Applied Soft Computing 110 (2021) 107600

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Attention-based C-BiLSTM for fake news detection

Tina Esther Trueman [a,*], Ashok Kumar J. [a], Narayanasamy P. [b], Vidya J. [a]

[a] *Department of Information Science and Technology, Anna University, Chennai 600025, India*
[b] *Department of Electrical and Electronics Engineering, PSG College of Technology, Coimbatore, India*

ARTICLE INFO

Article history:
Received 7 October 2020
Received in revised form 31 May 2021
Accepted 4 June 2021
Available online 15 June 2021

Keywords:
Fake news detection
Convolutional neural networks
Recurrent neural networks
Bi-directional long–short term memory
Attention mechanism

ABSTRACT

Social media platforms have radically transformed the creation and dissemination of news. Users can easily access this news in a fast and efficient manner. However, some users might post negative and fraudulent content in the form of comments or posts. Such content can constitute a threat to an individual or an organization. Therefore, the identification of fake news has become a major research field in natural language processing (NLP). The main challenge is to determine whether the news is real or fake. In this paper, we propose an attention-based convolutional bidirectional long short-term memory (AC-BiLSTM) approach for detecting fake news and classifying them into six categories. The evaluation of our proposed approach on a benchmarked dataset shows a significant improvement in accuracy rate in comparison with other existing classification models. In particular, this work contributes to the progress in the field of detecting fake news and confirms the feasibility of our proposed approach in classifying fake news on social media.

1. Introduction

Recent technological developments and the proliferation of the internet have had a tremendous impact on social interactions. Social media has become a popular way for people to interact and obtain information. People share their personal activities, interests, and opinions across various social media platforms. Social media offers an easy access to information with lower cost and faster information than the mainstream sources such as broadcast and print. Because of these features, many people search for news from social media rather than regular sources of news such as television or newspaper. As a result, social media news is rapidly replacing classical news sources. Although social media is very useful, the news content cannot be viewed always as trustworthy information. The source of the news is not authenticated as it would be if it was disseminated from a regular news source. This gives an opportunity to spread fake news. False information is misrepresented as truth. Disseminators of fake news, usually have an ulterior motive such as damaging the reputation of a person or making money through false claims [1–3]. Fake news is dangerous as it can cause damage in various ways. For instance, it can malign a person, organization, or even cause unrest and protests among the people. It can impact entire sections of society. Hence there is an increasing necessity to classify the news as truth or falsehood.

There are quite a few websites that are dedicated to verify the truthfulness of claims made by people. PolitiFact.com is one of them that grades claims as true, half true, barely true, mostly true, false, or pants on fire (outright fiction) as opposed to a binary classification of truth or falsehood. This is because certain statements are subtly changed which gives a false impression. Therefore, a multi-class classification problem has its own sets of challenges. In the field of machine learning, the discovery of fake news is one of the emerging issues that have attracted the attention of researchers around the world. Machine learning algorithms need feature selection to classify data accurately. However, the traditional machine learning algorithms fail to consider the semantic context representation. Also, the manual identification of feature selection is a very daunting task. Neural networks have achieved reasonable results compared to other machine learning methodologies [4,5]. Recently, the fake news has been detected using various deep learning techniques such as convolutional neural networks (CNN) and recurrent neural networks (RNN). The RNN includes LSTM (Long Short-term Memory Networks), BiLSTM (Bidirectional Long Short-term Memory Networks), GRU (Gated Recurrent Units (GRU), and BiGRU (Bidirectional Gated Recurrent Units) [6,7].

Moreover, deep learning techniques have the advantage of identifying the features on their own. These techniques identify the meaning of a word while considering its context. In particular, attention mechanisms have become one of the powerful techniques in natural language processing [8]. They are mainly used along with recurrent neural networks to predict the most relevant information in the input sequence. Although the research community has given considerable attention, a deficiency

* Corresponding author.
E-mail addresses: tina_trueman@auist.net (T.E. Trueman), jashokkumar83@auist.net (Ashok Kumar J.), dean.nc@psgtech.ac.in (Narayanasamy P.), vidyajeba@gmail.com (Vidya J.).

https://doi.org/10.1016/j.asoc.2021.107600
1568-4946/

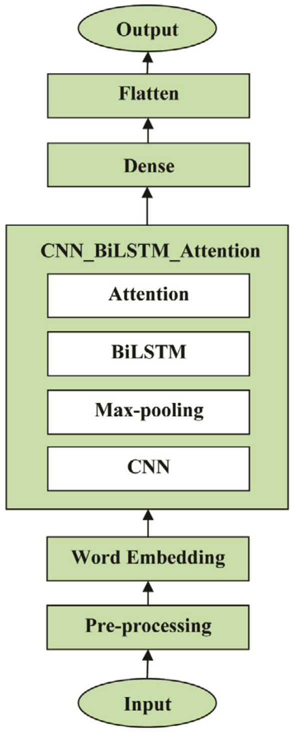

Fig. 1. AC-BiLSTM model for Fake News Detection.

in multi-class and context-based classification warrants further research. To address these issues, we propose an attention-based convolutional bidirectional long short-term memory model to detect fake news. In particular, this work has been motivated by the belief that consideration of the context significantly improves the accuracy of fake news identification. Therefore, this paper contributes to the following.

