

Fake News Detection Using BERT with Integration for Android

Sapna I. Narwade

Computer Science and Engineering
Deogiri Institute of Engineering and Management Studies
Chhatrapati Sambhajnagar, Maharashtra, India

Pranali Akhil Bhalekar

Computer Science and Engineering
Deogiri Institute of Engineering and Management Studies
Chhatrapati Sambhajnagar, Maharashtra, India

Abstract - The exponential growth of social media platforms has made the dissemination of misinformation a major concern. This study presents a machine learning-based approach for fake news detection using a fine-tuned BERT (Bidirectional Encoder Representations from Transformers) model. My work focuses on optimizing BERT alone for textual fake news detection and deploying the model in an Android environment using Tensor Flow Lite (TFLite). The model achieves high accuracy in real-time inference, making it practical for mobile deployment. Experimental results show that BERT performs robustly on fake news datasets, providing a reliable classification of news articles as real or fake.

Keywords - Fake news detection, BERT, Machine learning, Android app, Tensor Flow Lite, Text classification, Android Studio.

INTRODUCTION

With the surge in digital information sharing, fake news has emerged as a serious societal threat. Traditional machine learning approaches, including SVM, Naive Bayes, and Logistic Regression, struggle to capture contextual semantics in news text. Recent advancements in Natural Language Processing (NLP) through transformer-based models like BERT offer a powerful solution for understanding contextual relationships. This study focuses on fine-tuning a BERT model for text-based fake news detection and converting it into TFLite format for integration into a mobile Android application.

Social media platforms have reached exceptional success, unlocked new possibilities, and transformed the way information is created, shared, and consumed. As a result, they have become essential tools used for numerous purposes. The rapid enhancement of social media features across different platforms has encouraged even well-established news organizations and agencies to shift toward these networks.

However, the rising use of social media has also led to a significant surge in fake news and online misinformation. Fake news refers to fabricated content that resembles legitimate news but differs in purpose or the process behind its creation, with the intention of misleading readers [1]. Such misinformation spreads rapidly across social media, blogs, magazines, forums, and online newspapers, making it increasingly challenging to identify credible news sources.

The simplicity of generating and distributing information has turned social media into a perfect environment for creating, altering, and spreading false news. For example, Facebook reports that harmful actors were responsible for less than one-tenth of 1% of civic content posted on its platform [2].

In recent years, fake news has been linked to increased political polarization and conflicts between parties. It also affects topics such as vaccination, nutrition, and stock market prices. A study by researchers from Ohio State University [3] suggests that misinformation may have contributed to a decline in Hillary Clinton's support during the election. The study indicates that nearly 4% of Barack Obama's 2012 supporters chose not to vote for Clinton in 2016 due to believing in fake articles. Another example [4] shows how a false rumor about Tesla acquiring a lithium mining company caused its stock value to rise by almost 250%. According to [5], many tweets related to the COVID-19 pandemic contained false or unverified claims—24.8% and 17.4% respectively. Clearly, misinformation has a significant impact—either positive or negative—on public perception. Therefore, creating effective tools to analyze online content is critical to avoid harmful consequences for society, the economy, and politics.

Manual fact-checking relies on ongoing updates by crowd sourced contributors or small groups of experts and cannot perform automated learning [6]. Machine learning and deep learning methods have shown strong predictive capabilities and the ability to address complex issues [7–9]. Developing automatic, reliable, and accurate fake-news detection systems has become an important research area. Detecting fake news is a difficult NLP (natural language processing) task involving text classification to distinguish between real and false information. NLP has made tremendous progress in recent years, with transformer-based pre-trained language models becoming state-of-the-art solutions for many NLP tasks [10–12]. Nonetheless, research on fake-news detection using transformer-based approaches remains limited.

