Fake News Detection in Social Media: Hybrid Deep Learning Approaches

Fatoumata Wongbé Rosalie Tokpa 1,*, Beman Hamidja Kamagate 2, Vincent Monsan 3, and Souleymane Oumtanaga 4

1 Unité de Formation et de Recherche Mathématiques Informatique, Université Félix Houphouët-Boigny, Abidjan, Côte d´Ivoire
2 Laboratoire des Sciences et Technologies de l’Information et de la Communication, Ecole Supérieure Africaine des TIC, Abidjan, Côte d’Ivoire; Email: beman.kamagate@esatic.edu.ci (B.H.K.)
3 Université Félix Houphouët-Boigny, Abidjan, Côte d’Ivoire; Email: vmonsan@yahoo.fr (V.M.)
4 Institut National Polytechnique Félix Houphouët-Boigny, Yamoussoukro, Côte d’Ivoire; Email: oumtana@gmail.com (S.O.)

*Correspondence: tokpa.rosalie12@ufhb.edu.ci (F.W.R.T.)

Abstract—Social media refers to communication channels on Internet that enable the creation and publication of content generated by the user and interaction between users. Given the accessibility to these means of communication and their rapidity, people resort more to them comparatively to the traditional media including radio, television and newspapers. However, dubious pieces of information such as fake news are often disseminated for malicious purposes. The proliferation of fake news has a strong negative impact on a society such as damage to the reputation of a personality, an organization or the aggravation of conflicts between its members. Due to the proliferation of fake news on these websites, the notion of veracity of information becomes a crucial issue. Research based on machine learning is promising. However, one of the main limitations is the efficiency of predictions. As a solution to detect fake news, we have proposed two models based on hybrid deep learning and evaluated our models on the two real datasets, namely ISOT and FA-KES. An experience of the proposed models to detect fake news, allowed to obtain on ISOT an accuracy of 99% for both models and on FA-KES, we obtain an accuracy of 68% for one the models and an accuracy of 63% for other. Other experiments in generalizing models on these data sets have proposed. The results obtained are better than other machine learning models.

Keywords—veracity, fake news, social media, artificial intelligence, machine learning, convolution neural network, recurrent neural network

I. INTRODUCTION

Dixon [1] predicts that by 2027, over 5.8 billion people worldwide will use social media. Social media refers to communication channels on the Internet such as Wikinews, Google plus, Twitter, Facebook, etc., that enable the creation and publication of contents generated by user and interaction between users.

The political, economic and cultural stakes and contradictions between members of a society lead interest groups to use them to manipulate populations by producing fake news to control their state of mind or to maintain an anxiety-provoking climate for their own benefit. An increasing massification of the means of connection and the existence of numerous contents that are produced instantaneously at the time of real events such as conflicts, earthquakes, sports and elections, amplify the production of fake news on social media.

In recent years, fake news detection has received much attention from the scientific community. Several approaches to fake news detection have been proposed, including expert or crowd verification [2–5], ontology-based [6, 7] or machine learning-based [8–10].

To curb the proliferation of fake news on social media, approaches have proposed models of fake news detection based on machine learning [8–11]. However, one of the main limitations is the efficiency of predictions. To solve this problem, techniques based on hybrid deep learning models can be used for classification [12–15].

Our study aims to classify a news provided as true news or fake news.

In this paper, our main contributions are summarized below:

- To propose an hybrid deep learning model composed of a Convolutional Neural Network (CNN) and a recurrent neural network Long Short Term Memory (LSTM) called DeepCnnLstm which ensures the extraction of spatial and contextual features in the forward direction of text;
- To propose another derivative of the previous model composed of a Convolutional Neural Network (CNN) and a recurrent neural network Bidirectional Long Short Term Memory (Bi-LSTM) called DeepCnnBilstm which ensures the extraction of spatial and contextual features in the forward and backward direction of text;
- To train and to test these models on real data sets ISOT and FA-KES.

Manuscript received November 4, 2022; revised December 15, 2022, accepted February 16, 2023; published June 29, 2023.

doi: 10.12720/jait.14.3.606-615
• To train each proposed model on ISOT and to test each proposed model on FA-KES to evaluate the performance of the models in the generalization case;
• To compare the performances with those of baselines models [14] and other machine learning models.

