Fake News Detection using Deep Learning

Aaditya Khant
University of Texas at San Antonio
San Antonio, Texas, US
aaditya.khant@my.utsa.edu

Akshay Kulshrestha
University of Texas at San Antonio
San Antonio, Texas, US
akshay.kulshrestha@my.utsa.edu

Zhensen Wang
University of Texas at San Antonio
San Antonio, Texas, US
zhensen.wang@my.utsa.edu

Abstract

Tackling the critical challenge of fake news detection, our study confronts a pivotal issue in an era marked by an unprecedented surge of information. The rampant spread of fake news not only misleads the public but also poses a significant threat to the integrity of democratic systems, highlighting the urgent need for effective identification and mitigation strategies. We leverage cutting-edge deep learning technologies, including Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), and Bidirectional Encoder Representations from Transformers (BERT) [1], to dissect and pinpoint the nuanced features of fake news. Through rigorous training and testing across four diverse datasets, our goal is to significantly enhance our model's ability to generalize across different contexts and improve its detection accuracy. The rapid advancements in deep learning technology, coupled with the vast availability of data, make our proposed solution both timely and feasible. The culmination of our research will be the development of an automated tool capable of efficiently and accurately identifying fake news, offering robust support for media verification efforts and empowering the public with greater discernment in navigating information.

1. Introduction

In an era dominated by digital information dissemination, the spread of fake news has emerged as a formidable challenge, threatening the integrity of public discourse, democratic processes, and societal trust. The term "fake news" encompasses a spectrum of misleading, inaccurate, or deliberately fabricated information presented as legitimate news. With the rise of social media platforms and the ease of content creation and sharing, distinguishing fact from fiction has become increasingly complex and crucial. The goal of fake news detection is multifaceted, aiming to develop computational methods and frameworks capable of automatically identifying and flagging deceptive or misleading content across various digital platforms. By leveraging advancements in machine learning, natural language processing, and data analytics, researchers seek to equip algorithms with the ability to discern the subtle nuances between credible journalism and misinformation.

In this project, we propose three deep learning models consisting of CNN, LSTM and transformer-based architecture for the task of fake news detection. Models were selected to capture the sequential dependency in the input data. As there is no direct correlation between type of news and media content such as videos and images in the news article, only textual content is used from title and body of the article to reduce complexity in input data. This approach also makes models more power and time efficient. All three models were trained to classify given news article in one of the four classes – fake, reliable, clickbait and bias.

All three models were trained and tested using combination of four opensource datasets – Fake News Corpus [2], News dataset from TI-CNN [3], getting real about fake news [4] and fake news detection [5]. Size of the dataset was kept significantly large to cover news from all domains. Before training, datasets were pre-processed to extract textual features and to reduce any present bias. Trained models were evaluated using measures like accuracy, f1 score and confusion matrix.

2. Related Work

Advancements in Natural Language Processing (NLP), Machine Learning (ML) and Deep Learning (DL) have allowed researchers to develop novel techniques to tackle the problem of fake news detection. Paper by *Xinyi Zhou and Reza Zafarani* [6] gives comprehensive overview of work done up to 2020 in this domain. Approaches to tackle this problem can be categorized into four types.

2.1 Knowledge-Based

Also known as fact-checking aims to judge news authenticity by comparing the knowledge extracted from news content with known facts. Process can be further divided into manual and automatic fact-checking. In former one relies on human experts, websites [[7] and [8]] or crowd [9] to check the credibility of the news and in