In this work, Bidirectional Encoder Representations from Transformers (BERT) is used to process news articles and generate meaningful text representations. BERT is a powerful language-representation model known for delivering strong

results in various NLP applications. The proposed approach is tested on three fake-news datasets. Its performance is compared with multinomial Naive Bayes (MNB), linear regression (LR), linear support vector machines (LSVM), and long short-term memory (LSTM) models using different word-embedding techniques. The results show that the proposed method significantly outperforms existing techniques. The main contributions of this as follows:

- An automated fake-news detection framework is introduced for analyzing both news titles and full article content using a BERT-based model.
- The BERT model is employed to extract deep contextual representations from input text and classify the BERT-based embeddings as real or fake.
- The proposed method is tested on two fake-news datasets and compared with traditional machine-learning and deep-learning approaches.

RELATED WORK

A wide range of machine-learning methods have been developed to identify fake news. These approaches generally fall into two groups: traditional machine-learning methods and deep-learning methods.

Traditional approaches include models such as multinomial Naive Bayes (MNB), linear regression (LR), linear SVM (LSVM), decision trees (DT), and extreme gradient boosting (XGBoost). In [13], Ahmed et al. employed n-gram features with TF-IDF to detect fake news and compared the performance of six machine-learning algorithms. Their study found that LSVM achieved the highest accuracy of 92% on the ISOT dataset, although its performance on other datasets remains uncertain. Likewise, the author in [14] evaluated five machine-learning techniques using the same embedding method, where LSVM and XGBoost produced the best outcomes. In [15], Ozbay and Alatas used term frequency (TF) weighting and a document-term matrix to extract textual features, and they tested 23 supervised models for identifying fake news. DT performed best in their analysis. In another study [16], the same authors applied the salp swarm optimization (SSO) and grey wolf optimizer (GWO) algorithms instead of standard machine-learning models for fake-news detection. Kansal [17] examined writing styles using part-of-speech (POS) tags, feeding these features into XGBoost to build the first model. They then averaged TF-IDF and Word2Vec embeddings, passed them through a multilayer perceptron, and combined the results with the first model to generate final predictions. Nonetheless, traditional approaches often require large labeled datasets and complex feature-engineering techniques, and they may struggle to adapt to emerging forms of misinformation.

Deep learning has become a major focus area in machine learning and artificial intelligence due to its strong ability to learn patterns from data. Models such as CNNs and LSTM networks have grown increasingly popular. In [18], Nasir et al. proposed a hybrid CNN-LSTM framework, where CNN extracts local features and LSTM models long-term dependencies. They represented words using GloVe pre-trained embeddings, achieving better results than seven traditional machine-learning models. In [19], the authors introduced

FNDNet, a deep-learning architecture that uses GloVe embeddings followed by multiple convolution-pooling layers and dense layers for classification. In [20], Sastrawan et al. applied back-translation for data augmentation to handle class imbalance, used various pre-trained embeddings (Word2Vec, GloVe, fastText), and tested CNN, bidirectional LSTM, and ResNet to extract semantic features and detect fake news. Another deep-learning model, OPCNN-FAKE, was proposed in [21], consisting of an embedding layer, dropout layer, convolution-pooling layer, flattening layer, and output layer. Its performance was compared with RNN, LSTM, and six traditional machine-learning classifiers. In [22], Yang et al. developed a CNN-based model (TI-CNN) that integrates explicit and latent textual and visual features for fake-news detection, and they benchmarked it against LSTM, CNN, and GRU models. Similarly, a multimodal CNN architecture combining text and images was introduced in [23]. However, many of these approaches struggle to capture long-range contextual information, and static word embeddings often fail to represent context-specific meaning.

Recently, the transformer architecture has become a major breakthrough in NLP. In [24], the authors evaluated five transformer models—XLNet, BERT, RoBERTa, DistilBERT, and ALBERT—for fake-news detection across multiple hyperparameter settings and found that their performances were comparable. In [25], Kaliyar et al. introduced FakeBERT, which uses BERT embeddings followed by parallel convolutional layers with different kernel sizes. This model achieved better accuracy than traditional machine-learning approaches. In [26], researchers proposed a transformer-based system utilizing BART and RoBERTa embeddings, which were fed into LSTM- and CNN-based branches, then concatenated and passed through additional LSTM/CNN layers to produce final predictions. Qazi et al. [27] used an attention-based transformer model for classifying fake and real news, comparing it with a hybrid CNN that uses both text and metadata. The transformer model achieved superior accuracy. Despite their effectiveness, many transformer-based methods are computationally expensive and require extensive training resources. In contrast, the proposed approach leverages BERT, which is faster and demands fewer computational resources.

