regarding Covid-19 from social media using state-of-the-art deep learning models. The steps involved in the proposed methodology are shown in Fig. 2. The dataset used in this study is collected from various websites and social media sites. To make the dataset clean, some preprocessing were performed on the dataset by removing URLs, punctuations marks and empty columns. After text preprocessing, tokenization was performed to convert the larger text into words. The major part of this research is extracting the features from texts and then using these features for fake news detection instead of the texts. After feature extraction, these features are passed to state-of-the-art deep learning algorithms like RNN, LSTM, and GRU to train the model, while various evaluation metrics are used to evaluate the performance of our proposed model.

3.1. Dataset description

In this study, the COVID-19 dataset that were used was collected from [25]. Dataset consists of 586 true news and 578 fake news, and more than 1100 news articles and social media posts regarding COVID-19. The true news were gathered from Harvard Health Publishing, WHO, CDC, The New York Times and so on. Fake news were gathered from social media (Facebook) posts and other medical sites.

3.2. Feature extraction from text

The extraction of features is a reduction in dimensionality in which an original collection of raw data is condensed to more accessible classes for processing. By doing so, we reduce the compiler’s processing time and increase the rate of effectiveness in

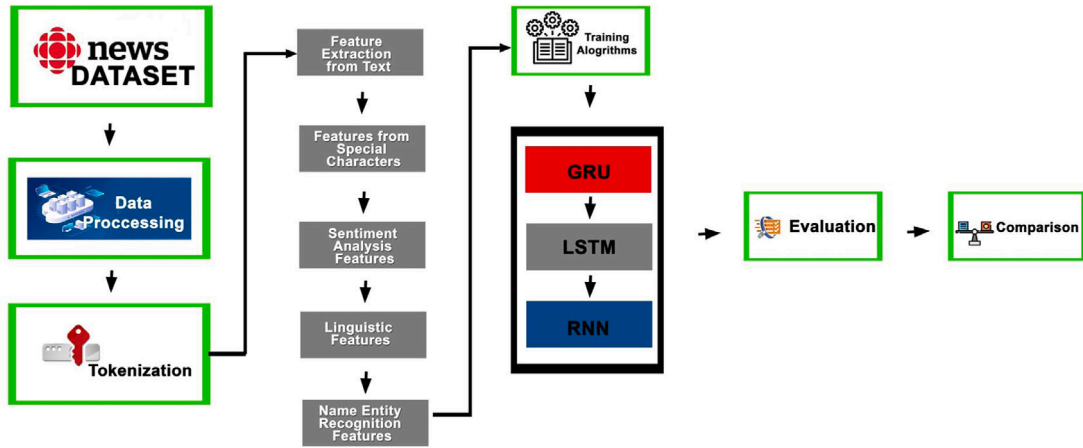


Fig. 2. Proposed model.

Table 1
Sentiment features from text.

Feature name	Data type
Positive sentiment score	Numeric
Negative sentiment score	Numeric
Neutral sentiment score	Numeric
Compound sentiment score	Numeric
Text polarity	Numeric

detecting the word value. A disadvantage of such broad data sets is that certain variables take a number of computing resources to handle. Thus, for this purpose we have created our own features for texts to detect fake news. These features have the number of stop words, number of upper case letters, number of small letters, upper case letters, number of numeric values, word count, character count, sentence count, and average sentence length of the texts. It also contains the average word length, sentence polarity score, sentiment scores, positive, negative, neutral, compound sentiment scores and the extracted named entity recognition features (NER) from text. Tables 1–3 present the features that we created from the texts and were used for classification.

3.3. Features of special characters

Features of special characters show the scenario that happens after data collection. Our model indicates that every input vector presents a special character. For example, min-char-rnn utilizes one-hot vectors to present different characters. Therefore, it allows our model to generate authentic training segments by predicting the next character based on previous characters. A special character allows for shorter text classification while integrating RNN, LSTM, and GRU with fully connected layers.

3.4. Sentiment analysis features

The fundamental task of applying the sentiment analysis features is the classification of the polarity of our model texts or sentences as positive, negative, neutral, or as compound sentiment. Here, our resultant text polarity of the data type in Table 1 is numeric. It presents the features that we created from the texts and were used for classification.

3.5. Linguistic features

The classification performed in this research entails categorizing a piece of Fake News text into a category. This is done by converting that piece of text into any other language. Table 2 shows the feature names arranged with their data types, displaying numeric and non-numeric characteristics.

3.6. Named entity recognition features

This is also known as entity identification, entity chunking, and entity extraction. The objective of Named Entity Recognition (NER) is to allocate and classify tokens in texts into predefined categories. Table 3 shows the NER with the feature names and data types.

Table 2
Linguistic features from text.

Feature name	Data type	Feature name	Data type
News Source	Non-Numeric	Num of ?	Numeric
Num of Stopwords	Numeric	Num of /	Numeric
Num of @	Numeric	Num of #	Numeric
Num of numeric values	Numeric	Num of uppercase characters	Numeric
Num of lowercase characters	Numeric	Num of all uppercase characters	Numeric
Text language	Numeric	Word count	Numeric
Character count	Numeric	Sentence count	Numeric
Average word length	Numeric	Average sentence length	Numeric

Table 3
NER features from text.

Feature name	Data type	Feature name	Data type
Person	Numeric	NORP	Numeric
FAC	Numeric	Organization	Numeric
GPE	Numeric	Location	Numeric
Product	Numeric	Event	Numeric
Work of art	Numeric	Law	Numeric
Language	Numeric	Date	Numeric
Time	Numeric	Percent	Numeric
Money	Numeric	Quantity	Numeric
Cardinal	Numeric	Ordinal	Numeric

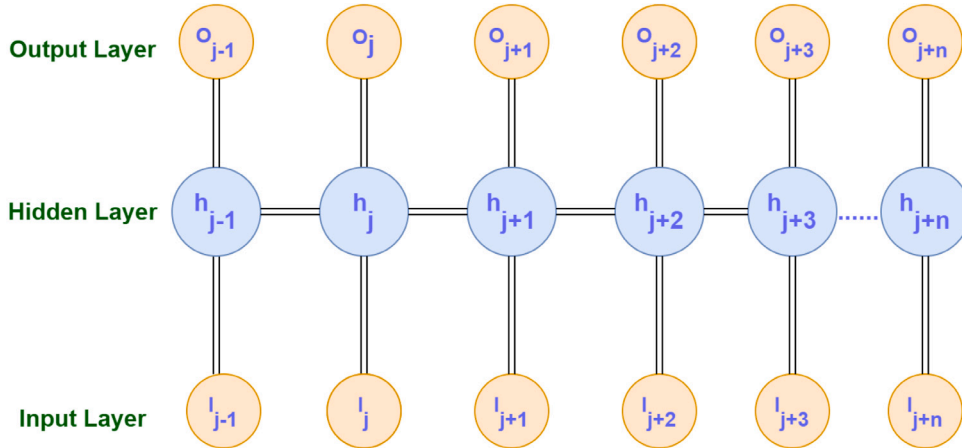


Fig. 3. RNN working model.

3.7. Recurrent neural networks (RNN)

Due to the missing gradient problem, conventional neural networks (NN) do not yield satisfactory results on time series data. To deal with the above-mentioned problem, RNN was proposed in 1982 by Hopfield. RNN has the strength of letting the NNs know the patterns over a period of time. RNN will forecast the sequential data behavior dependent on previous occurrences in a video, voice recordings, text, things, etc. Fig. 3 displays the RNN working system.