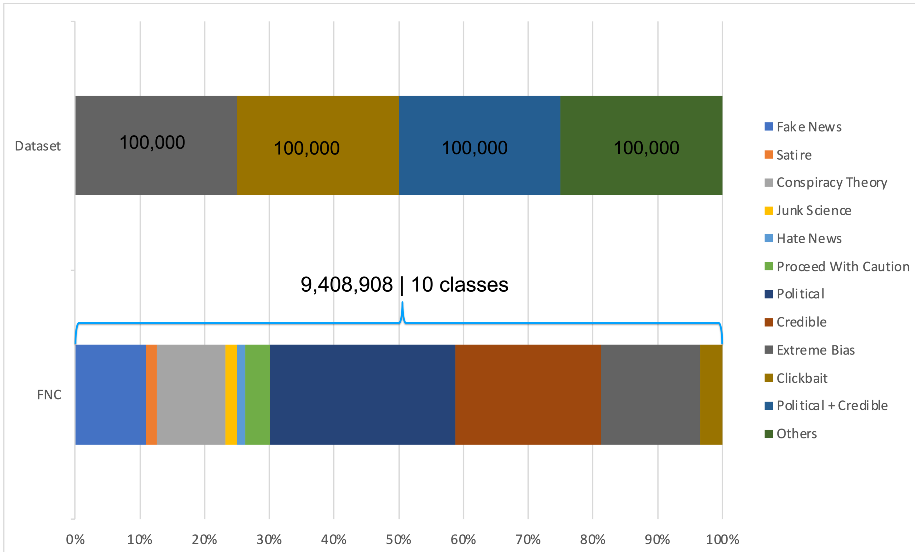

Figure 1 bar on top shows class distribution in training data and bar at the bottom shows class distribution in Fake News Corpus

former techniques such as Information Retrieval, NLP and network/graph theory [10] are used to validate news. While this method is known for its high accuracy, its effectiveness is impeded by the accessibility of information and the time required to verify news.

2.2 Style Based

In this approach, ML and DL techniques are used to classify news by analyzing general or latent textual features of the news. General textual features are often used to detect fake news within a traditional ML framework. These features describe content style from four language levels: lexicon [11], syntax [12], discourse [13], and semantic [14]. Latent textual features are often used for news text embedding conducted at word, sentence or document level. These embeddings can be directly used as the input to ML or DL based classifiers. ML models that can detect news type based on latent or non-latent representation can be supervised, semi-supervised, or unsupervised. Supervised methods mainly relied on Support Vector Machine (SVM) [15], Random Forest (RF) [11] and XGBoost [16]. In DL framework, embedding is processed by a well-trained neural network (e.g., CNNs [17] such as VGG-16/19 [18]; RNNs such as LSTMs [19]; and the Transformers such as BERT[20]) to classify news.

DL based models have showed the best results although they depend on the scalability of the dataset and model architecture. In this project we aim to train DL model on extensive dataset to achieve better performance compared to previous models.

2.3 Propagation Based

This approach focuses on tracking how news or information spreads through social networks, online communities, or traditional media channels. By examining factors such as the velocity, reach, and engagement levels of a piece of content, propagation-based detection aims to uncover patterns indicative of misinformation or disinformation campaigns. Techniques used in propagation-based detection include social network analysis, sentiment analysis, and anomaly detection algorithms. This approach is inefficient for fake news early detection, as it is difficult to detect fake news before it has been disseminated, or when limited news dissemination information is available. One also must rely on expert generated ground truth when predicting fake news by (semi-)supervised learning within graph optimization [21], traditional statistical learning [22], or deep neural networks [23].

2.4 Source Based

This approach focuses on assessing the reputation, expertise, and trustworthiness of the individuals, organizations, or media outlets responsible for producing and disseminating news content. Such an approach might seem arbitrary but is efficient [24], as evidence has revealed that many fake news stories come from either fake news websites that only publish hoaxes or from hyper-partisan websites that present themselves as publishing real news [25].

3. Methodology

Pipeline used in this project for fake news detection can be divided into two parts – data preparation and model deployment.

3.1. Data Preparation

Four opensource datasets are used for training and testing of the DL models. Datasets are cleaned and prepared before training. Firstly, only title, body and type of the news article are extracted from the datapoints. Next, from the textual data unnecessary elements like links, punctuation, non-alphanumeric characters, stop-words are removed.