PROPOSED METHOD

Fake news has become a major concern in modern society, as it can deceive people and negatively affect individuals, businesses, and even whole countries. Existing approaches to fake news detection often rely on manual factchecking or rule-based systems, which can be time-consuming and limited in their coverage. In this paper, I propose a novel method based on fine-tuned BERT to improve the accuracy and efficiency of fake news detection. My method leverages the power of BERT to capture complex linguistic patterns and the efficiency to optimize feature space and classification. By combining these two techniques, my aim to achieve better performance compared to other methods.

Figure 1 shows an overall view of the proposed system. I apply some pre-processing steps on the input text to get rid of unnecessary parts of the data. The input text is then tokenized to individual characters, sub words, and words that are good

enough to represent the input data to be fed to “fine- tuned” BERT. I extract the text embedding from the special token [CLS] of the last three hidden layers of the BERT. I train the model on the concatenated embedding vectors to get the final prediction.

Methodology

Dataset Acquisition and Preprocessing

I used open-source fake news datasets that include labeled text samples. Preprocessing involved cleaning text, removing stop words, and applying tokenization using the BERT tokenizer. The dataset was split into training and testing subsets (80:20 ratio).

Model Architecture

I utilized the pre-trained 'bert-base-uncased' model from Hugging Face Transformers. The model was fine-tuned for binary classification (real vs. fake) using a softmax output layer.

Formula for embedding extraction:

$$E = \text{BERT}(\text{Text Input})$$

$$P = \text{Softmax}(W * E[\text{CLS}] + b)$$

where $E[\text{CLS}]$ represents the embedding of the [CLS] token, W and b are trainable parameters.

Training Configuration

The model was trained using the AdamW optimizer, cross-entropy loss function, and a learning rate of $2e-5$ for 3 epochs. To reduce the risk of overfitting, techniques such as early stopping and dropout regularization were applied.

Model Conversion

After training, the model was exported to TensorFlow and converted into TensorFlow Lite (.tflite) format to enable deployment on Android devices.

Android Integration

The Android application integrates the TFLite model to classify news entered by the user in real time. The app features a simple interface allowing users to paste or type a news headline or paragraph, and it returns the prediction with a confidence score.

Experimental Results

The fine-tuned BERT model achieved an accuracy of approximately 95% on the test dataset. The confusion matrix and F1-score confirmed the model's strong performance in distinguishing fake and real news.

Metric	Value
Accuracy	95.2%
Precision	94.8%

Recall	95.5%
F1-score	95.1%

Flowchart depicting: Dataset Collection → Preprocessing → BERT Training → Model Evaluation → Conversion to TFLite → Android Integration

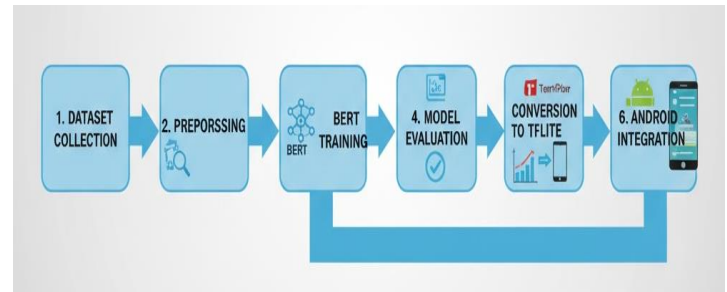


Figure 2.flowchart

Pre-processing

In this paper, I apply several pre-processing steps to clean the input data and reduce noise. All non-alphabet characters, tags, and URLs are filtered out from the text, since they may not provide much importance to understanding the text. Numbers are deleted as they represent quantified arguments in the news context and do not generally alter the meaning of the text. Stop words (e.g., “the”, “a”, and “is”, etc.) and punctuation (e.g., “!”, “?”, “-”, etc.) are removed, because they are more frequent and provide less helpful information. Case folding is applied to reduce all letters to lowercase. Finally, I exclude the news record from any analysis, if the number of words of its full text is less than ten words.