The remainder of this paper is organized as follows: Section II analyzes existing works in the field of fake news detection based on the neural network approach. Section III devotes to the methodology used. This part explains our approach and the proposed algorithms. Then, Section IV deals with the different experiments carried out during this work, analyzes them and interprets the main results. Finally, Section V gives an overview of our study and highlights future works.

II. RELATED WORKS

This section deals with the exploration of proposed studies on Fake News (FN), specifically based on the neural network learning approach.

These studies detect FN using the style of a news, style as subdivided in textual or visual features (image or video), extracted from the contents of the news article [4].

Textual features are extracted by the bag-of-words approach or by text embedding. In the bag-of-words approach, individual words or n-grams are analyzed to reveal indices that allow the veracity of the information to be estimated, such as the readability index and the frequency of words [16]. This analysis can be at the syntactic level using probabilistic Context Free Grammar [17], at the semantic level by introducing the description of the user's personal experience [18], as well as at the level of discourse analysis and rhetorical structure [19, 20]. The disadvantage of this approach is that n-grams are often separated from useful contextual information, so that resolving the meaning of a word is always problematic [16].

In the text embedding approach, features representing style are represented by vectors in order to capture relationships such as the distance between words, between sentences or between documents. The vector representation is constructed using various language models, some based on neural networks and others using co-occurrence statistics such as the models in [21–24].

However, given the existence of a database of tagged news articles, classification of FN can be performed by supervised algorithms to differentiate the language style of a fake and a true news article.

Chen and Liu et al. [25] presented a method based on a Convolutional Neural Network (CNN) to classify tweets in to categories such as supports, denials, queries, and comments. This classification is helpful to determine tweets veracity. An evaluation using datasets from [26] achieved an overall classification accuracy of 70%.

Girgis et al. [27] proposed to evaluate several neural network models such as standard recurrent neural networks (RNN), gated recurrent unit (GRU) and Long Short Term Memory (LSTM) architectures to determine whether information is truthful or misleading. These models were evaluated using the LIAR dataset containing 12800 labelled short records [28]. They found GRU to be better than the others.

Also, to find a solution for the detection of fake news, Popat and Mukherjee et al. built a neural network method called DeclarE (Debunking Claims with Interpretable Evidence) [13]. Using this method, the authors evaluated and explained the credibility of observations and news articles. The proposed model was composed of a bidirectional LSTM layer (Bi-LSTM) and two fully connected neural layers. It specifically addresses shortcomings in the literature such as the manual fact checking of sites. The authors evaluated this model by using datasets from different websites. The first dataset contains 4341 records extracted from Snopes (snopes.com) a general fact-checking web site. The second dataset containing 3568 records was published by PolitiFact (politifact.com) a political fact-checking site, containing 3568 records. The third dataset containing 5344 records was published by NewTrust which is a community for assessment of news article credibility [29]. Then, the last one contains 272 records evaluated and uploaded by SemEval-2017 [30]. This method was compared with other methods such as SVM, CNN and LSTM and it gave different results for the different datasets. On Snopes, SVM at high accuracy, on PolitiFact and SemEval, DeclarE obtains significantly better accuracy, and then with NewsTrust, they find the lowest mean square error.

Radhakrishnan and Vadavalli [31] proposed a Convolutional Neural Network model for authenticity search in textual corpora on Quora questions. They started from the premise that pre-existing iterative methods, probabilistic models and optimization models are not good in the case of feature extraction from unstructured data such as information published in social media. In addition, recurrent neural networks (RNN) cannot learn these features efficiently and require more time. Convolutional neural networks are effective for extracting features in a very short time. They observed a classification accuracy that is close to 60%.

Kumar et al. [32] proposed a hybrid neural network model consisting of a CNN and a Bidirectional LSTM (BiLSTM) to solve FNs. They use 1356 news articles collected from Twitter and PolitiFact. This method is then compared with the CNN, LSTM, and set methods. They concluded that the CNN-BiLSTM hybrid provided a better accuracy of 88.78% compared to [33]. In addition, Choudhary et al. [34] proposed a BerConvoNet deep learning model to classify news articles as fake or true with small errors. Their model converts article text into its vector representation using the BERT method [35]. They used a concatenation of several CNN layers with different kernel sizes to extract features from this representation. This model was tested on four datasets and an average classification accuracy of 90% was obtained for these sets.

