An Efficient Fake News Detection System Using Contextualized Embeddings and Recurrent Neural Network

Junaid Ali Reshi[1]*, Rashid Ali[1,2]

[1] Department of Computer Engineering, Aligarh Muslim University, Aligarh (India)
[2] Interdisciplinary Centre for Artificial Intelligence, Aligarh Muslim University, Aligarh (India)

Received 24 February 2022 | Accepted 12 January 2023 | Early Access 9 February 2023

Abstract

Fake news is detrimental for society and individuals. Since the information dissipation through online media is too quick, an efficient system is needed to detect and counter the propagation of fake news on social media. Many studies have been performed in last few years to detect fake news on social media. This study focuses on efficient detection of fake news on social media, through a Natural Language Processing based approach, using deep learning. For the detection of fake news, textual data have been analyzed in unidirectional way using sequential neural networks, or in bi-directional way using transformer architectures like Bidirectional Encoder Representations from Transformers (BERT). This paper proposes Contextualized Fake News Detection System (ConFaDe) - a deep learning based fake news detection system that utilizes contextual embeddings generated from a transformer-based model. The model uses Masked Language Modelling and Replaced Token Detection in its pre-training to capture contextual and semantic information in the text. The proposed system outperforms the previously set benchmarks for fake news detection; including state-of-the-art approaches on a real-world fake news dataset, when evaluated using a set of standard performance metrics with an accuracy of 99.9 % and F1 macro of 99.9%. In contrast to the existing state-of-the-art model, the proposed system uses 90 percent less network parameters and is 75 percent lesser in size. Consequently, ConFaDe requires fewer hardware resources and less training time, and yet outperforms the existing fake news detection techniques, a step forward in the direction of Green Artificial Intelligence.

Keywords

Contextualized Embeddings, Deep Learning, Fake News Detection, Natural Language Processing.

DOI: 10.9781/ijimai.2023.02.007

I. Introduction

Fake news has been a buzzword, often heard in journalistic discussions and political discourse. People from various backgrounds use it in different contexts and meanings, as per their understanding, to refer to misinformation, disinformation, rumors, and fake news, etc. There are varying definitions of fake news. Some definitions of fake news are so ambiguous that they eliminate the boundaries between the concepts of fake news, misinformation, disinformation, satire, or even improper and personally offensive news [1], [2]. Among other definitions in the literature, the most consistent definition of fake news is 'news that is intentionally and verifiably false' [3]–[5].

Fake news is a phenomenon that can cause serious consequences. These consequences may result in personal, national or global harm. Fake news is shown to spread more quickly than real news [6], and its impact has been studied in various situations, particularly elections [7], [8]. The rapid dissemination of fake news can have serious repercussions. The spread of fake news can cause the democratic processes to be undermined and create chaos. It can create distrust in neutral agencies and spread pseudo-science, thus hurting the communities on a large scale. It has been observed that anti-social elements spread fake news through social media to create law and order problems [8]. Early detection of fake news, is very important in this scenario as it somewhat helps mitigate the ill effects of fake news. It has also been observed that fake news, if encountered by debunking and presenting true news, the retractions fail to completely eradicate the influence of the misinformation[9],[10]. Therefore, a need is felt to detect Fake news at an early stage so that it will not propagate, and as such, we can reduce the harms of fake news spread to a great extent [11].

The problem of mitigating and detecting fake news has been studied by various researchers through different approaches. While some researchers are much concerned about the technicalities in the detection and mitigation of fake news on social media, social scientists have been focusing on various psychological aspects of fake news spread and the damage it causes [4], [12]. In empirical studies, many approaches of fake news detection have been experimented with and discussed. Different features have been utilized for the detection of fake news in online social networks through machine learning and deep learning frameworks. As an intelligent system through machine learning needs feature engineering [13], different works have focused on different features. Some of the works have focused on social context features [14], while some have used content-

* Corresponding author.
E-mail address: jreshi14@gmail.com

Please cite this article in press as:
J. A. Reshi, R. Ali. An Efficient Fake News Detection System Using Contextualized Embeddings and Recurrent Neural Network, International Journal of Interactive Multimedia and Artificial Intelligence, (2023), http://dx.doi.org/10.9781/ijimai.2023.02.007

based features [4], [12], [15], [16] to detect fake news. Among other approaches, the propagation aspect of fake news has also been studied and experimented with [17]. Fake news detection using diffusion networks, virality prediction based on network structure, and finding an influential node to determine the dynamics of fake news have also been studied [18], [19]. Fake news detection as a natural language processing problem has also been studied extensively [15], [20].

A. Our Contribution

- This paper proposes a novel fake news detection system – ConFaDe, that uses contextualized word embeddings generated through ELECTRA-based transformer model as an input to LSTM based deep neural network. This model is pre-trained through replaced token detection and masked language modeling tasks.
- The proposed system is evaluated on a well-known benchmark real-world fake news dataset based on the 2016 U.S. presidential elections. It outperforms the existing state-of-the-art (SOTA) Fake news detection system-FakeBERT [36].
- The proposed system uses 90 percent less parameters than FakeBERT and is 75 percent lesser in size than the same. While the FakeBERT model has 135.5M parameters, ConFaDe has only 14.09M. Moreover, ConFaDe utilizes ELECTRA-small as text encoder, which is about ¼ of the size of BERTbase, utilized by FakeBERT as text encoder.
- The proposed system achieves an accuracy of 99.9 percent and an F1 macro of 99.9 percent while training on fewer parameters, consuming fewer resources, and utilizing less hardware, making it an efficient and accurate model for detecting fake news. As the given model intuitively consumes less power, it leads to lesser carbon emission, therefore, taking a step forward in achieving sustainable models for Green Artificial Intelligence (AI).

The rest of this paper is structured as follows: Section II presents an essential background of relevant concepts. Section III contains a review of the pertinent literature. Section IV presents details about the approach and the proposed system. Section V details the experimental setup including the hardware, the software, and various parameters and configurations of the system. Section VI contains the results obtained through experiments and the discussion thereof. The discussion comprises a comparison with baselines, existing works, and the state-of-the-art system on various parameters. Section VII concludes the discussion and provides some future directions.

II. Background

In this section, we discuss various concepts we have used in our set of experiments. We also discuss different approaches to fake news detection and ensuing directions.

A. Word Vector Encodings and Embeddings

To process text, we need to do ample and tailored pre-processing. We cannot have raw text as an input to existing deep learning classifiers. Therefore, we pre-process text and convert it to a vector representation for further processing by deep learning classifiers. Apart from initial pre-processing like stop word removal, stemming, and lemmatization, we need to use a word encoding/embedding technique to create a text vector. There are different types of encodings and word embedding techniques. We briefly discuss GloVe-based embeddings as we further use them in our experiments.

1. GloVe

GloVe, expanded as 'Global Vectors for word representation,' is an unsupervised model for learning vector word representation through training on an aggregated global word-to-word co-occurrence matrix from a large text corpus. It can be used to find word relations like synonyms, antonyms, and other semantic relations like city-capitals, currency-capitals, role-salutation, etc. However, it is not efficient enough to determine word relations such as homonyms [21].

Although there are multiple versions of pre-trained GloVe word embeddings available online, we have used a 300-dimension vector GloVe model trained on 6 billion tokens and 400 thousand words vocabulary from Wikipedia 2014 Gigaword 5 corpus.

2. ELECTRA

'Efficiently Learning an Encoder that Classifies Token Replacements Accurately,' condensed as ELECTRA, is a Bidirectional Encoder Representations from Transformers (BERT) like pre-trained model that generates dense vector representations for natural language tasks. BERT is a Google-developed deep learning framework based on attention mechanism. It is pre-trained on Masked Language Model (MLM) and Next Sentence Prediction (NSP) tasks, useful for various downstream natural language processing tasks [22]. There are many versions of BERT, depending on the sentence length, corpus trained on, number of encoders, and number of attention heads used. In its pre-training, ELECTRA partially replaces the MLM task in BERT with the replaced token detection task (RTD) [23]. Somewhat like Generative Adversarial Networks (GANs), albeit with maximum likelihood, and not adversarial training, the ELECTRA model is trained to discriminate between 'real' and 'fake' input data, which is infused by corrupting some input tokens with plausible 'fake' tokens. The pre-training task requires the discriminator part of the model to determine the corrupted or intact tokens, which are fed by a small generator network. The addition of the RTD task has led to improvement in the model's performance for a given size, computing power, and data [23].

III. Literature Review

Machine Learning and deep learning find their application in most modern-day intelligent applications. Similarly, machine learning and deep learning models have found extensive usage in the task of fake news detection [24] [25]. With Machine Learning, feature engineering is the most crucial step, where working on the features is an essential step in improving the performance of a model [26], [27]. In early studies, to solve the problem of fake news detection through machine learning, a manual set of features were designed along with lexical and syntactical features [28], [29]. Among other machine learning algorithms, Random forest (R.F.), Support Vector Machines (SVM), and Naïve Bayes (N.B.) classifiers have been extensively used to detect fake news [1], [20], [24], [25], [30]. In machine learning models, apart from the pre-processing, text requires a lot of feature engineering before it is ready as an input. With the advent of deep learning driven technologies, deep learning models gradually replaced machine learning approaches. It was predominantly due to the fact that unlike machine learning models, deep learning models do not require explicit feature engineering to perform well. The input to these deep learning frameworks can be in the form of text, images, or videos, depending on the type of fake news detection approach [31]. In multimodal fake news detection, multiple architectures are combined to form a hybrid architecture, which is then used to detect fake news in text, images, and videos [32]. For text, the content is embedded at the word or sentence level [33]; for image, a pixel-based tensor is used as an input form to the deep learning framework [32]. After the preparation of input tensors, many neural network based architectures like Convolution Neural Networks (CNNs) [34]–[36] and Recurrent Neural Networks (RNNs) have been used. CNNs are primarily used to extract features from the text in a more efficient way. CNNs with average or max pooling find their use in fake news detection tasks. Recurrent Neural Networks such as Gated Recurrent Units (GRUs) [37], LSTMs [38], [39], and

Bidirectional recurrent neural networks (BRNNs) find great utility in text processing [40]. GRUs contain only two gates –reset and update and as such are easier to use than LSTM, which consists of an input gate, an output gate, a forget gate, and a cell. While LSTM can recall short-term memories for a long time, so as to aid in forming the context, it still processes the text in a unidirectional way[38]. It has also been observed that preference of GRUs over LSTM or LSTM over GRUs is decided by the computational resources available[41]. Recently, many works have focused on using transformers for extracting multimodal features of news content [42], [43].