Figure 1Pre-Processing of data

BERT

Incorporating a pre-trained language model attracted much attention to many NLP tasks such as paraphrasing [28], natural language inference [29, 30], named entity recognition [31] and question answering [32, 33]. BERT [10] is a pre- trained

bidirectional language model based on a transformer that produces language representations by combining both left and right contexts. It analyzes input text bidirectionally from left to right and right to left. BERT is a contextual model that considers the word position in a sentence to compute the representation of each word, unlike other word embedding techniques such as Word2Vec and GloVe which is a context-free model that produces the same word representation regardless of the word position in a sentence.

BERT is using the “masked language model” (MLM) pre-training objective to achieve bidirectional representations. The MLM is randomly masking 15% of the input tokens and the task is to retrieve the original token of the masked word. Moreover, the next sentence prediction task is also utilized to train the BERT model to capture the relationship between sentences by predicting the sentence that comes after the current sentence. Two large corpora of unlabeled text: BooksCorpus (800 million words) and English Wikipedia corpus (about 2.5 billion words) are used for the pre-training of the BERT model. Before processing the input to BERT, the input sentences are tokenized to individual characters, sub-words, and words that are good enough to represent the input data. BERT is using a fixed size of 30,000 token vocabularies. Special tokens [CLS] and [SEP] are added at the beginning and end of each sentence, respectively. WordPiece embeddings are a tokenization algorithm used by BERT. It works by first checking the whole word if it is in the vocabulary list,

it returns the corresponding word embedding vector. If not, it breaks up the word into the best-fit sub words and eventually individual characters that are in the vocabulary list. So that the original word representation will be as an average of all sub word embedding vectors. The output representation for a given token is computed by summing the position and segment embeddings to the token embedding.

BERT is a transformer-based model [34] which is different from RNN. A transformer is a deep-learning model that uses the attention mechanism to process all the input sequences at once without the need for RNNs. The attention mechanism examines the relationship between words, regardless of the places of words in a sentence. RNNs process input in a sequential manner, word by word, so it is difficult to parallelize the computation. This makes the training of RNNs inefficient when dealing with long sequences.

The original transformer model [34] is used in machine translation. It has two parts an encoder and a decoder. The encoder is producing an embedding representation for each word depending on their relationship to the other words in the sentence. The decoder takes the output embeddings of the encoder and turns them back into output text. BERT makes advantage of the transformer’s encoder only, since its purpose is to develop a model that can work well on a variety of NLP

tasks. Using the encoder part, the BERT is able to encode semantic and syntactic information in the embedding, which is required for a variety of jobs.

The base architecture of BERT adapted in this paper consists of a number of encoder layers ($L = 12$) and each layer has a hidden size (H) of 768 units, and the number of self-attention heads ($h = 12$). The total number of parameters is 110M. The input embeddings are passed through multiple layers of self-attention and feed forward networks called Transformer Block. The output of each layer is denoted by H_l , where l is the layer index\

$$H_l = \text{Transformer Block}(H_{l-1}). \quad (1)$$

The fundamental operation of the *Transformer Block* is to compute multi-head attention as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^O, \quad (2)$$

$$\text{where } \text{head}_i = \text{Attention}(QW^Q, KW^K, VW^V), \quad (3)$$

where Q, K , and V are the query, key, and value matrices, respectively. W^Q, W^K , and W^V are learnable weight matrices

for the i -th head. W^O is a learnable weight matrix to map the concatenated attention outputs back to the model

dimension. The output of multi-head attention is then fed through a feedforward network layer, which applies a nonlinear

transformation to the attention output. The residual connection and layer normalization are used to stabilize the training process and improve the overall performance of the network.

