applied sciences

MDPI

Article

A Novel Hybrid Deep Learning Model for Detecting COVID-19-Related Rumors on Social Media Based on LSTM and Concatenated Parallel CNNs

Mohammed Al-Sarem [1,2,*], Abdullah Alsaeedi [1,*], Faisal Saeed [1,3,*], Wadii Boulila [1,4] and Omair AmeerBakhsh [1]

[1] College of Computer Science and Engineering, Taibah University, Medina 41477, Saudi Arabia; wadii.boulila@riadi.rnu.tn (W.B.); oameerbakhsh@taibahu.edu.sa (O.A.)
[2] Information System Department, Saba'a Region University, Mareeb, Yemen
[3] Institute for Artificial Intelligence and Big Data, City Campus, Pengkalan Chepa, Universiti Malaysia Kelantan, Kota Bharu 16100, Kelantan, Malaysia
[4] RIADI Laboratory, National School of Computer Sciences, University of Manouba, Manouba 2010, Tunisia
* Correspondence: msarem@taibahu.edu.sa (M.A.-S.); aasaeedi@taibahu.edu.sa (A.A.); fsaeed@taibahu.edu.sa (F.S.)

check for updates

Citation: Al-Sarem, M.; Alsaeedi, A.; Saeed, F.; Boulila, W.; AmeerBakhsh, O. A Novel Hybrid Deep Learning Model for Detecting COVID-19-Related Rumors on Social Media Based on LSTM and Concatenated Parallel CNNs. *Appl. Sci.* **2021**, *11*, 7940. https://doi.org/10.3390/app11177940

Academic Editor: Arcangelo Castiglione

Received: 30 July 2021
Accepted: 24 August 2021
Published: 28 August 2021

Abstract: Spreading rumors in social media is considered under cybercrimes that affect people, societies, and governments. For instance, some criminals create rumors and send them on the internet, then other people help them to spread it. Spreading rumors can be an example of cyber abuse, where rumors or lies about the victim are posted on the internet to send threatening messages or to share the victim's personal information. During pandemics, a large amount of rumors spreads on social media very fast, which have dramatic effects on people's health. Detecting these rumors manually by the authorities is very difficult in these open platforms. Therefore, several researchers conducted studies on utilizing intelligent methods for detecting such rumors. The detection methods can be classified mainly into machine learning-based and deep learning-based methods. The deep learning methods have comparative advantages against machine learning ones as they do not require preprocessing and feature engineering processes and their performance showed superior enhancements in many fields. Therefore, this paper aims to propose a Novel Hybrid Deep Learning Model for Detecting COVID-19-related Rumors on Social Media (LSTM–PCNN). The proposed model is based on a Long Short-Term Memory (LSTM) and Concatenated Parallel Convolutional Neural Networks (PCNN). The experiments were conducted on an ArCOV-19 dataset that included 3157 tweets; 1480 of them were rumors (46.87%) and 1677 tweets were non-rumors (53.12%). The findings of the proposed model showed a superior performance compared to other methods in terms of accuracy, recall, precision, and F-score.

Keywords: rumor detection; deep learning; twitter analysis; convolution neural networks; LSTM; pretrained model; word embedding

1. Introduction

One of the cybercrimes that has recently occurred is the spreading of rumors on social media. According to [1], people help criminals spread false rumors due to the the insufficient credibility of governments and mainstream media. Some web media use attractive titles and some even spread unconfirmed rumors so that people cannot clearly distinguish between facts and rumors. For example, during pandemics, rumors often spread on social networks in terms of those who are suspected of infection [2]. Currently, several infectious diseases and pandemics have so far emerged and they are by nature rapidly evolving. In this context, COVID-19 is considered as one of the rapidly spreading pandemics through which almost all countries around the world have been infected. In Saudi Arabia, since the first outbreak of COVID-19 epidemic on 2 March 2020, the competent health authorities recommended a set of preventive and precautionary measures originating from their keenness to protect people's health and ensure their safety. Based on the World Health Organization's (WHO) statistics, even with such stringent precautionary measures adopted by the kingdom, the number of infections in Saudi Arabia was doubling every 6 days. However, this exponential growth in the total number of infected cases with COVID-19 has been reported by several countries. Therefore, with the exponential growth of the number of infected people and the rapid spread of the virus in different countries, users with different professions as well as laypeople can post and circulate any related news. This piece of medical news might cause a manic situation in a population as well as market disorder. To avoid this situation, competent health authorities and other government authorities issued several rules to criminalize posting and sharing any fake information on social media. To solve this issue, there is a need to propose an intelligent way to detect health-related COVID-19 information automatically on social media. Although a few intelligent methods have been used to detect rumors on social media, obtaining a high detection rate is still a large challenge, especially if the rumors are written in languages such as Arabic.

One of the main online social networking platforms is Twitter, which is currently used by several health and governmental authorities as one of the main sources of information and for announcing various information and awareness about emergency situations and regulations issued. Despite these efforts, it is noted that Twitter has become a breeding ground for several COVID-19 rumors. Thus, a broad number of researchers from different scientific fields such as computer science, social science, and psychology sought the reasons behind spreading rumors on social media platforms and for a way to combat rumors at the early stages [3]. Detecting rumors on social media platforms as a classification task has been addressed in several studies such as [3–10]. By analyzing the literature, the used techniques for detecting rumors can be easily categorized either as traditional machine learning-based (ML) or deep learning-based DL methods. The ML approaches require several preprocessing and feature engineering processes. Oppositely, DL methods can extract the informative features directly from the textual content without human assistance. Among these studies, only few researchers have investigated the detection of rumors that are written in Arabic on social media. The detection rate of the existing methods still needs to be improved. Because of this research problem, this paper proposes a deep learning based model to detect COVID-19-related rumors posted on Twitter using the Arabic language. We investigated different deep learning architectures using the publicly available dataset ArCOV-19 [11]. The proposed deep learning model connects long short-term memory (LSTM) with three parallel and concatenated conventional neural networks (PCNN). The experimental results showed that the proposed model outperformed all the other investigated models and produced satisfied results in terms of accuracy, recall, precision, and F-score.

The main contributions of this paper are:

- A new LSTM–PCNN architecture is proposed and extensive experiments are presented to demonstrate the performance of the proposed model.
- The impact of word embedding layers is investigated in order to select the appropriate scheme. For this purpose, we investigated the influence of static word embeddings such as word2vec, GloVe, and FastText on the proposed model.

The organization of the paper is as follows: In Section 2, we review the state-of-the-art techniques that address the rumor detection problem. Section 3 presents the architecture of the proposed model. In Section 4, the methodology of this study is described; the used dataset, preprocessing methods, evaluation metrics, and experimental design, and evaluation were highlighted. Section 5 gives the details of the experimental results to highlight our contribution. Section 6 concludes the whole paper by summarizing the contributions.

2. Related Studies

In the following subsections, we briefly present some of the notable works published during the COVID-19 pandemic that used ML and DL methods to detect COVID-19 related misinformation and rumors. We also report and summarize the results and limitations. This section also gives the reader the necessary background to understand the main characteristics of DL models that are investigated in this paper.

Rumor Detection Approaches

During the outbreak of COVID-19, especially when countries around the world began implementing a ban and a full lockdown, a wave of panic spread rapidly among citizens. Due to this, the WHO emphasized the need to fight against misinformation related to the virus and the methods of treatment [12]. To achieve this, the health authorities paid attention to debunking such rumors. However, verifying all these rumors required much human effort. Some researchers suggested developing AI techniques to fight against COVID-19 related misinformation/rumors on social media. Below are some of the notable works that used this approach.