In terms of the type of features used for fake news detection, many researchers approach the problem of fake news detection as different tasks like News content detection problem [4], [6], [44], News diffusion dynamics [45]–[47], and/or social context based problem[14]. As Social media content is primarily composed of text, news content-based features focusing on text are much helpful in fake news detection [48], [24].

In the existing literature, various configurations of deep learning networks combined with different features have been used to mitigate the fake news detection problem. These deep learning methods have been preferred as they are able to capture different patterns in text (news) implicitly, which machine learning algorithms are not capable of capturing until explicitly engineered. As raw text still cannot be placed as an input to the deep learning architecture, different word encoding and embedding schemes have been used as the first step[49]. Additionally, many efforts have been made to make programming configurations easier to implement and reduce the computational power needed [50]. In one of the pioneering works of using a deep learning framework for detecting fake news, Ma et al. use basic tf-idf based text encoding in their deep learning architecture for detecting fake news [51]. Their model performed better than many state-of-the-art (SOTA) machine learning based models at the time [51]. Many text embeddings have been used to simplify the problem of fake news detection. Document-level embeddings [52], Sentence-level embeddings [52], [53], and word-level embeddings have been utilized by various researchers to generate input vectors for a Machine learning or deep learning classifier [1]. Some studies use word-level embeddings like Word2Vec [54] and GloVe [55] to obtain text vectors for subsequent use by deep neural networks. Many language models that are based on RNNs and Transformers, like Embeddings from Language Models (ELMo) [56], FastText [57], and BERT [36] [58] have also been utilized for the generation of text embeddings. These text embeddings have been further used in deep learning models for fake news detection [58]–[61]. However, the problem with huge models like BERT is that it requires a lot of computational power and time to use them for downstream tasks, and thus, in the long run, we look for a better alternative.

Further, among deep learning models, some researchers have used CNNs, some have used RNNs, while some have used ensemble approaches for developing the frameworks to detect fake news [12], [16], [45]. Irrefutably, there has been a quest for getting the right approach in selecting a text embedding along with a neural network, as both are essential to the performance of a model/system. Improvement in either of these two research areas can lead to better systems for fake news detection. The recent techniques in fake news detection include use of Graph Neural Networks (GNN), Generational Adversarial Networks (GANs) and ensemble approaches. Graph neural networks operate on graph structure by recursive node classification. They process the global structural features better than other deep learning architectures [62]. Graph Convolutional Networks (GCN) and Propagation Graph Neural networks (PGN) are some important techniques that belong to Graph Neural Networks [63]. Generative adversarial networks are also used for fake news detection, albeit for images and videos only [63]. Adversarial training is done to generate synthetic fake images and videos. DeepFake is an area of applications of GANs where fake news detection can be indirectly achieved with the help of GANs [64]. Ensemble methods are created by combining several models for performing a single task at the end. CNN+LSTM ensembles have been used frequently for fake news detection [63]. Ensembles based on Bi-LSTM + CNN, RNN + SVM, Attention mechanism+RNN and other configurations have been tried in the literature [65], [66].

IV. Methodology

In this section, we discuss the proposed fake news detection system, starting from data preparation and culminating by describing the model in detail.

The core objective of this study is to develop an efficient and accurate fake news detection system. By using fewer resources, the system should be capable of detecting fake news with an accuracy that it may outperform the state-of-the-art system on this task. For its accuracy, we evaluate the system using different performance measures, as explained in the Section V.D. For measuring the efficiency, we compare the system with an already existing state-of the-art (SOTA) system in terms of parameters and resources used (details in Section VI.C).

We propose an LSTM-based model that leverages contextualized embeddings generated from a transformer-based architecture-ELECTRA, to detect Fake News. We call this model as ConFaDe. The process of fake news detection is illustrated in Fig. 1.

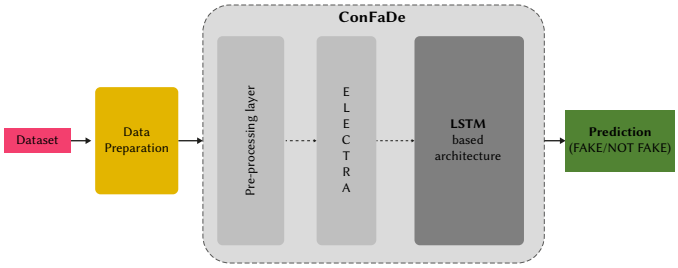

Fig. 1. Process of Fake News detection in the proposed system.

As seen in Fig. 1, we prepare the dataset at the first step of the process. The prepared dataset is then fed to ConFaDe architecture, which classifies the news as fake or not fake. The detailed process of data preparation is discussed in Section IV.A. Inside ConFaDe, before providing input to the LSTM based architecture, we pre-process the data through a layer to make data ready for our ELECTRA architecture. Subsequently, an LSTM based deep learning architecture classifies the news as 'fake' or 'not fake'. The detailed architecture of the proposed model is discussed in Section IV.C. Our model uses the word embeddings generated through the ELECTRA-Small++ model, pre-trained on a large uncased English text corpus by Google Research.

A. Data Preparation

Benchmark representative datasets are a standard in evaluating performance of a system [67],[68], [69], [70]. We select a well-known benchmark dataset of fake news, which contains fake and real news propagated during 2016 U.S. Presidential elections [36], [68], [69]. The raw input file is first examined for inconsistencies. Values that are in Arabic or are not legible and do not contain English language are removed, as our transformer model is trained on English corpus. After removing such values, we replace 'Null' and 'nan' values with blanks. We further perform initial pre-processing on text, removing punctuations, stopwords and URLs. After this process, we are left

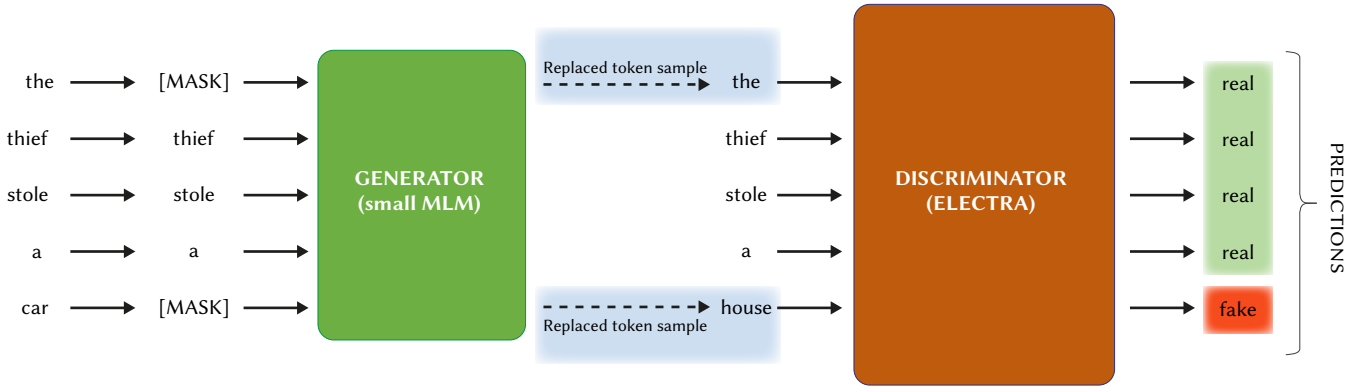

Fig. 2. Exemplifying Replaced Token Detection and prediction in ELECTRA.

with 20718 labeled instances. We further split the dataset in the ratio of 90:10 for Training + validation (18646 instances) and testing (2072 instances). Of the 18646 instances, 13053 were used for training, and 5594 were used for validation, in the ratio of 70:30.

B. ELECTRA Training and Hyperparameters

The ELECTRA model is trained using two encoder-based neural networks, Generator (G) and Discriminator (D). The first one is a Generator G, which maps a sequence of input tokens $\boldsymbol{a} = [a_1, a_2, \dots, a_n]$ into contextualized vector representations $c(x) = [c_1, c_2, ..., c_n]$. The Generator is trained to perform masked language Modeling (MLM). MLM randomly selects random positions (with integer values running from 1 to n) to mask out the original input $m = [m_1, m_2, \dots, m_k]$. When the positions are fixed, the corresponding tokens are replaced by a [MASK] token:

$$a^{masked} = REPLACE(a; m; [MASK]) \tag{1}$$

The masked out tokens are then replaced by generator samples:

$$a^{corrupt} = \text{REPLACE}(a; m; \hat{a}) \tag{2}$$

Where $\hat{a}$ represents a plausible generator sample and is given by

$$\hat{a}_i \sim p_G(a_i | a^{masked}) \; for \; i \in m \tag{3}$$

and model inputs m_i are constructed as:

$$m_i \sim unif\{1, n\} \; for \; i = 1 \; to \; k \tag{4}$$

The probability of generating a token a_t with a softmax layer at the Generator is given by:

$$P_G(a_t|a) = \exp(e(a_t)^T c_G(a)_t) / \sum_{a'} \exp(e(a')^T c_G(a)_t) \tag{5}$$

where t is a given position, e the token embeddings, and all other expressions hold the usual meaning. The Generator is specifically trained to increase the likelihood of masked out tokens and is not supplied with noise, like in adversarial training.