BERT fine-tuning

One of BERT’s most important capabilities is its support for transfer learning. A pre-trained BERT model—initially trained on large, general-purpose datasets—can be adapted to a specific downstream task through fine-tuning. For fake-news detection, fine-tuning adjusts BERT’s parameters so the model learns patterns relevant to distinguishing real from false content.

BERT encodes the complete input sequence into a vector representation derived from the final hidden state H_{CLS} of the first token, [CLS]. During fine-tuning, a fully connected layer followed by a softmax classifier is added to convert H_{CLS} into two output categories: *real* or *fake*. This fully connected layer has dimensions [768,2]. The model then computes the probability of each class using the softmax function:

$$p(C | H_{CLS}) = \text{softmax}(WH_{CLS}) \quad (4)$$

Here, W is the weight matrix of the added classification layer.

Both the BERT parameters and the matrix W are optimized

together by minimizing the negative log-likelihood of the true class labels.

Sentence Representations

After fine-tuning is complete, the next step is to generate fixed feature representations for each input sentence, which will be used to build the fake-news detection model. Several strategies exist for extracting embeddings from BERT, with most relying on the [CLS] token's hidden state from the final encoder layers. In this study, the [CLS] embeddings from the last **three** layers of BERT are concatenated to form a single feature vector:

$$E = \text{CONCAT}(H_{CLS}^L, H_{CLS}^{L-1}, H_{CLS}^{L-2})(5)$$

Gradient Boosting Decision Tree (GBDT)

GBDT is an ensemble technique that builds multiple weak learners (typically decision trees) one after another, where each tree attempts to correct the errors—represented as residuals or negative gradients—made by the previous trees. The model can be expressed as:

$$f(x) = \sum_{m=1}^M \gamma_m D(x; \theta_m)(6)$$

Here,

- M = total number of trees,
- γ_m = learning rate,
- $D(x; \theta_m)$ = the m -th decision tree with parameters θ_m .

Each tree is trained by minimizing a loss function L :

$$\theta_m = \arg \min_{\theta} \sum_{i=1}^N L(y_i, f_{m-1}(x_i) + \gamma_m D(x_i; \theta))(7)$$

where y_i is the true label, f_{m-1} is the model's previous prediction, and N is the number of samples. Gradient descent is typically used to optimize tree parameters.

Traditional GBDT models are powerful but computationally heavy—especially with large datasets—because they evaluate all data points to find the best feature split.

LightGBM

LightGBM is an optimized and highly efficient implementation of GBDT designed to reduce training time and memory usage while retaining strong accuracy. It speeds up computation by using a histogram-based algorithm, where continuous feature values are bucketed into discrete bins. This greatly reduces the cost of evaluating split points.

Additionally, LightGBM grows trees leaf-wise, focusing on expanding the leaf that produces the largest loss reduction. Although this may produce deeper, less balanced trees, it often results in stronger performance and faster training than level-wise methods like those used in XGBoost.

LightGBM also includes two major optimizations:

1. **Gradient-based One-Side Sampling (GOSS)**
 Instances with large gradients have greater influence on split decisions.
 - All high-gradient samples (top $a\%$) are kept.
 - A random sample ($b\%$) of small-gradient points is selected. This lowers the number of training samples without sacrificing model accuracy.
2. **Exclusive Feature Bundling (EFB)**
 Many high-dimensional datasets have sparse features that never take nonzero values at the same time. EFB groups such mutually exclusive features into a single feature, reducing dimensionality and improving training efficiency without harming performance.

Algorithm 1: BERT Approach for Fake News Detection

Input: Labeled fake-news dataset and a pre-trained BERT model

Output: Predicted label — *real* or *fake*

Steps:

1. Load the fake-news dataset and apply pre-processing to clean and format the text.
2. Load the pre-trained BERT model along with its tokenizer.
3. Tokenize all input text samples.
4. Fine-tune the BERT model on the training data using Equation (4).
5. Extract contextual embeddings using Equation (5).
6. Train the BERT classifier using the generated embeddings.
7. Use the trained model to make predictions on the test dataset.
8. Convert the trained model to TFLite format for deployment in an Android application.