Chen [13] embedded the pre-trained model of BERT with TextCNN and TextRNN models. The proposed model was trained on data with 3737 rumors collected from different Chinese platforms. The results showed that the proposed BERT model outperformed the other methods. In addition, all three models showed good results and could be used to defeat the COVD-19 related rumors.

Alqurashi et al. [14] conducted an extensive experiment on a dataset of COVID-19 misinformation written in Arabic spread on Twitter. The n-gram TF–IDF, word level TF–IDF feature representation, word2vec, and FastText word embedding were employed with several traditional ML and DL methods. As traditional ML methods, the random forest classifier, XGB, naïve Bayes, SGD, and SVM were investigated. In addition, the CNN, bi-LSTM, and CRNN models were used as DL methods. The findings showed that the TF–IDF word level performed well when employed with traditional ML methods comparing with n-gram TF–IDF. The FastText produced better results with ML methods and the CNN. The word2vec produced some improvement with CNN before optimizing the AUC score, while the RNN benefits more after optimizing the AUC. In [15] Wang et al. suggested combining text content [16], propagation patterns [17], and user feedback. They also analyzed the influence of these combinations on a deep attention model. The proposed model was tested on a set of publicly available datasets. They reported that when they tried to re-obtain the contents of some tweets, about of fifteen percent Twitter data has been lost. The results showed that this approach slightly improved rumor detection in the propagation cycle and achieved a good result with 94.2% accuracy.

In [18], Alsudias and Rayson collected around one million Arabic tweets related to COVID-19. Their aim was not only to detect rumors, but also to identify topics discussed during the period and to find the source of such rumors. For conducting rumor detection, the authors sampled only 2000 tweets and labeled them manually. After that, SVM, LR, and NB classifiers were used to distinguish rumor tweets from non-rumors. The highest achieved accuracy was 84.03%, which was achieved by LR with count vector and SVM with TF–IDF. They also examined the influence of word2vec and FastText on the classifiers' performance. They reported that applying the word embedding approaches did not impact positively on the classifiers' performance.

Apart from Coronavirus related rumors, a large amount of studies can be found in literature that addressed rumor detection via social media in general such as [3,5]. The subsection below briefly presents some DL techniques that are intensively used for detecting rumors via OSN.

A summary of the main existing methods on using machine learning and deep learning methods for detecting rumors is shown in Table 1. The proposed model extends the existing methods in the literature by proposing a new LSTM–PCNN architecture and conducting extensive experiments to demonstrate the performance of the proposed model. In addition, this study investigated the impact of word embedding layers to select the appropriate scheme such as the influence of static word embeddings (word2vec, GloVe, and FastText) on the proposed model. As a result, the proposed model provided interesting results and outperformed the other investigated models in terms of accuracy, recall, precision, and F-score.

Table 1. Summary of Existing COVID-19 Detection Models.

Paper	Dataset	Dataset Size	Technique Applied	Feature Representation	Findings	Limitation
[13]	Handcrafted COVID-19 dataset obtained from Chinese platforms	3737 rumor related data	TextCNN and TextRNN	BERT model	BERT model outperformed the other models. BERT, TextCNN, and TextRNN models all reached more than 90%	The dataset is not available which makes reconducting the model impossible.
[14]	Arabic COVID-19 obtained from Twitter	8786 Tweets	ML and DL methods	word2vec, FASTTEXT, TF–IDF, n-gram	TF–IDF word level was suitable more for ML, while FastText more reliable for DL methods. XGB classifier was the best classifier for identifying Arabic misinformation	Preprocessing limitation
[15]	Set of publicly available datasets	2313 rumor and 2351 non-rumor samples	Attention-based DL method	Not reported	The user feedback provided a clean signal for determining the trend of rumors	Need for optimizing the performance of model and reducing training time
[18]	Handcrafted COVID-19 dataset obtained from Twitter platform	2000 Tweets	ML methods	word2vec, FASTTEXT	The highest accuracy was 84.03% which was achieved by LR with count vector and SVM with TF–IDF	The annotation process is not clear enough

3. Methods

3.1. Deep Learning Techniques

Today, detecting rumors on OSN has gained a significant improvement due to applying DL. According to [19], the main advantage of DL-based techniques is that they do not require any feature engineering. The DL classifier extracts and obtains the useful features directly from the entered data during the training phase. Since there are many proposed DL models, we focused on the models that will be used in this paper to present the proposed model. First, we present an overview of the LSTM architecture and CNN. Then, the word embedding that we used as text representation is also presented in this section.

3.1.1. Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are a special class of recurrent neural networks (RNNs). Since the original RNNs are unable to learn the dependency found in input data especially when the gap is large, LSTM, due to the proposed gate functions, could handle such a problem well [20]. In practice, the powerful learning capacity of the LSTM method makes it one of the most used DL architectures and has been widely used in many fields, such as sentiment analysis [15,21,22], question answering systems [23], sentence embedding [24], and text classification [25].

A typical LSTM has three main gates: an input gate, a forget gate, and an output gate. In addition to the gates, LSTM uses a cell memory state to decide which information to save or discard. Figure 1 shows the original LSTM which was proposed by [26]. The original LSTM has been modified by several researchers. Variations include LSTM without a forget gate, LSTM with a forget gate [27], LSTM with a peephole connection, the gated recurrent unit (GRU) [28], Stacked LSTM [29], and Bi-LSTM [30].

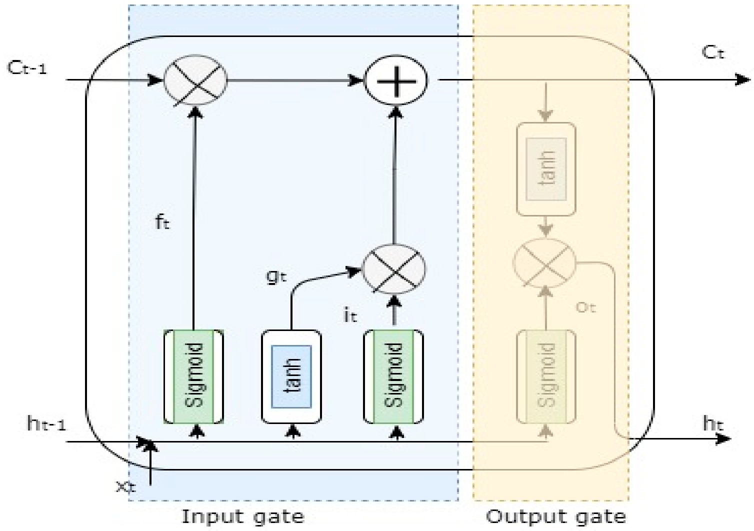

Figure 1. The original LSTM [20].

3.1.2. Convolutional Neural Network

The CNN is another type of DL architecture that has gained more attention in the last few years. The CNN is an unsupervised multilayer feed-forward neural network. It consists of one input layer, one output layer, and the hidden layer that can include any combination of the convolutional layer, nonlinearity, pooling layer, fully connected layer, and regularization. Figure 2 illustrates a typical CNN architecture for binary rumor detection.

