Applied Soft Computing 110 (2021) 107614

Contents lists available at ScienceDirect

Applied Soft Computing

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

BerConvoNet: A deep learning framework for fake news classification

Monika Choudhary, Satyendra Singh Chouhan [*], Emmanuel S. Pilli, Santosh Kumar Vipparthi

Department of CSE, MNIT Jaipur, 302017, India

ARTICLE INFO

Article history:
Received 8 September 2020
Received in revised form 23 April 2021
Accepted 7 June 2021
Available online 24 June 2021

Keywords:
Deep learning
Fake news
BERT
CNN
Ablation study

ABSTRACT

Fake news has become a major concern over the Internet. It influences people directly and should be identified. In the recent years, various Machine Learning (ML) and Deep Learning (DL) based data-driven approaches have been suggested for fake news classification. Most of the ML based approaches use hand-crafted features extracted from input textual content. Moreover, in DL based approaches, an efficient word embedding representation of input data is also a major concern. This paper presents a deep learning framework, *BerConvoNet*, to classify the given news text into fake or real with minimal error. The presented framework has two main building blocks: a news embedding block (NEB) and a multi-scale feature block (MSFB). NEB uses Bidirectional Encoder Representations from Transformers (BERT) for extracting word embeddings from a news article. Next, these embeddings are fed as an input to MSFB. The MSFB consists of multiple kernels (filters) of varying sizes. It extracts various features from news word embedding. The output of MSFB is fed as an input to a fully connected layer for classification. To validate the performance of *BerConvoNet*, several experiments have been performed on four benchmark datasets and various performance measures are used to evaluate the results. Furthermore, the ablative experiments with respect to news article embedding, kernel size, and batch size have been carried out to ensure the quality of prediction. Comparative analysis of the presented model is done with other state of the art models. It shows that *BerConvoNet* outplays other models on various performance metrics.

1. Introduction

With the increase in accessibility and diversity of online interaction and information sharing platforms, the content shared online in the form of text, images, and multimedia, reaches out to the masses swiftly. According to a survey by American Press Institute, 59% of people between the age group 18–34 years and 56% of people between the age group 35–49 years use digital medium as a source of news and information [1]. The content shared for online readers in Tweets, Facebook, and Reddit posts and information published on websites can be a source for the spread of misinformation and falsified data. This can affect and influence readers negatively. For example, The Daily Mail, which is read by over a million readers on a regular basis, had one of its headlines in January 2020 as "Finland to introduce a four-day working week and SIX-HOUR days under plans drawn up by 34 year-old prime minister Sanna Marin" [2]. However, the story has since been denounced as fake news, and the Finland Government has tweeted over its official Twitter handle to clarify it.

Sharing misleading and fake information can greatly scrape public trust and confidence in online information. As the users are not always able to judge the correctness of information shared online, there should be mechanisms to know whether they are coming across true or fake information. Manual assessment of interminable online data is nearly impossible. Due to this reason, there have been various attempts to detect fake news by analyzing the content of news articles with intelligent systems. Recently, machine learning approaches have been used for this task. In machine learning, researchers often detect fake news by relying on either latent (via neural networks) [3,4] or non-latent (usually hand-crafted) features [5–8] of the content.

The proposed work emphasizes on deep learning-based approach for fake news identification. Work done so far in this direction has been explored in depth. In [9] authors used information based on interconnection between news articles, creators of news articles, and the various news subjects. They proposed a deep network that assimilated the information related to network structure and incorporated this for model learning. In [10], Term Frequency–Inverse Document Frequency (Tf–IDF) and dense neural network are used to anticipate the stance between headline and article body pair. Though their model is effective, the general-

[*] Corresponding author.
E-mail addresses: 2019rcp9186@mnit.ac.in (M. Choudhary), sschouhan.cse@mnit.ac.in (S.S. Chouhan), espilli.cse@mnit.ac.in (E.S. Pilli), skvipparthi@mnit.ac.in (S.K. Vipparthi).

https://doi.org/10.1016/j.asoc.2021.107614
1568-4946/© 2021 Elsevier B.V. All rights reserved.

ized model created for fake news identification may be unable to recognize the context of ironic or satirical news articles. In [11], the authors proposed an ensemble machine learning algorithm with a deep neural network model for fake news classification. A larger part of existing methods uses a single prototype embedding model. These models cannot represent polysemous words. Polysemous words have more than one interpretation or meaning. Therefore, using a single prototype embedding for each word can be troublesome for intrinsically polysemous words.

This paper presents a hybrid framework, *BerConvoNet*, based on the concatenation of Bidirectional Encoder Representations from Transformers (BERT) embedding with convolutional neural network [12,13]. To the best of our knowledge, we are the first to experiment with a combination of BERT embedding with CNN for credibility assessment of online news in textual format. The presented framework has two main building blocks: a news embedding block (NEB) and a multi-hale feature block (MSFB). NEB takes a news article as input and uses BERT to extract word embeddings from it. The embeddings generated are fed as an input to MSFB. The MSFB consists of multiple kernels, with each having a different size. It extracts various features from news word embeddings. The multiscale filters in BerConvoNet contain parallel convolutions, which make it possible to extract multiscale features. Finally, these feature vectors are forwarded to a fully connected layer, which classifies the news into real or fake.

We investigate and evaluate *BerConvoNet* framework by performing an experimental study on four different fake news dataset [14–16]. We perform a holistic evaluation of model performance using seven different performance evaluation measures like Accuracy, Precision, Recall, F1-score, Mathew's Correlation Coefficient (MCC), Specificity, and G-mean. Moreover, the ablation study with different embedding approaches shows that BERT embedding with CNN outperforms other state-of-the-art techniques.

The main contributions of this paper are as follows.

- It presents a Deep learning framework, *BerConvoNet*, based on the combination of BERT embedding and CNN that achieves significant performance improvement in fake news identification.
- In addition, multiscale feature block (MSFB) is use to extract the various features of a news article. Here, four types of filter size are used for fake news identification.
- Ablation study of *BerConvoNet* is performed which is focused on optimizing the hyper-parameters involved at different layers in the presented model.
- Performance comparison of *BerConvoNet* with other state-of-the-art models. This analysis helps in assessing the viability of the presented framework.

The rest of the paper is organized as follows. Section 2 briefs the previous work done for fake news detection. The proposed methodology, including the BerConvoNet framework is presented in Section 3. Section 4 presents the experiments and results. It first describes the experimental setup, the dataset characteristics, performance evaluation metrics and then summarize our experimental outcomes. Lastly, we perform a result analysis and conclude the proposed work in Section 5.

2. Related work

The task of fake news detection has been explored by several researchers using disparate technical approaches. All the contributions can be broadly categorized into three groups: Linguistic based, Network based (Network theory based) and Machine learning based Techniques.

2.1. Linguistic based techniques

In this approach, the extracted news content is analyzed to associate language patterns and identify fake News. In [17] authors highlighted the usage of 'unusual' language in fake stories and were able to draw insightful indications from them. In [18], authors used natural language processing techniques to solve the fake news detection problem. Linguistic based methods often use word-based analysis by using word features such as N-grams and Part-of-Speech (POS) to identify the fake news [6,19]. Some researchers use syntax's structure of the document to identify fake news. These approaches considered textual documents and were able to determine hierarchical structure in them for fake news detection [20].

Recently, in [21], the authors propose a methodology that builds a Discourse-level structure to identify fake news. Additionally, their work recognizes insightful structure-related properties that clarify their discovered structure and enhance the understanding of fake news.