Evaluation Metrics

Five different evaluation metrics were used to assess model performance: **Accuracy (Acc %)**, **Precision (Pre %)**, **Recall (Rec %)**, **F1-Score (F1 %)**, and **Area Under the Curve (AUC)**.

- **Accuracy** measures the proportion of correctly predicted instances out of all predictions:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (8)$$
- **Precision** represents the proportion of correctly identified positive samples among all positive predictions:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (9)$$
- **Recall** (or sensitivity) indicates the percentage of true positives correctly identified by the model:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (10)$$
- The F1-score represents the harmonic mean of precision and recall, offering a balanced evaluation of both metrics.

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

CONCLUSION

This study introduces a novel model designed for the automatic detection of fake news. The proposed system integrates BERT for contextual word embeddings for efficient and accurate classification. This research successfully demonstrates an efficient fake news detection system using BERT integrated within an Android application. By leveraging deep contextual representations, the system effectively classifies textual news articles as real or fake in real time. Future work includes extending the model to handle multimodal inputs (text + images) and supporting multilingual fake news detection for broader usability.

REFERENCES

- [1]. Lazer DMJ, Baum MA, Benkler Y, Berinsky AJ, Greenhill KM, Menczer F, Metzger MJ, Nyhan B, Pennycook G, Rothschild D, Schudson M, Sloman SA, Sunstein CR, Thorson EA, Watts DJ, Zittrain JL (2018) The science of fake news. *Science* 359(6380):1094–1096.
- [2]. Weedon J, Nuland W, Stamos A (2017) Information operations and Facebook. Retrieved from Facebook: Accessed 15 Aug 2022
- [3]. Gunther R, Beck PA, Nisbet EC (2018) Fake news may have contributed to trump's 2016 victory. Ohio state university, [Online]. Accessed 24 Aug 2022
- [4]. The Economic Times (2022) Fake news of Tesla acquiring lithium miner sent its stock up over 250%. [Online]. Accessed 24 Aug 2022
- [5]. Kouzy R, Abi Jaoude J, Kraitem A, El Alam MB, Karam B, Adib E, Zarka J, Traboulsi C, Akl EW, Baddour K (2020) Coronavirus goes viral: quantifying the covid-19 misinformation epidemic on twitter. *Cureus* 12(3):e7255
- [6]. Seddari N, Derhab A, Belaoued M, Halboob W, Al-Muhtadi J, Bouras A (2022) A hybrid linguistic and knowledge-based analysis approach for fake news detection on social media. *IEEE Access* 10:62097–62109
- [7]. Essa E, Xie X, Jones J-L (2015) Minimum s-excess graph for segmenting and tracking multiple borders with hmm. In: International conference on medical image computing and computer-assisted intervention, pp 28–35
- [8]. Essa E, Xie X (2018) Phase contrast cell detection using multi-level classification. *Int J Numer Methods Biomed Eng* 34(2):2916.
- [9]. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaria J, Fadhel MA, Al-Amidie M, Farhan L (2021) Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *J Big Data* 8(1):1–74
- [10]. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol 1 (long and short papers), pp 4171–4186.
- [11]. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV (2019) Xlnet: generalized autoregressive pretraining for language understanding. In: Advances in neural information processing systems, vol 32, pp 5753–5763
- [12]. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: a robustly optimized bert pretraining approach. *arXiv preprint*.
- [13]. Ahmed H, Traore I, Saad S (2017) Detection of online fake news using n-gram analysis and machine learning techniques. In: International conference on intelligent, secure, and dependable systems in distributed and cloud environments, pp 127–138
- [14]. Wijeratne Y (2021) How much bullshit do we need? benchmarking classical machine learning for fake news classification. *LIRNEasia*, [Online]. Accessed 24 Aug 2022
- [15]. Ozbay FA, Alatas B (2020) Fake news detection within online social media using supervised artificial intelligence algorithms. *Phys A* 540:123174.