Dataset	Number of articles	Number of classes
Fake News Corpus	9,408,908	11
Getting Real about Fake News	12,999	5
Fake News Detection	3,352	2
News dataset from TI-CNN	20,015	2

Table 1 Dataset

Table 1 gives the overview of all four datasets. Subset of 200,000 news articles from Fake News Corpus dataset was used to train the models and remaining three datasets were used to evaluate the model. This way we can test the model on completely never seen data before. Figure 1 shows distribution of training dataset among four classes. We have kept the number of news article in each class significantly large yet computationally optimum. Bias in the trained model was maintained minimum by using balanced training set. While generating training set classes conspiracy, junksci, rumor, unknown, satire, hate, unreliable and fake were considered as fake; classes reliable and political were considered as valid; bias as bias; and clickbait as clickbait. 50,000 news articles were selected for each of the four classes.

For CNN and LSTM models, tokenized tile and content further converted into vectors using Global Vectors for Word Representation (GLoVe) embeddings [26]. GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training of GLoVe was performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space. Particularly, we used glove.6B.50d variant, which represents each word using 50 floating point numbers. glove.6B was trained on Wikipedia 2014 and Gigaword 5 corpus, which combined contains 6B uncased tokens and vocabulary of size 400k [27]. Maximum length of title is set to 15 and content is set to 1600. String containing tokens more than limit is truncated and less than limit is padded with zeros.

3.2. Model Deployment

Processed data was used to train and evaluate three deep learning models. Training and evaluation details of these models are as following.