- Tackles fake news detection problem in a multi-class environment.
- Proposes attention mechanisms in conjunction with convolutional bidirectional recurrent neural networks.
- Improves the accuracy of fake news detection.

The rest of this paper is organized as follows. Section 2 discusses the related works in fake news detection. The design methodology of the proposed model is presented in Section 3. In Section 4, we present the experimental results and discussions and their significant comparison with the state-of-the-art (SOTA) results. Finally, we conclude this work with possible future enhancements in Section 5.

2. Related works

In this section, we discuss the deep learning methods that have been used in the existing works for fake news detection. In particular, the CNN and LSTM methods have already been successfully used in fake new tasks as the following works illustrate. Wang et al. [6] have put forward a fake news detection method using a hybrid CNN on a benchmarked dataset. Specifically, the authors combined the metadata with text. The hybrid CNN produced more accuracy than other existing models such as SVM (Support Vector Machine), Logistic Regression, and BiLSTM. However, the BiLSTM did not perform well due to the overfitting. Girgis et al. [7] presented a work to identify fake news using deep learning models. They built a hybrid classification model using GRU and CNN. The authors used the LIAR dataset to show the efficiency of their model. They have obtained the accuracy of 21.7%, 21.66%, and 21.5% with the GRU, LSTM, and vanilla models, respectively. The authors sought to increase the accuracy by proposing a hybrid model between the GRU and CNN methods on the same data set.

Olivieri et al. [9] proposed a methodology that employs task-generic features, paired with textual features to detect fake news and categorize them into six classes. Specifically, the task-generic features were created on the metadata attached to answers from Google's search engine and crowdsourcing. Rashkin et al. [10] suggested an LSTM based method to detect fake news in the context of political fact-checking. To compare the linguistic features of an unreliable text, they compared the original news language with sarcasm, fraud, and propaganda. Overall performance analysis of different approaches in three different datasets is presented by Khan et al. [11]. They showed that neural network-based models trained by a dataset with fewer than 100 news articles can achieve the same results as Naïve Bayes with N-grams. Finally, they conducted a topic-based analysis, which exposes the difficulty of finding out the fraudulent news in politics, health, and research.

Roy et al. [12] developed CNN and BiLSTM models with an accuracy of 42.89% and 42.65%, respectively. The authors assigned the obtained representation of both CNN and BiLSTM to an MLP (multi-layer perceptron) model to yield the final classification accuracy (44.87%). Long et al. [13] proposed an attention-based LSTM model to integrate speaker profiles for detecting fake news. Specifically, the authors included speaker profiles such as credit history, speaker title, party affiliation, and location in the attention model and the input representation. Their study indicated that the proposed model improves accuracy by 14.5% for the benchmarked fake news dataset. From the above literature, it is obvious that deep learning methods work comparatively better than classical machine learning algorithms. Moreover, the deep learning methods identify the best feature set automatically for a specific problem. However, the attention mechanism concept plays an important role to detect the most relevant information in the sentence. Therefore, we propose a combined CNN and BiLSTM models with an attention mechanism to improve the accuracy of fake news detection.

3. Attention-based C-BiLSTM for fake news detection

In this section, we present an attention-based Convolutional Bidirectional Long Short-Term Memory (AC-BiLSTM) model to detect fake news in a multiclass environment. The architecture of this model is illustrated in Fig. 1. It depicts the main components of our model such as input data, pre-processing, word embeddings, convolutional neural networks (CNN), bidirectional long short-term memory networks (BiLSTM), and attention mechanism. We discuss each of these components as follows.

3.1. Input data

We demonstrate the proposed model using the LIAR dataset which is publicly available for detecting fake news [6,14]. This dataset contains 12.8 K statements that are gathered from PolitiFact.com. Currently, it is the largest dataset available for fake news detection. Table 1 shows the structure of the dataset. It contains three separate groups of data for training, validation, and testing. Each of these groups includes 10,269, 1284, and 1283 statements, respectively. They consists of six labeled categories, namely, true, half-true, mostly true, barely true, false, and pants on fire.