For a given position t, the discriminator D predicts whether the token a_t is corrupted or not, i.e., whether it is from the Generator and not the original data distribution, in specific terms; whether $a^{corrupt}$ matches the original input a.

$$D(a, t) = \text{sigmoid}(w^T c_D(a)_t) \tag{6}$$

The whole process of Masked Language Modeling, Replaced Token Detection, and prediction is illustrated through an example in Fig. 2.

In Fig. 2, the text 'the thief stole a car' is the input text in which some tokens are replaced with [MASK] token. The Masked-out tokens are replaced (corrupted) by Generator samples and provided as input to the Discriminator. The Discriminator then predicts whether the given token is 'fake' (corrupted/replaced) or 'real' (not corrupted/ original). The token 'house' is a corrupted token, the original being 'car'. The ELECTRA model in the example predicts the last token 'house' as 'fake' (corrupted) and the rest as 'real'.

After pre-training, Generator is not used, and the Discriminator is trained on the downstream tasks. In our case, it was pre-trained on a large uncased English text corpus by Google Research. The text corpus is not public and hence not available for any experimenting. The combined loss in the model is minimized as:

$$\min_{\theta_G} \theta_D \; \sum_{a \in X} \; L_{MLM}(a, \theta_G) + \lambda L_{Disc}(a, \theta_D) \tag{7}$$

Where loss function of MLM is given as:

$$L_{MLM}(a, \theta_G) = \mathbb{E}\left(\sum_{i \in m} -\log p_G\left(a_i | a^{masked}\right)\right) \tag{8}$$

and loss function of Discriminator is given as:

$$L_{Disc}(a, \theta_D) = \mathbb{E}\begin{pmatrix} \sum_{t=1}^{n} 1(a_t{}^{corrupt} = a_t) \log D(a^{corrupt}, t) \\ + \\ 1(a_t{}^{corrupt} \neq a_t) \log 1 - D(a^{corrupt}, t) \end{pmatrix} \tag{9}$$

Where the symbols used hold their usual contextual meaning.

The pre-trained model used in our work has an additional dense layer on the top of CLS token and has been initialized by an identity matrix. The fine-tuned hyperparameters for the training of ELECTRA Small are given in Table I.

TABLE I. Fine-tuned Parameters for ELECTRA Small

Hyperparameter	Value
Batch size	32
Learning Rate	1e-4
Adam β_1	0.900
Adam ε	1e-6
Adam β_2	0.999
Learning rate decay	Linear
Layer-wise L.R. decay	0.800
Dropout	0.100
Attention dropout	0.100
Warmup fraction	0.100
Weight Decay	0

C. Proposed Model: ConFaDe - The Fake News Detection System

The proposed model - ConFaDe consists of a transformer-based model for generation of text embeddings and LSTM based deep learning architecture for further classification of news as fake or not. The details of the ConFaDe architecture are shown in Table II and illustrated in Fig. 3.

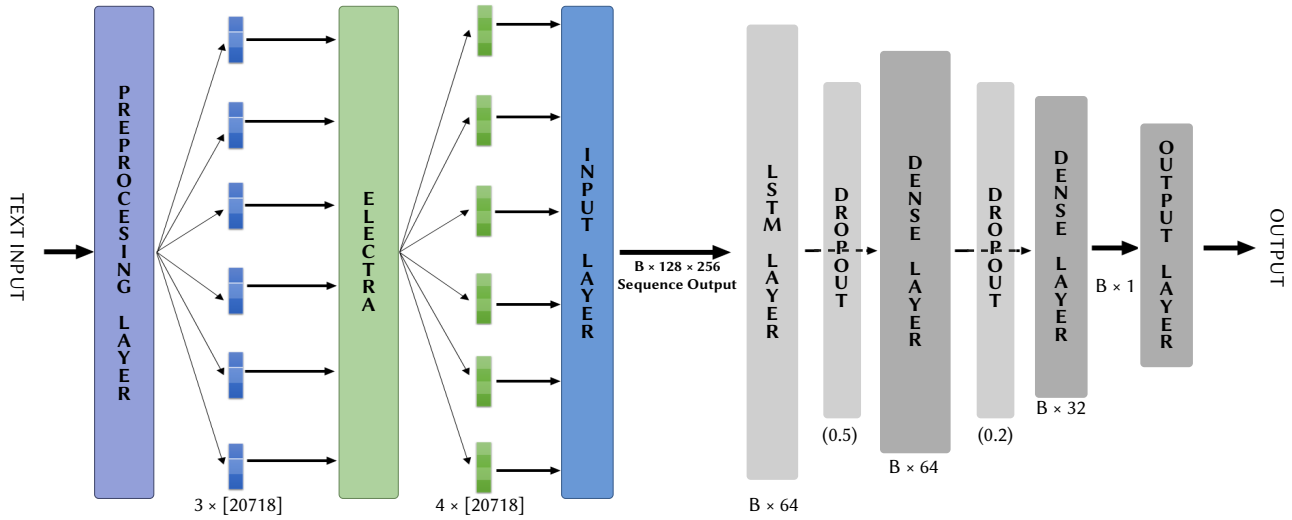

Fig. 3. Layered Architecture of ConFaDe.

TABLE II. ConFaDe Layered Architecture

Layer	Input Dimensions	Output dimensions	Parameter number
Pre-processing	Plain Text 20718 rows	Dict[3]×Tensor[20718]	-
ELECTRA	Dict[3]×Tensor[20718]	Dict[4]×Tensor[20718]	110M
Input	Dict[4]×Tensor[20718]	128×256	-
LSTM	64×128×256	64×64	82176
Dropout	64×32	64×32	0
Dense	64×64	64×64	4160
Dropout	64×32	64×32	0
Dense	64×64	64×32	2080
Dense	64×32	64×1	33

Table II lists different layers with input dimensions, their output dimensions, and the number of parameters. It can be observed that the parameters in different layers of classifier architecture are limited. The batch size in the following experiments is set to 64. For the sake of explanation and visualization, the layered architecture of ConFaDe is illustrated in Fig. 3.

As can be seen from Fig. 3, ConFaDe consists of various layers, the description of which is as follows:

Pre-processing layer: This layer is used to pre-process text. It maps a string Tensor to a dictionary of numeric tensors, which is the required input for the transformer architecture. It performs the basic operations on the text, including some pre-processing. It provides, as an output, a dictionary of numeric values mapped to the text string of the shape of batch size, as defined by the architecture.

ELECTRA layer: This layer accepts a dictionary of tensors as an input and performs training on the tensors for embedding computation. It returns a dictionary of computed outputs on the text. There are four outputs in this layer. The output under key 'Pooled_ output' contains the embedding for each sentence as it appears in the corpus and is two-dimensional. The 'Sequence_output' provides contextualized word-level embedding for each sentence limited to the maximum word length. It is three-dimensional. The output under 'encoder_output' provides the output from each encoder. It is noticeable that the last layer of encoder output is actually the sequence output as it logically should be. Under the key 'default' is a Tensor of shape [batch_size× dimension].

Input layer: This layer is used to prepare input for the subsequent deep learning layer architecture. It gives, as an output, sequence_ output of the previous layer to be processed by the LSTM layer. It performs no other operations on the data.

LSTM layer: LSTM is a kind of recurrent neural network, which has the capability of learning long-term dependencies [39]. The hyperparameters used in the architecture of LSTM are listed in Table IV. With 64 as batch size, it takes 128×256 shaped tensor and provides a tensor with shape as 64 to the next dense layer.

Dense Layer: A layer of neurons connected together, with each neuron receiving input from the previous layer, is called a dense layer. A dense layer is capable of learning representations based on the input. A dense layer, in essence, carries out matrix-vector multiplication and provides an output as the application of activation function to dot product of input (data) and kernel (weight matrix), with the addition of bias. The activation function used in these layers is Rectified Linear Unit (ReLU).

Dropout: The dropout layer is used for the regularization of a neural network, to avoid overfitting. During training, some random neurons are ignored by not activating them in the forward pass and not updating their weights in the backward pass so that every neuron contributes to learning, and only some neurons may not remember the pattern. In ELECTRA pre-training, the dropout is set to 0.1. In dense layers, it is set to 0.5 and 0.2.

Output layer: This layer consists of the neurons, which are fired according to the prediction on the data. This layer consists of a different activation function, 'Sigmoid', as the predictions are not continuous but binary.

Activation functions: This function is a non-linear transformation applied to the inputs from the previous layers to provide output. In our model, we use Sigmoid activation function at the output layer and ReLU as an activation function in other layers.

Loss function: It is a function to calculate the gradients for updating weights in a neural network. We have used binary cross-entropy loss function, which is mathematically calculated as:

$$Loss = -\frac{1}{output\ size}\sum_{i=1}^{output\ size} \alpha_i \,.\log \hat{\alpha}_i + (1-\alpha).\log(1-\hat{\alpha}_i) \quad (10)$$

Where α_i is the target value, $\hat{\alpha}_i$ is the value of i^{th} scalar, and the output size is the number of scalar values in the model output.

Optimizer: This function is used to update model parameters like weights and learning rate, and minimize loss functions to achieve maximum performance in a deep learning algorithm. We have used Adam optimization algorithm, a version of the stochastic gradient descent method based on adaptive estimation of lower order (first and second) moments [71].The parameters used are ε = 1e-08, decay=0.0, beta_1=0.9, learning rate=0.001, and beta_2=0.999.