Nasir et al. focused on the classification of FN [14]. They proposed CNN-RNN a machine learning method combining a convolutional neural network CNN and a recurrent neural network LSTM. Their method was
evaluated using two databases. The first is FA-KES [5], consisting of 804 news articles on the war in Syria. The second is ISOT [36], consisting of 44,898 news articles on facts from Wikipedia and PolitiFact. For the training on ISOT they reach an accuracy of more than 90%, on the other one they reach 60% at most. In their work, they aimed to solve the problem of generalization of machine learning models by training their approach on ISOT and testing on FA-KES. However, they did not consider some parameters that can improve the performance of the model and they achieved an accuracy of 50% and a precision of 48%. These parameters are the regularization of the data and the appropriate choice of an activation function.

Considering the impact of the proliferation of fake news, deep learning models can be of great help in detecting fake news on the web.

Based on this literature, several deep learning models have been experimented and evaluated on the same topic, but suffer from overlearning and a bias related to the dataset specific to their topic of interest, so they perform poorly on other topics. The hybridization deep learning approach appears to be more efficient on the detection of fake news. Some hybridization deep learning models not yet considered could give better results.

So, we propose two hybrid deep learning architectures and evaluate their performance on news articles datasets.

### III. METHODOLOGY

In this section, we explain our approach by showing the techniques used to build our models and the data used to evaluate them.

A. Definition of the Problem and Approach to Solving It

Current research on fake news such as Zhou and Zafarani [4], Allcott and Gentzkow [37], Shu et al. [38], adopt a definition according to which, a fake news is “a news published by a news outlet that is intentionally and verifiably false”. Two aspects that emerge from this definition are the intent and the authenticity. According to the study of Zhou and Zafarani, intent can be malicious or no [4]. Intent is sometimes difficult to detect and is not explicitly available as most current fake news datasets do not make it clear whether annotations in the datasets take into account the intent of the news [4, 38]. In this study we focus on detecting the authenticity of news articles.

However, the content of a news article represented by features such as the source or publisher of the article, the title and the body text which elaborates on the details of the news. These features mentioned can be helpful to detect authenticity of news articles [4].

But, let \( \mathcal{A} \) be a set of news articles consisting of \( N \) articles. Each article \( a_{\text{sisn}} \) is labelled 1 if it is true and labelled 0 if it is fake. Any article \( a_{\text{sisn}} \) is composed of \( M \) words \( w_1, \ldots, w_M \).

However, the problem in this work is to find a function \( S \) that by training on \( \mathcal{A} \) extracts the features and best classifies a news article \( a_{\text{new}} \) into fake or true. That is a binary classification problem. But, in some situation news can take multi class, that are true, false, partially false or partially true. Our study is based on binary classification.

To solve this problem, we proceed by steps described in Fig. 1 and detailed in the sections that follow.

B. Datasets

For the evaluation of our models, we used accessible datasets such as ISOT [36] and FA-KES [5]. These datasets were collected from real-world sources.

FA-KES contains articles about the war in Syria published on the web and verified by experts. In this dataset, an article is labelled as 0 if it is fake and 1 if it is true. This dataset is grouped by article ID, article text, source, publication date, publication location and label. It contains 804 records of which we have 426 true news articles and 378 fake news articles.

The ISOT dataset comprises 44898 news articles and covers different topics. Within this dataset 47.70% of the articles are true and 52.30% are fake. The true articles were collected from Reuter.com a news articles site and the fake articles are collected from unreliable sites reported by PolitiFact (a fact checking organization) and Wikipedia. As with FA-KES, in ISOT each news article is described by the title, text, subject, source and date on which the news document was published.

However, before using this data, we cleaned it up and put it into a usable format. We perform the following steps described in Fig. 1.

![Figure 1. Our approach description.](image)

C. Data Preprocessing

Data preprocessing is a technique that transforms data into a meaningful and understandable format. Datasets from the web are often noisy, incomplete, and inconsistent because of their origin from different people or different sources. Preprocessing consist of two many steps.

The first one is data cleaning and splitting which consist of the following sub steps:

- Put the text of the article in the lower case;
- Replace the line break and tab characters with a single space;
- Remove white space;
- Expand contracted words. Ex: “can’t” become “can not”;
- Delete digits;
Delete all characters that are neither numbers nor letters;
Delete stopwords.
After the cleaning stage, both the ISOT and FA-KES sets are subdivided into two parts, 80% of the set used for training and the remaining 20% for testing. That is, for FA-KES, we use 643 news articles for training and 161 news articles for testing. For ISOT, we use 35918 news articles for training and 8980 news articles for testing.