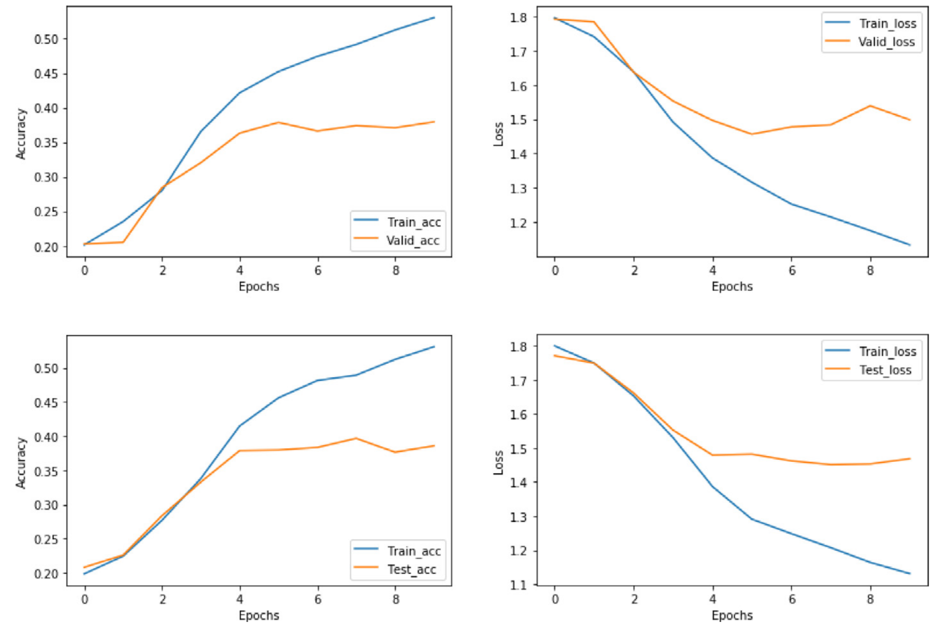

Fig. 2. The learning curve for AC-LSTM (F).

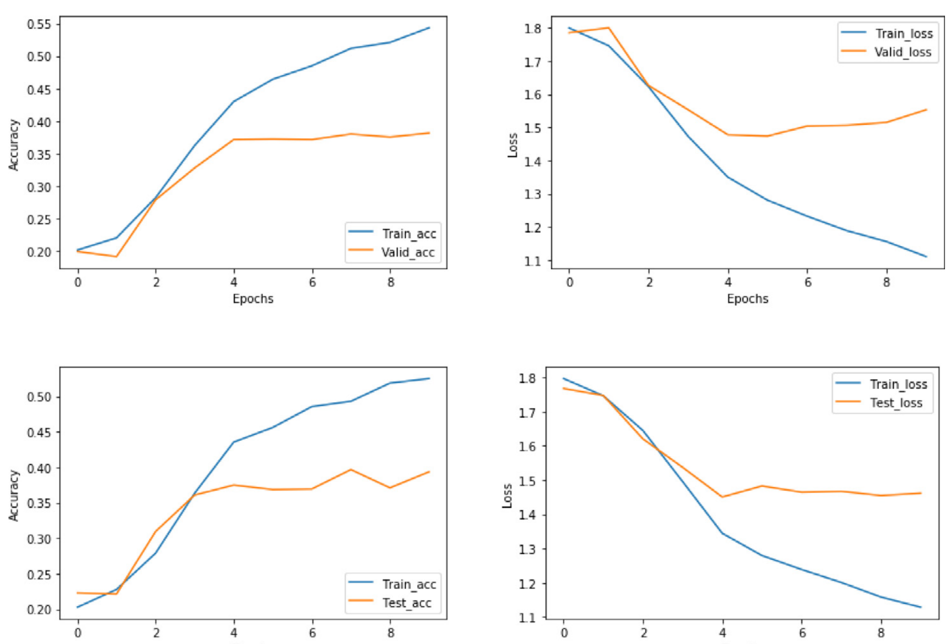

Fig. 3. The learning curve for AC-LSTM (B).

Table 1
Fake news dataset structure.

Categories	Training	Validation	Testing
True	1683	169	211
Half true	2123	248	267
Mostly true	1966	251	249
Barely true	1657	237	214
False	1998	263	250
Pants-fire	842	116	92
Total	10,269	1284	1283

3.2. Pre-processing

The input data is pre-processed using various steps such as a case converter, tokenization, and removing symbols [7]. First, the case converter is used to convert text data into lowercase letters. Second, all symbols are removed from the text data. Finally, the tokenization process is applied to divide the given input text into smaller sections called tokens. These steps help to transform the given input data into integer sequences.

3.3. Word-embeddings

The obtained or preprocessed data is passed to the word embedding layer. The primary goal of this layer is to generate

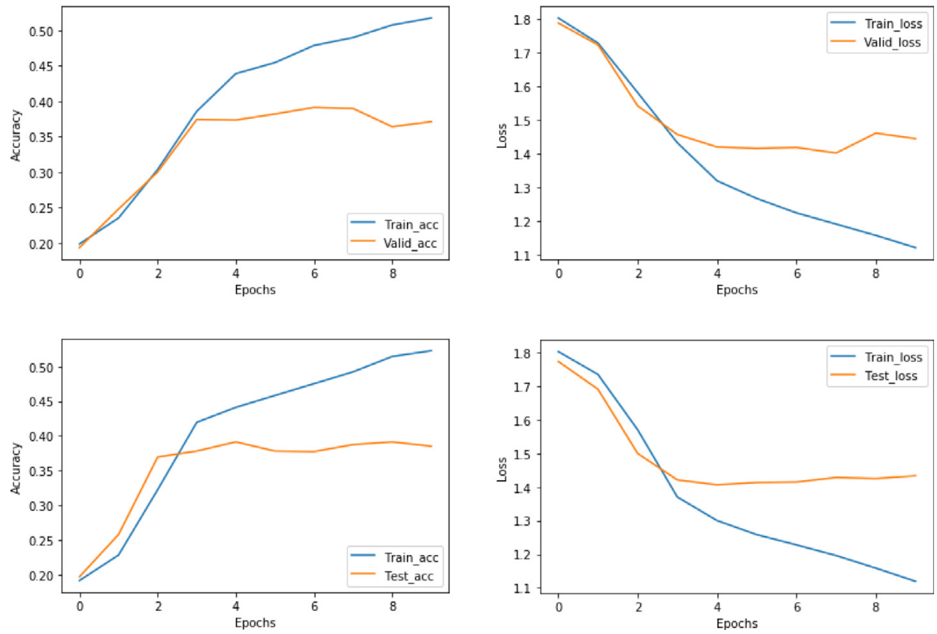

Fig. 4. The learning curve for AC-BiLSTM.