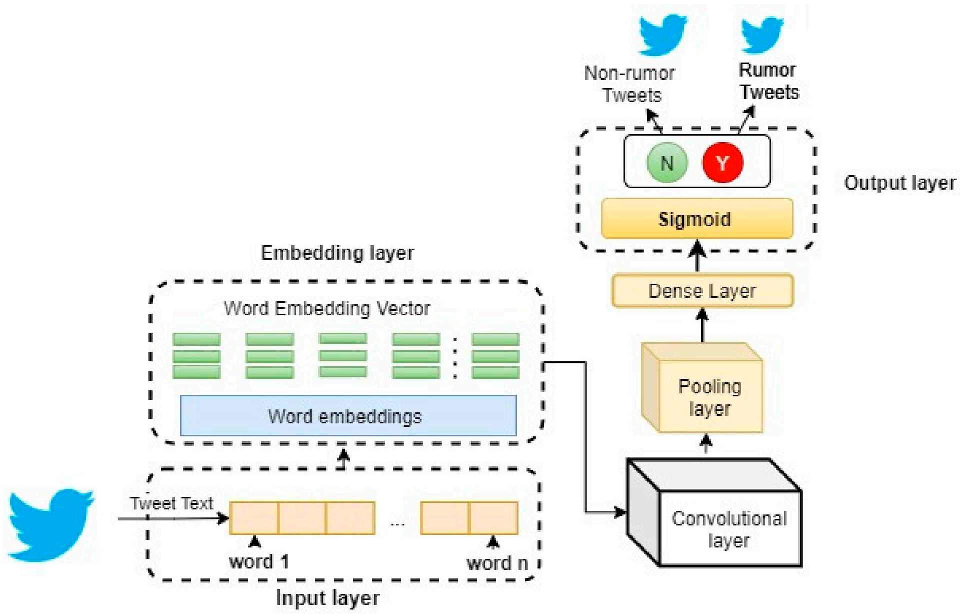

Figure 2. Illustration of a typical CNN architecture for rumor detection.

The CNN has been proven to perform effectively in image classification. Researchers found it a powerful method also in the natural language processing field, such as text classification [31–33]. In [3], authors investigated the influence of CNN on rumor detection task. They found that the CNN can capture rumor features well when the hidden layer is tuned gradually.

- Convolutional layer: For textual data, a convolutional layer is connected to the input layer for extracting features around a particular window, *h*, of words, *w*, referred to as a filter. To capture the useful features, the filter slides across the data. The length of the filter is called the kernel size or window size. Once the features are extracted, the output is passed forward to the next layer.
- Nonlinearity: Here, the goal is to include nonlinear properties in the network. The most used nonlinearity functions in CNN are tanh, sigmoid, and relu. Alsaeedi and Al-Sarem [3] found that the tanh activation function yielded better results compared to sigmoid and relu. Thus, in this paper, we followed their recommendation and empirically assessed the results.
- Pooling layer: Often, the convolutional layer generates feature maps with high dimensionality. Thus, the role of the pooling layer is to reduce the dimensionality by applying a function such as max pooling, average pooling, and stochastic pooling.
- Regularization layers: Similar to the traditional ML methods, the DL also suffers from an overfitting problem. Regularization methods such as early stopping, dropout, and weight penalties are type of techniques that are used for reducing the testing error [34].

3.2. Word Embeddings

Word embedding (WE) is a representation technique of a text where the words with the same meaning have a similar representation. Recently, there have been several word embeddings widely used in ML and DL models. In the literature, there are many pre-trained WEs that can be categorized into two groups [10]: static representation models and contextual models. Word2vec, GloVe, and FastText are types of static WEs that can convert a text into vectors of meaningful representation.

- *Word2vec* works as a language model [35], which is widely used for many NLP tasks. In general, the word2vec embeddings can be obtained using either skip gram or common bag of words (CBOW) [36]. The skip-gram model computes the conditional probability of a word by predicting the surrounding context words given the central target word. The CBOW does the opposite of skip-gram, by computing the conditional probability of a target word give the context words surrounding it across a window of size *k* [23]. Mathematically, both CBOW (Equation (1)) and skip-gram (Equation (2)) models are trained as follows:

$$J = \frac{1}{V}\sum_{t=1}^{V} \log p(w_t | w_{t-c}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+c}) \tag{1}$$

$$J = \frac{1}{V}\sum_{t=1}^{V} \sum_{i=t-c,\ i\neq j}^{t+c} \log p(w_i | w_t) \tag{2}$$

where J is the loss function, $[-c,\ c]$ is the word context of the target word w_t, and V-vocabulary size. In this work, we used both models and the results of their influence on the proposed model was examined. The pre-trained word2vec word embeddings have 300 features which trained on 100 billion words.
- *GloVe* is an unsupervised training “count-based” model [23]. Opposite to word2vec, the *GloVe* word embedding generates the embedding vector using word occurrences. Formally, the space vector is computed using a weighted least-squares method (Truşcă et al., 2020) as follows:

$$J = \sum_{\mathrm{i}=1}^{V}\sum_{\mathrm{j}=1}^{V} \mathrm{f}(\mathrm{X}_{ij}) \left(\mathrm{w}_i^{\mathrm{T}}\, \mathrm{w}_j + \mathrm{b_i} + \mathrm{b_j} - \log(\mathrm{X}_{ij})\right)^2 \tag{3}$$

where V is the vocabulary size and $\mathrm{f}(\mathrm{X}_{ij})$ is a weighting function. The smallest package of the embedding is 822Mb, called “glove.6B.zip”. The GloVe model is trained on a dataset having one billion words with a dictionary of 400 thousand words. There are different embedding vector sizes, with 50, 100, 200, and 300 dimensions for processing. In this paper, we used the 100 dimensional version.

– *Fast Text* is an extension of the word2vec approach where the word embedding is represented using n-gram [37]. Once the word has been represented using n-grams, a skip-gram model or CBOW is trained to learn the embeddings. Today, the pre-trained FastText word vector supports 157 languages. The main parameters that need to be adjusted before using the FastText word embedding are the dimension and the range of subwords size. By default, the size of 100 dimensions is used. However, it is allowed to have a value in the 100–300 range. In this paper, we set the dimensionality of word embeddings to 300.

3.3. The Proposed Method

In this work, we propose a hybrid deep learning-based model LSTM–PCNN to detect rumors on Twitter. The proposed model hybridizes LSTM architecture with three parallel CNN models. The structure of the LSTM–PCNN model is shown in Figure 3.