However, using only linguistic features to identify fake news may not be sufficient and often these techniques are merged with machine learning techniques to identify fake news. Some other notable works in fake news detection using linguistic base approaches are given in Ref. [22–25]. Since the proposed work is not relevant to linguistic-based techniques, they are discussed in brief.

2.2. Network based techniques

Analyzing network structure and behavior is another approach to identify deception. Checking facts based on the relationship among entities by developing a knowledge graph can be useful. In [5], the researchers have proposed the concept of 'network effect' variables. The authors claim 61% to 95% accuracy of methods based on knowledge graph analysis.

In [26], the Authors proposed fake news detection approaches based on incomplete and imprecise knowledge graphs. In [27], authors used graph-based approximations for studying the relations between users who share news and the path which the shared content follows in order to mitigate its potential deception effects. Recently, in [28] a graph-based approach for fake news detection is proposed. This approach is based upon graph-based concepts such as biclique identification, graph-based feature vector learning and label spreading.

Some other notable works in network-based approaches are given in Ref. [29–33]. Another promising research direction is exploiting the social network behavior to identify the fake news.

2.3. Machine learning based techniques

Here, some of the significant work done using Machine Learning and Deep Learning techniques are discussed.

In [34], the authors proposed a system that performed credibility assessment of user input by considering language features. They also focused on the stance of the text and the trustworthiness of the source. Aspects of retrieved articles that contributed most in determining credibility were indicated in their work. Similar to this work, other researchers have considered related articles from the internet [35]. This work focused on capturing the similarity of input news articles with other relevant articles by calculating numerical features. To determine the article similarity, the authors proposed a hybrid of N-gram, TFIDF, and Cosine Similarity measures. Authors in [36] use different machine learning algorithms for supervised model building for fake news classification. Hybrid classification models have also been proposed for fake news detection [37]. The authors use Random Forest and K-Nearest Neighbor (KNN) algorithms. They claimed eight percent accuracy improvement over Support Vector Machine (SVM).

Deep Learning-based Techniques Due to the increase in size and complexity of the datasets, deep learning techniques have been used lately for text classification. Notable work was presented by Kim [38] using CNN for sentence classification. In the proposed model, the author uses an embedding layer to convert words into feature vectors and applies single-layer convolution. Multiple kernels with size 3, 4, 5 are used in this work to obtain multiple features. Max pooling is applied on the feature maps generated in the previous step to register the most important features to which a fully connected layer with dropout is applied along with softmax activation for obtaining final classification. Several authors have extended this work for developing text classification and fake news classification. models [39,40].

In [41], the authors proposed an embedding method that captures both social proximity and community structures. Their classification method utilized the underlying network of Long Short Term Memory (LSTM) and Recurrent Neural Networks (RNN). In [42] the proposed model analyzes the relationship between the news headline and its textual body. Additional news data is gathered to pre-train the model. This model used weighted cross-entropy for the classification of input data. Authors in [43] present a capture, score, and integrate model where they focus upon the source news article. Their approach also includes the news article's response by using RNN for analyzing the temporal pattern of user activity on the news article under consideration.

In the light of the above work, we present a hybrid deep learning-based framework, *BerConvoNet*, for fake news classification.

3. Methodology

This section discusses the formal problem definition and formulation. It also presents the *BerConvoNet* framework in details.

3.1. Problem definition and formulation

A news article can be termed as a fake news if it deliberately conveys verifiable incorrect information. Fake news can be define as follows [44].

Definition 1. Given a dataset of news articles $\mathcal{N}$ consisting of labeled news articles. They have labels L as either Real (1) or Fake (0). A news article $n \in \mathcal{N}$ consists of sentences and words. Let us say, news article n_i contain m sentences $s_1, \ldots, s_m$ and sentence $s_j (0 \leqslant j \leqslant m)$ contain X words. Words in a sentence s_j can be represented as $W_j = w_1, \ldots, w_{X_j}$. Here X_j denotes number of words in sentence s_j.

The definition not only includes intentionally forged and falsified news articles but also considers those articles which appear on satirical online portals. Such articles could be misunderstood as fact based, particularly when viewed in isolation like on social networking websites like Twitter and Facebook. The objective of this research is to design a model M that can correctly classify new news article n_{new} into label $L \in \{Real(1), Fake(0)\}$.

3.2. BerConvoNet framework

To perform text processing and analysis, input text words have to be translated into numerical representations. Word embedding is one of the most widely used methods to convert the word into its vector representation. In our experiment, we have chosen BERT (Bidirectional Encoder Representation from Transformers) to generate word embeddings. We have used BERT to imbibe the capability of identifying homographs (words with the same spelling but different meaning) into our proposed model. BERT is a pretrained, transformer-based unsupervised language model introduced by Google. It is trained on a huge book corpus and Wikipedia. As Bert is deeply bidirectional and achieves a deeper sense of context, we presume that it can give good results for determining sentence credibility because credibility is not word-specific; instead, it is a sentence-specific assessment. The word embeddings are generated in the NEB block in our model. Next, we slide the word embeddings over convolution to extract features by applying kernel or filter. The feature maps obtained from different kernel sizes are passed through the activation function to generate a non-linear relationship for output. This is followed by Pooling to reduce the dimensional complexity. The MSFB block is used to perform the convolution and pooling operations. Lastly, the fully connected layer and sigmoid activation function are used to classify an input article as fake or real.

Hence, the proposed *BerConvoNet* works in three stages. In the first stage, the NEB block is used for generating word embeddings for news text. Next, the MSFB block generates various feature maps from the news word embeddings. Various feature maps are concatenated and fed into a fully connected layer. The overall framework is shown in Fig. 1.

3.2.1. News embedding block (NEB)

In NEB, pretrained BERT model is adopted for generating word embeddings. Given the input news article sentence as token sequence $W_j = \{w_1, \ldots, w_{X_j}\}$ of length X, BERT transformer is employed with T number of Layers. Token level representations are calculated by BERT embedding layer using information from the entire sentence. The input features are packed as $R^0 = \{c_1, \ldots, c_X\}$, where $c_x (x \in [1, X])$ is the amalgamation of the token, position and segment embedding corresponding to the input token w_x. In order to refine the token-level features, T transformer layers are introduced. Specifically, the representations, $R^t = \{r_1^t, \ldots, r_X^t\}$ at the tth transformer layer $((0 \leq t \leq T))$ can be shown in accordance to the following equation

$$R^t = Transformer_t(R^{t-1}) \qquad (1)$$

So, R^t is the contextualized representations of the input tokens. The output from BERT in the form of contextualized representations R^T are fed as an input to MSFB block for feature extraction where R^T is

$$R^T = \{r_1^T, \ldots, r_X^T\} \in \mathbb{R}^{X \times dim_r} \qquad (2)$$

3.2.2. Multiscale feature block (MSFB)

In multiscale feature block, different size of kernel such as 2, 3, 4 and 5 are used. Corresponding to each kernel, 50 filters are used. After getting feature maps from all the convolution (and Pooling) operations, the resultant vectors are concatenated and forwarded to fully connected layer as shown in Fig. 1. The detailed explanation of MSFB block is given as follow.

3.2.3. Convolution layer

The convolution layer is an essential part of convolution neural network architecture. It is used for feature extraction. This layer consists of linear operations like convolution and nonlinear operations like using the activation function. Multiple kernels are applied by repeating this procedure to form a random number of feature maps. These feature maps provide an insight into the internal representations and reflect disparate characteristics of the specific sentence. Different kernels can, thus, be considered as different feature extractors. The size and number of kernels are the two main tuning parameters of the convolution operation.