V. Experimental Setup

We carried our experimentation using DELL PowerEdgeR740 Server P.C. with Intel Xeon Silver 4114 CPU with 20 core(s). We use NVIDIA Quadro P4000 GPU with 1972 CUDA cores, peak single precision of 5.3 TFLOPS, DDR5 memory of 8 G.B., Memory bandwith of 243 Gb/S, Memory Interface of 256 Bits, and a Maximum power consumption of 105W. We trained the model using Tensorflow, and Python 3.8.8, and CUDA version 11.4.

A. Dataset Description

The dataset that has been used for experimentation is openly available and has been used by various researchers in their experiments [36]. It contains a collection of labeled fake news and real news propagated during the U.S. General presidential Election - 2016. The dataset can be downloaded from the internet. It comprises two data files:

(i) train.csv: This file contains training data with the following attributes:
- id: unique id for a news article
- title: the title of a news article
- author: author of the news article
- text: the text of the article
- label: label of the corresponding news article, having two values as:
  - 1: Fake.
  - 0: Real.

(ii) test.csv: This contains testing data with no labels.

After initial data preparation, as earlier explained, the dataset contains 20718 instances, the description of which is listed in Table III.

TABLE III. Dataset Description

Feature	Number of Instances (raw dataset)	Number of instances (Processed dataset)
id	20800	20718
title	20242(excluding missing, including null)	20160
author	18843(excluding nan)	20718
text	20761(excluding missing)	20679
label	20800	20718

The pre-processed dataset consists of 10369 instances with class '1' and 10349 instances of class '0'. The ground truth has been labeled by the contributor of the dataset. The detailed process of collection of dataset is not available. As this dataset has been used extensively in the literature, we also use it in our experiments.

B. Experimental Configurations

We conducted various experiments with different embedding and architectures to present a baseline. We categorize the experiments based on the type of embeddings used. We use different types of encodings and embeddings. We start with one hot encoding, then move on to integer encoding. After that we try GloVe pre-trained embeddings, and at last we use ELECTRA generated encodings. We classify our experiments as transformer based models and non-transformer based models, depending upon the use of transformers in any sub-task. We try different configurations of these models and report the performance of only those models whose performance was best and had comparatively few parameters. The model hyperparameters for these models were set to default and the batch size was set to 64 for all of them. The Training: Validation: Testing split was the same for all of the experiments (as explained in Section IV.A). The evaluation metrics used are detailed in sub-section V.D.

C. Model Hyperparameters

Hyperparameter selection is one of the most important aspects of a deep learning architecture. Optimal hyperparameters are very important for a deep learning framework to perform well, while reducing cost and memory utilization. For manually selecting optimal hyperparameters, knowledge of the problem, domain, and deep learning is required. Table IV provides the optimal hyperparameters used in our LSTM architecture.

TABLE IV. Hyperparameters of LSTM Based Deep Learning Architecture (ConFaDe)

Hyperparameter	Values
Dropout rate	0.5, 0.2
Activation function	ReLU, Sigmoid
Learning rate	0.001
Loss function	Binary Crossentropy
Optimizer	Adam
Batch size	64
No of epochs	13
Recurrent activation	Sigmoid
Recurrent_initializer	orthogonal
Bias	True
Bias_initializer	zeros
Kernel_initializer	Glorot_uniform
Recurrent Dropout	0
Unit_forget_bias	True

D. Evaluation Parameters

To evaluate the performance of a classifier for the task at hand, various performance metrics or evaluation parameters are used. For a classification task, the confusion matrix is an important performance measure. In multi-level classification, it clearly represents the number of classifications or miss-classification an algorithm does by assigning the number of instances that the algorithm thinks to belong to a particular class versus the actual class the instance belongs to, in a tabular format. In binary classification problems, the confusion matrix is an important performance depicter. In the context of fake news detection, the confusion matrix consists of the following values:

True Positive (T.P.): When the algorithm characterizes a news article as fake when it is actually labeled as fake.

False Positive (F.P.): When the algorithm characterizes a news article as fake when it is actually labeled as true.

False Negative (F.N.): When an algorithm characterizes a news article as true when it is actually labeled as fake.

True Negative (T.N.): When the algorithm characterizes a news article as true when it is actually labeled as true.

Apart from these, different metrics are used in evaluating the performance of classifiers [72], which are as below:

- **Accuracy:** It gives a measure of similarity between predicted fake news and actual fake news.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (11)$$

- **Precision:** It measures the objective of the classifier, which here is to detect fake news, and quantifies the fraction of all predicted fake news that is actually labeled as fake news. A value of precision closer to 1 or 100% is best. This measure is often used in conjunction with the Recall, as the precision will automatically be high with few positive (fake news) predictions. A classification model that does not produce any false positives has the maximum value for precision.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (12)$$

- **Recall:** It quantifies the sensitivity measure or the fraction of positive (fake news) articles that are categorized/predicted as fake news. It gives the measure of the degree of correctness of a classifier with respect to predicting only a particular class (positive/fake) and does not take into consideration the false positives. A model with no False negatives will have a maximum value of 1 or 100% for the Recall.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (13)$$

- **F1 Score:** It is also called as F-measure. It is the harmonic mean of precision and Recall, which provides an overall measure of the prediction performance of a classifier in predicting fake news.

$$F-measure = 2\ \frac{precision * recall}{precision + recall} \quad (14)$$

- **F1-Macro:** It is almost the same as F-measure but differs only in how it is calculated. For a binary or multi-class classification, when we tend to take precision, Recall, and F measure of individual classes, for a total performance measure, we use F1-macro. It is defined as the mean of the class-wise F-measure values and gives equal weightage to all classes (fake and real) in the dataset [73].
- **FNR:** Also called as miss rate. It gives a measure of how many fake news articles were misclassified by the classification algorithm.

$$\text{False Negative Rate (FNR)} = \frac{FN}{TP+FN} \quad (15)$$

- **FPR:** Also called as fall-out, it is the proportion of the negative classes (real news) identified as positive classes (fake news).

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP+TN} \quad (16)$$

VI. Results and Discussion

In this paper, we experiment with various text embeddings and deep learning models for fake news detection. We particularly investigate the use of contextualized embeddings for fake news detection with the help of an LSTM based deep learning model. We use ELECTRA for generating text embeddings and name this model as ConFaDe. The ConFaDe model performs better than the existing best models with an accuracy of 99.9 % and consumes less time and resources than the existing state-of-the-art models. We also conducted some experiments with different vector representations and embeddings to draw a comparison. We used DNN based classification model and LSTM based classification model with different embeddings. The confusion matrix for each experiment is given in the corresponding section, and performance on different evaluation measures is given in appropriate section.

A. Non-Transformer Based Models

We conducted several experiments with deep learning models and different word embeddings. We use a simple deep neural network (DNN) and an LSTM based network to estimate the overall efficiency in tandem with different word embeddings. For a better comparison, we start with one-hot encodings at the word level. After that, we use integer encodings with an embedding layer for learning word embeddings.

With one hot encoding, vocabulary built on top-512 words, the classifier performs just fine. This gives an intuition that the news headlines may play an important role in helping to classify fake news. However, this conclusion is not definitive, as the limitation of vocabulary leads to the loss of much information and the vectors are too sparse when built through the strategy. With the accuracy of 72.15% and 74.18 % of the basic DNN and LSTM based network respectively, the models did not achieve outstanding results. The Recall in both cases is low, with DNN having a recall of 59.7 % for fake class and LSTM having a Recall of 66.67%. This points out to the fact that the classifiers missed many of the fake classes. The corresponding confusion matrices are listed in Table V and Table VI.

TABLE V. Confusion Matrix for One Hot- Encoding With DNN

	Predicted Fake	Predicted True
Actual Fake	643	434
Actual True	143	852

TABLE VI. Confusion Matrix for One Hot Encoding With LSTM

	Predicted Fake	Predicted True
Actual Fake	719	358
Actual True	177	818

With integer encoding, maximum length fixed at 512 words, and vocabulary limited to 5000 words, we build DNN and LSTM models. We use an embedding layer with an output dimension of 100 to learn the encoding. The performance of the DNN based model and the LSTM based model with integer encoding is observable from the confusion matrix in Table VII and Table VIII, respectively.

TABLE VII. Confusion Matrix for Integer Encoding With DNN

	Predicted Fake	Predicted True
Actual Fake	1029	48
Actual True	55	940

TABLE VIII. Confusion Matrix for Integer Encoding With LSTM

	Predicted Fake	Predicted True
Actual Fake	1001	76
Actual True	70	925

The DNN based model shows an accuracy of 95.02%, and the LSTM based model shows an accuracy of 92.95%. The F1 Macro, indicative of the overall performance of DNN based model and LSTM based model, is also impressive, with values equal to 95.01% and 92.94%, respectively. It implies that the encoding approach of creating an embedding layer to learn the embeddings is a better method than one-hot encoding in this case.

We also use a pre-trained word embedding- GloVe, with an embedding dimension of 300, to observe the behavior of the classification algorithms. As explained earlier, it uses 6 billion tokens and has a dimension of 300. With classification models using a static embedding matrix initialized with the GloVe embedding, and the

layer set to non-trainable, both DNN and LSTM based models had a comparable accuracy of 95.02%, with their F1 scores being 93.70% and 93.23% respectively. The Confusion matrix for models using Glove embeddings in conjugation with DNN and LSTM are listed in Table IX and Table X, respectively.