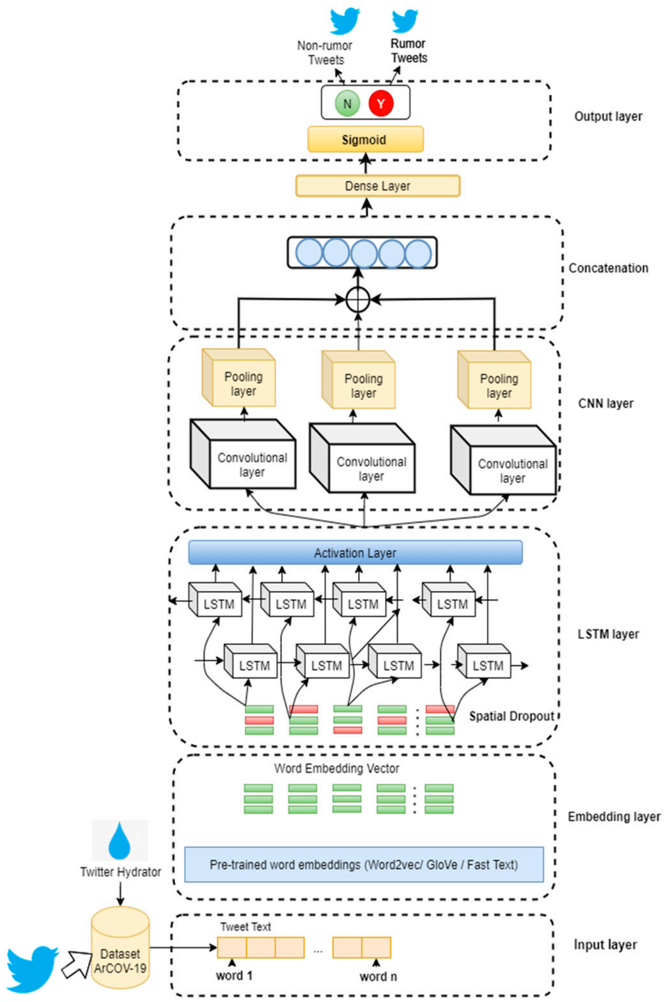

Figure 3. Structure of the LSTM–PCNN model.

3.3.1. Input Layer

There are several publicly available datasets. Table 2 presents the existing available publicly datasets. In this work, we used the ArCOV-19 dataset. The original dataset contains 95,000 tweets; out of them, 3612 tweets were annotated (the full description of the dataset is presented in Section 4.1). Since the maximum length of a tweet written in Arabic is 280 characters, the input layer was set to cover the maximum length as shown in the input layer in Figure 3. Before feeding the tweet into the next layer, a set of preprocessing techniques were applied. The complete process is discussed in Section 4.2.

Table 2. Publicly Available Arabic Datasets.

Dataset Name	Size	Dataset Purpose	Link
ArCorona-ver1	8000	Understanding public behavior, Sources of tweets, Topics of interest, and Requests from governments.	https://alt.qcri.org/~hmubarak/ArCorona-ver1.tsv last access: 26 August 2021
COVID-19-FAKES	10,000	Fake information detection	https://github.com/mohaddad/COVID-FAKES/tree/COVID-FAKES-A (last access: 26 August 2021)
COVID-19-Arabic-Tweets-Dataset	94,000	Sources of tweets and Topics of interest	https://github.com/SarahAlqurashi/COVID-19-Arabic-Tweets-Dataset https://github.com/mohaddad/COVID-FAKES/tree/COVID-FAKES-A (last access: 26 August 2021)
GeoCoV19	5.4 M	Understanding public reactions and sentiment	https://crisisnlp.qcri.org/covid19 (last access: 26 August 2021)
ArCOV-19 dataset	2.7 M	Propagation networks, Fake information detection	https://gitlab.com/bigirqu/ArCOV-19 last access 26 August 2021

3.3.2. Embedding Layer

As shown in Figure 3, we employed three different pre-trained embedding layers, namely, word2vec, GloVe and Fast Text model. Each word embedding was fed separately into the LSTM layer. Table 3 shows the tuned hyper parameters of each used model. It is important to highlight that GloVe word embedding is a pre-trained model, while the other models are trained from the training data.

Table 3. Configuration of the used Word Embedding.

Word Embedding Model	Configuration
Word2vec	Word vector dimension = 300, window = 3, min_count = 1, sg = {0,1}
GloVe	Default
FastText	Default

3.3.3. Long Short-Term Memory Layer

The output of the embedding layer was a vector with a predefined size in which the words per tweet w_t were embedded. However, before feeding the output to the LSTM, we used a spatial dropout layer [38]. The spatial dropout layer has proven its benefit for improving the performance of CNN architecture [39] and avoiding overfitting in LSTM [40,41]. In this work, we suggest adding one spatial dropout layer before feeding the output into the LSTM layer. Table 4 presents the layered architecture of LSTM model.

Table 4. LSTM Layered Architecture Layer.

Layer	Input Dimension	Output Dimension	Activation Function	Dropout
LSTM	400	250	*tanh*	0.3

3.3.4. Convolutional Neural Network Layer

As shown in Figure 3, the LSTM layer was followed by three parallel CNN layers. Each block generated a 150-dimensional vector F_t^i that indicated word features, where i is the number of CNN block and F_t represents features obtained by each block. The configuration of each CNN block is presented as shown in Table 5.

Table 5. CNN Layered Architecture.

Layer	Input Dimension	Output Dimension	Kernel Size	Padding	Activation Function
Conv1D	250	150	1	valid	relu
Conv1D	250	150	2	valid	relu
Conv1D	250	150	3	valid	relu

3.3.5. Concatenation Layer

As described earlier, each CNN block generated a 150-dimensional vector. So, we concatenated each feature F_t obtained by each block. As a result, we obtained a 450-dimensional vector F. Thus, the vector F is given by:

$$F = F_t^1 \oplus F_t^2 \oplus F_t^3 \tag{4}$$

3.3.6. Output Layer

Finally, the vector F was passed into the output layer. Since, the rumor detection task can be considered as a binary classification task, the vector F was passed into the Sigmoid function, which can take a value of either 0 or 1 as follows:

$$p = Sigmoid(F) \tag{5}$$

$$y = \begin{cases} 0, & p \in [0,\ 0.5) \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

where p is the possibility that the tweet is a rumor or non-rumor. The y is the classification result where $y = 0$ indicates that the tweet is non-rumor and $y = 1$ indicates a rumor tweet.

4. Experimental Design

In this paper, the experiments were conducted to evaluate the performance of the proposed LSTM–PCNN model. Therefore, we implemented two baseline DL-based models: (i) LSTM, and (ii) Parallel CNN. The experimental part of this work was performed on the Keras 2.2.4 API with TensorFlow backend using Python 3.6 with Windows 10 operating system. In addition, the used dataset, preprocessing methods, and the evaluation metrics are presented and explained in this section.

4.1. Data Sets

The ArCOV-19 dataset is a collection of Arabic tweets about the COVID-19 pandemic, considering the most common public dataset covering the period from 27 January to 30 April 2020. The Twitter API were used to collect the Arabic tweets based on manually-entered queries targeting COVID-19 topics, including keywords such as “Corona,” hashtags such as “#coronavirus,” or phrases such as “COVID-19 pandemic.” The search queries were customized to remove all retweets, avoid duplicate tweets, and return Arabic tweets only in chronological order. The ArCOV-19 dataset comprised 94K tweets. The original dataset contained 3612 tweets (Last access was on 10 March 2021. Thus, the number of collected tweets might have increased.). Since the ArCOV-19 dataset complied with the Twitter content redistribution policy, only the tweet IDs were published publicly. Therefore, the full object of tweets was obtained using the Hydrator tool to obtain tweets in JSON format for the given tweets’ IDs. Due to the inaccessibility of some tweets (deleted tweets or deactivated accounts), the total number of tweets we retrieved was reduced to 3157 tweets, including 1480 rumors (46.87%) and 1677 non-rumors (53.12%). The dataset included several types of rumors related to COVID-19 such as social, political, health, and religious rumors. The main motivations for distributing these rumors were to provide health and social awareness and information about COVID-19. Some of these rumors were political and used to distribute misinformation against specific countries, while others tried to circulate rumors about the treatment of COVID-19 by taking the form of religious advice. By reviewing the rumors in this dataset, the majority of these rumors fell into the sociological and political types. Table 6 shows examples of these rumors.