In Fig. 4, h_j depicts the weight vector for the hidden layer, O_j denotes the weight vector for the output layer, I_j presents the input word vector. The time stamp for the hidden layer j is expressed using Eq. (1) and the output layer final value for RNN is calculated using Eq. (2).

$$h_j = \sigma (M * 0_j + N * h_{j-1}) \tag{1}$$

$$o_j = \sigma (K * h_j) \tag{2}$$

where $\sigma(\cdot)$ presents the activation function. Tanh, Relu or sigmoid, one of them can be used as the activation function. After every timestamp j , the hidden state h_j is calculated by using Eq. (2), with the corresponding input parameters.

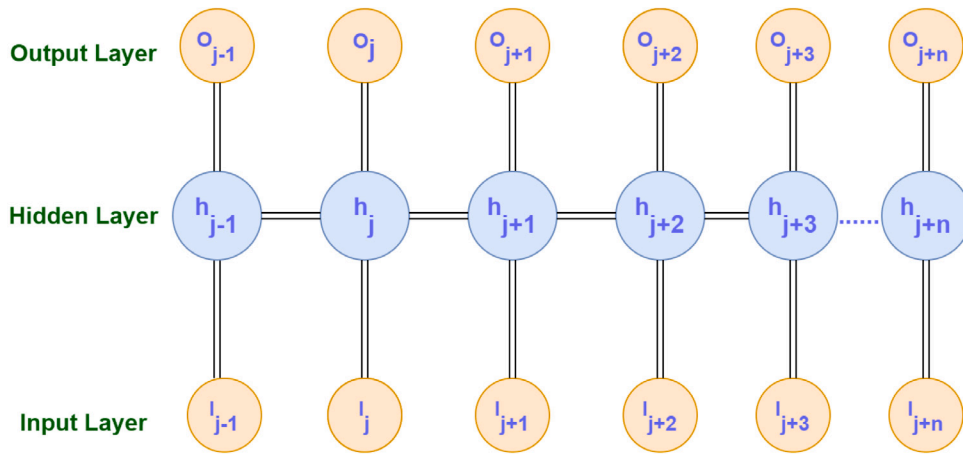


Fig. 4. RNN propose structure.

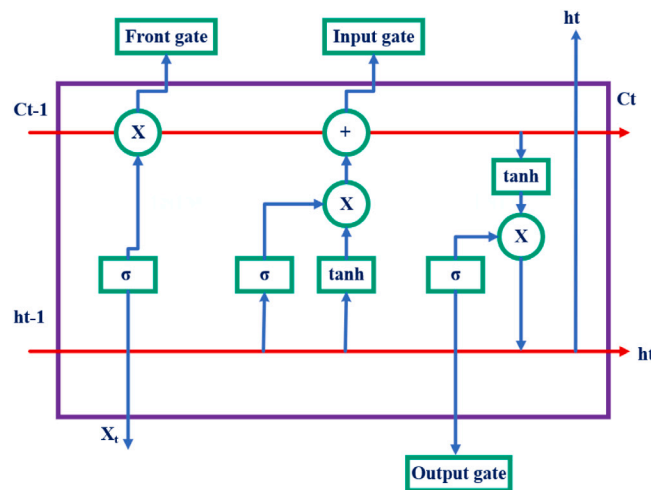


Fig. 5. LSTM working model.

3.8. Long short term memory

RNN suffers from short-term memory. If a series is long enough, they may find it difficult to bring knowledge from earlier stages in time to later ones. Thus, if you are trying to process a paragraph to do prediction, RNN can leave crucial details out from the start. The RNN suffers from the vanishing gradient issue during back propagation. Gradients are the values used to update the weights of a neural network. The problem with the vanishing gradient is that the gradient shrinks as it propagates back over time. If the gradient value is exceedingly small, the learning does not help that much. LSTM was created as the short-term memory solution. They have got internal systems called gates that can control the information flow. These gates may know which data is necessary to retain or to throw away in a sequence. In doing so, it will transfer important knowledge down the long sequence chain to render predictions. With these two networks, almost all state-of-the-art results were achieved based on recurrent neural networks. LSTM consists of 3 gates such as input gate, forget gate, and output gate. LSTM graphical presentation is given in Fig. 5. Sequence value D_i concatenated with the previous cell O_{i-1} output.

The first move for this combined input is to have it squashed through a **tano** layer. The second step is to transfer this input via an input gate. The sigmoid function output range is in between 0 and 1, so the weights that bind the input to those nodes can be trained to output values close to zero to “delete” certain input values (or, conversely, outputs below 1 to “transfer” certain values). The next step is data flow through the forget gate loop. LSTM cells consist of an internal state variable e_i . This variable is updated after one time phase i.e. e_{i-1} is applied to the input data in order to establish an efficient recurrence layer. Lastly, we have an output layer **tano** squashing function, whose output is controlled by an output gate. This gate decides what values are technically allowed as cell O_i outputs. Table 4 shows the LSTM params that can be articulated through an equation:

$$g = (b^g + D_i W^g + O_{i-1} U^g) \tag{3}$$

Table 4
LSTM proposed tabulation.

Layer (type)	Output shape	Param #
lstm_13 (LSTM)	(None, None, 100)	56400
lstm_14 (LSTM)	(None, None, 50)	30200
lstm_15 (LSTM)	(None, 25)	7600
dropout_17 (Dropout)	(None, 25)	0
dense_21 (Dense)	(None, 10)	260
dense_22 (Dense)	(None, 10)	110
dense_23 (Dense)	(None, 1)	11
Total params: 94,581		
Trainable params: 94,581		
Non-trainable params: 0		

b^g presents input bias, while W^g is the weight for input and U^g is the weight of previous output cell. Exponents g do not reflect an increased power, but rather imply that these are input weights and bias values. Such squashed input which is then element-wise multiplied by the output of input gate as mentioned above, is a set of activated sigmoid nodes:

$$I = \sigma(b^I + D_i W^I + O_{i-1} U^I) \quad (4)$$

The output of the LSTM cell input section is shown in Eq. (3).

$$g \circ I \quad (5)$$

The \circ operator depicts element-wise multiplication. Mathematical form of forget gate is shown in Eq. (6).

$$f = \sigma(b^f + D_i W^f + O_{i-1} U^f) \quad (6)$$

The output of the previous state's element-wise product and the forget gate shall be presented as

$$e_{i-1} \circ f \quad (7)$$

Output of the forget gate is shown in Eq. (8).

$$s_i = s_{i-1} \circ f + g \circ I \quad (8)$$

Mathematical form of the output gate is shown in Eq. (9).

$$e_i = e_{i-1} \circ f + g \circ I \quad (9)$$

Therefore, the cell's final output can be seen, with the **tano** squashing, as:

$$R = \sigma(b^R + D_i W^R + O_{i-1} U^R) \quad (10)$$

3.9. Gated recurrent unit (GRU)

Many variants were built to solve the problem of the Vanishing–Exploding gradients often encountered during the development of a simple Recurrent Neural Network. One of the most popular variants is the LSTM. The GRU is one of the lesser-known but equally powerful variants. GRU, unlike LSTM only consists of just three gates and does not retain an Internal Cell state. Information that is contained in an LSTM recurrent unit in the Internal Cell State is incorporated into the Gated Recurrent Unit's hidden state. The cumulative information would be moved on to the next Recurrent Gated Unit. GRU with various gates functions are as outlined below:-

Update Gate: It dictates how much of the historical information has to be transferred into the future. It is similar to the Input Gate in a recurrent cell of the LSTM.