word vectors for the downstream tasks. However, it is a challenging task to construct word vectors from scratch with the billions of words using a large dataset. Therefore, we use a pre-trained model that reduces the amount of time required to clean, train, and process the model. The predefined models (Word2Vec or GloVe) create a unique word embedding representation of each word in the vocabulary. In particular, we use GloVe word vectors with a fixed dimension. It is trained as word-to-word co-occurrence statistics from a corpus of one billion words with a vocabulary of 400,000 words [7,15]. Specifically, we update the learned weights in our model.

3.4. Convolutional neural networks (CNN)

The CNN is a deep learning model that processes information to perform image classification, automatic natural language processing, question answering, feedback analysis, and text summarization tasks with remarkable results [16–18]. Specifically, CNN has a unique structure that facilitates detecting higher-level features. The convolutional layer is the main working block of CNN and it performs feature detection based on matrix coefficients [19]. In particular, this layer contains a set of kernels or filters. These filters help to process a limited part of the input sequence over all the input data through an activation function called ReLU (Rectified Linear Unit). The ReLU is the ellipse of the corrected linear unit [20] mainly set to zero on the network for deleting negative values from the activation map. Therefore, this activation function is computationally more efficient for solving the invisible gradient problem than the sigmoid and tanh functions.

3.5. Max-pooling layer

The features detected through the convolutional layer are passed to a pooling layer. The pooling layer summarizes the features detected in the input and produces a new feature map. It is also called a nonlinear down-sampling method [21,22]. This method progressively reduces the feature dimension to reduce the number of parameters in the computational network. In this paper, we use a maximum pooling layer to select the maximum value in a feature map with the help of filter size.

3.6. Bidirectional long short-term memory networks (BiLSTM)

A recurrent neural network (RNN) is mainly used to process sequential data. It stores the relevant parts of the input data and uses this information to predict the future output. The RNN contains memory cells, which are used to store the most relevant information from the past inputs. Also, it fails to handle long-term dependencies. Therefore, a special type of RNN is introduced to solve the long-term dependencies called a long short-term memory network (LSTM) [23,24]. This is implemented using three gating concepts [16,25,26], namely, input (*IG*), output (*OG*), and forget gates (*FG*). These gates control the information flow (read, write, and reset) over the gradient along with the candidate hidden state (*CHS*), current state (*CS*), and hidden sequence (*HS*) as in Eqs. (1)–(6). Specifically, the LSTM network processes information unidirectionally either from left to right or right to left. It leads to the problem of obtaining future information. In this case, a bidirectional LSTM is introduced to learn the original input sequence from the beginning to the end and from the end to the beginning as in Eq. (7).

$$\mathrm{IG_t} = \sigma(W_{IG}x_t + R_{IG}h_{t-1} + b_{IG}) \tag{1}$$

$$\mathrm{OG_t} = \sigma(W_{OG}x_t + R_{OG}h_{t-1} + b_{OG}) \tag{2}$$

$$\mathrm{FG_t} = \sigma(W_{FG}x_t + R_{FG}h_{t-1} + b_{FG}) \tag{3}$$

$$\mathrm{CHS_t} = \tanh(W_{CHS}x_t + R_{CHS}h_{t-1} + b_{CHS}) \tag{4}$$

$$\mathrm{CS_t} = FG_t \otimes CS_{t-1} + IG_t \otimes HS_t \tag{5}$$

$$\mathrm{HS_t} = OG_t \otimes \tanh(CS_t) \tag{6}$$

$$\mathrm{y_t} = V(\overrightarrow{\mathrm{H}}\,\mathrm{S_t}\text{:}\ \overleftarrow{H}\,S_t) \tag{7}$$

Where W_{IG}, W_{OG}, $W_{FG,}$ and W_{CHS} are referring to the weight matrices of the current input x_t. R_{IG}, R_{OG}, R_{FG}, and R_{CHS} are referred to as the weight matrices of previous state h_{t-1}. b_{IG}, b_{OG}, b_{FG}, and b_{CHS} are denoted as the bias value. y_t represents the output of the forward LSTM and backward LSTM units.

Table 2
Model hyperparameters.

Hyper parameters	Size	Hyper parameters	Size
GloVe dimension	100	BiLSTM Units	150
Input sequence length	150	Attention activation	Sigmoid
Vocabulary size	10,000	Dense layer	32
Dropout rate	0.5	Batch size	64
Filters	300	Output activation	Softmax
Window size	5	Optimizer	Adam (0.001)
Pooling size	4	Loss function	Categorical cross-entropy

Table 3
Training and validation confusion matrix for AC-LSTM (F).