As result of this step, we obtain representative features of the corpus, and we can now proceed the second step.
The second step is article embedding it consist of text encoding. The encoding text is a necessary task when it has to be processed digitally. In this paper on classification of FN, we represent texts using numerical vectors. To get there we use one of the pre-trained methods of Glove [22].
Glove (Global vectors) is a word vector representation model. It is a logarithmic bilinear regression model that uses statistics of word occurrence in the same context and ratios of co-occurrences of these words, forming matrices of 50 dimensions, 100 dimensions or 300 dimensions. This model is trained on over 42 billion words of web data to provide a vocabulary of 400,000 words that are most commonly used. In our case, we train our numerical vectors using glove 100 dimensions. However, to use Glove, we must define a fixed length of text to be digitized. As the news article contents are of variable length, we will use a fixed length Post Padding of 300, i.e., after the Padding each article will be composed of 300 words.
At the end of this stage, we obtain a representation formed by a matrix of dimension \(300 \times 100\) for each article, which will be used as input for the first convolution layer of our architectures. This matrix is obtained following the execution of Algorithm 1.

Algorithm 1. EmbeddingGlove

Input:
GloveIndex Ý Glove’s dictionary
Maxlen = 300
DataTrain
DataTest
Output:
TrainGlove,
TestGlove
embeddingMatrix
WordIndex
Begin
WordIndex = Tokenization (DataTrain)
\(nb\)Word = len (WordIndex)
TrainGlove = Pad_Sequence (DataTrain, maxlen)
TestGlove = Pad_Sequence (DataTest, maxlen)
embeddingMatrix = numpy.zero (\(nb\)Word +1, 100)
For word, i in WordIndex.item ():
If embeddingVector is not None:
embeddingMatrix [i] = embeddingVector
End If
End For
End

D. Fake News Classification

The models DeepCnnLstm and DeepCnnBilstm that we propose are a concatenation of neural networks consisting of:

- Embedding layer
- Dropout layer
- Convolution layer
- Batch normalization layer
- Dropout layer
- Pooling layer
- Recurrent neural network layer (Bi-LSTM for DeepCnnBilstm or LSTM for DeepCnnLstm)
- Two dense layers.

This architecture is shown in Fig. 2 and Fig. 3.

![Figure 2. DeepCnnLstm summary.](image)

![Figure 3. DeepCnnBilstm summary.](image)

1) Embedding layer
Let \(D = [w_1, ..., w_n] \) be, a news article containing \(n\) words. Let \(W_i \in \mathbb{R}^d\) the numerical representation of word \(w_i\). The output of this layer is a matrix \(M = [W_1, ..., W_n] \in \mathbb{R}^{n \times d}\) which is the concatenation \(W_i\).
2) **Convolution**

Originally the CNN was built for image pattern recognition tasks [39]. Since then, it has also been used for other tasks such as text classification [40], or for automatic language processing. A CNN is a deep multi-layer neural network consisting of several filters defined by the size of the convolution layer windows. These filters have a feature extraction function. Since in the case of text classification by a CNN it is recommended to use a 1-dimensional representation, and then as a convolution layer we use the Conv1D architecture, an activation function, batch normalization and a Dropout operation.

We consider the result of the embedding layer obtained in the encoding layer which is the matrix \( M \in \mathbb{R}^{n \times d} \).

The filter is slide \( f_j \in \mathbb{R}^{i \times d} \) over \( M \) with a window size \( l \), by splitting \( M \) into ngrams of \( l \)-words \( u_i \in \mathbb{R}^{i \times d} \), \( u_i \) is the concatenation of \( l \) words. For the filter \( f_j \) and for an \( l \)-words \( u_i \), we obtain characteristic \( c_i \) by computing as in Eq. (1).

\[
e_i = f(b, f_j > + b) \in \mathbb{R} \tag{1}
\]

The bias \( b \in \mathbb{R} \) and \( f \) is an activation function which can be the sigmoid, the hyperbolic tangent, ReLu or the Elu activation function, etc. For this layer we used the Elu activation function [41]. The Elu activation function speeds up learning in deep neural networks and leads to higher classification accuracies [42]. Elu is an extension of ReLU activation function. ReLu is the identity function for positive arguments and zero otherwise.