Table 6. Rumor Types and Examples.

Rumor in Arabic	Translated Text	Type of Rumor
قرية إندونيسية تستعين بالأشباح لإجبار الناس على البقاء في منازلهم في ظل فيروس كورونا	An Indonesian village uses ghosts to force people to stay indoors during the Corona virus	Social
وباء قاتل كل 100 سنة وسر خطير في رقم 20	A fatal epidemic every 100 years and a dangerous secret in number 20	Social
سبب انتشار فيروس كورونا يعود لإنتشار شبكات الجيل الخامس5Gالصيني	The cause of the spread of the Corona virus is due to the spread of Chinese 5G networks	Political
في الاكوادور بسبب انهيار النظام الصحي و قلة الموارد و ايضا فساد وضعف المسؤولين يتم رمي جثث المصابين بفايروس كورونا في الشوارع وفي المستشفيات يتم وضعها باكياس الزبالة ورميها لدرجة ان الغربان والطيور اصبحت تنهشها	In Ecuador, due to the collapse of the health system and the lack of resources, as well as the corruption and weakness of officials, the corpses of people infected with the Corona virus are thrown in the streets and in hospitals, they are placed in rubbish bags and thrown to the point that crows and birds are eating them	Political
الغرغرة بالملح و تنظيف الأنف طريقة فعالة لحمايتك من فيروس كورونا	Gargling with salt and cleaning the nose is an effective way to protect you from the Corona virus	Health
!ترامب يقترح استخدام المطهرات لحقن مصابي #كورونا	Trump suggests using antiseptics to inject Corona patients!	Health
الوضوء هو اللي يحمي الانسان من الامراض المعديه	Ablution is what protects a person from infectious diseases	Religious

4.2. Data Preprocessing

Several preprocessing steps were performed to prepare the tweets' texts before feeding them into the embedding layer and the proposed deep learning classification models. First, we handled URLs by replacing them with "رابط" meaning "hyperlink, mention character (@) removal, hashtag character (#) removal, handling words with repeating characters, numbers removal, and emoticon handling by replacing positive emoticons with a 'إيجابي' meaning 'positive' word and negative emoticons with a 'سلبي' meaning 'negative' word. In addition, we removed punctuation and additional white spaces. We also normalized non-Arabic letters by converting them into Arabic using manually crafted translator. After that, we utilized the PyArabic library to normalize both "hamza" and ligature, and to strip both "tatweel" and "tashkeel". In addition to the above steps, we performed the stemming process using the snowball stemmer and removed stop words from the text. Figure 4 shows an illustrative example of a tweet after applying some of the preprocessing techniques.

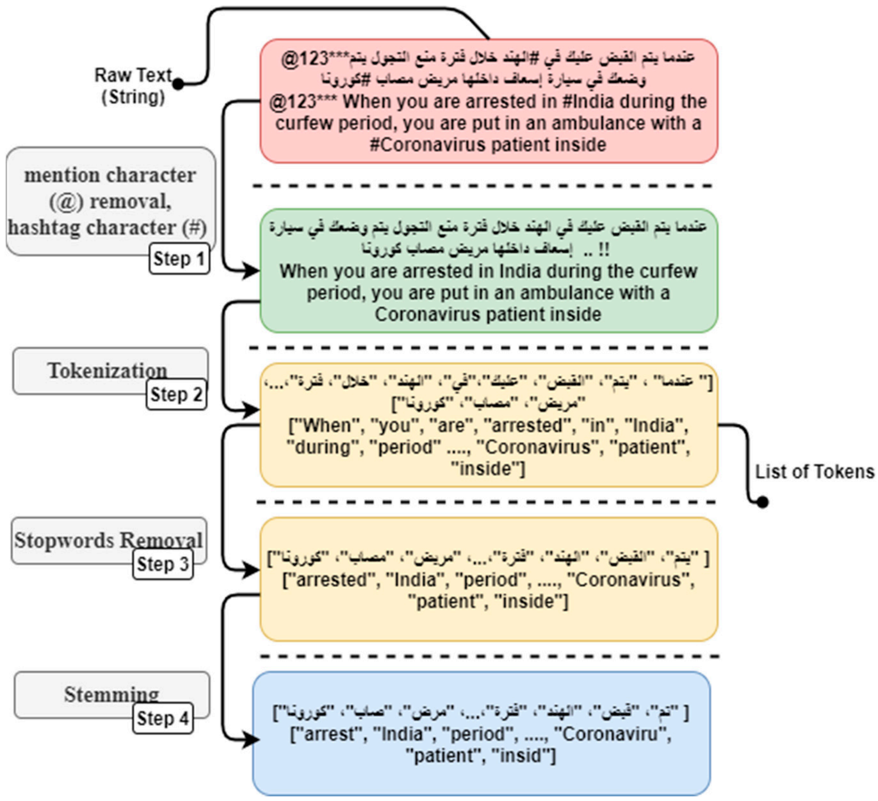

Figure 4. Preprocessing phase: an illustrative example.

4.3. Evaluation Metrics

To evaluate the performance of the proposed model, the following performance measures were used: classification accuracy, precision, recall, and F1 score. In addition, we present the confusion matrix per each fold (refer to Table 7 for more details). These measures are commonly used by researchers to evaluate the performance of a rumor detection system. In order to precisely assess the proposed method, all the conducted experiments were validated using fivefold cross-validation.

Table 7. Confusion matrix.

	Predicted Negative	**Predicted Positive**
Actual Negative	True negative (TN)	False positive (FP)
Actual Positive	False negative (FN)	True positive (TP)

5. Experimental Results

The results shown in this section are the average value of each experiment that was repeated, as stated earlier, five times independently.

5.1. Evaluation of the Embeddings

To choose the appropriate embedding extractor in the LSTM–PCNN model, we applied different static word embedding models: word2vec, GloVe, and FastText. The structures of the other parts in the DL models remain unchanged. Later, in the next section, we examined the performance of adding more dense layers to the models under investigation. The performance of baselines models with different embeddings is shown in Table 8.

Table 8. The performance of baseline models with different embeddings.

Word Embedding	Baseline Classifier	Accuracy	Precision	Recall	F-Score
Word2Vec (CBOW)	PCNN	0.8209	0.8257	0.8209	0.8213
	LSTM	0.8304	0.8334	0.8304	0.8295
Word2Vec (Skip-Gram)	PCNN	0.8320	0.8319	0.8320	0.8319
	LSTM	0.8526	0.8527	0.8526	0.8526
Fast Text (CBOW)	PCNN	0.8399	0.8405	0.8399	0.8401
	LSTM	0.8352	0.8422	0.8352	0.8349
Fast Text (Skip-Gram)	PCNN	0.8542	0.8569	0.8542	0.8545
	LSTM	0.8447	0.8448	0.8447	0.8447
GloVe	PCNN	0.8542	0.8558	0.8542	0.8541
	LSTM	0.8479	0.8494	0.8479	0.8478

It is important to report that we have trained all the word embedding models (without finetuning) on AraCOV−19 data for the fairness of the experiment. As shown in Table 8, FastText outperformed both word2vec and GloVe. The PCNN model benefitted more from the Fast Text and GloVe embeddings compared to word2vec. However, LSTM showed an improvement when the word2vec skip-gram model is used. Therefore, at this stage it was difficult to decide which model we should use. For this reason, we investigated the impact of these word embeddings on the proposed LSTM–PCNN model. As shown in Figure 5, the proposed LSTM–PCNN, unlike expected, achieved the highest performance when the word2vec skip-gram model was used. In addition, comparing the proposed model with the other baseline models, LSTM–PCNN achieved the best result among all the models. Figures 6 and 7 present the structure of LSTM and PCNN models, respectively.