Class	Training						Validation					
	0	1	2	3	4	5	0	1	2	3	4	5
0	1428	47	240	47	96	140	126	24	35	12	56	10
1	146	530	149	85	746	27	41	47	21	13	43	4
2	328	69	1051	73	86	50	54	24	93	20	40	6
3	243	127	250	771	720	12	30	34	26	73	73	4
4	65	137	88	63	1608	5	50	25	29	21	122	4
5	163	2	55	2	2	618	41	11	11	3	24	26

Table 4
Training and testing confusion matrix for AC-LSTM (F).

Class	Training						Testing					
	0	1	2	3	4	5	0	1	2	3	4	5
0	1514	44	144	121	74	101	133	10	21	51	31	4
1	167	587	100	322	490	17	58	43	26	38	46	0
2	419	52	861	236	60	29	45	8	61	48	46	6
3	265	90	116	1285	357	10	58	10	18	117	60	4
4	107	123	47	298	1388	3	50	22	20	47	110	0
5	167	2	45	7	2	619	22	6	8	19	6	31

Table 5
Training and validation confusion matrix for AC-LSTM (B).

Class	Training						Validation					
	0	1	2	3	4	5	0	1	2	3	4	5
0	1384	104	176	106	163	65	123	36	7	43	53	1
1	95	719	66	121	669	13	34	46	19	29	40	1
2	324	98	873	177	165	20	45	35	74	48	34	1
3	170	205	96	991	656	5	40	43	9	110	46	0
4	46	215	20	77	1605	3	44	28	15	40	123	1
5	193	4	57	11	4	573	40	20	3	18	21	14

Table 6
Training and testing confusion matrix for AC-LSTM (B).

Class	Training						Testing					
	0	1	2	3	4	5	0	1	2	3	4	5
0	1690	58	39	116	32	63	132	31	10	52	23	2
1	218	662	56	286	442	19	54	55	18	43	40	1
2	611	79	655	236	44	32	44	24	50	57	31	8
3	346	155	44	1271	294	13	64	28	9	129	35	2
4	119	199	20	360	1265	3	49	32	9	53	106	0
5	226	7	25	12	1	571	20	6	1	26	6	33

Table 7
Training and validation confusion matrix for AC-BiLSTM.

Class	Training						Validation					
	0	1	2	3	4	5	0	1	2	3	4	5
0	1439	109	113	75	89	173	144	64	6	12	30	7
1	112	1079	55	123	290	24	40	69	15	15	24	6
2	490	162	741	137	51	76	58	66	65	20	18	10
3	320	524	98	881	278	22	56	79	8	71	27	7
4	101	799	26	111	925	4	57	67	6	14	102	5
5	178	5	29	11	0	619	45	33	1	5	7	25

Table 8
Training and testing confusion matrix for AC-BiLSTM.

Class	Training						Testing					
	0	1	2	3	4	5	0	1	2	3	4	5
0	1533	32	219	114	39	61	177	2	12	37	20	2
1	192	443	102	434	498	14	69	38	20	46	38	0
2	452	25	978	146	31	25	69	9	74	39	19	4
3	269	34	265	1359	187	9	97	14	27	106	23	0
4	160	55	120	508	1119	4	87	12	23	56	71	0
5	242	3	62	13	0	522	31	4	12	14	3	28

Table 9
The performance of the AC-LSTM (F).

Class	Training			Validation			Training			Testing		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
0	0.60	0.71	0.65	0.36	0.48	0.41	0.57	0.76	0.65	0.36	0.53	0.43
1	0.58	0.31	0.41	0.28	0.28	0.28	0.65	0.35	0.45	0.43	0.20	0.28
2	0.57	0.63	0.60	0.43	0.39	0.41	0.66	0.52	0.58	0.40	0.29	0.33
3	0.74	0.36	0.49	0.51	0.29	0.37	0.57	0.61	0.59	0.37	0.44	0.40
4	0.49	0.82	0.62	0.34	0.49	0.40	0.59	0.71	0.64	0.37	0.44	0.40
5	0.73	0.73	0.73	0.48	0.22	0.31	0.79	0.74	0.76	0.69	0.34	0.45
MAC	0.62	0.60	0.60	0.40	0.36	0.36	0.64	0.61	0.61	0.44	0.37	0.38
MIC	0.58	0.58	0.58	0.38	0.38	0.38	0.61	0.61	0.61	0.39	0.39	0.39
WA	0.61	0.58	0.58	0.40	0.38	0.38	0.62	0.61	0.60	0.41	0.39	0.38

Table 10
The performance of the AC-LSTM (B).

Class	Training			Validation			Training			Testing		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
0	0.63	0.69	0.66	0.38	0.47	0.42	0.53	0.85	0.65	0.36	0.53	0.43
1	0.53	0.43	0.47	0.22	0.27	0.24	0.57	0.39	0.47	0.31	0.26	0.28
2	0.68	0.53	0.59	0.58	0.31	0.41	0.78	0.40	0.52	0.52	0.23	0.32
3	0.67	0.47	0.55	0.38	0.44	0.41	0.56	0.60	0.58	0.36	0.48	0.41
4	0.49	0.82	0.61	0.39	0.49	0.43	0.61	0.64	0.63	0.44	0.43	0.43
5	0.84	0.68	0.75	0.78	0.12	0.21	0.81	0.68	0.74	0.72	0.36	0.48
MAC	0.64	0.60	0.61	0.45	0.35	0.35	0.64	0.59	0.60	0.45	0.38	0.39
MIC	0.60	0.60	0.60	0.38	0.38	0.38	0.60	0.60	0.60	0.39	0.39	0.39
WA	0.62	0.60	0.59	0.43	0.38	0.38	0.62	0.60	0.59	0.42	0.39	0.39

Table 11
The performance of the AC-BiLSTM.