For an X-words input news article $w_1, \ldots, w_X$ BERT embed each symbol as dim_r dimensional vector, resulting in $R^T =$

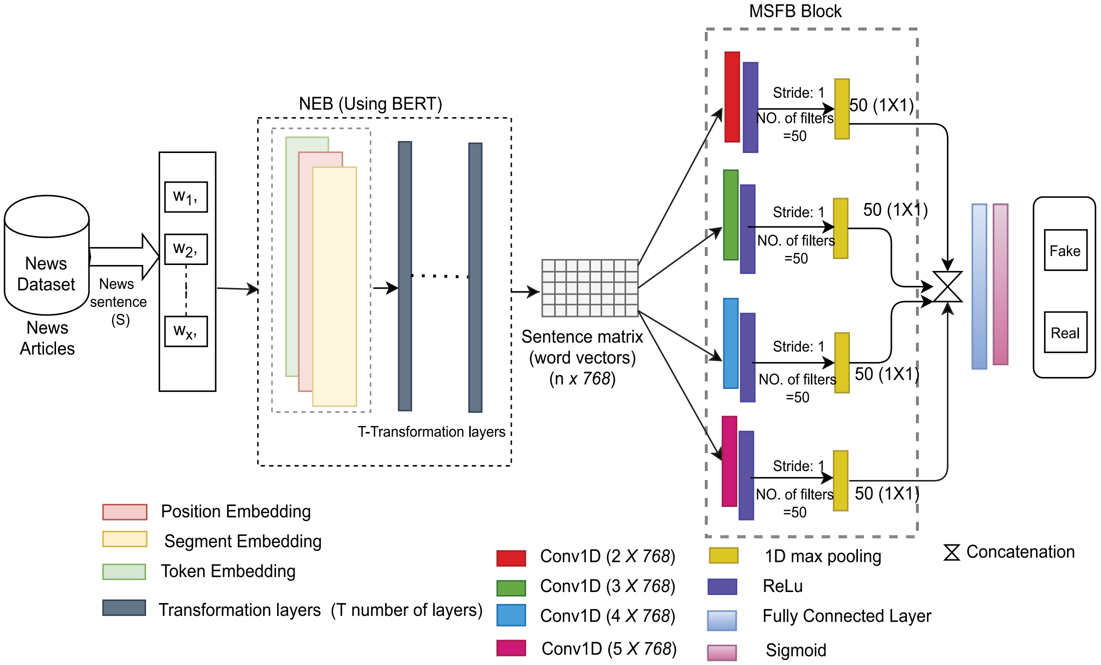


Fig. 1. *BerConvoNet* framework.

$\{r_1^T, \dots, r_X^T\} \in \mathbb{R}^{X\times dim_r}$ (Definition 1). where T is the number of transformer layers used in BERT. For simplicity, $R = \{r_1, \dots, r_X\}$ is used as the output embedding of BERT. The resulting $X \times dim_r$ matrix, R is then fed into a convolutional layer where a sliding window is passed over the text. For each l-words embedding:

$$u_i = [r_i, \dots, r_{i+l-1}] \in \mathbb{R}^{l\times dim_r}; 0 \leqslant i \leqslant X - l \tag{3}$$

And for each filter $f_j \in \mathbb{R}^{l\times dim_r}$ $\langle u_i, f_j\rangle$ is calculated; the convolution results in matrix $F \in \mathbb{R}^{X\times m}$ and the resultant is fed to RELU activation layer.

$$F_{ij} = \langle u_i, f_j\rangle \tag{4}$$

$$c_j = ReLU(F_{ij}) \tag{5}$$

3.2.4. Max pooling and concatenation

Applying max-pooling across the embedding dimension results in $p \in \mathbb{R}^m$.

$$p_j = \max(c_j) \tag{6}$$

In *BerConvoNet*, multiple window (kernel) sizes are used $\ell \in L, L \subseteq \mathbb{N}$ by using multiple convolution filters in parallel and concatenating the resulting p^ℓ vectors.

$$P = p_j^2 \oplus p_j^3 \oplus p_j^4 \oplus p_j^5 \tag{7}$$

In Eq. (7), the superscript 2, 3, 4 and 5 are kernel size and j varies from 1 to the total number of filters corresponding to a particular kernel size.

3.3. Fully connected layer

Like multi-perceptron neural networks, nodes are fully connected to all activations from the previous layer in a fully connected layer. In *BerConvoNet*, a linear fully connected layer $W \in R^{c\times m}$ produces the distribution over classification classes (fake or real) from which the strongest class is presented as final output. In this layer used *sigmoid* activation function is used for classification. Formally,

$$Y_{real/fake} = sigmoid(W_P) \tag{8}$$

The flow diagram of BerConvoNet which tries to explain the process flow in a phased manner is shown in Fig. 2. It depicts the flow in the form of Pre-processing, Word embedding, Feature Extraction and Classification phases for better understanding.

3.4. Analysis of BerConvoNet framework

In *BerConvoNet*, BERT is used as an encoder and convolution neural network as a classifier. The BERT model's two common variants are BERT Base with 12 layers (transformer blocks), 12 attention heads, 110 million parameters, and BERT Large with 24 layers, 16 attention heads, and 340 million parameters [9]. BERT base model is used in this experiment. BERT is used over other pretrained embeddings as it is bidirectional and combines mask language model with next sentence prediction to understand the context of the text. To begin with, each word in an input post is transformed into a vector of 1 × 768 dimension as the length of a BERT embedding is 768. This operation is repeated for all n words in a post, and as a result, a matrix of $n \times 768$ is obtained. Here n represents the number of words corresponding to an input post. However, there are less than n words in one input post in actuality as BERT inserts specific tokens to represent the beginning of each sentence's first sentence and end. A news article's length is also limited to L words, where L is an arbitrary number. The news articles whose length is less than L words are padded by adding 0 vectors at the end. The input is processed and tokenized, padded, and converted into PyTorch Tensors. Then BERT embedding is used to transform the text to embeddings.

Next, word embeddings generated corresponding to a sentence in the previous step are considered input with the $n \times m$ dimension. Here n denotes the maximum number of words in a sentence, and m represents embedding length. Four kernels of sizes 2, 3, 4, and 5 are used in order to capture the different characteristics of the input text. Unlike CNN for images, which use square convolution $[n \times n]$ on embeddings, here convolution