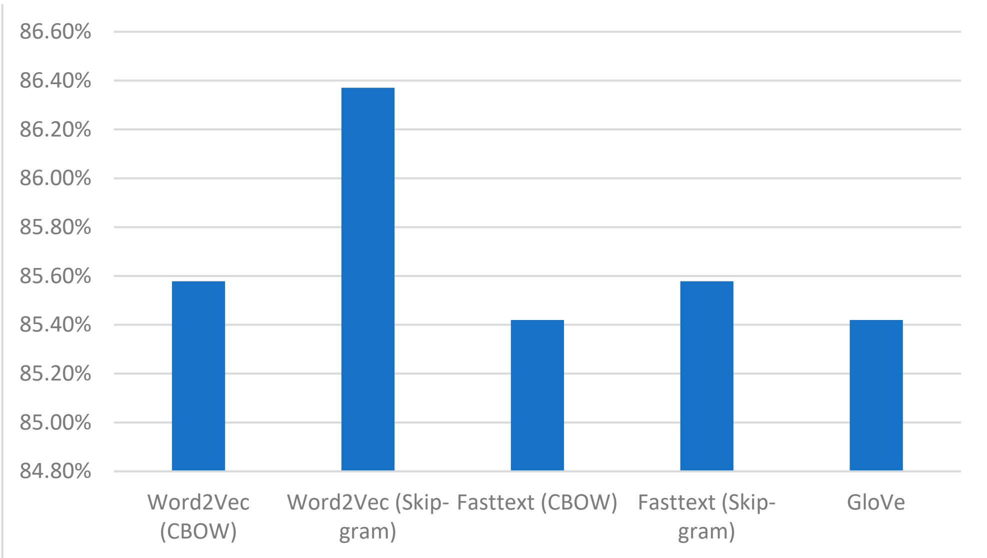

Figure 5. Accuracy of the LSTM–PCNN model with different word embedding.

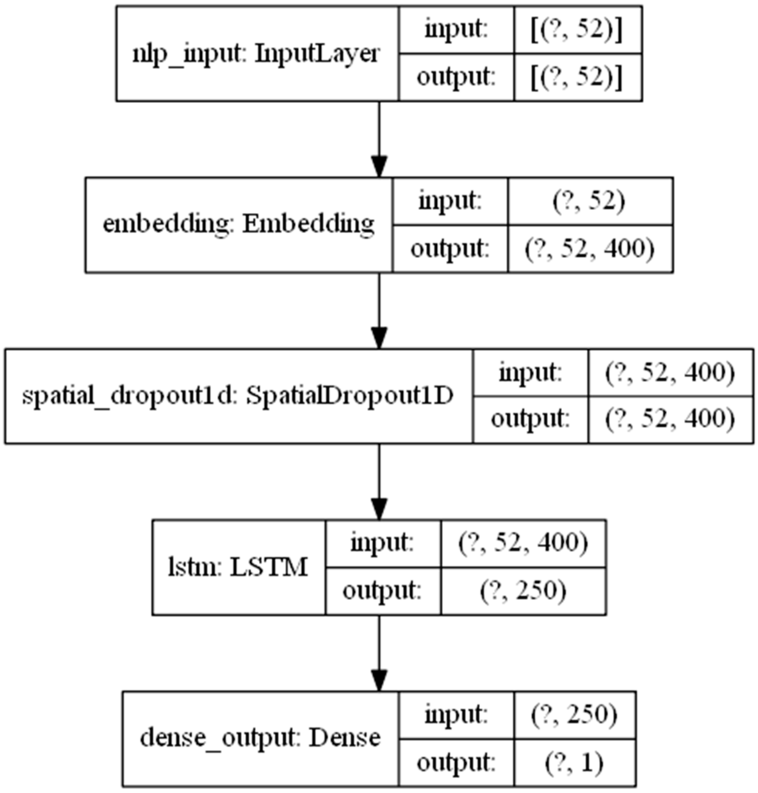

Figure 6. Architecture of the baseline: LSTM model.

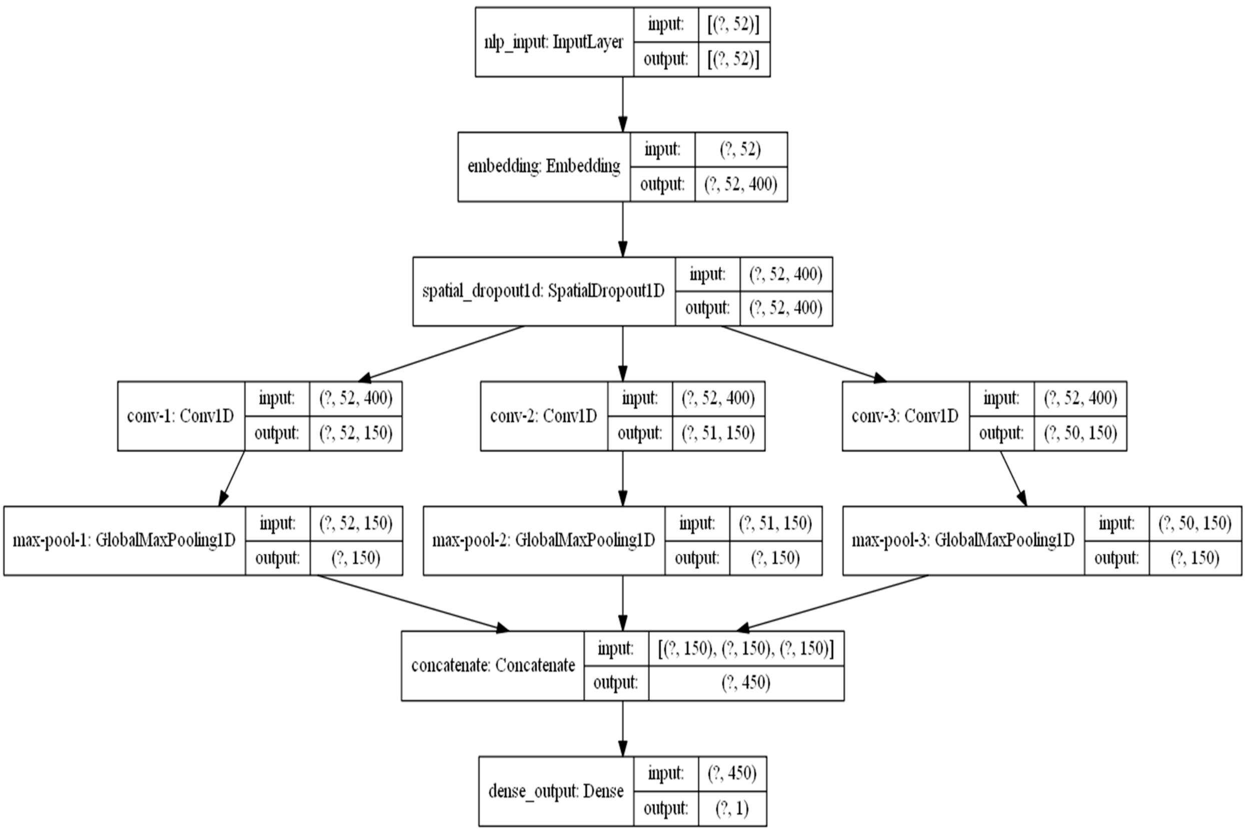

Figure 7. Architecture of the baseline: PCNN model.