Class	Training			Validation			Training			Testing		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
0	0.55	0.72	0.62	0.36	0.55	0.43	0.54	0.77	0.63	0.33	0.71	0.45
1	0.40	0.64	0.49	0.18	0.41	0.25	0.75	0.26	0.39	0.48	0.18	0.26
2	0.70	0.45	0.55	0.64	0.27	0.38	0.56	0.59	0.57	0.44	0.35	0.38
3	0.66	0.41	0.51	0.52	0.29	0.37	0.53	0.64	0.58	0.36	0.40	0.39
4	0.57	0.47	0.51	0.49	0.41	0.44	0.60	0.57	0.58	0.41	0.29	0.34
5	0.67	0.74	0.70	0.42	0.22	0.28	0.82	0.62	0.71	0.82	0.30	0.44
MAC	0.59	0.57	0.56	0.44	0.36	0.36	0.63	0.57	0.58	0.47	0.37	0.38
MIC	0.55	0.55	0.55	0.37	0.37	0.37	0.58	0.58	0.58	0.39	0.39	0.39
WA	0.58	0.55	0.55	0.45	0.37	0.38	0.61	0.58	0.57	0.43	0.39	0.44

3.7. Attention mechanism

The attention layer creates a context vector to the learned input vectors. It plays an influential role in machine translation, image recognition, text summarization, text classification, and question answering systems. In this paper, the attention layer is built on the top of the BiLSTM network for updating weights. Specifically, the attention layer selectively assigns higher weights to the most relevant and important words from the input sequence. The advantage of this mechanism is to preserve longer input sequences [8,27].

3.8. Dense layer

The dense or fully-connected layer performs a linear operation that connects all input values to every neuron in the next layer by weight. The role of this layer is to provide all combinations of features from the previous layer to the next layer [16,22,28].

Table 12
The mean accuracy of the proposed model.

Models	Training	Validation	Training	Testing
AC-LSTM (F)	0.3963	0.3243	0.3923	0.3351
AC-LSTM (B)	0.4022	0.3250	0.3987	0.3389
AC-BiLSTM	0.4007	0.3384	0.4065	0.3513

3.9. Flatten layer

The flattening layer transforms the dimension of the dense layer into a single dimension without affecting the batch size. For instance, the 2D dimensional features are transformed into

Table 13
Result comparison of the proposed model.

Authors	Models	Accuracy		F1-Score		Micro F1		Macro F1		Weighted F1	
		Valid	Test	Valid	Test	Valid	Test	Valid	Test	Valid	Test
Wang et al. [6]	Majority	0.204	0.208	–	–	–	–	–	–	–	–
	SVM	0.258	0.255	–	–	–	–	–	–	–	–
	Log_Reg	0.257	0.247	–	–	–	–	–	–	–	–
	BiLSTM	0.223	0.233	–	–	–	–	–	–	–	–
	CNN	0.260	0.270	–	–	–	–	–	–	–	–
	Hybrid CNN	0.247	0.274	–	–	–	–	–	–	–	–
Olivieri et al. [9]	NN	–	–	0.273	–	–	–	–	–	–	–
	SVM	–	–	0.308	–	–	–	–	–	–	–
Rashkin et al. [10]	Majority BL	–	–	–	–	–	–	–	0.06	–	–
	NB (text+LIWC)	–	–	–	–	–	–	0.21	0.17	–	–
	MaxEnt (text+LIWC)	–	–	–	–	–	–	0.21	0.22	–	–
	LSTM (text+LIWC)	–	–	–	–	–	–	0.22	0.19	–	–
	LSTM (text)	–	–	–	–	–	–	0.21	0.20	–	–
Girgis et al. [7]	SVM	–	0.255	–	–	–	–	–	–	–	–
	Log_Reg	–	0.247	–	–	–	–	–	–	–	–
	BiLSTM	–	0.233	–	–	–	–	–	–	–	–
	CNN	–	0.270	–	–	–	–	–	–	–	–
	Vanila	–	0.215	–	–	–	–	–	–	–	–
	GRU	–	0.217	–	–	–	–	–	–	–	–
	LSTM	–	0.217	–	–	–	–	–	–	–	–
Roy et al. [12]	BiLSTM	–	0.427	–	0.41	–	–	–	–	–	–
	CNN	–	0.429	–	0.42	–	–	–	–	–	–
	RNN_CNN	–	0.449	–	0.43	–	–	–	–	–	–
Yunfei Long [13]	Base LSTM	0.250	0.255	–	–	–	–	–	–	–	–
	Hybrid LSTM	0.407	0.415	–	–	–	–	–	–	–	–
Alhindi et al. [29]	Log_Reg	–	–	0.38	0.37	–	–	–	–	–	–
	SVM	–	–	0.35	0.35	–	–	–	–	–	–
	BiLSTM	–	–	0.34	0.32	–	–	–	–	–	–
	P_Bi_LSTM	–	–	0.37	0.36	–	–	–	–	–	–
Proposed	AC-LSTM (F)	0.324	0.335	0.36	0.38	0.38	0.39	0.36	0.38	0.38	0.38
	AC-LSTM (B)	0.325	0.339	0.35	0.39	0.38	0.39	0.35	0.39	0.38	0.39
	AC-BiLSTM	0.338	0.351	0.36	0.38	0.37	0.39	0.36	0.38	0.38	0.37

a 1D dimensional feature. The output of this layer is applied to the softmax output layer [16,28].