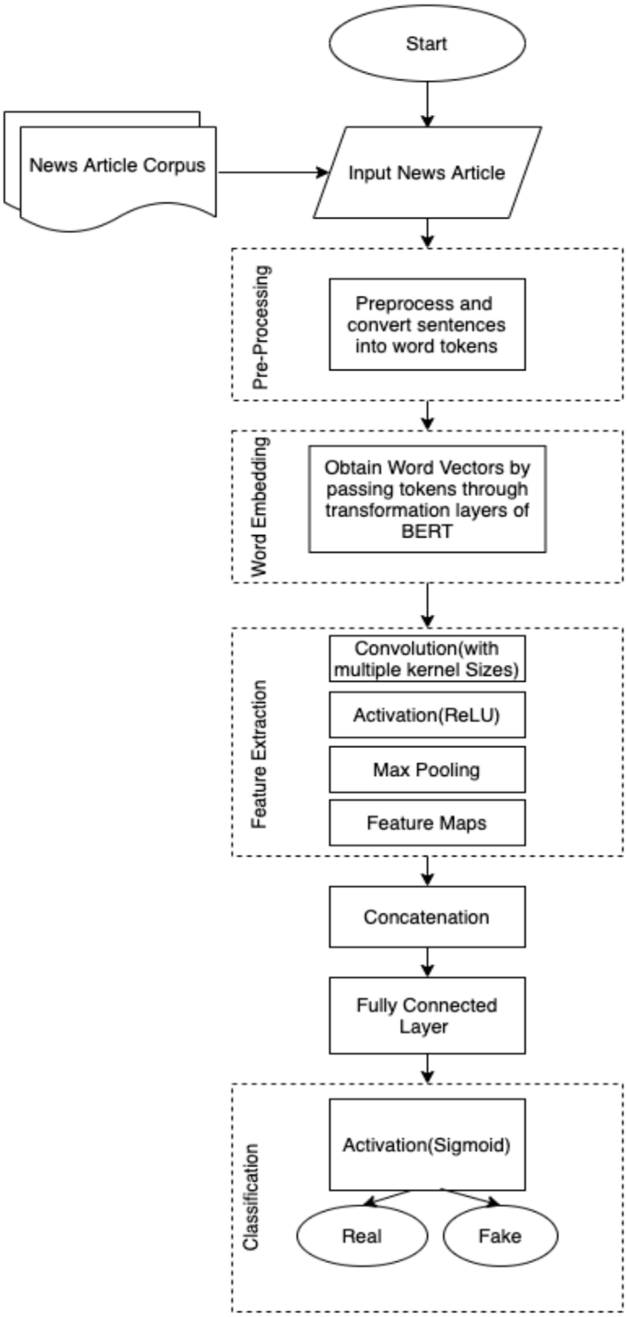


Fig. 2. Flow diagram of *BerConvoNet*.

of different sizes [2 × m], [3 × m], [4 × m], and [5 × m] is applied on embeddings to model combinations of 2 words, 3 words and so on. To train the neural networks, Rectified Linear Unit (ReLU) is applied in order to use stochastic gradient descent with backpropagation of errors. In text classification, CNN based models are shallow in nature. Similarly, in BerConvoNet, the network is not very deep; the effect of vanishing gradients is not significant in our case. Thus, we preferred using ReLU activation over other activation functions like Sigmoid or tanh. Also, it is not computationally intensive. This is followed by 1-max pooling to down-sample the input representation and prevent overfitting. Downsampling also helps in reducing the computational cost. Vectors from previous operations are concatenated into a single vector, and a dropout layer is appended to deal with overfitting. Lastly, the sigmoid activation function is used to scale unnormalized projections between 0 and 1 for each class.

4. Experiments and results

Several experiments are conducted to evaluate the performance of the presented approach for credibility analysis on four benchmark datasets. In this section, first, the details of datasets used are discussed (Section 4.1). Next, the baseline methods and performance metrics are discussed (Sections 4.2 and 4.3). Finally, the performance evaluation and comparison results are presented (Section 4.5).

The implementation has been done using the python language and various Deep learning libraries, including Keras, Scikit learn, matplotlib and pandas. Model is trained on a batch size of 100 and learning rate of 0.001. Adam optimizer is used with the Binary Cross-Entropy loss function (BCE). BCE loss allows this model to assign independent probabilities to the labels. The list specifying the different hyperparameters and their values is shown in Table 1. Several experiments were carried out to decide the optimum hyperparameter values like the number of Kernel, Kernel Size, and batch size. (Section 4.4).

4.1. Dataset description

To validate the performance of the presented model, existing publicly available datasets have been used. The first fake news dataset used is George McIntire Dataset. It is referred as Dataset-1 in this paper. This dataset contains 3164 fake news and 3171 real articles. The second dataset with news articles is sourced from Kaggle. It has 20,800 news articles in the training set and 5200 articles in the testing set. It is denoted as Dataset-2 in this paper throughout. The third and fourth fake news datasets used are from FakeNewsNet Repository. In this repository, fact-checking websites like Gossipcop and Politifact are used to obtain fake and true news. The dataset from Gossipcop is referred as Dataset-3. It contains 4947 fake and 16694 real news. It is balanced to remove bias by considering equal instances of fake and real news from this dataset. The dataset from Politifact is referred as Dataset-4. It contains 420 fake and 528 real news. The data used for training and testing is preprocessed. We have used a 10-fold cross-validation scheme to split the dataset into the training and testing subsets and evaluate model prediction performance. (See Table 2.)

4.2. Experimental procedure

In this section, we describe the overall process followed to build the proposed BerConvoNet model. In our experimental procedure, we have considered four datasets. We have applied the BerConvoNet model to four labeled datasets and evaluated the model performance. We have also used different combinations of word embedding and deep learning classification techniques as baseline models. These baseline models are evaluated, and their performance is compared with BerConvoNet on the four datasets considered.

4.3. Comparison models

We compared *BerConvoNet* with some baseline models. A brief description of these models is given below.

- *Random Embedding with CNN [45,46]*: Keras has a word Embedding called Random Embedding or Embedding Layer. This embedding learns along with a neural network model on a particular natural language processing assignment. Vectors are initialized with small random numbers, and vector

Table 1
List of hyper-parameters in a convolutional neural network (CNN).

Hyperparameters	Environment settings	Remarks
Number of kernels	4	Filter used to extract features
Kernel size	2, 3, 4, 5	Models different word combinations (eg. 2 words, 3 words...)
Activation function	ReLU (Rectified Linear Unit)	Act as piece-wise linear function to train neural networks
Pooling method	1-Max- pooling	Down samples input and prevents overfitting
Optimizer	Adam Optimizer	Used to update network weights iteratively
Learning rate	.001	Step size
Loss function	Binary cross entropy	Assigns independent loss probabilities to the labels
Batch size	100	Number of training samples used to work through before updating the internal model parameters

Table 2
Dataset description.

Attribute	Dataset-1	Dataset-2	Dataset-3	Dataset-4
Total number of posts	6335	20761	21641	948
Number of fake posts	3171	10374	4947	420
Number of real posts	3164	10387	16694	528
Mean length of sentence	767.055	771	11.171	9.685
Max length of sentence	20897	15198	31	53
Standard deviation length of sentence	873.89	844.82	3.88	5.20

space sizes like 50, 100, or any such value are defined as part of the model. The embedding layer is present at the front side of the neural network, and the Backpropagation algorithm is used to fit it in a supervised manner.

- *Static & Dynamic GloVe with CNN [47,48]*: The Global Vectors for Word Representation (GloVe) algorithm is an addition to the word2vec embedding model. It is used for efficiently and systematically learning word vectors. GloVe benefits from local context-based learning like word2vec and takes advantage of matrix factorization techniques' global statistics. Unlike word2vec, GloVe considers word context to construct a word co-occurrence matrix. The learning model built upon this results in better word embeddings. In Dynamic Glove, the embeddings are trainable. This has the added benefits of updating the words which were randomly assigned a vector.
- *Random Embedding with LSTM [45,49]*: Long Short-Term Memory networks (LSTM) belong to a particular type of Recurrent Neural Network (RNN). LSTMs are capable of learning the relationships between elements in an input sequence. The Keras Embedding Layer is used along with LSTM for classification.
- *Elmo with Neural Networks [50,51]*: ELMo models the complex characteristics of word usage, its semantics, along with how the nuances vary across different contexts. These word vectors are developed using bidirectional language models obtained from pre-trained models. To provide an analogy, bidirectional models read the sentence in both directions contrary to unidirectional processing. This embedding layer is added to a neural network for text classification.