Reset gate: It determines how much past knowledge you should forget about. It is identical to combining the Input Gate and the Forget Gate in a recurrent LSTM unit.

$$U^j = \sigma(W_{U^j} X^j + Z_{U^j} h_{j-1} + b_{U^j}) \quad (11)$$

$$R^j = \sigma(W_{R^j} X^j + Z_{R^j} h_{j-1} + b_{R^j}) \quad (12)$$

Updated gate is presented by U^j , sigmoid function is depicted by $\sigma(\cdot)$. Similarly W , Z and b are matrices and vector parameters. h_j presents output vector, input vector is presented by X^j . The upgrade gate executes identical forget gate and an LSTM input gate functions. This is the duty of choosing which information to remove and which information to add. It is the duty of the reset gate to decide how many of the previous data will be forgotten. Because GRU has less gates relative to LSTM, the cycle of training is typically quicker.

Table 5
GRU propose tabulation.

Layer (type)	Output shape	Param #
gru_1 (GRU)	(None, None, 100)	42300
gru_2 (GRU)	(None, None, 50)	22650
dropout_1 (Dropout)	(None, None, 50)	0
gru_3 (GRU)	(None, 25)	5700
dropout_2 (Dropout)	(None, 25)	0
dense_1 (Dense)	(None, 10)	260
dense_2 (Dense)	(None, 1)	11
Total params: 70,921		
Trainable params: 70,921		
Non-trainable params: 0		

Current Memory Gate: This is often overlooked during typical Gated Recurrent Unit Network discussions. It is integrated into the Reset Gate as the Input Modulation Gate is a subpart of the Input Gate that is used to add a certain non-linearity in the data and even to render the Zero-mean response.

Current Memory Gate calculation process is a bit different. Firstly, the Reset Gate Hadamard product and the previous hidden state vector are calculated. This function is then parameterized and then applied to the input vector of the parameterized current as shown in Table 5.

$$\tilde{h}_j = \tanh(W \odot I_j + W \odot (R_j \odot h_{j-1})) \quad (13)$$

First, a vector of ones and the same dimensions as that of the input is defined to calculate the current hidden state. This vector is named as ones but presented as 1 mathematically. Measure the update gate's hadamard element and past hidden state function. Subtract the upgrade gate from ones to create a new vector and then, measure the newly created vector hadamard product using the current memory gate.

$$h_j = U_j \odot h_{j-1} + (1 - U_j) \odot \tilde{h}_j \quad (14)$$

The back-propagation for a GRU network through time algorithm is similar to that of a LSTM network and differs only in the formation of the differential chain. Let \bar{y}_j be the expected performance at every stage of the period, and y_j be the actual performance at every step. The error is then computed at each step of the time by:-

$$E_j = -y_j \log(\bar{y}_j) \quad (15)$$

Therefore the cumulative error of all time phases is calculated by summing up the errors.

$$E = \sum_j E_j \quad (16)$$

$$E = \sum_j -y_j \log(\bar{y}_j) \quad (17)$$

Likewise, on every time stage, the value $\frac{\partial E}{\partial W}$ can be determined as the summation of the gradients.

$$\frac{\partial E}{\partial W} = \sum_j \frac{\partial E_j}{\partial W} \quad (18)$$

Using the rule of the chain and the fact that \bar{y}_j is a feature of h_j and which is also a feature of \tilde{h}_j , the following statement arises:-

$$\frac{\partial E_j}{\partial W} = \frac{\partial E_j}{\partial \bar{y}_j} \frac{\partial \bar{y}_j}{\partial h_j} \frac{\partial h_j}{\partial h_{j-1}} \frac{\partial h_{j-1}}{\partial h_{j-2}} \cdots \frac{\partial h_0}{\partial W} \quad (19)$$

Therefore the complete gradient of the error is supplied by:-

$$\frac{\partial E}{\partial W} = \sum_j \frac{\partial E_j}{\partial \bar{y}_j} \frac{\partial \bar{y}_j}{\partial h_j} \frac{\partial h_j}{\partial h_{j-1}} \frac{\partial h_{j-1}}{\partial h_{j-2}} \cdots \frac{\partial h_0}{\partial W} \quad (20)$$

Remember that the gradient equation requires a ∂h_j chain that appears similar to a simple Recurrent Neural Network but this method functions differently owing to the internal workings of the h_j derivatives.

4. Experimental set-up

In order to make the dataset clean, preprocessing is performed on the dataset as described in Section 3.4. After extracting features from text, these features are normalized using min-max normalization, the mathematical expression for min-max normalization is given below:

$$f' = \frac{f - \min(N)}{\max(N) - \min(N)} (\text{new_max}(N) - \text{new_min}(N)) + \text{new_min}(N) \quad (21)$$

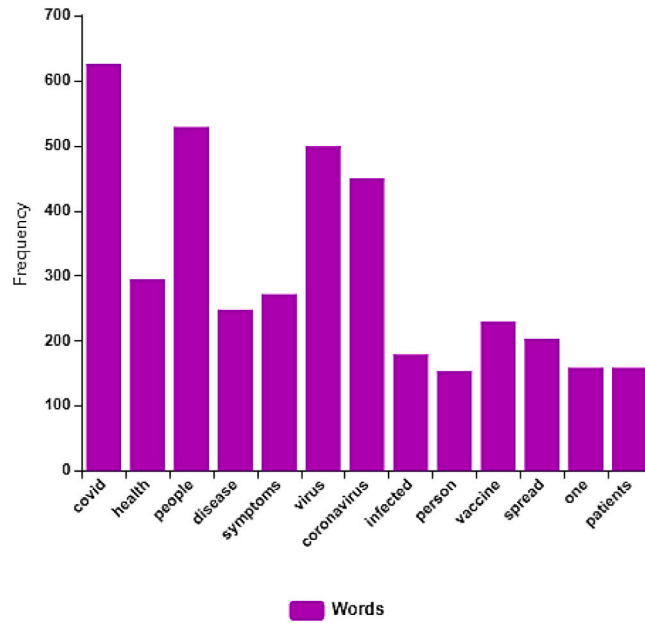


Fig. 8. Frequency bar graph for COVID-19 real news.

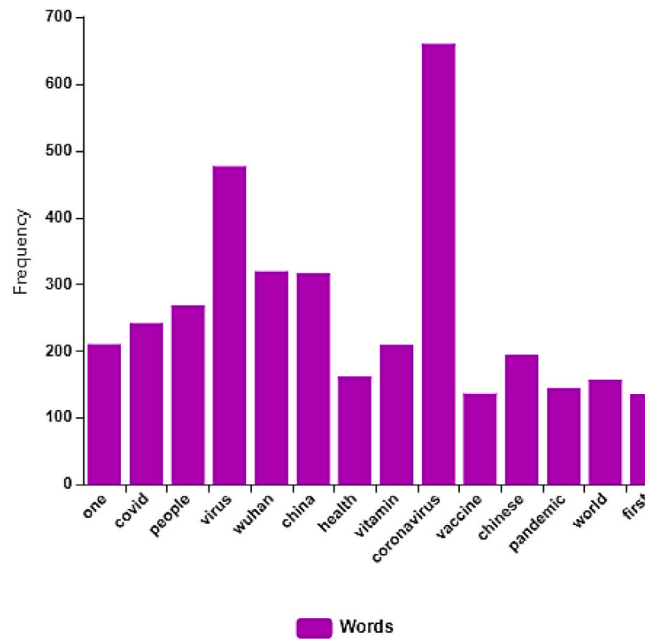


Fig. 9. Frequency bar graph for COVID-19 fake news.