TABLE IX. Confusion Matrix for Glove Embedding With DNN

	Predicted Fake	Predicted True
Actual Fake	1025	52
Actual True	78	917

TABLE X. Confusion Matrix for Glove Embedding With LSTM

	Predicted Fake	Predicted True
Actual Fake	1008	69
Actual True	71	924

B. Transformer Based Models

The transformer-based models for generating word representations have shown promise in solving downstream tasks. We use ELECTRA based transformer model to generate word embeddings for the sentences. The word embeddings we generate are contextual in nature and quite powerful for downstream tasks.

a) Simple DNN Models

We train a simple Deep Neural Network (DNN) with two dense layers of size 64 and 32 and an output layer. Fig. 4 contains the loss curve as observed during the training of the DNN based model.

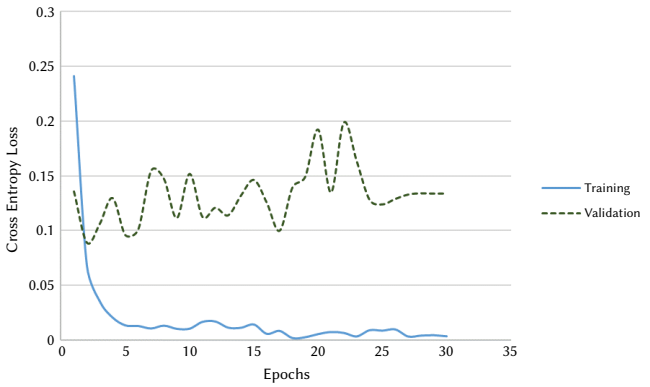

Fig. 4. Loss graph of ELECTRA based DNN based model.

It is pretty clear by observing the loss curve in Fig. 4 that the training loss and validation loss, both are converging, indicative of model learning. As it can be observed that the training loss is almost constant at epoch 30, validation loss also becomes somewhat steady and lowest around epoch 30. We use early stopping criteria for training as no substantial decrease in training loss for five consecutive epochs (patience=5) and minimum delta=0.0001. On the mentioned configuration, the optimized time taken for training the model was about 3 minutes (2.7 minutes) and the whole process takes about 1.15 hours (excluding the time for optimizing the environment). Fig. 5 contains the accuracy plot of ELECTRA based DNN model.

In Fig. 5, we can observe the accuracy curve of training and validation of simple DNN based model. By observing the accuracy plot of the ELECTRA based DNN model, it can be noted that the validation accuracy and the training accuracy tend to converge and achieve a plateau around epoch 30. The graphs are indicative of the point that the model is not overfitting on the data. The Confusion matrix obtained during model testing is given in Table XI.

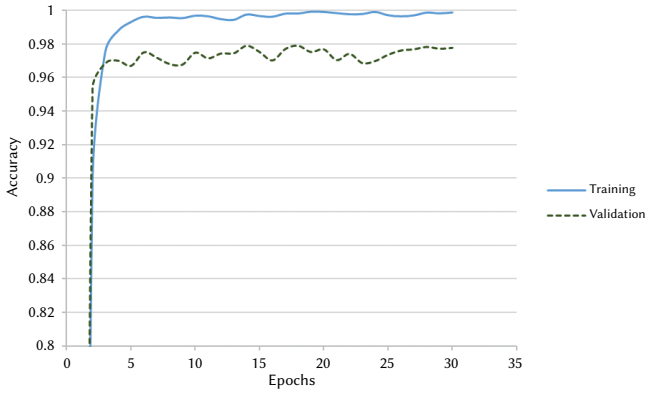

Fig. 5. Accuracy of ELECTRA based DNN based model.

TABLE XI. Confusion Matrix for ELECTRA Based Embedding With DNN

	Predicted Fake	Predicted True
Actual Fake	1055	22
Actual True	25	970

The DNN based model, when used with the ELECTRA-generated word embeddings, had an accuracy of 97.73% and an F1 score of 97.78%.

b) LSTM Based Model (ConFaDe)

On using LSTM based neural network model and trying different configurations, the best configuration yielded an accuracy of 99.9% with an F1 score of 99.9%. On examining the training graph of the ConFaDe, we can see that the validation loss and training loss almost converge near the 13th epoch. This is the optimal point for stopping the model training.The Loss and accuracy curves associated with the model are given in Fig. 6 and Fig. 7, respectively.

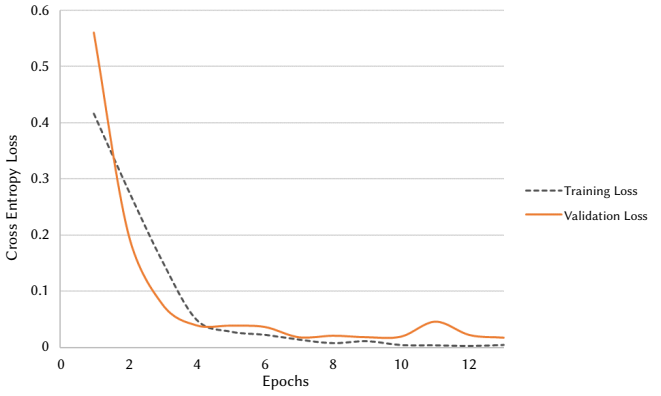

Fig. 6. Loss graph of ConFaDe model.

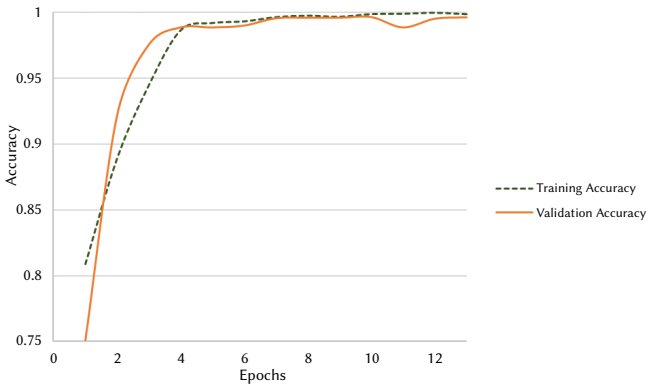

Fig. 7. Accuracy graph of ConFaDe model.

As observable from Fig. 6, the model sees a steady decline in both the training loss and the validation loss, and achieves a plateau around 13 epochs, the stopping criteria being no further substantial decrease in loss for 3 epochs (patience=3, min_delta=0.0001).

From the accuracy plot of training and validation of ConFaDe in Fig. 7, the curve is indicative of model achieving maximum learning at the 13th epoch, as the validation accuracy and training accuracy almost converge. These graphs, when combined together, are also indicative of the fact that the model has not overfit on the data.

For obtaining the actual performance metrics of the model, we run the model on unseen test data, as already explained earlier, to get the results. The confusion matrix obtained by testing the model on the unseen data is given in Table XII.

TABLE XII. Confusion Matrix for ELECTRA Based Embedding With LSTM (ConFaDe)

	Predicted Fake	Predicted True
Actual Fake	1075	2
Actual True	0	995

As is evident from the confusion matrix in Table XII, the total number of False Positives is 0, and that of False Negatives is 2. This is self-explanatory of the performance of the model as it does not create a false alarm and has a very low miss rate.

To evaluate the performance of our model in detail, we have used various performance measures like precision, Recall, and F1 score of individual classes, and F1 macro and Accuracy of different models. Table XIII presents the details of the performance of each model.

It is quite evident from the reported metrics in Table XIII that the model outperforms other model configurations on the same dataset with less loss, more accuracy, Precision, Recall, and F1 score.

To give an idea of how the performances vary with the change in embeddings, we also provide a comparison between the accuracy and F1-macro in different models, using different embeddings in Fig.8 and Fig. 9, respectively.

As we know, LSTM processes the sequence of text singularly to remember context of words. Therefore, it is expected to show some improvement in performance when augmented with a better encoding scheme. DNN just remembers the patterns as a whole and doesn't include any context. A better encoding scheme which captures context in a better way may also increase the performance of DNN. One hot encoded vectors suffer from sparsity problem. Integer encoding performs well because we create our own embedding matrix in it and it is also trained to learn better representations. Nevertheless, it may also require more data to perform better. We also use pre-trained GloVe embeddings for both the models. As we set the embedding static, the performance in both the models seems alike. At the end, we use ELECTRA to generate contextual embeddings which learn context of whole sentences both ways, and at once. The results in the performance of both the models is excellent as compared to other baselines.

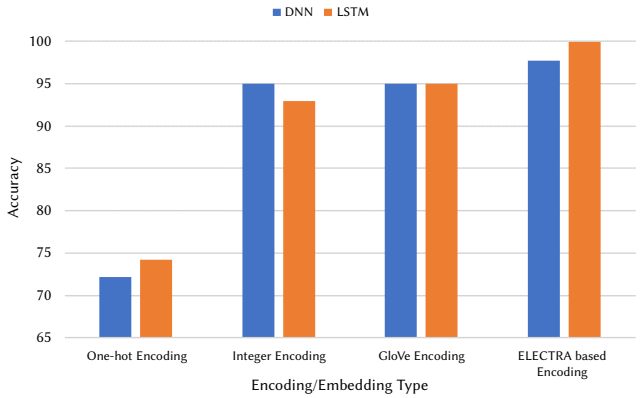

Fig. 8. Comparison of accuracy of different model configurations.

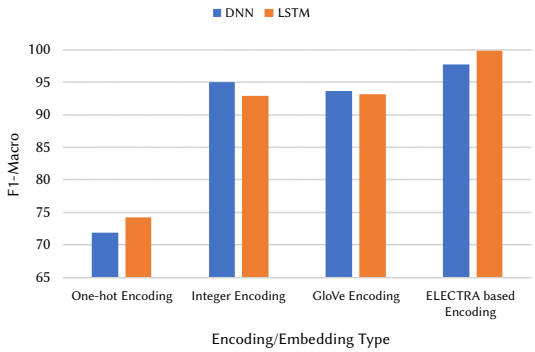

Fig.9. Comparison of F1-Macro of different model configurations.