4.4. Performance metrics

The confusion matrix is used to characterize the output of a classification model over a collection of test data for which the actual values are known to us. The confusion matrix imparts the following information:
True positive (TP) — Positive cases that are predicted as Positive.
False-positive (FP) — Negative cases that are predicted as Positive
False Negative: (FN) — Positive cases that are predicted as Negative.
True Negative (TN) — Negative cases that are predicted as Negative.

The other performance metrics based on the confusion matrix are shown in Table 3. These performance metrics are used to compare the presented approach with other models. Accuracy is a good performance metric when the dataset is well balanced in terms of labels or categories. As the number of fake and real posts in our datasets are similar, hence it is an important metric in our consideration. We have calculated Precision and Recall to minimize false negatives and false positives one by one and analyze our results. Specificity is essentially our Sensitivity to detecting Negative class examples, which is an important aspect of classification analysis. F1 score is a key metric to determine how precise and robust our classifier is by considering both Precision and Recall together. Mathews Correlation Coefficient (MCC) is more informative over accuracy and F1 score as it considers all four confusion matrix categories together.

4.5. Hyper-parameter selection

We have performed tuning of different hyperparameters using the random search method. Initially, we generate training and test datasets for each of the sections of the k-fold cross-validation. This is followed by the random selection of the hyperparameters to be tested. We train the BerConvoNet model for each hyperparameter combination using the training data and use these trained models on the test data and analyze the performance. Some of the parameters are trivial to understand. For example, since the addressed problem is a binary classification problem, binary cross-entropy is used as a loss function. The Adaptive Moment Estimation (Adam) optimizer is used for optimization. We choose Adam optimizer with a moderate learning rate to avoid unstable training or failure in the training model. However, two parameters, namely kernel size, and batch size are important and can affect the presented model's performance. Thus, to identify the right kernel size and batch size, several experiments of BerConvoNet on Dataset-1 have been performed with varying kernel size and batch size. We considered different batch sizes for each of the kernel sizes, like batch size 20, 40, 60, 80, and 100. Fig. 3 graphically represents the experimental results. From the results, the following findings have been observed:

- Accuracy of *BerConvoNet* with kernel size $\langle 2, 3, 4, 5\rangle$ is significantly better than other kernel sizes irrespective of batch size. Moreover, higher accuracy is achieved with batch size 100 for kernel size $\langle 2, 3, 4, 5\rangle$.
- Precision of *BerConvoNet* with kernel size $\langle 2, 3, 4, 5\rangle$ is better than other kernel sizes irrespective of batch size. In kernel size $\langle 2, 3, 4, 5\rangle$ higher precision is achieved with respect to batch size 30.

Table 3
List of performance metrics used.

Performance metrics	Formula	Description
Accuracy	$\frac{TP+TN}{TP+FP+TN+FN}$	Accuracy refers to the amount of accurate assumptions the algorithm produces for forecasts of all sorts.
Precision	$\frac{TP}{TP+FP}$	Precision is the percentage of successful cases that were reported correctly.
Recall	$\frac{TP}{TP+FN}$	It is the number of right positive outcomes divided by the number of all related samples (including samples that were meant to be positive).
F1-score	$\frac{2\times P\times R}{P+R}$	It is the harmonic mean of precision and recall values.
MCC	$\frac{(TP*TN-FP*FN)}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}}$	MCC calculates the correlation coefficient between the actual values of the class label and the predicted values of the class label.
Specificity	$\frac{Tn}{TP+FN}$	It is used to calculate the fraction of negative values correctly classified
G-mean	$\sqrt{TP \times TN}$	It stands for geometric means. G-mean is calculated as the square root of the product of recall and specificity.

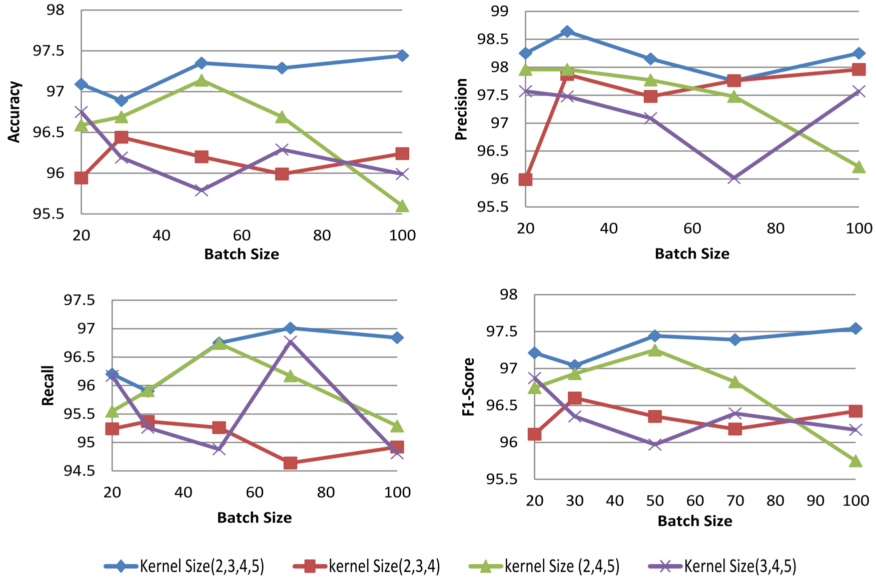


Fig. 3. Experimental result of *BerConvoNet* with different parameters.

- Recall of *BerConvoNet* with kernel size ⟨2, 3, 4, 5⟩ is better than other kernel sizes irrespective of batch size.
- Similarly, significantly better F1-score is achieved with kernel size ⟨2, 3, 4, 5⟩ and batch size 100.
- Overall, it can be observed that kernel size ⟨2, 3, 4, 5⟩ is a better parameter compared to other kernel sizes.

Thus for the performance evaluation of *BerConvoNet* model and comparison with other model, kernel size ⟨2, 3, 4, 5⟩ and batch size 100 have been selected in this work.

4.6. Abalative experiments and results

The methodology discussed in Section 3 and the experimental procedure discussed in the above subsections have been used for the experimental evaluation. Table 4 shows the comparison results of the presented approach with other models. From the results, it can be observed that:

- The proposed model shows the highest accuracy among the other baseline models. It gives an accuracy of 94.25% on Dataset-1 and 97.45% accuracy in Dataset-2. Accuracy on Dataset-3 and Dataset-4 is 75.18% and 90.27%, respectively.
- Precision of the BerConvoNet model is significantly better with 95.13%, 96.85%, and 74.54% precision on Dataset-1, Dataset-2, and Dataset-3, respectively. It is comparable to Elmo with the neural network model on Dataset-4
- While the Recall of the presented model is comparable with Random Embeddings with LSTM model over Dataset-1, the Static Glove Embeddings with CNN shows the highest Recall measure. The predicted labels with high recall but low precision provide many results riddled with incorrect labels compared to training labels. The presented model has the highest Precision and Recall measure over Dataset-2. An ideal system would have high recall (for returning many results) along with high precision (results labeled correctly). Unlike other algorithms which show high recall and low precision (like Static Glove+CNN on Dataset-1, Dynamic Glove+CNN on Dataset-3, and Random Embedding+CNN on Dataset-4), the proposed BerConvoNet model has shown high precision and high recall on all four datasets.
- F1 score of BerConvoNet is highest on the first three datasets and is comparable to Elmo with the NN model on Dataset-4.
- Overall the proposed model, BerConvoNet, performs better than the other models on all four fake news datasets.