Thus, a filter \( f_j \) slipped over all \( l \)-words \( u_i \) produces a feature map \( C_j \) as shown in Eq. (2).

\[
C_j = (c_i) \in \mathbb{R}^{n-l+1}. \tag{2}
\]

For the convolution layer, we use \( m \) filters to extract various features. These \( m \) filters generate \( m \) differents features maps \( C \in \mathbb{R}^{(n-l+1) \times m} \). Let \( C \) be a vector of the \( C_j \), as shown in Eq. (3). The are used by the next layer.

\[
C = [C_0, ..., C_m]. \tag{3}
\]

3) **Batchnormalization**

Batch input normalization is a layer that refocuses and rescales each element in a batch. This method improves the accuracy and speed of training [43].

Formally given an input \( x \) and a batch size \( k \), for batch normalization, we first calculate for each batch \( L \) of \( x \) formed, the mean \( \mu_L \) and the variance \( \sigma_L^2 \) as in Eq. (5) and Eq. (6).

\[
x = (x_i, ..., x_N) \tag{4}
\]

\[
\mu_L = \frac{1}{k} \sum_{i=1}^{k} x_i \tag{5}
\]

\[
\sigma_L^2 = \frac{1}{k} \sum_{i=1}^{k} (x_i - \mu_L)^2 \tag{6}
\]

Then each element \( x_i \) in the batch is transformed into an element \( \hat{x}_i \), where \( \hat{x}_i \) is computed as in Eq. (7).

\[
\hat{x}_i = \gamma \frac{(x_i - \mu_L)}{\sigma_L^2 + \epsilon} + \beta, i \in [1, ..., k] \tag{7}
\]

In Eq. (7), \( \epsilon \) is a small constant added to the denominator for numerical stability. \( \gamma \) and \( \beta \) are respectively the scale factor initialized to 1 and the offset factor initialized to 0. These two factors are updated automatically during training and can be disabled [44].

4) **Dropout**

Dropout is a regularization technique that solves the overfitting problem and facilitates the combination of several neural network architectures. It allows to temporarily disable neurons in the network as well as its incoming and outgoing connections [45]. Each neuron is activated with probability \( p \) of being equal to 1. Dropout can be applied to both the input and output of a layer. Let \( l \) be a layer on which dropout is applied, let \( Y^l \) be its output vector and let \( r \) be a vector of Bernoulli random variables, each of which has probability \( p \). The dropout applied to this layer gives Eq. (8) and the input of the layer that follows is given by Eq. (9).

\[
\widetilde{Y}^l = r \times Y^l \tag{8}
\]

\[
Z^{l+1} = W\widetilde{Y}^l + b^{l+1} \tag{9}
\]

5) **Pooling**

This operation consists in reducing the size of the vector generated by the convolution layer while keeping the most important features. Here we apply the pooling operation MaxPooling1D.

6) **Recurrent neural network**

Unlike other neural networks, RNN are suitable for sequential data such as words in a sentence that depend on each other. They are used for various tasks such as speech recognition [46]. Stock market prediction [47], also used for text classification [48]. However, when dealing with a long sequence, the traditional RNN has the problem of leakage gradient. To overcome this problem, the LSTM neural network has been proposed [49]. The LSTM is able to learn the dependencies in a short or long text by capturing the contextual features in that text.

An LSTM sequence is composed of cells linked by a time steps \( t \). The output of each cell is controlled by a set of activation functions \( \{f(t), i(t), o(t)\} \) also called gates as shown in Fig. 4. They have values in \( \mathbb{R}^T \) where \( T \) is the dimension (the number of cells in the sequence) of the LSTM.

The function \( f(t) \), also called the forget gate, defines the extent to which the information of the previous cell \( c_{t-1} \) is eliminated. The input gate \( i(t) \) defines the proportion of the information that will be stored in cell \( c_t \). The output gate \( o(t) \) controls the proportion of the internal state transmitted to the next cell.
The result of a bidirectional LSTM layer is the concatenation of the outputs in both directions, as shown in Eq. (13).

\[ H = [h_T, h_T'] \]  

IV. EXPERIMENTATION, RESULTS AND DISCUSSION

A. Experimental Framework

1) Models implementation in Keras

For the implementation of our models, we worked on an Hp core i3 computer, with 4 GB of Ram and a 64 bits operating system on which we connected to the Google Colab environment. This environment is adapted to machine learning, data analysis, it is a hosted service of Jupyter notebooks. These notebooks can be shared with other people. In addition, we used the Keras library which is an API written in python for neural network modelling. In addition, we had to access the Numpy and Pandas packages for scientific calculations and array manipulation operations. Furthermore, we used Scikit-learn, Regular Expression, and NLTK (Natural Language Toolkit) for data preprocessing.