3.10. Softmax layer

The softmax layer measures probability for all possible classes in a multi-class problem. It transforms the learned input values between zero and one. This probability assignment helps the model to converge quickly [16,22,28]. The softmax function is defined as follows in Eq. (8).

$$\mathrm{softmax(z)_i} = \frac{e^{z_i}}{\sum_{i=1}^{k} e^{z_i}} \tag{8}$$

Where z_i refers to the elements of the input vector that takes any real value and k refers to the number of classes. The denominator of the function normalizes all output values in the range 0 to 1. The highest probability value is the output of the class.

4. Results and discussion

In this section, we present the performance of the proposed AC-BiLSTM model using the standard evaluation metrics such as confusion matrix, accuracy, precision, recall, F1-score, macro average, micro average, and weighted average [30]. A confusion matrix or an error matrix is a table that represents the performance of a classifier based on true positives, false positives, false negatives, and true negatives. Accuracy measures the total correctly identified documents out of all documents. Precision calculates the ability of a classifier to identify all relevant documents. Recall measures the ability of a classifier to return only relevant documents. F1-score calculates the harmonic mean of precision and recall. Macro average score calculates metric for each class with equal weights. It does not consider the problem of class imbalance. Micro average score measures metric for each instance with equal weights based on the total true positives, false positives, and false negatives. Weighted average score calculates metrics for each class based on the number of true instances. It is the same as the macro average score, but it considers the problem of class imbalance. In this paper, we used a LIAR benchmark dataset for fake news detection. The LIAR dataset contains 10,269 instances for training, 1284 instances for validation, and 1283 instances for testing. We then employed pre-processing techniques such as characters and symbols removal, character conversion, and tokenization for obtaining the quality of input documents. These documents are transformed into a sequence of integers. Later, the GloVe pre-trained word embedding is used to generate word vectors for each word in the training document. These word vectors are supplied as input sequences to the proposed AC-BiLSTM model. Table 2 shows the model hyperparameters. The model is implemented in a CPU environment with a windows system using Anaconda software with various machine learning libraries such as Keras and scikit. The performance of the proposed model is evaluated with the above-stated metrics in the forward pass, backward pass, and both forward and backward pass. In particular, the proposed AC-BiLSTM model is implemented with 10 epochs. Tables 3 to 8 show the confusion matrix of the proposed model for both training and validation and training and testing. The performance of the proposed model is shown in Tables 9 to 11 for the forward pass, backward pass, and both forward and backward passes. These tables indicate the precision (P), recall (R), F1-score (F1), micro average (MIC), macro average (MAC), and weighted average (WA) scores. Table 12 shows the mean training, validation, and testing accuracy. The learning curve of the proposed model is shown in Fig. 2, Fig. 3, and Fig. 4. We have compared our proposed AC-BiLSTM model with various existing models that have used the same dataset and the same label categories as shown in Table 13. In this table, Rashkin et al. [10] used labeled instances from PolitiFact and PunditFact sites. Roy et al. [12] have shown higher results. They have used the LIAR benchmark dataset with reduced number of instances. Furthermore, Yunfei Long [13] experimented with a limited number of parameters of the LIAR dataset. Therefore, we estimate that the proposed model comparatively performs better than the state-of-art results.

5. Conclusion

In this paper, we proposed an attention-based convolutional long short-term memory network for automatically detecting fake news using the LIAR dataset. This dataset provided an authentic and original brief statement from different speakers with different contexts. Therefore, it influenced the development of a fake news detection method in a broad coverage. First, we preprocessed the LIAR dataset using the case conversion, removing symbols and punctuation, and tokenization techniques. Second, we generated word vectors for the given input data using GloVe pre-trained word embedding. Finally, the proposed AC-BiLSTM model is employed on these word vectors to predict fake news in the multi-class environment. Specifically, the proposed model captures local, global, and temporal meaning of the sentence using C-BiLSTM and then the attention mechanism helps to memorize long input sequence. Therefore, our proposed hybrid model achieves greater accuracy (35.1%) and micro F1-score (39%) than other existing models. However, the fake news detection method still remains as an open challenge problem in terms of data, model, feature, and application. In future works, we intend to predict fake news based on gender and age group using audio, video, image, and text data. Moreover, the transformer-based models can be used to predict fake news on this information fusion. Also, the identification of key elements in the spread of fake news using graph neural networks can be another possible direction for future research.

CRediT authorship contribution statement

Tina Esther Trueman: Conceptualization, Methodology, Writing - original draft, Writing - review & editing, Supervision. **Ashok Kumar J.:** Methodology, Software, Validation, Visualization. **Narayanasamy P.:** Supervision. **Vidya J.:** Software, Validation, Investigation, Visualization.