Visualization of the true positive rate against the false-positive rate can be done using Receiver Operating Characteristic (ROC) Curve. This curve shows the correlation between sensitivity and Specificity. Sensitivity and Specificity have negative co-variance

Table 4
Comparison results of the presented approach with baseline models.

Models	Precision	Recall	F1 Score	MCC	Accuracy	Specificity	G-mean
DATASET-1							
Random+ CNN	0.7854	0.7093	0.7454	0.5548	0.7800	0.8388	0.7713
Static-GloVe+ CNN	0.5191	**0.9799**	0.6787	0.1674	0.5380	0.0996	0.3124
Dynamic Glove+ CNN	0.7795	0.8049	0.7920	0.5844	0.7920	0.7795	0.7921
Random+LSTM	0.8115	0.9399	0.8710	0.7307	0.8608	0.7816	0.8571
ELMo+NN	0.9512	0.9215	0.9362	0.8722	0.9358	0.9508	0.9360
BERT+ CNN	**0.9513**	0.9355	**0.9433**	**0.8851**	**0.9425**	**0.9598**	**0.9426**
DATASET-2							
Random+CNN	0.8622	0.7886	0.8238	0.6698	0.8340	0.8780	0.8321
Static-Glove+ CNN	0.5958	0.9116	0.7206	0.3499	0.6480	0.3865	0.5936
Dynamic Glove+ CNN	0.8475	0.8333	0.8403	0.6920	0.8462	0.8583	0.8457
Random+LSTM	0.9498	0.9190	0.9342	0.8709	0.9352	0.9514	0.9351
ELMo+NN	0.8784	0.8996	0.8889	0.7758	0.8877	0.8760	0.8877
BERT+CNN	**0.9685**	**0.9825**	**0.9754**	**0.9490**	**0.9745**	**0.9659**	**0.9742**
DATASET-3							
Random+CNN	0.6951	0.6513	0.6724	0.3936	0.6980	0.7404	0.6944
Static-Glove+ CNN	0.8055	0.3610	0.4986	0.3398	0.6500	0.9189	0.5759
Dynamic Glove+ CNN	0.6116	**0.9113**	0.7326	0.3832	0.6667	0.4200	0.6194
Random+LSTM	0.6970	0.7823	0.7322	0.4455	0.7211	0.6599	0.7185
ELMo+NN	0.7168	0.7536	0.7347	0.4704	0.7347	0.7168	0.7350
BERT+CNN	**0.7454**	0.7634	**0.7543**	**0.5038**	**0.7518**	**0.7403**	**0.7518**
DATASET-4							
Random+CNN	0.7391	0.3542	0.4788	0.2835	0.6300	0.8846	0.5597
Static-Glove+ CNN	0.8064	0.6410	0.7143	0.5711	0.7989	0.9008	0.7599
Dynamic Glove+ CNN	0.8105	0.8750	0.8415	0.7088	0.8543	0.8378	0.8562
Random+LSTM	0.7600	**0.9047**	0.8261	0.6305	0.8095	0.7143	0.8039
ELMo+NN	**0.8993**	0.8803	**0.8897**	0.7875	0.8938	0.9067	0.8934
BERT+CNN	0.8696	0.8889	0.8791	**0.7978**	**0.9027**	**0.9118**	**0.9003**

such that an increase in specificity results in a decrease in sensitivity. The test accuracy increases as the graph move closer towards the top and left-hand borders. It can be observed from Figs. 4, 5 that ROC curve is better for the *BerConvoNet* model for all the four the datasets. It should be noted that the accuracy of the graph increases as it approaches the diagonal. If a graph goes perpendicular from zero up to top-left and then linearly parallel to the horizontal, it is said to be a perfect test. Fig. 4 and 5 shows the ROC plots of all the models.

4.7. Comparison with existing state of the art work

We have compared the performance of presented approach, *BerConvoNet*, with the existing state-of-the art work. The summary of the comparative analysis is given in Table 5.

We have compared the results of the best performing techniques with our best performing presented model in the table. In [52], authors propose a Two-Level Convolutional Neural Network with User Response Generator (TCNN-URG). Their model reported 89.84% accuracy on the Weibo dataset and 88.83% accuracy on the self-collected dataset for fake news detection. They only present accuracy metrics for performance evaluation. Authors in [36] use handcrafted features with the existing classifier and report that RF and XGB classifiers have comparable accuracy of 85% and 85%, respectively. This is very low compared to our presented work. In their work [3] and [53] present accuracy, precision, recall, and f1-score metrics on different datasets.

From Table 5, it can be observed that the presented work has shown better results than all the existing works for all the used performance measures. Further, we have used seven different performance measures to do an extensive evaluation for fake news detection, which any previous works have not considered.

5. Conclusion

This paper presents a hybrid *BerConvoNet* model for fake news classification. Firstly, this model generated news embedding representations from news articles. Next, this embedding was given into the proposed convolutional neural network for classification. The presented model adopts a multiscale feature learning from news articles. It is based on the concatenation of BERT and convolutional neural networks. The effectiveness of the presented model is tested on four benchmark datasets. From the experimental results, it is evident that *BerConvoNet* is effective in identifying the truthfulness of a news article.

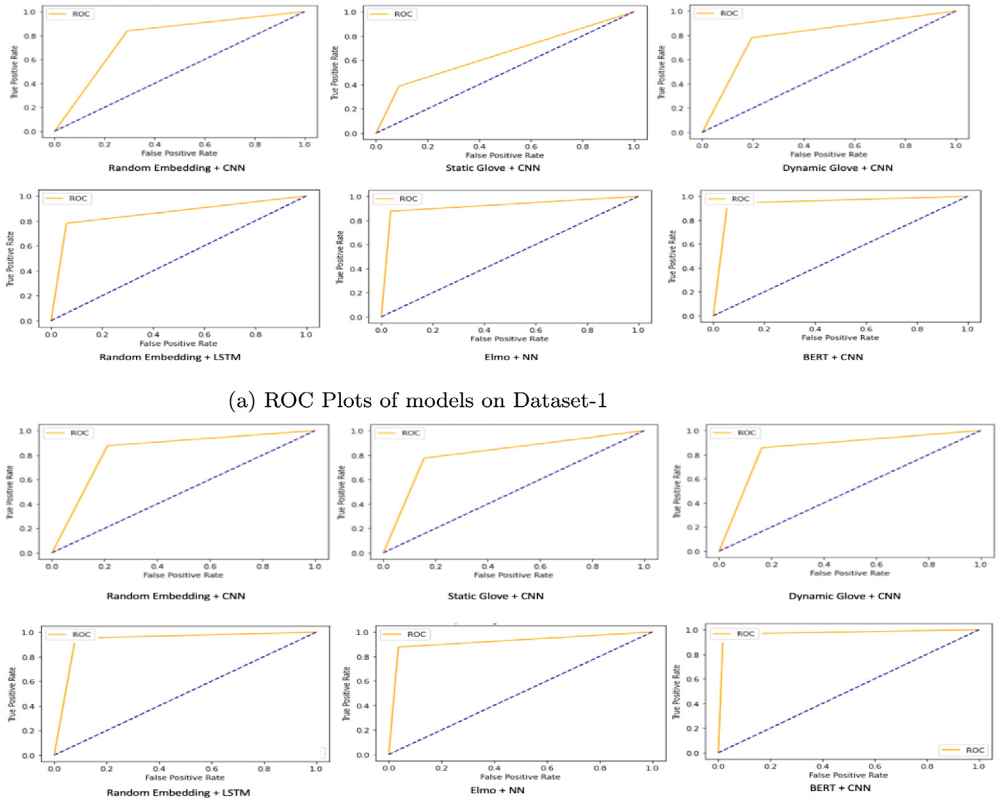
In the future, multi-model learning could be a possible extension of the proposed work. In multi-model learning, images associated with the news text are also considered for the fake news classification.