However, we built our models, DeepCnnBilstm and DeepCnnLstm, using the Keras sequential API. It consists of:

- An embedding layer. This layer uses the Glove method described previously. It transforms all the 300d vectors obtained after post-padding into a matrix of dimension 300 \times 100,
- A Dropout layer with probability 0.7,
- A Conv1D convolution layer of 128 filters of size 3 and ELu activation function,
- A Conv1D convolution layer of 128 filters of size 5 and ELu activation function,
- A Batchnormalisation layer,
- A Dropout layer with probability 0.5,
- A Bidirectional LSTM layer of 32 cells,
- A Concatenation layer,
- A Dense layer of 32 neurons with a ReLu activation function,
- A dense layer of 1 neuron using the Sigmoid activation function for 1 or 0 classification.

In the case of DeepCnnLstm, we change Bidirectional LSTM layer of 32 cells by LSTM layer of 32 cells.

2) Compared models

We compared the performance of our models with other supervised classification methods tested on the ISOT and FA-KES datasets. These models are listed in Table I.
3) Metrics
To evaluate our models, we used several metrics as performance indicators such as accuracy, precision, recall and F1-score described in Table II.

In our context, a news article is true if it is labelled 1 and labelled 0 if it is Fake News. So we can posit:
- Positive = 1
- Negative = 0
- TP = True Positive
- TN = True Negative
- FP = False Positive
- FN = False Negative

TP is when a true news article is classified as true news article by the method.

TN is when a false news is classified as fake news by the method.

FP is when a fake news is classified as true news article by the method.

FN is when a true news article is classified as fake news by the method.

### TABLE I. COMPARATIVE MODEL

<table>
<thead>
<tr>
<th>Short Name</th>
<th>Classifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN only</td>
<td>Convolutional Neural Network</td>
</tr>
<tr>
<td>LSTM only</td>
<td>Long Short Term Memory</td>
</tr>
<tr>
<td>Bi-LSTM only</td>
<td>Bidirectional Long Short Term Memory</td>
</tr>
<tr>
<td>LR</td>
<td>Logistic Regression</td>
</tr>
<tr>
<td>RF</td>
<td>Random Forest</td>
</tr>
<tr>
<td>MNB</td>
<td>Multinomial Naïve Bayes</td>
</tr>
<tr>
<td>SGD</td>
<td>Stochastic Gradient Decend</td>
</tr>
<tr>
<td>KNN</td>
<td>K Nearest Neighbors</td>
</tr>
<tr>
<td>DT</td>
<td>Decision Tree</td>
</tr>
<tr>
<td>AB</td>
<td>Ada Boost</td>
</tr>
</tbody>
</table>

### TABLE II. MEASURES USED AND THEIR DESCRIPTION IN THE CASE OF FAKE NEWS DETECTION

<table>
<thead>
<tr>
<th>Metric</th>
<th>Formula</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>( \frac{TP + TN}{TP + FP + TN + FN} )</td>
<td>Accuracy is the proportion of news articles well classified that the model produces for classifications of all kinds.</td>
</tr>
<tr>
<td>Precision</td>
<td>( \frac{TP}{TP + FP} )</td>
<td>The percentage of true news articles well classified by the model. The higher it is the more the number of false news classified as true news article is minimised.</td>
</tr>
<tr>
<td>Recall</td>
<td>( \frac{TP}{TP + FN} )</td>
<td>The number of true news articles well classified by the model divided by the number of all related samples. The higher it is the more the number of true news articles classified as fake news is minimised.</td>
</tr>
<tr>
<td>F1-score</td>
<td>( \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} )</td>
<td>the harmonic mean of recall and precision</td>
</tr>
</tbody>
</table>

### B. Results and Discussion

Experiments were conducted on both FA-KES and ISOT datasets. The training performed out during 10 epochs with a batch size of 64. Our results were compared with those in [14].