4.3. Evaluation metrics

Various performance metrics are used to evaluate the proposed solution, including precision, recall, F1-Measure and accuracy. Above mentioned performance metrics are based on Actual Positive (AP), False Positive (FP), False Negative (FN) and Actual Negative (AN). Accuracy is used to measure how many instances are correctly classified as normal and attacks classes. Accuracy is achieved by summing correctly classified instances and dividing by the total instances as presented in Eq. (22).

$$Accuracy = \frac{AP + AN}{AP + FP + FN + AN} \tag{22}$$

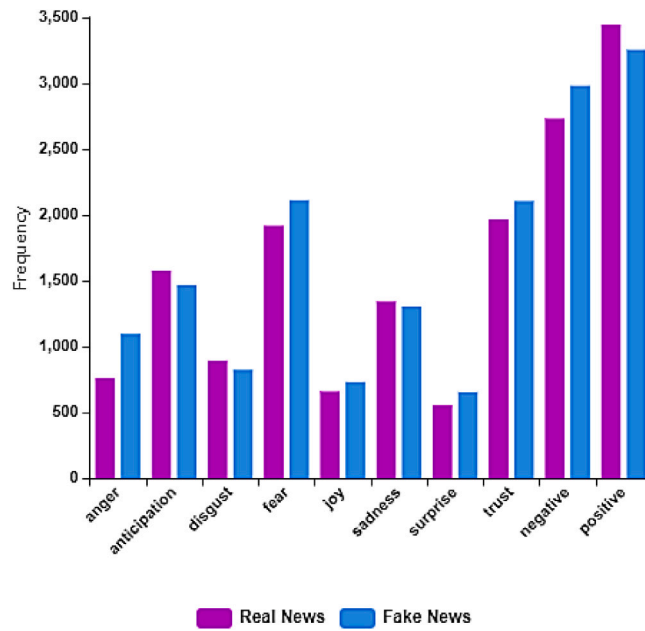


Fig. 10. Emotion mining for COVID-19 news.

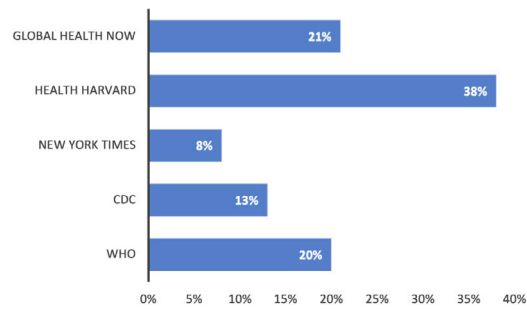


Fig. 11. Real news sources for COVID-19.

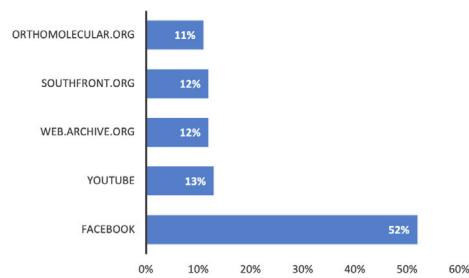


Fig. 12. Fake news sources for COVID-19.

$$DR = \frac{AP}{AP + FN} \tag{23}$$

Precision's objective is to evaluate the True Positive (TP) entities in relation to False Positive (FP) entities.

$$Precision = \frac{AP}{AP + FP} \tag{24}$$

Table 6
Performance of the ML algorithms before feature extraction.

Algorithm	Prediction accuracy	Precision	Recall	F1-measure
AdaBoost classifier	79.88	76.76	86.36	81.82
Decision tree classifier	67.81	70.51	62.50	66.26
KNeighbors classifier	62.06	72.91	39.77	51.47

Table 7
Accuracy and loss of all the classifiers.

Classifier	Training loss	Training accuracy	Testing loss	Testing accuracy
GRU	0.05	98.29	4.11	59.20
LSTM	0.31	95.07	6.90	55.72
RNN	0.00	100.00	3.30	57.71

Table 8
Classification report before features extraction.

Model	Label	Precision	Recall	F1-measure
GRU	Fake news	55	58	56
	Real news	63	60	62
LSTM	Fake news	51	60	55
	Real news	61	52	56
RNN	Fake news	53	62	57
	Real news	63	55	59

The purpose of recall is to evaluate True Positive (TP) entities in relation to (FN) False Negative entities that are not at categorized. The mathematical form of recall is mentioned in the equation below

$$Recall = \frac{AP}{AP + FN} \quad (25)$$

Sometimes performance assessment may not be suitable with accuracy and recall. For instance, if one mining algorithm has low recall but high precision then another algorithm would be needed. This problem could be solved by using F1-score that gives an average recall and precision. F1-score can be calculated as shown in equation.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (26)$$

5. Results and discussion

All these experiments are performed on google colab. The system specification is core I3 system with 8 GB RAM and 2.7 GHz processor.

5.1. Results without features extraction

From Table 6 we can depict that Ada-boost classifier outperformed the other machine learning algorithms like Decision Tree (DT) and K Nearest Neighbor (KNN) in terms of accuracy, precision, recall and F1-Measure score. 79.88% prediction accuracy is achieved by Ada-boost before feature extraction which is highest among all the machine learning classifiers. Similarly, Ada-boost also achieved 76.76% precision, 86.36% recall and 81.82% F1-Measure score respectively. Prediction and accuracy for DT and KNN are 67.81% and 62.06%, respectively. Precision, recall and F1-Measure for DT are 70.51%, 62.50% and 66.26%, respectively. KNN achieved 72.91% precision, 39.77% recall and 51.47% F1-Measure respectively. We can see that the traditional machine learning algorithms performed very well compared to our proposed deep learning models before creating our own features as stipulated in Section 3. Therefore, after the creation of our novel features from text, our proposed model is expected to outperform the machine learning models.

From Tables 7 and 8 it can be observed that the Precision, Recall and F1-Score for GRU model without feature extraction are 55%, 58%, and 56% for fake news, 63%, 60% and 62% respectively for Real News using GRU model. Similarly, training loss for GRU model is 0.05 and training accuracy is 98.29%. This is typically expected for a testing accuracy to be lower than the training accuracy due to the refinement of the data. Also, prediction loss and prediction accuracy scores are 4.11 and 59.20%. Using LSTM model, we have achieved 0.31 training loss and 6.90 testing loss. Training and testing accuracy for LSTM model are 95.07% and 55.72%, respectively. Precision, recall and F1-Measure for fake news using LSTM model are 51%, 60% and 55%, respectively. Similarly for real news precision, recall and F1-Measure scores are 61%, 52% and 56%, respectively. For RNN model 0.00 training loss and 3.30 testing loss is recorded, respectively. Training and prediction accuracy for RNN model are 100% and 57.71%, respectively. Overall RNN model achieved 0.00 training loss and training accuracy 100% which is better compared to GRU and LSTM training loss and training accuracy. Similarly, RNN achieved low loss for prediction which is 3.30 and GRU achieved high prediction accuracy

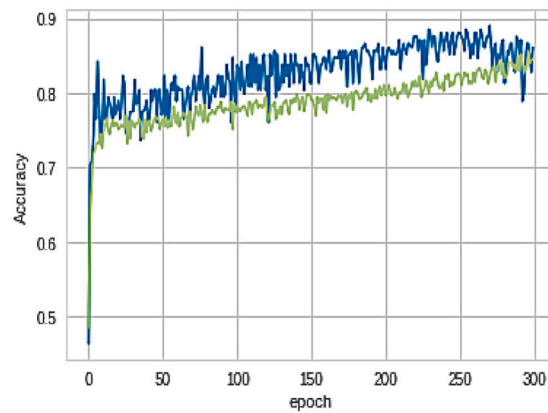


Fig. 13. GRU model training and testing accuracy.