CRediT authorship contribution statement

Monika Choudhary: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Writing - original draft. **Satyendra Singh Chouhan:** Conceptualization, Formal analysis, Investigation, Supervision, Validation, Writing - review & editing. **Emmanuel S. Pilli:** Conceptualization, Formal analysis, Investigation, Supervision, Validation, Writing - review & editing. **Santosh Kumar Vipparthi:** Validation, Writing - review & editing.

Declaration of competing interest

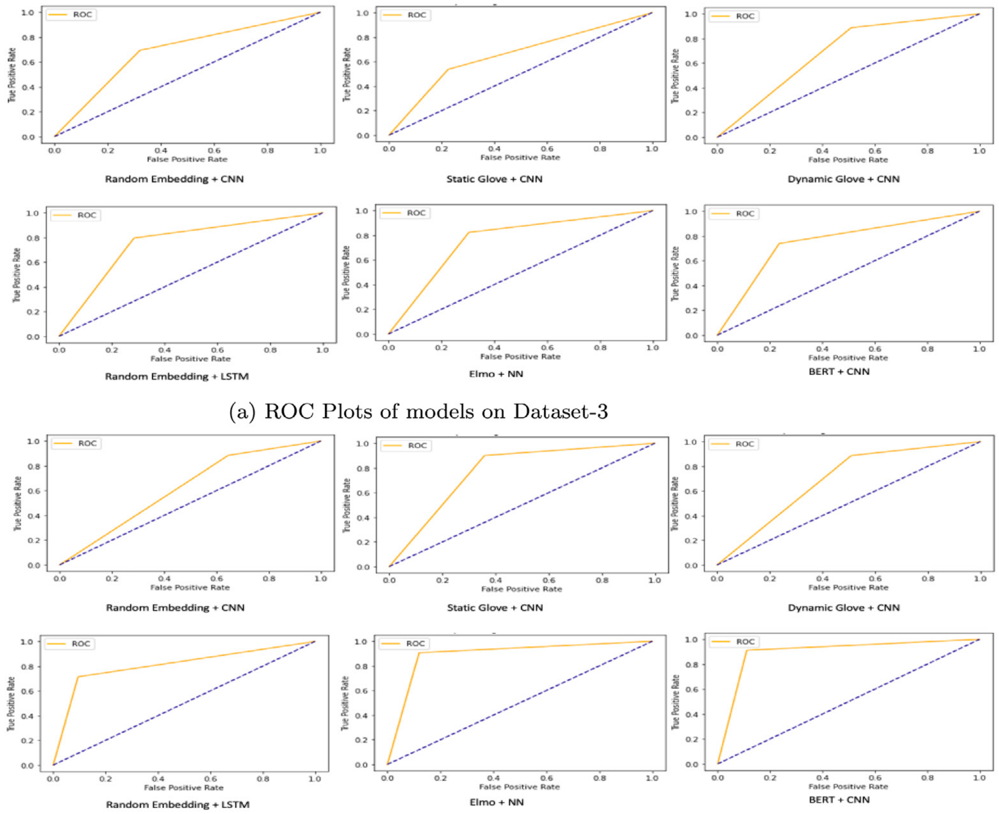
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.



(a) ROC Plots of models on Dataset-1

(b) ROC Plots of models on Dataset-2

Fig. 4. ROC Plots of different Models on Dataset-1 and Dataset-2.



(a) ROC Plots of models on Dataset-3

(b) ROC Plots of models on Dataset-4

Fig. 5. ROC Plots of different Models on Dataset-3 and Dataset-4.

Table 5
Comparison of presented work with other related work for fake news identification.

Related work	Dataset used	Learning model	Result reported	Remarks
Qian et al. [52]	Weibo & self collected Twitter dataset	TCNN-URG	Accuracy — 89.8% (best result on Weibo dataset)	This work relies on user response that may not be available for early fake news detection
Reis et al. [36]	Buzzfeed	XGB	Accuracy — 86% F1 score — 81%	
Khan et al. [53]	Liar, George M and self made dataset	Char level LSTM	Precision — 95% Recall — 95% Âccuracy — 89% F1 score — 95%	When dataset size is fixed, the accuracy of the model is proportional to the average article length of news. Hence, may not perform well for short news articles
Ruchansky et al. [43]	Twitter, Weibo	CSI hybrid deep model	Accuracy — 95.3% F1 score — 95.4% (best result on Weibo dataset)	The results are promising but the context in news articles can be explored further for better understanding
Singhania et al. [54]	Mixed corpus	Three level hierarchical attention network	Accuracy — 96.8%	The feature extraction process could be elaborated with more detail on different performance metrics
Shu et al. [3]	Gossipcop, Politifact	Sentence-comment co-attention network	Precision — 90.2% Recall — 95.6% Âccuracy — 90.4% F̂1 score — 92.8%	The proposed model is promising and can be explored to further improve results.
Presented work	George M., Kaggle, GossipCop, Politifact	BerConvoNet	Precision — 96.85% Recall — 98.25% Âccuracy — 97.45% F̂1 score — 97.54% M̂CC — 94.90% Ŝpecificity-96.59% Ĝ-Mean — 97.42%	Presents better results on different performance metrics in comparison with other related work.

References

[1] A.P. Institute, Social and demographic differences in news habits and attitudes, 2014.

[2] D.M. Dougall, How Finland's fake four-day week became a 'fact' in Europe's media, 2020.

[3] K. Shu, L. Cui, S. Wang, D. Lee, H. Liu, defend: Explainable fake news detection, in: Proceedings of ACM Sigkdd International Conference on Knowledge Discovery & Data Mining, 2019, pp. 395–405.

[4] Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, J. Gao, Eann: Event adversarial neural networks for multi-modal fake news detection, in: Proceedings of ACM Sigkdd International Conference on Knowledge Discovery & Data Mining, 2018, pp. 849–857.

[5] G.L. Ciampaglia, P. Shiralkar, L.M. Rocha, J. Bollen, F. Menczer, A. Flammini, Computational fact checking from knowledge networks, PLoS One 10 (2015) e0128193.

[6] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, R. Mihalcea, Automatic detection of fake news, 2017, arXiv preprint arXiv:1708.07104.

[7] B. Shi, T. Weninger, Discriminative predicate path mining for fact checking in knowledge graphs, Knowl.-Based Syst. 104 (2016) 123–133.

[8] N. Sitaula, C.K. Mohan, J. Grygiel, X. Zhou, R. Zafarani, Credibility-based fake news detection, in: Disinformation, Misinformation, and Fake News in Social Media, Springer, 2020, pp. 163–182.

[9] J. Zhang, B. Dong, S.Y. Philip, Fakedetector: Effective fake news detection with deep diffusive neural network, in: Proceedings of International Conference on Data Engineering, ICDE, 2020, pp. 1826–1829.

[10] A. Thota, P. Tilak, S. Ahluwalia, N. Lohia, Fake news detection: A deep learning approach, SMU Data Sci. Rev. 1 (2018) 10.