C. Comparison With Previous Works

We compare our model - ConFaDe with different models that exist in literature. We also make comparison of our model with state-of-the-art (SOTA) model- FakeBERT on different parameters, as listed in Table XIV.

From Table XIV, it is evident that ConFaDe architecture uses a smaller number of parameters than the other transformer-based architecture -FakeBERT. ConFaDe architecture has only 14M encoding parameters, in comparison to 110M encoding parameters of FakeBERT architecture. In addition, ConFaDe has only 88.4K parameters in its classification model whereas FakeBERT has 25.5M parameters in its classification model. In entirety, FakeBERT has 135.5M parameters while ConFaDe only has 14.09M parameters.

On comparing our work with the previous BERT based SOTA model, it can be seen that our model performs better. Fig 10 and Fig 11. provides the comparison of cross entropy loss and accuracy of the two models.

TABLE XIII. Performance Metrics of Various Classifiers and Vector Representations

Vector representation	Classification Model type	Fake Class(1)			Real Class(0)			Accuracy	F1 Macro
		Precision	Recall	F1	Precision	Recall	F1		
One hot encoding	DNN	81.8	59.7	69.03	66.25	85.63	74.70	72.15	71.87
One hot encoding	LSTM	80.25	66.76	72.88	69.56	82.21	75.36	74.18	74.12
Integer Encoding	DNN	94.92	95.54	95.23	95.14	94.47	94.80	95.02	95.01
Integer Encoding	LSTM	93.64	92.94	93.20	92.40	92.96	92.68	92.95	92.94
GloVe	DNN	92.92	95.72	94.04	94.63	92.16	93.38	95.02	93.70
GloVe	LSTM	93.41	93.35	93.50	93.05	92.86	92.95	95.02	93.23
ELECTRA	DNN	97.69	97.96	97.82	97.78	97.49	97.63	97.73	97.78
ELECTRA	**LSTM**	**100**	**99.81**	**99.90**	**99.80**	**100**	**99.89**	**99.90**	**99.90**

TABLE XIV. COMPARISON OF CONFADE WITH FAKEBERT

PARAMETER NAME	FakeBERT	ConFaDe
Number of Transformer blocks/layers	12	12
Encoding Hidden Size	768	256
Encoding model Parameter number	110M	14M
Classification model Parameter number	25.5M	88.4K
Total Parameter number	135.5M	14.09M

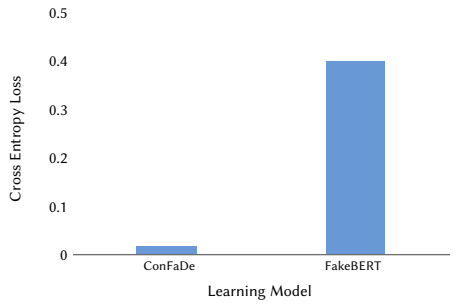

Fig. 10. Loss comparison with FakeBERT.

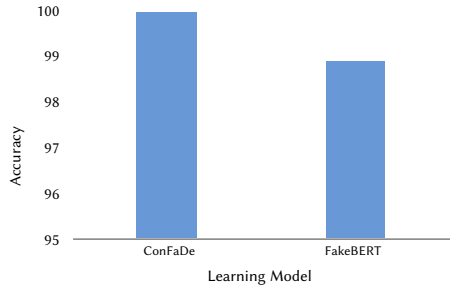

Fig. 11. Accuracy comparison with FakeBERT.

From Fig. 10 and Fig. 11, it can be observed that our model has less cross entropy loss (binary) than FakeBERT based model and its accuracy is higher than SOTA BERT based model – FakeBERT. In Fig. 12 and Fig. 13, we present the comparison with FakeBERT with False Negative Rate (FNR) and False Positive Rate (FPR) as performance measures.

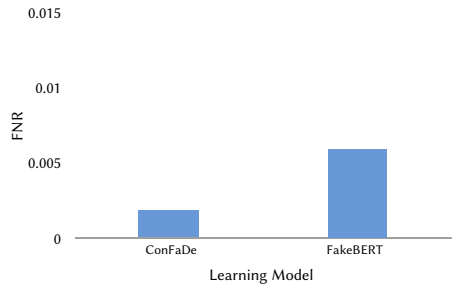

Fig.12. FNR comparison with FakeBERT.

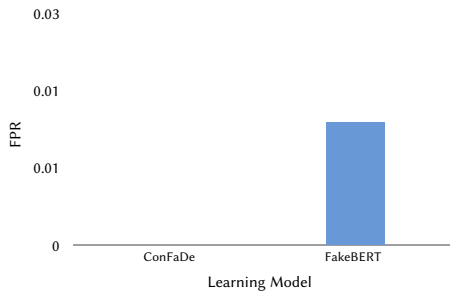

Fig. 13. FPR comparison with FakeBERT.

On comparing both the models on the parameters of FPR and FNR, it can be observed that ConFaDe model has a lower False Negative Rate than FakeBERT and a zero False Positive Rate. It is indicative of the superior performance of the model.

Compared with other existing models that have worked on real-world fake news data, including machine learning models, CNN based models, vanilla LSTM based models, and Hybrid architecture models, the proposed model performs better. The comparison is reported in Table XV.

TABLE XV. COMPARISON WITH EXISTING WORKS ON REAL-WORLD FAKE NEWS DATASETS [36]

Paper	Accuracy Reported (in %)	Technique/Name
(Ghanem et al., 2019)	48.80	SVM,RF,NB, DNN
(Singh et al., 2017)	87.00	SVM
(Ahmed et al., 2017)	89.00	LR- unigram model
(Ruchansky et al., 2017)	89.20	CSI Model
(Ahmed et al., 2017)	92.00	LSVM model
(Liu & Wu, 2018)	92.10	RNN+CNN
(O'Brien et al., 2018)	93.50	Word2Vec+Deep Learning
(Kaliyar et al., 2021)	98.90	FakeBERT(BERT+CNN)
This Paper	**99.90**	**ConFaDe(ELECTRA+LSTM)**

Different models for fake news prediction have been tried in previous works. As is evident from Table XV, machine learning models tend to have low accuracy, owing to the dependence on manual feature engineering. As the deep learning models are used for fake news detection, the accuracy increases sharply. The area of improvement for the detection of fake news in natural language processing condenses to the task of finding an efficient representation of text. Lately, the use of transformers in downstream tasks has been prominently explored. Transformer architectures like BERT are pre-trained on large corpora and afterward used for various downstream tasks, including fake news detection [36].

The central theme of our study is to create efficient as well as accurate architecture for fake news detection. The concerns about the cost and carbon emissions of huge NLP models are reasonably valid, but overlooked in current research. Bender et al. [77] delve into the concerns regarding the environmental and financial costs of NLP models. On training the model once, BERT-base with 110M parameters has an energy consumption of 1507 kWh and carbon emission of 1438 lbs, with a cost of USD 3751-12571 [78]. This study tries to address the problem of fake news through a model that is computationally less expensive, innately simpler, and sustainable in the longer run. The proposed system utilizes a lighter version of ELECTRA, which is about ¼ of the size of BERT, moving a step forward towards Green AI. Though we did not measure the power consumption and carbon emission of the ConFaDe model, it only has 14.9 M parameters [23] (95.1M less parameters than BERT base), and intuitively, its carbon emission and energy consumption will be far less than BERT based models.

The proposed model performs well on a benchmark dataset, outperforming the existing state-of-the-art model in terms of accuracy and efficiency. Although there are many studies related to fake news detection through natural language processing, these studies have not used transformer architecture trained on replaced token detection task to capture context. Previous studies on fake news detection have resulted in creating complex deep learning architectures, adding more layers, and making the process more complex than concentrating on simple solutions. This study focuses on designing simple and efficient system for fake news detection which achieves best results while consuming fewer resources.

Although this study outperforms the existing state-of-the-art model on the same task, there are some limitations. The model being pre-trained on English corpus, is limited to the English language. Fake news detection models need to be developed for multi-lingual fake news where a single news item on internet or social media contains different languages. This study only considers text of the fake news, and as such, multimodal fake news is not covered in this. With respect to the embedding size, fake news detection for long texts is not possible for this model. But, as headlines are an effective cue in detecting the fake news, this model can be applied to the headlines of such lengthy articles. For tackling fake news in different languages like Hindi, Urdu, Arabic, Tamil etc., separate architectures pre-trained for the task are needed.

The technique proposed in the current paper can find its utility in the fake news detection on internet. After creating a deployable model based on the techniques proposed, it can be used to detect fake news on internet portals, microblogs, and social media. Going by the main idea in the proposed technique, small applications for less capable hardware can be developed and deployed. The effective application of the technique proposed may lead to early detection of fake news and thus alleviate the harms caused by the fake news.

VII. Conclusion and Future Scope

In this study, we try to tackle the problem of fake news detection on social media through transformer based contextualized word embeddings. We also conduct experiments with various word embeddings and deep learning models to evaluate the efficiency of each embedding model. We utilize a version of BERT based model-ELECTRA Small++, which differs from original BERT model in the pre-training task and is lighter as far as training and resource consumption is concerned. We generate the word embeddings for LSTM based architecture using the same model. Our word-embedding model is pre-trained on a large English corpus. The results are suggestive of the efficient performance of our model, ConFaDe, over the current state-of-the-art model FakeBERT with the same real-world fake news dataset. We also compare the performance of our model with the FakeBERT model through FPR, FNR and F1 macro and it performs very well on the task with an accuracy of 99.9% and F1 macro of 99.9%.