5.2. Evaluation of Adding more Dense Layers

Similar to what we completed in the previous section, the influence of adding dense layers on the performance of the implemented DL models was investigated. Thus, to evaluate the contributions of these layers to the models, we added them gradually in turn to the model. Tables 9 and 10 show the median values obtained by adding more layers to the proposed LSTM–PCNN. The rest of the results are in "Appendix A" (see Tables A1 and A2).

Table 9. The Median Values Obtained using One Dense Layer.

Embedding Layers	Accuracy	Precision	Recall	F-Score
Word2Vec (CBOW)	85.10%	0.85093	0.85103	0.85097
Word2Vec (Skip-gram)	85.74%	0.86054	0.85737	0.85767
Fast Text (CBOW)	85.42%	0.85577	0.85420	0.85388
Fast Text (Skip-gram)	83.99%	0.84059	0.83994	0.83891
GloVe	85.42%	0.85584	0.85420	0.85449

Table 10. The Median Values Obtained using Three Dense Layers.

Embedding Layers	Accuracy	Precision	Recall	F-Score
Word2Vec (CBOW)	84.79%	0.84762	0.84786	0.84760
Word2Vec (Skip-gram)	84.79%	0.84971	0.84786	0.84775
Fasttext (CBOW)	85.26%	0.85560	0.85261	0.85294
Fasttext (Skip-gram)	86.05%	0.86406	0.86054	0.86084
GloVe	85.10%	0.85553	0.85103	0.85136

6. Conclusions

The paper proposed a novel hybrid deep learning model for detecting COVID-19-related rumors on social media based on a long short-term memory and concatenated parallel convolutional neural networks (LSTM–PCNN). The conducted experiments used three static word embedding models, which are word2vec, GloVe, and FastText. The experimental results showed that the proposed LSTM–PCNN model achieved the highest performance when the word2vec skip-gram model was used, and it outperformed the other baseline models, where the obtained detection accuracy reached 86.37%. The experiments also investigated adding more dense layers to the architecture of the proposed model leads. It was found that, in most cases, this adding degraded the overall performance. Statistical analysis was conducted using the Mann-Whitney-Wilcoxon test and the Wilcoxon signed-rank test and the findings showed that adding more "Dense layers" did not improve the performance of the proposed model. As the rumors have negative impact on the social and political aspects of many countries, the proposed model can help the health and other governmental authorities to automatically detect fake information about COVID-19 on social media and mitigate this impact. In future work, other datasets with Arabic tweets could be used, and other deep learning-based methods could be proposed and investigated to enhance the detection of health-related rumors in Arabic and other languages.

Author Contributions: Conceptualization, M.A.-S., A.A. and F.S.; methodology, M.A.-S., A.A. and F.S.; Software, A.A. and M.A.-S.; validation, W.B. and O.A.; formal analysis, F.S., M.A.-S., W.B. and O.A.; investigation, M.A.-S., F.S. and W.B.; data curation, M.A.-S. and F.S.; writing—original draft, M.A.-S.; writing—review & editing, F.S., W.B., O.A. and A.A.; visualization, M.A.-S.; supervision, M.A.-S.; project administration, M.A.-S. and F.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Deanship of Scientific Research at Taibah University, Saudi Arabia, project number (CSE—4).

Acknowledgments: The authors would like to thank the Deanship of Scientific Research at Taibah University, Saudi Arabia, for funding this research project number (CSE—4).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. The Median Values obtained using Various Classifiers with One Dense Layer.

Word Embedding	Classifier	Accuracy	Precision	Recall	F-Score
Word2Vec-CBOW	LSTM-PCNN	0.851030111	0.850930411	0.851030111	0.850971622
	PCNN	0.820919176	0.824961489	0.820919176	0.821206478
	LSTM	0.839936609	0.839783925	0.839936609	0.839841079
Word2Vec -Skip grams	LSTM-PCNN	0.857369255	0.860538827	0.857369255	0.857674224
	PCNN	0.82725832	0.829704392	0.82725832	0.827321158
	LSTM	0.844690967	0.849820666	0.844690967	0.844944908
Fasttext-CBOW	LSTM-PCNN	0.854199683	0.855772847	0.854199683	0.853882674
	PCNN	0.839936609	0.839878863	0.839936609	0.839905606
	LSTM	0.838351823	0.8395801	0.838351823	0.838661839
Fasttext-Skip grams	LSTM-PCNN	0.839936609	0.840589106	0.839936609	0.8389057
	PCNN	0.838351823	0.846783629	0.838351823	0.838476882
	LSTM	0.841521395	0.841705837	0.841521395	0.840747605
Glove	LSTM-PCNN	0.854199683	0.855841279	0.854199683	0.854492222
	PCNN	0.856369255	0.861367633	0.857369255	0.857680946
	LSTM	0.852614897	0.86374224	0.852614897	0.85273262

Table A2. The Median Values obtained using Various Classifiers with Three Dense Layers.

Word Embedding	Classifier	Accuracy	Precision	Recall	F-Score
Word2Vec-CBOW	LSTM-PCNN	0.847860539	0.847620683	0.847860539	0.847602354
	PCNN	0.816164818	0.832481927	0.816164818	0.815732473
	LSTM	0.851030111	0.85332578	0.851030111	0.851323063
Word2Vec -Skip grams	LSTM-PCNN	0.847860539	0.849708961	0.847860539	0.84775121
	PCNN	0.846275753	0.857998622	0.846275753	0.84559449
	LSTM	0.841521395	0.851922253	0.841521395	0.84144736
Fasttext-CBOW	LSTM-PCNN	0.852614897	0.855598804	0.852614897	0.852936834
	PCNN	0.838351823	0.838488951	0.838351823	0.838411901
	LSTM	0.851030111	0.853978739	0.851030111	0.851469463
Fasttext-Skip grams	LSTM-PCNN	0.860538827	0.864062368	0.860538827	0.860840177
	PCNN	0.830427892	0.848173415	0.830427892	0.830666543
	LSTM	0.836767036	0.846330271	0.836767036	0.836393332
Glove	LSTM-PCNN	0.851030111	0.855528458	0.851030111	0.851356306
	PCNN	0.852614897	0.855580581	0.852614897	0.852399986
	LSTM	0.844690967	0.844797684	0.844690967	0.8447358