- [16]. Ozbay FA, Alatas B (2021) Adaptive salp swarm optimization algorithms with inertia weights for novel fake news detection model in online social media. *Multimed Tools Appl* 80(26):34333–34357
- [17]. Kansal A (2021) Fake news detection using pos tagging and machine learning. *J Appl Secur Res*.
- [18]. Nasir JA, Khan OS, Varlamis I (2021) Fake news detection: a hybrid cnn-rnn based deep learning approach. *Int J Inf Manag Data Insights* 1(1):100007.
- [19]. Kaliyar RK, Goswami A, Narang P, Sinha S (2020) Fndnet—a deep convolutional neural network for fake news detection. *Cogn Syst Res* 61:32–44.
- [20]. Sastrawan IK, Bayupati IPA, Arsa DMS (2022) Detection of fake news using deep learning cnn-rnn based methods. *ICT Express* 8(3):396–408.
- [21]. Saleh H, Alharbi A, Alsamhi SH (2021) Openn-fake: optimized convolutional neural network for fake news detection. *IEEE Access* 9:129471–129489. 2021
- [22]. Yang Y, Zheng L, Zhang J, Cui Q, Li Z, Yu PS (2018) TI-CNN: convolutional neural networks for fake news detection. *arXiv preprint*.
- [23]. Raj C, Meel P (2021) Convnet frameworks for multi-modal fake news detection. *Appl Intell* 51(11):8132–8148
- [24]. Schütz M, Schindler A, Siegel M, Nazemi K (2021) Automatic fake news detection with pre-trained transformer models. In: International conference on pattern recognition, pp 627–641
- [25]. Kaliyar RK, Goswami A, Narang P (2021) Fakebert: fake news detection in social media with a Bert-based deep learning approach. *Multimed Tools Appl* 80(8):11765–11788
- [26]. Truica C-O, Apostol E-S (2022) Misrobaerta: transformers versus misinformation. *Mathematics* 10(4):569
- [27]. Qazi M, Khan MUS, Ali M (2020) Detection of fake news using transformer model. In: 2020 3rd international conference on computing, mathematics and engineering technologies (iCoMET), pp 1–6.
- [28]. Arase Y, Tsujii J (2021) Transfer fine-tuning of BERT with phrasal paraphrases. *Comput Speech Lang* 66:101164
- [29]. Williams A, Nangia N, Bowman SR (2017) A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint*.
- [30]. Storks S, Gao Q, Chai JY (2019) Recent advances in natural language inference: a survey of benchmarks, resources, and approaches. *arXiv preprint*.
- [31]. Souza F, Nogueira R, Lotufo R (2019) Portuguese named entity recognition using bert-crf. *arXiv preprint*.
- [32]. Yang W, Xie Y, Lin A, Li X, Tan L, Xiong K, Li M, Lin J (2019) End-to-end open-domain question answering with bertserini. *arXiv preprint*.
- [33]. Yang Z, Garcia N, Chu C, Otani M, Nakashima Y, Takemura H (2020) Bert representations for video question answering. In: Proceedings of

- the IEEE/CVF Winter conference on applications of computer vision (WACV)
- [34]. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, vol 30
- [35]. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. *Ann Stat* 29(5):1189–1232
- [36]. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu T-Y (2017) Lightgbm: a highly efficient gradient boosting decision tree. In: Advances in neural information processing systems, vol 30
- [37]. Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd Acm Sigkdd international conference on knowledge discovery and data mining, pp 785–794
- [38]. Szpakowski M (2020) Fake News Corpus. [Online]. Accessed 24 Aug 2022
- [39]. Ramos J (2003) Using tf-idf to determine word relevance in document queries. In: Proceedings of the first instructional conference on machine learning, vol 242, pp 29–48
- [40]. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
- [41]. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. *J Mach Learn Res* 12:2825–2830
- [42]. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M (2020) Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, pp 38–45
- [43]. Smith LN (2018) A disciplined approach to neural network hyperparameters: part 1—learning rate, batch size, momentum, and weight decay. arXiv preprint.
- [44]. Akiba T, Sano S, Yanase T, Ohta T, Koyama M (2019) Optuna: a next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International conference on knowledge discovery and data mining, pp 2623–2631