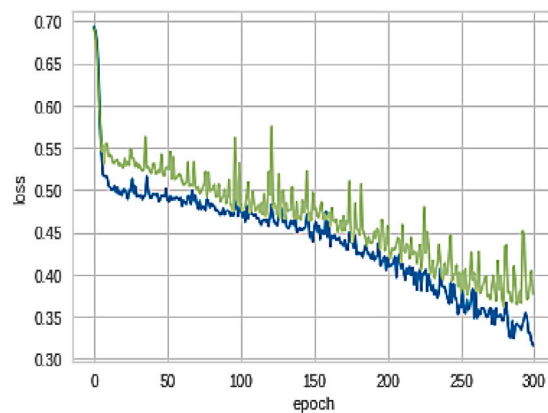


Fig. 14. GRU model training and testing loss.

59.20% compared to LSTM and RNN. For fake news, high precision is 55% using GRU model. Best recall and F1-Measure for fake news recorded are 62% and 57% using RNN model, respectively. Optimal precision, recall and F1-Measure for real news are 63%, 60% and 62% using GRU mode. Training accuracy is high because 70% dataset is used for training and only 30% dataset is used for testing. It also shows that deep learning algorithms performs well on large features and large dataset.

Fig. 13 presents the prediction accuracy and training accuracy for GRU model. The green curve depicts the training accuracy. At 1st epoch, training accuracy is 45% and after 100th epoch, we achieved 98.29% training accuracy which is highest training accuracy. Similarly at 1st epoch, prediction accuracy using GRU model is 47% and after 100th epoch, maximum prediction accuracy reported is 59.20%. Fig. 14 presents the GRU model loss for training and testing. The blue curve depicts the training loss and the green curve presents the testing loss this time. At 1st iteration, training loss is around 0.70 and 0.05 after 10th iteration. Similarly for prediction loss at 1st epoch, prediction loss is 0.5 and after 100th epoch, prediction loss is 0.41 respectively using GRU model.

Fig. 15 presents the prediction accuracy and training accuracy for LSTM model. The green curve depicts the training accuracy, at 1st epoch training accuracy is 47% and after 100th epoch, we achieved 95.07% training accuracy. Similarly at 1st epoch, prediction accuracy using LSTM model is 56% and after 100th epoch, maximum prediction accuracy reported is 55.72%. Fig. 16 presents the LSTM model loss for training and testing. The blue curve depicts the training loss and green curve present the testing loss this time. At 1st epoch training loss is around 0.68 and after 100th iteration 0.00. Similarly for prediction loss at 1st epoch, prediction loss is 0.67 and after 100th epoch, prediction loss is 0.30 respectively using RNN model.

Fig. 17 presents the prediction accuracy and training accuracy for RNN model. The blue curve depicts the training accuracy, at 1st epoch training accuracy is 53.32% and after 100th epoch, we achieved 100% training accuracy which is highest training accuracy. Similarly at 1st epoch, prediction accuracy using LSTM model is 60.70% and after 10th epoch, maximum prediction accuracy reported is 57.71%. Fig. 18 presents the RNN model loss for training and testing. At 1st epoch training loss is around 0.70 and after 10th iteration 0.58. Similarly for prediction loss at 1st epoch prediction loss is 0.72 and after 10th epoch prediction loss is 0.68 respectively using GRU model.

Fig. 19 presents AUC curve for GRU classifier and AUC for GRU is around 59.05%. Similarly Fig. 20 presents AUC curve for LSTM and Fig. 21 depicts AUC for RNN model. LSTM and RNN both have 56.11% and 57.99% AUC, respectively. GRU outperformed in terms of AUC compared to other classifiers.

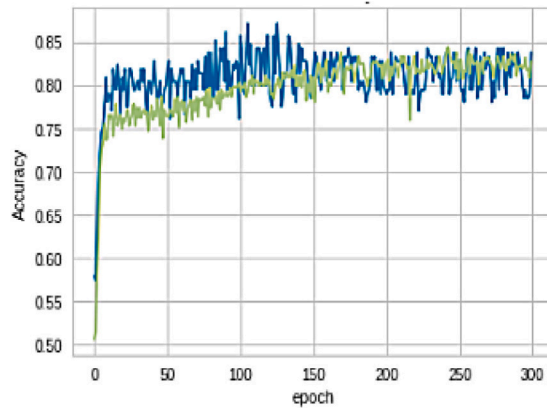


Fig. 15. LSTM model training and testing accuracy.

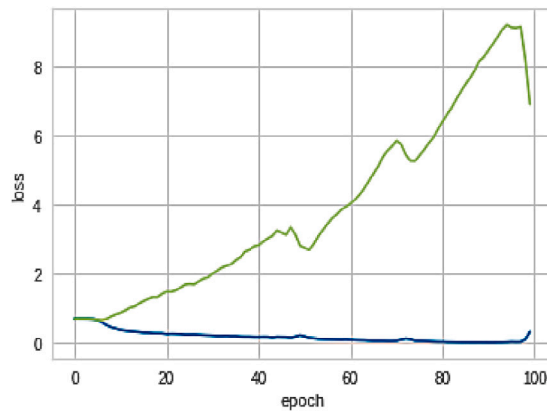


Fig. 16. LSTM model training and testing loss.

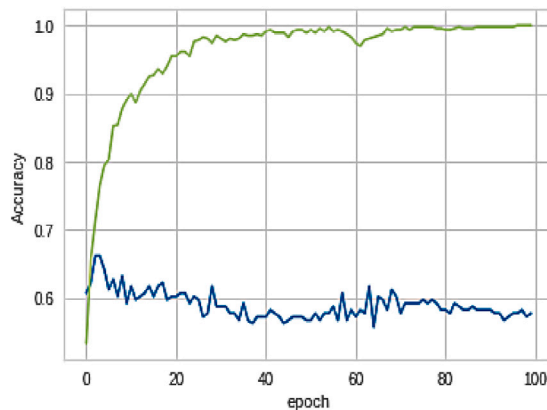


Fig. 17. RNN model training and testing accuracy.

5.2. Results after features extraction

Table 9 depicts the that Ada-boost classifier achieved 82.75% prediction accuracy, while DT and KNN achieved 77.58% and 69.54% prediction accuracy, respectively. Similarly, precision, recall and F1-Measure scores are 79%, 89.77% and 84.04%, respectively. 72.47%, 89.77% and 80.20% precision, recall and F1-Measure scores are achieved using DT, respectively. KNN classifier achieved 62.41% precision, 100% recall and 76.85% F1-Measure score, respectively. After feature extraction with both DL and

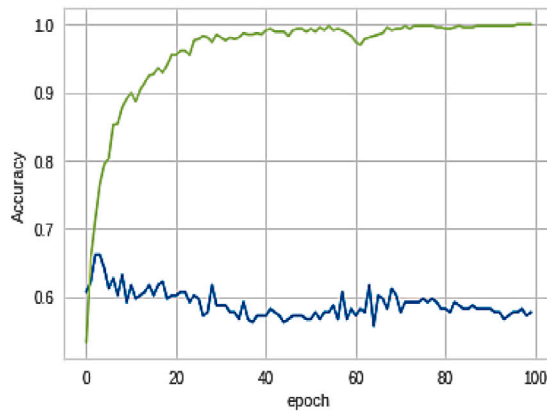


Fig. 18. RNN model training and testing loss.

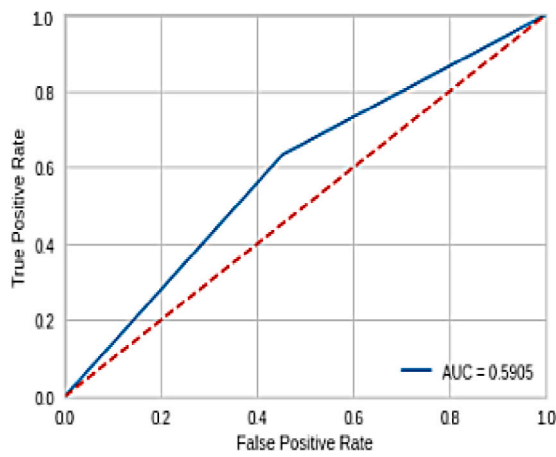


Fig. 19. GRU ROC curve.