References

1. Zhang, C.D.; Pan, Y.P. Critical Information Detection for the Prevention and Control of Burst Cybercrime Events Under the Background of Big Data. In Proceedings of the 3rd International Conference on Wireless Communication and Sensor Networks, Wuhan, China, 10–11 December 2016; pp. 365–368.
2. Fontanilla, M.V. Cybercrime pandemic. *Eubios J. Asian Int. Bioeth.* **2020**, *30*, 161–165.
3. Alsaeedi, A.; Al-Sarem, M. Detecting Rumors on Social Media Based on a CNN Deep Learning Technique. *Arab. J. Sci. Eng.* **2020**, *45*, 10813–10844. [CrossRef]
4. Sun, S.; Liu, H.; He, J.; Du, X. Detecting Event Rumors on Sina Weibo Automatically. In *Proceedings of the Asia-Pacific Web Conference, Sydney, Australia, 4–6 April 2013*; Springer: Berlin, Heidelberg, 2013; pp. 120–131.
5. Zubiaga, A.; Liakata, M.; Procter, R. Exploiting Context for Rumour Detection in Social Media. In *Proceedings of the International Conference on Social Informatics, Oxford, UK, 13–15 September 2017*; Springer: Cham, Switzerland, 2017; pp. 109–123.
6. Santhoshkumar, S.; Babu, L.D. Earlier detection of rumors in online social networks using certainty-factor-based convolutional neural networks. *Soc. Netw. Anal. Min.* **2020**, *10*, 1–17. [CrossRef]
7. Xu, F.; Sheng, V.S.; Wang, M. Near real-time topic-driven rumor detection in source microblogs. *Knowl.-Based Syst.* **2020**, *207*, 106391. [CrossRef]
8. Alkhodair, S.A.; Ding, S.H.; Fung, B.C.; Liu, J. Detecting breaking news rumors of emerging topics in social media. *Inf. Process. Manag.* **2020**, *57*, 102018. [CrossRef]

9. Providel, E.; Mendoza, M. Using Deep Learning to Detect Rumors in Twitter. In *Proceedings of the International Conference on Human-Computer Interaction, Copenhagen, Denmark, 19–24 July 2020*; Springer: Cham, Switzerland, 2020; pp. 321–334.
10. Kaliyar, R.K.; Goswami, A.; Narang, P. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimed. Tools Appl.* **2021**, *80*, 11765–11788. [CrossRef]
11. Haouari, F.; Hasanain, M.; Suwaileh, R.; Elsayed, T. Arcov-19: The first Arabic covid-19 twitter dataset with propagation networks. *arXiv* **2020**, arXiv:2004.05861.
12. Luo, J.; Xue, R.; Hu, J. COVID-19 infodemic on Chinese social media: A 4P framework, selective review and research directions. *Meas. Control.* **2020**, *53*, 2070–2079. [CrossRef]
13. Chen, S. Research on Fine-Grained Classification of Rumors in Public Crisis—Take the COVID-19 incident as an example. In *E3S Web of Conferences*; EDP Sciences: Les Ulis, France, 2020; Volume 179, p. 02027.
14. Alqurashi, S.; Hamoui, B.; Alashaikh, A.; Alhindi, A.; Alanazi, E. Eating garlic prevents covid-19 infection: Detecting misinformation on the Arabic content of twitter. *arXiv* **2021**, arXiv:2101.05626.
15. Wang, L.; Wang, W.; Chen, T.; Ke, J.; Tang, B. Deep Attention Model with Multiple Features for Rumor Identification. In *Proceedings of the International Conference on Frontiers in Cyber Security, Tianjin, China, 15–17 November 2020*; Springer: Singapore, 2020; pp. 65–82.
16. Takahashi, T.; Igata, N. Rumor detection on twitter. In *Proceedings of the 6th International Conference on Soft Computing and Intelligent Systems, and the 13th International Symposium on Advanced Intelligence Systems, Kobe, Japan, 20–24 November 2012*; IEEE: Piscataway, NJ, USA, 2012; pp. 452–457.
17. Ma, J.; Gao, W.; Wong, K.F. Detect rumors in microblog posts using propagation structure via kernel learning. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 708–717.
18. Alsudias, L.; Rayson, P. COVID-19 and Arabic Twitter: How can Arab World Governments and Public Health Organizations Learn from Social Media? In Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020, Seattle, WA, USA, 9–10 July 2020.
19. Al-Sarem, M.; Boulila, W.; Al-Harby, M.; Qadir, J.; Alsaeedi, A. Deep learning-based rumor detection on microblogging platforms: A systematic review. *IEEE Access* **2019**, *7*, 152788–152812. [CrossRef]
20. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef] [PubMed]
21. Zhang, Y.; Tiwari, P.; Song, D.; Mao, X.; Wang, P.; Li, X.; Pandey, H.M. Learning interaction dynamics with an interactive LSTM for conversational sentiment analysis. *Neural Netw.* **2021**, *133*, 40–56. [CrossRef] [PubMed]
22. Yang, J.; Zou, X.; Zhang, W.; Han, H. Microblog sentiment analysis via embedding social contexts into an attentive LSTM. *Eng. Appl. Artif. Intell.* **2021**, *97*, 104048. [CrossRef]
23. Huang, Z.; Xu, S.; Hu, M.; Wang, X.; Qiu, J.; Fu, Y.; Wang, C. Recent trends in deep learning based open-domain textual question answering systems. *IEEE Access* **2020**, *8*, 94341–94356. [CrossRef]
24. Palangi, H.; Deng, L.; Shen, Y.; Gao, J.; He, X.; Chen, J.; Ward, R. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 694–707. [CrossRef]
25. Li, Y.; Wang, X.; Xu, P. Chinese text classification model based on deep learning. *Future Internet* **2018**, *10*, 113. [CrossRef]
26. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
27. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471. [CrossRef]
28. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
29. Fernández, S.; Graves, A.; Schmidhuber, J. Sequence labelling in structured domains with hierarchical recurrent neural networks. In Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI 2007, Hyderabad, India, 6–12 January 2007; pp. 774–779. Available online: http://ijcai.org/Proceedings/07/Papers/124.pdf (accessed on 28 June 2021).
30. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
31. Alhawarat, M.; Aseeri, A.O. A Superior Arabic Text Categorization Deep Model (SATCDM). *IEEE Access* **2020**, *8*, 24653–24661. [CrossRef]
32. El-Alami, F.Z.; El Alaoui, S.O.; En-Nahnahi, N. Deep neural models and retrofitting for Arabic text categorization. *Int. J. Intell. Inf. Technol.* **2020**, *16*, 74–86. [CrossRef]
33. Zhang, Y.; Wallace, B. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In Proceedings of the Eighth International Joint Conference on Natural Language Processing, Taipei, Taiwan, 27 November–1 December 2017; pp. 253–263.
34. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA USA, 2016.

35. Truşcă, M.M.; Wassenberg, D.; Frasincar, F.; Dekker, R. A hybrid approach for aspect-based sentiment analysis using deep contextual word embeddings and hierarchical attention. In *Proceedings of the International Conference on Web Engineering: 20th International Conference, ICWE 2020, Helsinki, Finland, 9–12 June 2020*; Springer Nature: Gewerbestrasse, Switzerland, 2020; Volume 12128, pp. 365–380.
36. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013*; Curran Associates Inc.: Red Hook, NY, USA, 2013; Volume 2, pp. 3111–3119.
37. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
38. Tompson, J.; Goroshin, R.; Jain, A.; LeCun, Y.; Bregler, C. Efficient object localization using convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 648–656.
39. Chammas, E.; Mokbel, C. Fine-tuning Handwriting Recognition systems with Temporal Dropout. *arXiv* **2021**, arXiv:2102.00511.
40. Sarkar, K. Sentiment polarity detection in Bengali tweets using LSTM recurrent neural networks. In *Proceedings of the 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), Gangtok, India, 25–28 February 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
41. Ranasinghe, T.; Zampieri, M.; Hettiarachchi, H. BRUMS at HASOC 2019: Deep Learning Models for Multilingual Hate Speech and Offensive Language Identification. 2019, pp. 199–207. Available online: http://ceur-ws.org/Vol-2517/T3-3.pdf (accessed on 30 July 2021).