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] Zeynep Tufekci, It's the (democracy-poisoning) golden age of free speech. Wired, https://www.wired.com/story/free-speech-issue-tech-turmoil-new-censorship/. Published: 16.01.2018, (Accessed 02 March 2020).

[2] David Leonhardt, Stuart A. Thompson, Trump's lies, 14, The New York Times, 2017.

[3] Nolan Higdon, The Anatomy of Fake News: A Critical News Literacy Education, University of California Press, 2020.

[4] Yann LeCun, Yoshua Bengio, Geoffrey Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[5] Tomas Mikolov, Ilya Sutskever, Kai. Chen, Greg S. Corrado, Jeff Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.

[6] William Yang Wang, Liar liar pants on fire: A new benchmark dataset for fake news detection, 2017, arXiv preprint arXiv:1705.00648.

[7] Sherry Girgis, Eslam Amer, Mahmoud Gadallah, Deep learning algorithms for detecting fake news in online text, in: 2018 13th International Conference on Computer Engineering and Systems (ICCES), IEEE, 2018, pp. 93–97.

[8] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, Neural machine translation by jointly learning to align and translate, 2014, arXiv preprint arXiv:1409.0473.

[9] Alex Olivieri, Shaban Shabani, Maria Sokhn, Philippe Cudré-Mauroux, Creating task-generic features for fake news detection, in: Proceedings of the 52nd Hawaii International Conference on System Sciences, 2019.

[10] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, Yejin Choi, Truth of varying shades: Analyzing language in fake news and political fact-checking, in: Proceedings of the 2017 conference on empirical methods in natural language processing, 2017, pp. 2931-2937.

[11] Junaed Younus Khan, Md Khondaker, Tawkat Islam, Anindya Iqbal, Sadia Afroz, A benchmark study on machine learning methods for fake news detection, 2019, arXiv preprint arXiv:1905.04749.

[12] Arjun Roy, Kingshuk Basak, Asif Ekbal, Pushpak Bhattacharyya, A deep ensemble framework for fake news detection and classification, 2018, arXiv preprint arXiv:1811.04670.

[13] Yunfei Long, Fake news detection through multi-perspective speaker profiles, Association for Computational Linguistics, 2017.

[14] https://www.politifact.com/.

[15] Jeffrey Pennington, Richard Socher, Christopher D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532-1543.

[16] Yoon Kim, Convolutional neural networks for sentence classification, 2014, arXiv preprint arXiv:1408.5882.

[17] Pierre Sermanet, Soumith Chintala, Yann LeCun, Convolutional neural networks applied to house numbers digit classification, in: Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), IEEE, 2012, pp. 3288–3291.

[18] Hussein Mouzannar, Yara Rizk, Mariette Awad, Damage Identification in Social Media Posts using Multimodal Deep Learning. in: ISCRAM, 2018.

[19] Botao Zhong, Xuejiao Xing, Peter Love, Xu Wang, Hanbin Luo, Convolutional neural network: Deep learning-based classification of building quality problems, Adv. Eng. Inform. 40 (2019) 46–57.

[20] Yuanzhi Li, Yang Yuan, Convergence analysis of two-layer neural networks with relu activation, in: Advances in Neural Information Processing Systems, 2017, pp. 597–607.

[21] Yanming Huang, Y. Jiang, Touhidul Hasan, Q. Jiang, Chao Li, A topic BiLSTM model for sentiment classification, in: Proceedings of the 2nd International Conference on Innovation in Artificial Intelligence, 2018, pp. 143-147.

[22] Vijay Vasudevan, Barret Zoph, Jonathon Shlens, Quoc V. Le, Neural architecture search for convolutional neural networks. U.S. patent no. 10,521,729, 2019, 31 Dec..

[23] Sepp Hochreiter, Jürgen Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[24] Zhiyong Cui, Ruimin Ke, Ziyuan Pu, Yinhai Wang, Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction, 2018, arXiv preprint arXiv:1801.02143.

[25] Shlok Gilda, Notice of violation of IEEE publication principles: Evaluating machine learning algorithms for fake news detection, in: 2017 IEEE 15th Student Conference on Research and Development (SCOReD), IEEE, 2017, pp. 110–115.

[26] Jinlong Sun, Wenjuan Shi, Zhutian Yang, Jie Yang, Guan Gui, Behavioral modeling and linearization of wideband RF power amplifiers using BiLSTM networks for 5G wireless systems, IEEE Trans. Veh. Technol. 68 (11) (2019) 10348–10356.

[27] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, Eduard Hovy, Hierarchical attention networks for document classification. in: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, 2016, pp. 1480-1489.

[28] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand. Thirion, Olivier Grisel, Mathieu. Blondel, et al., Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[29] Tariq Alhindi, Savvas Petridis, Smaranda Muresan, Where is your Evidence: Improving Fact-checking by Justification Modeling. in: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), 2018, pp. 85-90.

[30] Margherita Grandini, Enrico Bagli, Giorgio Visani, Metrics for Multi-Class Classification: An Overview, 2020, arXiv preprint arXiv:2008.05756.