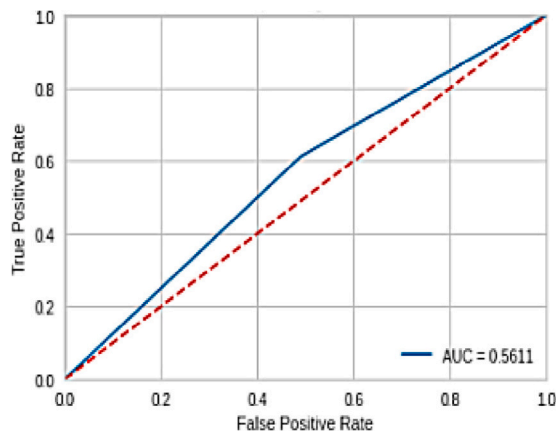


Fig. 20. LSTM ROC curve.

ML, we can deduce that deep learning outperformed machine learning algorithms in terms of precision, recall, F1-Measure and prediction accuracy. Table 10 shows Accuracy and Loss of all the classifiers and Table 11 shows Classification Report before feature extraction.

Table 11 depicts that the Precision, Recall and F1-Score for GRU model after feature extraction are 84%, 83%, and 83% for fake news, 88%, 88% and 88% respectively for Real News using GRU model. Similarly, training loss for GRU model is 0.31, training

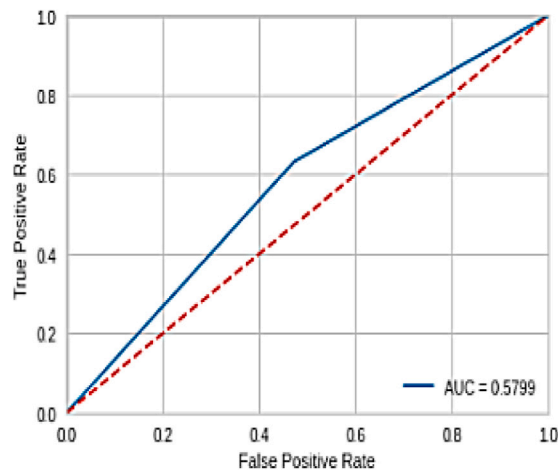


Fig. 21. RNN ROC curve.

Table 9
Performance of the ML algorithms after feature extraction.

Algorithm	Prediction accuracy	Precision	Recall	F1 score
AdaBoost	82.75	79.00	89.77	84.04
Decision tree	77.58	72.47	89.77	80.20
KNN	69.54	62.41	100	76.85

Table 10
Accuracy and loss of all the classifiers.

Classifier	Training loss	Training accuracy	Testing loss	Testing accuracy
GRU	0.31	84.57	0.37	86.12
LSTM	0.38	82.72	0.45	83.73
RNN	0.31	84.57	0.34	85.65

Table 11
Classification report before features.

Model	Label	Precision	Recall	F1-measure
GRU	Fake news	84	83	83
	Real news	88	88	88
LSTM	Fake news	84	76	80
	Real news	84	89	86
RNN	Fake news	85	80	82
	Real news	86	90	88

accuracy is 84.57% respectively. prediction loss and prediction accuracy scores are 0.37 and 86.12%. Using LSTM model, we have achieved 0.38 training loss and 0.45 testing loss. Training and testing accuracy for LSTM model are 82.72% and 83.73%, respectively. Precision, recall and F1-Measure for fake news using LSTM model are 84%,76% and 80%, respectively. Similarly for real news precision, recall and F1-Measure scores are 84%, 89% and 86% respectively. For RNN model 0.31 training loss and 0.34 testing loss is recorded, respectively. Training and prediction accuracy for RNN model are 84.57% and 85.65%, respectively. Overall GRU and RNN models achieved 0.31 training loss and training accuracy 84.57% which is better compared to LSTM training loss and training accuracy, respectively. Similarly, RNN achieved low loss of 0.34 and GRU achieved high prediction accuracy of 86.12% compared to LSTM and RNN. For fake news, high precision is 85% using RNN model. Best recall and F1-Measure for fake news recorded are 83% using GRU model. 88% optimal precision is recorded using GRU for real news. RNN model achieved 90% recall for real news while 88% F1-measure is recorded using GRU and RNN.

Fig. 22 presents the prediction accuracy and training accuracy for GRU model. The green curve depicts the training accuracy. At 1st epoch training, accuracy is 45% and after 100th epoch, we achieved 84.57% training accuracy which is highest training accuracy. Similarly at 1st epoch, prediction accuracy using GRU model is 44% and after 100th epoch, maximum prediction accuracy reported is 86.12%. Fig. 23 presents the GRU model loss for training and testing. The blue curve depicts the training loss and the green curve presents the testing loss. At 1st epoch, training loss is around 0.55 and 0.31 after 100th iteration. Similarly, for prediction loss at 1st epoch, prediction loss is 0.70 and after 100th epoch, prediction loss is 0.37.

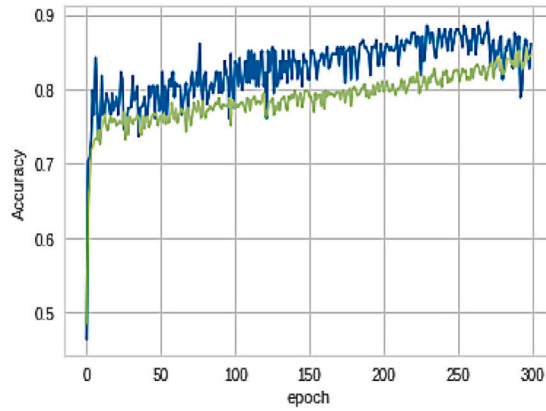


Fig. 22. GRU model training and testing accuracy.

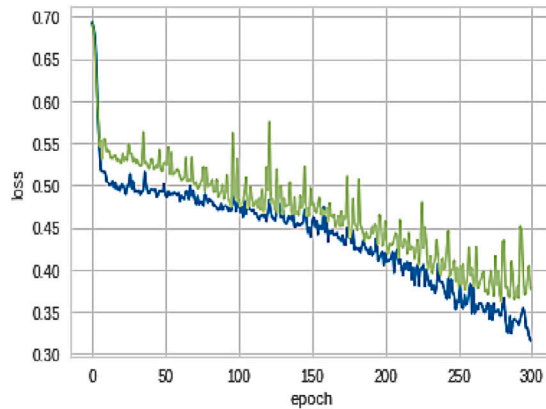


Fig. 23. GRU model training and testing loss.

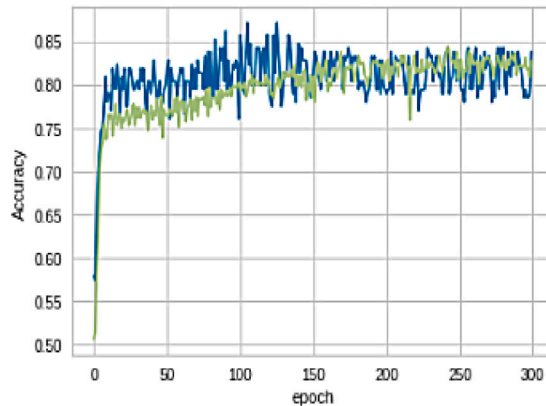


Fig. 24. LSTM model training and testing accuracy.