Table III shows the results of the neural network models applied to FA-KES. In this table, we can see that DeepCnnBilstm has an accuracy of 68%, a precision of 64%, a recall of 88% and an F1-score of 74%. Given this accuracy, this model outperforms the CNN only, LSTM only and Bi-LSTM only models by more than 15% and the CNN-RNN [14] hybrid model by more than 8%. The precision of this model is higher than that of the models used alone by more than 8% and the hybrid models CNN-RNN [14] by more than 2%. The recall of DeepCnnBilstm is higher than that of the models used alone by more than 15% and the hybrid model by more than 15%. The accuracy of DeepCnnLSTM outperforms the CNN only, LSTM only and Bi-LSTM only models by over 10% and the precision by over 6%. Furthermore, the accuracy and precision of DeepCnnBilstm are slightly higher than those of DeepCnnLSTM, respectively by 5% and 2%.

In Table IV, the experiments on ISOT show that the values of accuracy, precision and recall are 99% to 100% except in the case of LSTM only. The latter has an accuracy value of 98%, a precision of 98%, a recall of 98% and an F1-score of 98%.

We also compare the performance of DeepCnnBilstm using the accuracy of the other classical machine learning models listed in Table I. These models have been used extensively in the literature but often induce more false positives (FP) and false negatives (FN). Table V and Table VI, shows the performance of these models respectively on FA-KES and ISOT. On the figures Fig. 6 and Fig. 7, the accuracy of the neural network based models, especially DeepCnnBilstm and DeepCnnLSTM (i.e., our models) higher than the other machine learning methods.

### TABLE III. RESULTS OF NEURAL NETWORK MODELS ON THE FA-KES DATASET

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN only</td>
<td>0.50</td>
<td>0.55</td>
<td>0.50</td>
<td>0.52</td>
</tr>
<tr>
<td>LSTM only</td>
<td>0.50</td>
<td>0.51</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>Bi-LSTM only</td>
<td>0.53</td>
<td>0.56</td>
<td>0.73</td>
<td>0.63</td>
</tr>
<tr>
<td>CNN-RNN [14]</td>
<td>0.60</td>
<td>0.59</td>
<td>0.60</td>
<td>0.59</td>
</tr>
<tr>
<td>CNN_Bi-LSTM</td>
<td>0.59</td>
<td>0.62</td>
<td>0.73</td>
<td>0.67</td>
</tr>
<tr>
<td>DeepCnnLSTM</td>
<td>0.63</td>
<td>0.62</td>
<td>0.98</td>
<td>0.76</td>
</tr>
<tr>
<td>DeepCnnBilstm</td>
<td>0.68</td>
<td>0.64</td>
<td>0.88</td>
<td>0.74</td>
</tr>
</tbody>
</table>

### TABLE IV. RESULTS OF NEURAL NETWORK MODELS ON THE ISOT DATASET

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN only</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>LSTM only</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
</tr>
<tr>
<td>Bi-LSTM only</td>
<td>0.99</td>
<td>0.99</td>
<td>1</td>
<td>0.99</td>
</tr>
<tr>
<td>CNN-RNN [14]</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>DeepCnnLSTM</td>
<td>0.99</td>
<td>1</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>DeepCnnBilstm</td>
<td>0.99</td>
<td>0.99</td>
<td>1</td>
<td>0.99</td>
</tr>
</tbody>
</table>
To know the ability of our models to generalize, we used ISOT for training and FA-KES for testing because generalization is a difficult problem in the field of machine learning. We observe the generalization results in Table VII. In this table, DeepCnnBilstm and DeepCnnLstm have a value of accuracy equal to 54% and 52% respectively. DeepCnnBilstm has a recall and F1-score of 93% and 67% respectively. Table VIII shows examples of predictions taken from the FA-KES dataset. In the Table VIII, the news item in example number one is labelled “1”, the model has classified it in the same label (it is True Positive). The news item in example number two is labelled “0”, the model has classified it in label “1” (it is False Positive). Looking at the confusion matrix in Fig. 8, out of 378 news items labelled “0” the model predicted 38 news items labelled “0” (10% True Negative) and out of 426 news items labelled “1” the model predicted 396 news items labelled “1” (92% True Positive). In the case of generalization, our models have better accuracy, recall and F1-score than [14]. Also as shown in that work, our models DeepCnnLstm and DeepCnnBilstm work well on specific datasets (see Table III, Table IV), but do not generalize well (see Table VII). However, compared to [14] which first suggested generalization, our models generalize better.