[11] R.K. Kaliyar, A. Goswami, P. Narang, DeepFake: Improving fake news detection using tensor decomposition-based deep neural network, J. Supercomput. (2020).

[12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.

[13] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification, in: Proceedings of AAAI Conference on Artificial Intelligence, 2015.

[14] L. Hamel, Fake-news, 2018, https://raw.githubusercontent.com/lutzhamel/fake-news/master/data/fake_or_real_news.csv.

[15] Kaggle, Fake news, 2018, https://www.kaggle.com/c/fake-news/data.

[16] K. Shu, Fakenewsnet, 2019, https://github.com/KaiDMML/FakeNewsNet.

[17] B.M. DePaulo, J.J. Lindsay, B.E. Malone, L. Muhlenbruck, K. Charlton, H. Cooper, Cues to deception, Psychol. Bull. 129 (2003) 74.

[18] R. Mihalcea, C. Strapparava, The lie detector: Explorations in the automatic recognition of deceptive language, in: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, 2009, pp. 309–312.

[19] H. Ahmed, I. Traore, S. Saad, Detection of online fake news using n-gram analysis and machine learning techniques, in: Proceedings of International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments, Springer, 2017, pp. 127–138.

[20] S. Feng, R. Banerjee, Y. Choi, Syntactic stylometry for deception detection, in: Proceedings of Annual Meeting of the Association for Computational Linguistics, vol. 2: Short Papers, 2012, pp. 171–175.

[21] H. Karimi, J. Tang, Learning hierarchical discourse-level structure for fake news detection, 2019, arXiv preprint arXiv:1903.07389.

[22] V.L. Rubin, Y. Chen, N.K. Conroy, Deception detection for news: Three types of fakes, Proc. Assoc. Inf. Sci. Technol. 52 (2015) 1–4.

[23] V.L. Rubin, N.J. Conroy, Y. Chen, Towards news verification: Deception detection methods for news discourse, in: Proceedings of International Conference on System Sciences, 2015, pp. 5–8.

[24] Y.R. Tausczik, J.W. Pennebaker, The psychological meaning of words: LIWC and computerized text analysis methods, J. Lang. Soc. Psychol. 29 (2010) 24–54.

[25] G. Gravanis, A. Vakali, K. Diamantaras, P. Karadais, Behind the cues: A benchmarking study for fake news detection, Expert Syst. Appl. 128 (2019) 201–213.

[26] J.Z. Pan, S. Pavlova, C. Li, N. Li, Y. Li, J. Liu, Content based fake news detection using knowledge graphs, in: Proceedings of International Semantic Web Conference, Springer, 2018, pp. 669–683.

[27] K. Shu, H.R. Bernard, H. Liu, Studying fake news via network analysis: Detection and mitigation, in: Emerging Research Challenges and Opportunities in Computational Social Network Analysis and Mining, Springer, 2019, pp. 43–65.

[28] S.C.R. Gangireddy, C. Long, T. Chakraborty, Unsupervised fake news detection: A graph-based approach, in: Proceedings of ACM Conference on Hypertext and Social Media, 2020, pp. 75–83.

[29] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proceedings of Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.

[30] Z. Jin, J. Cao, Y. Zhang, J. Luo, News verification by exploiting conflicting social viewpoints in microblogs, in: Proceedings of Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[31] S. Vosoughi, D. Roy, S. Aral, The spread of true and false news online, Science 359 (2018) 1146–1151.

[32] S. Gupta, R. Thirukovalluru, M. Sinha, S. Mannarswamy, CIMTDetect: A community infused matrix-tensor coupled factorization based method for fake news detection, in: Proceedings of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2018, pp. 278–281.

[33] K. Shu, S. Wang, H. Liu, Beyond news contents: The role of social context for fake news detection, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019, pp. 312–320.

[34] K. Popat, S. Mukherjee, A. Yates, G. Weikum, Declare: Debunking fake news and false claims using evidence-aware deep learning, 2018, arXiv preprint arXiv:1809.06416.

[35] G. Jodhani, Fake News Detection using Text Similarity Approach (Ph.D. thesis), GB Pant University of Agriculture and Technology, Pantnagar, Uttarakhand, 2019.

[36] J.C. Reis, A. Correia, F. Murai, A. Veloso, F. Benevenuto, Supervised learning for fake news detection, IEEE Intell. Syst. 34 (2019) 76–81.

[37] P. Kaur, R.S. Boparai, D. Singh, Hybrid text classification method for fake news detection, Int. J. Eng. Adv. Technol. 8 (2019) 2388–2392.

[38] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1746–1751, http://dx.doi.org/10.3115/v1/D14-1181, URL: https://www.aclweb.org/anthology/D14-1181.

[39] S. Wang, M. Huang, Z. Deng, Densely connected CNN with multi-scale feature attention for text classification, in: IJCAI, 2018, pp. 4468–4474.

[40] C. Li, G. Zhan, Z. Li, News text classification based on improved Bi-LSTM-CNN, in: 2018 9th International Conference on Information Technology in Medicine and Education, ITME, IEEE, 2018, pp. 890–893.

[41] L. Wu, H. Liu, Tracing fake-news footprints: Characterizing social media messages by how they propagate, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, pp. 637–645.

[42] H. Jwa, D. Oh, K. Park, J.M. Kang, H. Lim, ExBAKE: Automatic fake news detection model based on bidirectional encoder representations from transformers (BERT), Appl. Sci. 9 (2019) 4062.

[43] N. Ruchansky, S. Seo, Y. Liu, Csi: A hybrid deep model for fake news detection, in: Proceedings of ACM on Conference on Information and Knowledge Management, 2017, pp. 797–806.

[44] H. Allcott, M. Gentzkow, Social media and fake news in the 2016 election, J. Econ. Perspect. 31 (2017) 211–236.

[45] N. Ketkar, Introduction to keras, in: Deep Learning with Python, 2017, pp. 97–111.

[46] H. Jangid, S. Singhal, R.R. Shah, R. Zimmermann, Aspect-based financial sentiment analysis using deep learning, in: Companion Proceedings of the the Web Conference, 2018, pp. 1961–1966.

[47] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proceedings of Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1532–1543.

[48] S. De Sarkar, F. Yang, A. Mukherjee, Attending sentences to detect satirical fake news, in: Proceedings of International Conference on Computational Linguistics, 2018, pp. 3371–3380.

[49] C. Baziotis, N. Pelekis, C. Doulkeridis, Datastories at semeval-2017 task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis, in: Proceedings of International Workshop on Semantic Evaluation, 2017, pp. 747–754.

[50] W. Liu, B. Wen, S. Gao, J. Zheng, Y. Zheng, A multi-label text classification model based on elmo and attention, in: MATEC Web of Conferences, vol. 309, 2020, p. 03015.

[51] Z. Huang, W. Zhao, Combination of ELMo representation and CNN approaches to enhance service discovery, IEEE Access 8 (2020) 130782–130796.

[52] F. Qian, C. Gong, K. Sharma, Y. Liu, Neural user response generator: Fake news detection with collective user intelligence, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, vol. 18, 2018, pp. 3834–3840.

[53] J.Y. Khan, M. Khondaker, T. Islam, A. Iqbal, S. Afroz, A benchmark study on machine learning methods for fake news detection, 2019, arxiv preprint arXiv:1905.04749.

[54] S. Singhania, N. Fernandez, S. Rao, 3han: A deep neural network for fake news detection, in: Proceedings of International Conference on Neural Information Processing, 2017, pp. 572–581.