Fig. 24 presents the prediction accuracy and training accuracy for LSTM model. The green curve depicts the training accuracy. At 1st epoch, training accuracy is 52% and after 100th epoch, we achieved 82.72% training accuracy which is highest training accuracy for this model. Similarly at 1st epoch, prediction accuracy using GRU model is 54% and after 100th epoch, maximum prediction accuracy reported is 83.73%. Fig. 25 presents the LSTM model’s loss for training and testing. The blue curve depicts the training loss and the green curve presents the testing loss. At 1st epoch, training loss is around 0.68 and 0.38 after 100th iteration. Similarly for prediction loss at 1st epoch, prediction loss is 0.67 and after 100th epoch, prediction loss is 0.45.

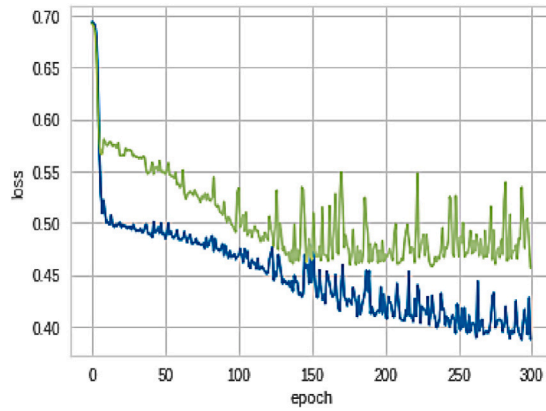


Fig. 25. LSTM model training and testing loss.

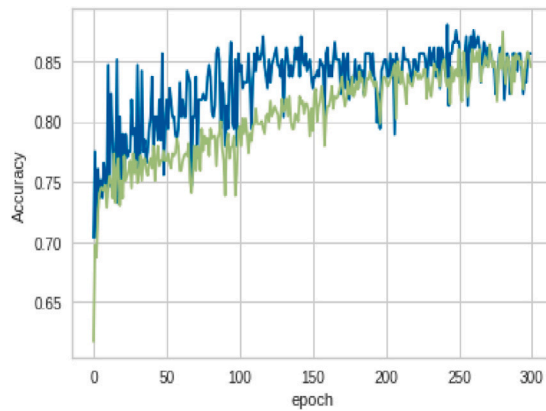


Fig. 26. RNN model training and testing accuracy.

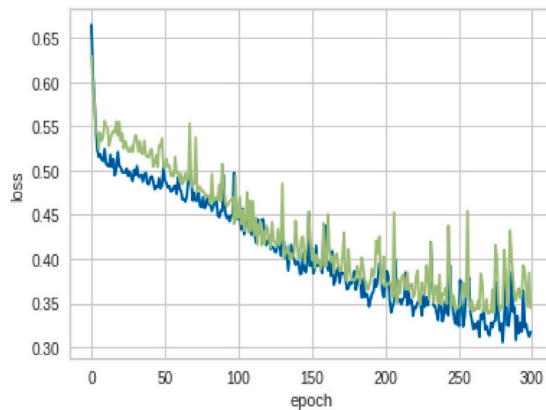


Fig. 27. RNN model training and testing loss.

Fig. 26 presents the prediction accuracy and training accuracy for RNN model. The green curve depicts the training accuracy. At 1st epoch, training accuracy is 61% and after 100th epoch, we achieved 84.57% training accuracy. Similarly, at 1st epoch, prediction accuracy using the RNN model is 71% and after 100th epoch, maximum prediction accuracy reported is 85.65%. Fig. 27 presents the RNN model's loss for training and testing. The blue curve depicts the training loss and the green curve presents the testing loss. At 1st epoch, training loss is around 0.68 and 0.31 after 100th iteration. Similarly, for prediction loss at 1st epoch, prediction loss is 0.65 and after 100th epoch, prediction loss is 0.34.

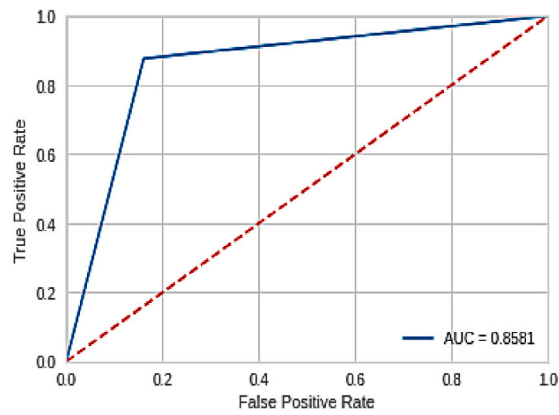


Fig. 28. GRU ROC curve.

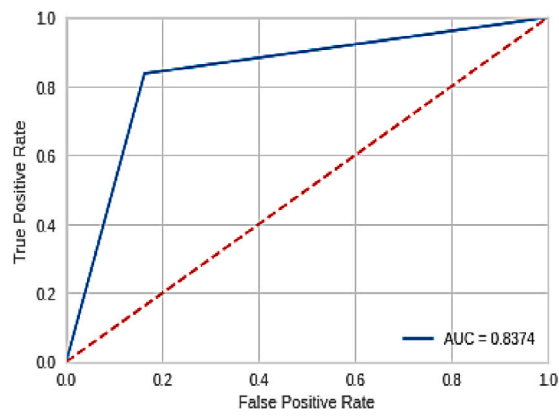


Fig. 29. LSTM ROC curve.

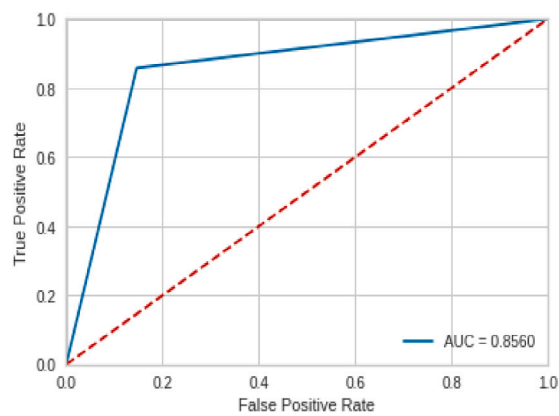


Fig. 30. RNN ROC curve.

Figs. 28–30 show that GRU has the highest area under the curve (AUC) which is 85.81% while LSTM and RNN have 83.74% and 85.60%, respectively. Comparing these values before feature extraction, AUC after feature extraction is much higher with a 20% increase in AUC after feature extraction.

6. Conclusion

At a time where the use of masks and adherence to social distancing policies due to the COVID-19 pandemic has a profound power to slow down or even eliminate the coronavirus outbreak. Why has it been so hard to get people to adopt these simple measures? There are many reasons, but this research has identified misinformation on the internet as a major factor. Consequently, people are quite unsure and struggle to understand the exponential spread of the virus. The outbreak of the virus was paralleled by an outbreak of misinformation about the virus. This misinformation ranges from false origin of the virus, conspiracy theories, fake cures to harmful health advice. Misinformation jeopardises public health responses. This research presented a novel system utilizing 39 features used in detecting fake news about COVID-19. The model uses an information fusion process to obtain social media data and applies state-of-the-art deep learning models such as GRU, LSTM, and RNN. We have proposed sentiment features, linguistic features, and named entity-based features. After extracting features from the text, our new features detect fake news about COVID-19 with an accuracy of 86.12%. Thus, the accuracy is increased by 20%. Overall high precision is 85% using RNN model. The results further show the best recall and F1-Measure for fake news to be 83% using GRU model. Similarly, for real news, precision, recall and F1-Measure are 88%, 90% and 88%, respectively. Our results outperformed the standard machine learning algorithms on the same dataset.