The model can easily be applied to the English language, as the pre-training task is done on the English corpus. However, for multi-lingual or resource-scarce languages like Urdu and Hindi, we first need to pre-train our transformer model on a corpus of the language. This type of model is best suited for online micro-blogging sites like Twitter and other social media as the sentence size is limited in these platforms.

As a future task, we can incorporate different social and psychological theories and approach the problem with a more data-centric approach towards data-scarce languages to develop a model, which can efficiently tackle the problem of fake news. With a more multi-disciplinary approach, the problem of fake news detection can also be tackled on multiple fronts- from users to network, and the interaction thereof. Another front of work is detecting the fake news in different forms-like picture, text, and video (multimodal). Efficient multimodal fake news detection may be achieved by exploring ensemble of efficient models.

References

[1] X. Zhou and R. Zafarani, "A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities," ACM Computing Surveys, vol. 53, no. 5, pp. 1–40, 2020, https://doi.org/10.1145/3395046.

[2] J. Golbeck et al., "Fake news vs satire: A dataset and analysis," in WebSci 2018 - Proceedings of the 10th ACM Conference on Web Science, pp. 17–21, 2018, https://doi.org/10.1145/3201064.3201100.

[3] X. Zhou, R. Zafarani, K. Shu, and H. Liu, "Fake News: Fundamental Theories, Detection Strategies and Challenges," In *Proceedings of the twelfth ACM international conference on web search and data mining*, pp. 836-837, 2019, https://doi.org/10.1145/3289600.3291382.

[4] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake News Detection on Social Media: A Data Mining Perspective," vol. 19, no.1, pp.22-36, 2017, https://doi.org/10.1145/3137597.3137600.

[5] X. Zhou and R. Zafarani, "Fake News: A Survey of Research, Detection Methods, and Opportunities," ACM Computing Surveys, vol. 53, no.5, pp 1–40, 2018, https://doi.org/10.1145/3395046.

[6] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," Science, vol. 359, no. 6380, pp. 1146–1151, 2018, https://doi.org/10.1126/science.aap9559.

[7] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," Journal of Economic Perspectives, vol. 31, no. 2, pp. 211–236, 2017, https://doi.org/10.1257/jep.31.2.211.

[8] M. H. Alkawaz and S. A. Khan, "Use of Fake News and Social Media by Main Stream News Channels of India," in Proceedings - 2020 6th IEEE International Colloquium on Signal Processing & Its Applications, pp. 93–97, 2020, https://doi.org/ 10.1109/CSPA48992.2020.9068673.

[9] S. Lewandowsky, U. K. H. Ecker, C. M. Seifert, N. Schwarz, and J. Cook, "Misinformation and Its Correction: Continued Influence and Successful Debiasing," Psychological Science in the Public Interest, Supplement, vol. 13, no. 3, pp. 106–131, 2012, https://doi.org/10.1177/15291006124510.

[10] N. Walter and R. Tukachinsky, "A Meta-Analytic Examination of the Continued Influence of Misinformation in the Face of Correction: How Powerful Is It, Why Does It Happen, and How to Stop It?," Communication Research, vol. 47, no. 2, pp. 155–177, 2020, https://doi.org/10.1177/0093650219854600.

[11] Y. Tsfati, H. G. Boomgaarden, J. Strömbäck, R. Vliegenthart, A. Damstra, and E. Lindgren, "Causes and consequences of mainstream media dissemination of fake news: literature review and synthesis,", vol. 44, no. 2, pp. 157–173, 2020, https://doi.org/10.1080/23808985.2020.1759443.

[12] R. Zafarani, X. Zhou, K. Shu, and H. Liu, "Fake news research: Theories, detection strategies, and open problems," in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 3207–3208, 2019, https://doi.org/10.1145/3292500.3332287.

[13] T.A. Shaikh, W.A. Mir, I. Mohammad, and R. Ali, "An Intelligent Healthcare System for Automated Alzheimer's Disease Prediction and Personalized Care", International Journal of Next-Generation Computing, vol. 12, no. 2, pp.240-253, 2021, https://doi.org/10.47164/ijngc.v12i2.196.

[14] K. Shu, S. Wang, and H. Liu, "Beyond news contents: The role of social context for fake news detection," in WSDM 2019 - Proceedings of the 12th ACM International Conference on Web Search and Data Mining, pp. 312–320, 2019, https://doi.org/10.1145/3289600.3290994.

[15] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," in *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55-75, 2018, https://doi.org/10.1109/MCI.2018.2840738.

[16] N. Ruchansky, S. Seo, and Y. Liu, "CSI: A hybrid deep model for fake news detection," in International Conference on Information and Knowledge Management, Proceedings, Nov. 2017, pp. 797–806, 2017, https://doi.org/10.1145/3132847.3132877.

[17] C. Song, K. Shu, and B. Wu, "Temporally evolving graph neural network for fake news detection," Information Processing & Management, vol. 58, no. 6, 2021, https://doi.org/10.1016/j.ipm.2021.102712.

[18] L. Weng, F. Menczer, and Y. Y. Ahn, "Virality prediction and community structure in social networks," Scientific Reports, vol. 3, no. 1, pp. 1–6, 2013, https://doi.org/10.1038/srep02522.

[19] D. Centola, "The spread of behavior in an online social network experiment," Science, vol. 329, no. 5996, pp. 1194–1197, 2010, https://doi.org/10.1126/science.1185231.

[20] R. Oshikawa, J. Qian, and W. Y. Wang, "A survey on natural language processing for fake news detection," in LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings, European Language Resources Association (ELRA), 2020, pp. 6086–6093.

[21] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, pp. 1532–1543, 2014, http://dx.doi.org/10.3115/v1/D14-1162.

[22] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, pp. 4171–4186, 2019, https://doi.org/10.18653/v1/n19-1423.

[23] K. Clark, M.-T. Luong, Q. v. Le, and C. D. Manning, "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators," *arXiv preprint arXiv:2003.10555, 2020*, https://doi.org/10.48550/arXiv.2003.10555.

[24] A. Bondielli, F. Marcelloni "A survey on fake news and rumour detection techniques," Information Sciences, vol. 497, pp. 38–55, 2019, https://doi.org/10.1016/j.ins.2019.05.035.

[25] V. K. Singh, R. Dasgupta, D. Sonagra, K. Raman, and I. Ghosh, "Automated Fake News Detection Using Linguistic Analysis and Machine Learning," In International conference on social computing, behavioral-cultural modeling, & prediction and behavior representation in modeling and simulation (SBP-BRiMS), pp. 1-3, 2017, https://doi.org/10.13140/RG.2.2.16825.67687.

[26] S. P. Yadav, "Emotion recognition model based on facial expressions," Multimedia Tools and Applications, vol. 80, no. 17, pp. 26357–26379, 2021, https://doi.org/10.1007/s11042-021-10962-5.

[27] S. P. Yadav, "Vision-based detection, tracking, and classification of vehicles", IEIE Transactions on Smart Processing and Computing, vol. 9, no. 6, pp. 427–434, 2020, https://doi.org/10.5573/IEIESPC.2020.9.6.427.

[28] V. L. Rubin, Y. Chen, and N. J. Conroy, "Deception Detection for News: Three Types of Fake News," in Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community, pp.1–4, 2015, https://doi.org/10.1002/pra2.2015.145052010083.

[29] C. Castillo, M. Mendoza, and B. Poblete, "Predicting information credibility in time-sensitive social media," Internet Research, vol. 23, no. 5, pp. 560–588, 2013, https://doi.org/10.1108/IntR-05-2012-0095.

[30] H. Ahmed, I. Traore, and S. Saad, "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques," in Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science, vol. 10618, pp. 127–138, 2017, https://doi.org/10.1007/978-3-319-69155-8_9.

[31] A. Giachanou, G. Zhang, and P. Rosso, "Multimodal fake news detection with textual, visual and semantic information," In Proceedings of Text, Speech, and Dialogue: 23rd International Conference, Springer-Verlag, Berlin, Heidelberg, pp. 30–38, 2020, https://doi.org/10.1007/978-3-030-58323-1_3.

[32] S. Singhal, R. R. Shah, T. Chakraborty, P. Kumaraguru and S. Satoh, "SpotFake: A Multimodal Framework for Fake News Detection," 2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM), Singapore, pp. 39-47, 2019, https://doi.org/10.1109/BigMM.2019.00-44.

[33] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space", 1st International Conference on Learning Representations, Workshop Track Proceedings, USA, 2013.

[34] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, 2016, https://doi.org/10.1109/CVPR.2016.90.

[35] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261-2269, 2017, https://doi.org/10.1109/CVPR.2017.243.

[36] R. K. Kaliyar, A. Goswami, and P. Narang, "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach," Multimedia Tools and Applications, vol. 80, no. 8, pp. 11765–11788, 2021, https://doi.org/10.1007.

[37] K. Cho, B.van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1724–1734, 2014, https://doi.org/10.3115/v1/d14-1179.

[38] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber, "LSTM: A Search Space Odyssey," in IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2222-2232, Oct. 2017, https://doi.org/10.1109/TNNLS.2016.2582924.

[39] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997, https://doi.org/10.1162/neco.1997.9.8.1735.

[40] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," in IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, 1997, https://doi.org/10.1109/78.650093.

[41] S. Girgis, E. Amer and M. Gadallah, "Deep Learning Algorithms for Detecting Fake News in Online Text," 2018 13th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 2018, pp. 93-97, https://doi.org/10.1109/ICCES.2018.8639198.