TABLE V. RESULTS OF OTHER MACHINE LEARNING ON THE FA-KES DATASET

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>0.49</td>
<td>0.5</td>
<td>0.49</td>
<td>0.49</td>
</tr>
<tr>
<td>RF</td>
<td>0.53</td>
<td>0.56</td>
<td>0.53</td>
<td>0.54</td>
</tr>
<tr>
<td>MNB</td>
<td>0.38</td>
<td>0.39</td>
<td>0.38</td>
<td>0.32</td>
</tr>
<tr>
<td>SGD</td>
<td>0.47</td>
<td>0.49</td>
<td>0.47</td>
<td>0.48</td>
</tr>
<tr>
<td>KNN</td>
<td>0.57</td>
<td>0.58</td>
<td>0.57</td>
<td>0.57</td>
</tr>
<tr>
<td>DT</td>
<td>0.55</td>
<td>0.56</td>
<td>0.55</td>
<td>0.55</td>
</tr>
<tr>
<td>AB</td>
<td>0.47</td>
<td>0.49</td>
<td>0.47</td>
<td>0.47</td>
</tr>
</tbody>
</table>

TABLE VI. RESULTS OF OTHER MACHINE LEARNING ON THE ISOT DATASET

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>0.52</td>
<td>0.5</td>
<td>0.52</td>
<td>0.42</td>
</tr>
<tr>
<td>RF</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
</tr>
<tr>
<td>MNB</td>
<td>0.60</td>
<td>0.60</td>
<td>0.60</td>
<td>0.60</td>
</tr>
<tr>
<td>SGD</td>
<td>0.52</td>
<td>0.52</td>
<td>0.52</td>
<td>0.52</td>
</tr>
<tr>
<td>KNN</td>
<td>0.60</td>
<td>0.67</td>
<td>0.61</td>
<td>0.56</td>
</tr>
<tr>
<td>DT</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
</tr>
<tr>
<td>AB</td>
<td>0.92</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
</tr>
</tbody>
</table>

TABLE VII. RESULTS OF GENERALIZATION

<table>
<thead>
<tr>
<th>Models</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN-RNN [14]</td>
<td>0.50</td>
<td>0.48</td>
<td>0.48</td>
<td>0.46</td>
</tr>
<tr>
<td>DeepCnnLstm</td>
<td>0.52</td>
<td>0.55</td>
<td>0.67</td>
<td>0.60</td>
</tr>
<tr>
<td>DeepCnnBilstm</td>
<td>0.54</td>
<td>0.53</td>
<td>0.93</td>
<td>0.67</td>
</tr>
</tbody>
</table>

In summary with the results obtained in Table III, Table IV and Table V, the performance of the convolutional neural network and recurrent neural network hybridization outperforms the performance of the models without hybridization and the classical machine learning models. In contrast to CNN-RNN [14], we concatenated filters of different sizes and used parameters such as the Elu activation function and the dropout. This parameterization allowed our models to perform better than those of CNN-RNN [14].
V. CONCLUSION AND FUTURES WORKS

The proliferation of fake news on social media is a problem that society is facing. To provide a solution to this we have in this paper, proposed two new hybrid neural network models for fake news detection. Our models are composed of convolutional neural network and recurrent neural network: DeepCnnLstm which ensures the extraction of spatial and contextual features in the forward direction of the text and DeepCnnBilstm which ensures the extraction of spatial and contextual features in the forward and backward direction of text. These models perform better than other existing machine learning models and DeepCnnBilstm performs better than the DeepCnnLstm. The experimental results show that such hybrid models can indeed improve the performance of news classification in social media.

Future works will aim at taking into account the problem of multi-class classification, as a news article can be partially false. In addition, a news article can contain both textual and visual features, so we will consider this aspect as well.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Souleymane Ouamtanga, Vincent Monsan, Beman Hamidja Kamagaté conducted the research by defining research frameworks, designing research methodology. Fatoumata Wongbé Rosalie Tokpa analyzed, modeled, and prepared the data for modeling and writing the paper. In addition, Beman Hamidja Kamagaté conducted the research by verifying and preparing the paper, reviewing models, and discussing the findings. All authors had approved the final version.

REFERENCES


Keras. [August 2022]. [Online]. Available: https://keras.io/api/layers/activation_layers/elu


Copyright © 2023 by the authors. This is an open access article distributed under the Creative Commons Attribution License (CC BY-NC-ND 4.0), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.