CRedit authorship contribution statement

Celestine Iwendi: Conceptualization of this study, Methodology, Software, Writing first draft. **Senthilkumar Mohan:** Validation of Data. **Suleman Khan:** Data curation, Writing - Original draft preparation. **Ebuka Ibeke:** Reviewing and Editing of final draft and data validation. **Ali Ahmadian:** Supervision and Reviewing Original draft. **Tiziana Ciano:** Reviewing and Editing of final draft and Validation of Data.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] Petchot Kamonwan. 'Fake news' in the time of COVID-19. 2020, UNESCO Bangkok. URL <https://bangkok.unesco.org/index.php/content/press-provides-antidote-fake-news-time-covid-19>. [Accessed 10 July 2020].
- [2] Bedi A, Pandey N, Khatri SK. A framework to identify and secure the issues of fake news and rumours in social networking. In: 2019 2nd international conference on power energy, environment and intelligent control. 2019. p. 70–3.
- [3] Qi P, Cao J, Yang T, Guo J, Li J. Exploiting multi-domain visual information for fake news detection. In: 2019 IEEE international conference on data mining. 2019. p. 518–27.
- [4] Bonato A, Nazareth A. Facts or fake news: Revealing patterns in the COVID-19 tweets of Trudeau and Trump. 2020, URL <https://theconversation.com/facts-or-fake-news-revealing-patterns-in-the-covid-19-tweets-of-trudeau-and-trump-142139>. [Accessed 10 July 2020].
- [5] Ding Jin, Sun Hailong, Wang Xu, Liu Xudong. Entity-level sentiment analysis of issue comments. In: 2018 IEEE/ACM 3rd international workshop on emotion awareness in software engineering. 2018. p. 7–13.
- [6] Snell N, Fleck W, Traylor T, Straub J. Manually classified real and fake news articles. In: 2019 international conference on computational science and computational intelligence. 2019. p. 1405–7.
- [7] Uppal A, Sachdeva V, Sharma S. Fake news detection using discourse segment structure analysis. In: 2020 10th international conference on cloud computing, data science engineering. 2020. p. 751–6.
- [8] Manzoor SI, Singla J, Nikita. Fake news detection using machine learning approaches: A systematic review. In: 2019 3rd international conference on trends in electronics and informatics. 2019. p. 230–4.
- [9] Ramezani M, Rafiei M, Omranpour S, Rabiee HR. News labeling as early as possible: Real or fake? In: 2019 IEEE/ACM international conference on advances in social networks analysis and mining. 2019. p. 536–7.
- [10] Vinit Bhoir S. An efficient fake news detector. In: 2020 international conference on computer communication and informatics. 2020. p. 1–9.
- [11] Sa-nga-ngam P, Mayakul T, Srisawat W, Kiattisin S. Fake news and online disinformation. A perspectives of Thai government officials. In: 2019 4th technology innovation management and engineering science international conference, TIMES-ICON. 2019. p. 1–4.
- [12] Barua R, Maity R, Minj D, Barua T, Layek AK. F-NAD: An application for fake news article detection using machine learning techniques. In: 2019 IEEE Bombay section signature conference. 2019. p. 1–6.
- [13] Das D, Clark AJ. Satire vs fake news: You can tell by the way they say it. In: 2019 first international conference on ?transdisciplinary AI. 2019. p. 22–6.
- [14] Kang S, Hwang J, Yu H. Multi-modal component embedding for fake news detection. In: 2020 14th international conference on ubiquitous information management and communication. 2020. p. 1–6.
- [15] Bhutani B, Rastogi N, Sehgal P, Purwar A. Fake news detection using sentiment analysis. In: 2019 twelfth international conference on contemporary computing. 2019. p. 1–5.
- [16] Balestrucci A, Nicola RD. Credulous users and fake news: A real case study on the propagation in Twitter. In: 2020 IEEE conference on evolving and adaptive intelligent systems. 2020. p. 1–8.
- [17] Sear RF, Velázquez N, Leahy R, Restrepo NJ, Oud SE, Gabriel N, et al. Quantifying COVID-19 content in the online health opinion war using machine learning. *IEEE Access* 2020;8:91886–93.
- [18] Wang X, Deng X, Fu Q, Zhou Q, Feng J, Ma H, et al. A weakly-supervised framework for COVID-19 classification and lesion localization from chest CT. *IEEE Trans Med Imaging* 2020;1.
- [19] Chamola V, Hassija V, Gupta V, Guizani M. A comprehensive review of the COVID-19 pandemic and the role of IoT, drones, AI, blockchain, and 5G in managing its impact. *IEEE Access* 2020;8:90225–65.
- [20] Rustam F, Reshi AA, Mehmood A, Ullah S, On B, Aslam W, et al. COVID-19 future forecasting using supervised machine learning models. *IEEE Access* 2020;8:101489–99.

- [21] Cui L, Wang S, Lee D. SAME: Sentiment-aware multi-modal embedding for detecting fake news. In: 2019 IEEE/ACM international conference on advances in social networks analysis and mining. 2019. p. 41–8.
- [22] Han W, Mehta V. Fake news detection in social networks using machine learning and deep learning: Performance evaluation. In: 2019 IEEE international conference on industrial internet. 2019. p. 375–80.
- [23] Hirlekar VV, Kumar A. Natural language processing based online fake news detection challenges – A detailed review. In: 2020 5th international conference on communication and electronics systems. 2020. p. 748–54.
- [24] Iwendi C, Jalil Z, Javed AR, G. TR, Kaluri R, Srivastava G, et al. KeySplitWatermark: Zero watermarking algorithm for software protection against cyber-attacks. *IEEE Access* 2020;8:72650–60.
- [25] Li Susan. Explore COVID-19 infodemic. *Medium*, 2020, Towards Data Science. URL <https://towardsdatascience.com/explore-covid-19-infodemic-2d1ceaae2306>. [Accessed 10 July 2020].

Celestine Iwendi (Senior Member, IEEE) is a Senior Lecturer with the school of Creative Technologies, University of Bolton, United Kingdom. He received his second master's degree from Uppsala University, Sweden in 2008 and Ph.D. degree in electronics from the University of Aberdeen, U.K. in 2013. He has 22 years of teaching and industry experience with 60+ articles in peer-reviewed journals.

Senthilkumar Mohan received a master's degree in Software engineering in 2007, a master's degree in IT-Networking in 2013, and Ph.D. degree in Big Data in 2017 from VIT University, India. He is currently working as Associate Professor with the Department of software and system engineering, School of Information Technology and Engineering (SITE), Vellore Institute of Technology (VIT), Vellore.

Suleman Khan received the master's degree from the Department of Computer Science, Air University, Islamabad, Pakistan, in 2019. He is currently a PhD student at Linköping University, Sweden. His research interests include network security, machine learning, and data science.

Ebuka Ibeke is a Lecturer in Business/Data Analytics at Robert Gordon University. He holds a FHEA, UK, a BSc in Computer Science from Nnamdi Azikiwe University, Nigeria, an MSc in Information Engineering (with Distinction) from Robert Gordon University, UK, and a PhD in Computing Science from the University of Aberdeen, UK. His research interests revolve around Artificial Intelligence.

Ali Ahmadian received the Ph.D. degree from Universiti Putra Malaysia (UPM) in 2014. He was a Senior Lecturer at the National University of Malaysia. He is an Adjunct faculty member at Kean University, Wenzhou Campus, China; His published more than 200 articles in the field of applied mathematics.

Tiziana Ciano is a graduate of Economics from the "Mediterranea" University of Reggio di Calabria where she is currently a PhD Student in Law and Economics. Her specializations include Stochastic and Nonlinear Programming – Dynamical Systems and Epidemics Models – Machine Learning and Date Science. She is a Visiting PhD Student at University of Portsmouth (UK) and Politehnica University of Bucharest (Romania).