[42] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding In Proceedings of NAACL-HLT, pp. 4171-4186, 2019, https://doi.org/10.18653/v1/n19-1423.

[43] A. Vaswani et al., "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017, pp. 5999–6009.

[44] S. A. Alkhodair, B. C. M. Fung, S. H. H. Ding, W. K. Cheung, and S. C. Huang, "Detecting High-Engaging Breaking News Rumors in Social Media," ACM Transactions on Management Information Systems, vol. 12, no. 1, pp. 1-16, 2021, https://doi.org/10.1145/3416703.

[45] T. Bian et al., "Rumor detection on social media with bi-directional graph convolutional networks," in In Proceedings of the AAAI conference on artificial intelligence vol. 34, no. 01, pp. 549-556, 2020.

[46] J. Shin, L. Jian, K. Driscoll, and F. Bar, "The diffusion of misinformation on social media: Temporal pattern, message, and source," Computers in Human Behavior, vol. 83, pp. 278–287, 2018, https://doi.org/10.1016/j.chb.2018.02.008.

[47] X. Zhou and R. Zafarani, "Network-based Fake News Detection: A Pattern-driven Approach," ACM SIGKDD Explorations Newsletter, vol. 21, no. 2, pp. 48–60, 2019, https://doi.org/10.1145/3373464.3373473.

[48] C. Song, C. Yang, H. Chen, C. Tu, Z. Liu and M. Sun, "CED: Credible Early Detection of Social Media Rumors," in IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 8, pp. 3035-3047, 2021, https://doi.org/10.1109/TKDE.2019.2961675.

[49] H. G. Oliveira, T. Sousa, A. Alves, "Assessing Lexical-Semantic Regularities in Portuguese Word Embeddings", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, no. 5, pp. 34-46, 2021, https://doi.org/10.9781/ijimai.2021.02.006.

[50] S. P. Yadav, S. Zaidi, A. Mishra, and V. Yadav, "Survey on Machine Learning in Speech Emotion Recognition and Vision Systems Using a Recurrent Neural Network (RNN)," Archives of Computational Methods in Engineering, vol. 29, no. 3, pp. 1753–1770, 2021, https://doi.org/10.1007/s11831-021-09647-x.

[51] J. Ma et al., "Detecting rumors from microblogs with recurrent neural networks," In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, pp. 3818–3824, 2016, https://dl.acm.org/doi/10.5555/3061053.3061153.

[52] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in Proceedings of the 31st International Conference on International Conference on Machine Learning, 2014, pp. 1188–1196. https://dl.acm.org/doi/10.5555/3044805.3045025.

[53] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," In Proceedings of International conference on learning representations, 2017.

[54] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," In Proceedings of the 26th International Conference on Neural Information Processing Systems, vol. 2, pp. 3111–3119, 2013.

[55] A. Agarwal, M. Mittal, · Akshat Pathak, · Lalit, and M. Goyal, "Fake News Detection Using a Blend of Neural Networks: An Application of Deep Learning," S.N. Computer Science, vol. 1, no. 3, 2020 https://doi.org/10.1007/s42979-020-00165-4.

[56] G. K. W. Huang and J. C. Lee, "Hyperpartisan News and Articles Detection Using BERT and ELMo," 2019 International Conference on Computer and Drone Applications (IConDA), Kuching, Malaysia, pp. 29-32, 2019, https://doi.org/10.1109/IConDA47345.2019.9034917.

[57] R. M. Silva, R. L. S. Santos, T. A. Almeida, and T. A. S. Pardo, "Towards automatically filtering fake news in Portuguese," Expert Systems with Applications, vol. 146, pp. 113199, 2020, https://doi.org/10.1016/j.eswa.2020.113199.

[58] M. Samadi, M. Mousavian, and S. Momtazi, "Deep contextualized text

representation and learning for fake news detection," Information Processing & Management, vol. 58, no. 6, pp. 102723, 2021, https://doi.org/10.1016/j.ipm.2021.102723.

[59] M. Hoang, O. A. Bihorac, and J. Rouces, "Aspect-Based Sentiment Analysis using BERT," In Proceedings of the 22nd Nordic conference on computational linguistics, pp. 187-196, 2019.

[60] A. Alessa, M. Faezipour and Z. Alhassan, "Text Classification of Flu-Related Tweets Using FastText with Sentiment and Keyword Features," In Proceedings of IEEE International Conference on Healthcare Informatics, pp. 366-367, 2018, https://doi.org/10.1109/ICHI.2018.00058.

[61] B. Büyüköz, A. Hürriyetoğlu, and A. Özgür, "Analyzing ELMo and DistilBERT on Socio-political News Classification," In Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020, pp. 9-18, 2020.

[62] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner and G. Monfardini, "The Graph Neural Network Model," in IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61-80, 2009, https://doi.org/10.1109/TNN.2008.2005605.

[63] M. F. Mridha, A. J. Keya, M. A. Hamid, M. M. Monowar and M. S. Rahman, "A Comprehensive Review on Fake News Detection with Deep Learning," in IEEE Access, vol. 9, pp. 156151-156170, 2021, https://doi.org/10.1109/ACCESS.2021.3129329.

[64] B. Khoo, R. C. W. Phan, and C. H. Lim, "Deepfake attribution: On the source identification of artificially generated images," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 12, no. 3, p. e1438, 2022, https://doi.org/10.1002/widm.1438.

[65] J. F. Low, B. C. M. Fung, F. Iqbal, and S. C. Huang, "Distinguishing between fake news and satire with transformers," Expert Systems with Applications, vol. 187, pp. 115824, 2022, https://doi.org/10.1016/j.eswa.2021.115824.

[66] J. A. Reshi and R. Ali, "Rumor proliferation and detection in Social Media: A Review," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 1156-1160, https://doi.org/10.1109/ICACCS.2019.8728321.

[67] J. Lies, "Marketing Intelligence: Boom or Bust of Service Marketing?", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 7, no. 7, pp. 115-124, 2022, https://doi.org/10.9781/ijimai.2022.10.001.

[68] S. A. Alameri and M. Mohd, "Comparison of Fake News Detection using Machine Learning and Deep Learning Techniques," In Proceedings of 2021 3rd International Cyber Resilience Conference (CRC), pp. 1-6, 2021, https://doi.org/10.1109/CRC50527.2021.9392458.

[69] M. Z. Khan and O. H. Alhazmi, "Study and analysis of unreliable news based on content acquired using ensemble learning (prevalence of fake news on social media)," International Journal of Systems Assurance Engineering and Management, vol. 11, no. 2, pp. 145–153, 2020, https://doi.org/10.1007/s13198-020-01016-4.

[70] V.L. Rubin, "Artificially Intelligent Solutions: Detection, Debunking, and Fact-Checking," In Misinformation and Disinformation, pp. 207-263, 2022, https://doi.org/10.1007/978-3-030-95656-1_7

[71] D. P. Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization," arXiv preprint arXiv:1412.6980, 2014.

[72] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Information Processing and Management: an International Journal, vol. 45, no. 4, pp. 427–437, 2009, https://doi.org/10.1016/j.ipm.2009.03.002.

[73] Z. C. Lipton, C. Elkan, and B. Naryanaswamy, "Optimal thresholding of classifiers to maximize F1 measure," in Lecture Notes in Computer Science, vol. 8725, pp. 225–239, 2014, https://doi.org/10.1007/978-3-662-44851-9_15.

[74] B. Ghanem, P. Rosso, and F. Rangel, "Stance Detection in Fake News A Combined Feature Representation," in Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), pp. 66–71, 2018, https://doi.org/10.18653/v1/W18-5510.

[75] Y. Liu and Y.-F. B. Wu, "Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, 2018, https://doi.org/10.1609/aaai.v32i1.11268

[76] N. O'Brien, S. Latessa, G. Evangelopoulos, and X. Boix, "The Language of Fake News: Opening the Black-Box of Deep Learning Based Detectors," In Proceedings of workshop on "A.I. for Social Good," 32nd Conference on Neural Information Processing Systems, 2018.

[77] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" in Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, pp. 610–623, 2021, https://doi.org/10.1145/3442188.3445922.

[78] E. Strubell, A. Ganesh, and A. McCallum, "Energy and Policy Considerations for Deep Learning in NLP," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3645–3650, 2019, https://doi.org/10.18653/v1/P19-1355.

Junaid Ali Reshi

Junaid Ali Reshi received his B.Tech in Computer Science and Engineering (2015) from University of Kashmir and M. Tech in Computer Science and Technology (2017) from Central University of Punjab, India. He is currently a doctoral research scholar in the department of Computer Engineering, Aligarh Muslim University, India. He has served as reviewer for many papers in WoS indexed journals and top conferences, and is actively involved in research. His areas of interest are Social Network analysis, Natural Language processing, and Computational Social science.

Rashid Ali

Prof. Rashid Ali obtained his B.Tech. and M.Tech. from Aligarh Muslim University, India in 1999 and 2001 respectively. He obtained his PhD in Computer Engineering in February 2010 from Aligarh Muslim University, India. He is currently serving as full Professor in the department of Computer Engineering, Aligarh Muslim University. He is also serving as the Coordinator, Interdisciplinary Center for Artificial Intelligence, Aligarh Muslim University. Apart from being a member of various international societies, he is also a senior member of IEEE. He has authored more than 125 papers in various International Journals and conferences of repute. He has also chaired sessions at some International conferences. He reviews articles for some of the reputed International Journals and conferences. He has supervised more than 25 M.Tech Dissertation and 6 PhD Thesis. His research interests include Web Searching, Web Mining, Soft computing Techniques (Rough-Set, Artificial Neural Networks, fuzzy logic etc.), Recommender Systems, and Online Social Network Analysis.