3.2.1 CNN

CNN are a class of deep neural networks primarily used for image recognition and processing, but they have also shown effectiveness in other tasks such as natural language processing and speech recognition. Figure 2 shows architecture detail of CNN model used for this project.

```
Layer (type)                 Output Shape          Param #   Connected to
==================================================================================
title_input (InputLayer)     [(None, 15, 50)]      0         []

content_input (InputLayer)   [(None, 1600, 50)]    0         []

Conv1Title (Conv1D)          (None, 6, 5)          1005      ['title_input[0][0]']

Conv2Content (Conv1D)        (None, 397, 80)       64080     ['content_input[0][0]']

Pool1Title (MaxPooling1D)    (None, 3, 5)          0         ['Conv1Title[0][0]']

Pool2Content (MaxPooling1D   (None, 99, 80)        0         ['Conv2Content[0][0]']
)

flatten (Flatten)            (None, 15)            0         ['Pool1Title[0][0]']

flatten_1 (Flatten)          (None, 7920)          0         ['Pool2Content[0][0]']

Dense1Title (Dense)          (None, 50)            800       ['flatten[0][0]']

Dense1Content (Dense)        (None, 100)           792100    ['flatten_1[0][0]']

concatenate (Concatenate)    (None, 150)           0         ['Dense1Title[0][0]',
                                                              'Dense1Content[0][0]']

dense (Dense)                (None, 50)            7550      ['concatenate[0][0]']

dropout (Dropout)            (None, 50)            0         ['dense[0][0]']

dense_1 (Dense)              (None, 50)            2550      ['dropout[0][0]']

dropout_1 (Dropout)          (None, 50)            0         ['dense_1[0][0]']

dense_2 (Dense)              (None, 4)             204       ['dropout_1[0][0]']

==================================================================================
Total params: 868289 (3.31 MB)
Trainable params: 868289 (3.31 MB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 2 CNN Architecture

Training dataset contained 180,000 training and 10,000 validation news articles. Model was trained for 10 epochs using cross entropy loss function, RMSprop optimizer and batch size of 500. Dataset was shuffled after each epoch. Figure 3 shows training and validation loss after each epoch.

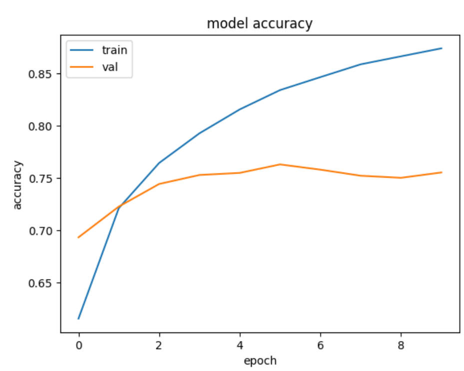

Figure 3 CNN training

3.2.2 LSTM

LSTM networks are a type of recurrent neural network (RNN) architecture designed to capture long-term dependencies and handle the vanishing gradient problem often encountered in traditional RNNs. Figure 4 shows architecture detail of LSTM model used for this project.

Layer (type)	Output Shape	Param #	Connected to
title_input (InputLayer)	[(None, 15, 50)]	0	[]
content_input (InputLayer)	[(None, 1600, 50)]	0	[]
bidirectional (Bidirectional)	(None, 46)	13616	['title_input[0][0]']
bidirectional_1 (Bidirectional)	(None, 300)	241200	['content_input[0][0]']
concatenate (Concatenate)	(None, 346)	0	['bidirectional[0][0]', 'bidirectional_1[0][0]']
dense (Dense)	(None, 73)	25331	['concatenate[0][0]']
dropout (Dropout)	(None, 73)	0	['dense[0][0]']
dense_1 (Dense)	(None, 24)	1776	['dropout[0][0]']
dropout_1 (Dropout)	(None, 24)	0	['dense_1[0][0]']
dense_2 (Dense)	(None, 4)	100	['dropout_1[0][0]']

Total params: 282023 (1.08 MB)
Trainable params: 282023 (1.08 MB)
Non-trainable params: 0 (0.00 Byte)

Figure 4 LSTM architecture

Same hyperparameters and training dataset as CNN were used for LSTM training. Figure 5 shows training and validation loss after each epoch.

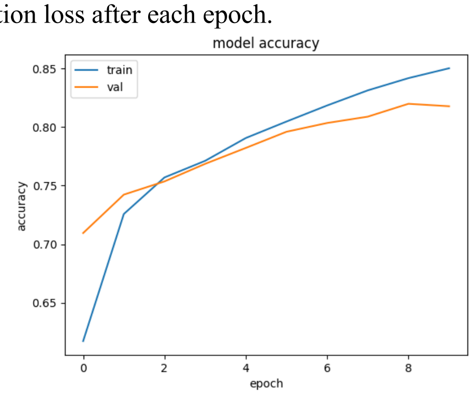

Figure 5 LSTM Training

3.2.3 BERT

BERT and other Transformer encoder architectures have been wildly successful on a variety of tasks in NLP (natural language processing). They compute vector-space representations of natural language that are suitable for use in deep learning models. The BERT family of models uses the Transformer encoder architecture to process each token of input text in the full context of all tokens before and after. BERT models are usually pre-trained on a large corpus of text, then fine-tuned for specific tasks.

Title and content were embedded using SavedModel from TensorFlow hub [28]. A preprocessor SavedModel provides a way to map single-segment text inputs directly to encoder inputs.

Small Bert model with 4 layers, 512 hidden size and 8 self-attention heads was selected because of computational limitation [29]. Model was fine-tuned using 50,000 news articles and two epochs. Adam west optimizer with batch size of 32 was used.

4. Results

Each of the three trained models were evaluated on 10,000 news articles. Confusion matrix, precision, recall and f1-score were used to evaluate the models. Figure 6 shows confusion matrix for all three models. Figure 7 shows precision, recall and f1-score for all three models. From the results we can clearly see that there is no bias or over fitting present in trained models. Models CNN, LSTM and BERT achieved accuracy of 76%, 83% and 78% respectively. Although BERT model had significantly more parameters and took significantly more time to train, it achieved compatible accuracy in just two epochs and 50,000 news articles.

Note: To duplicate the results, python code used for this project is uploaded at:
https://github.com/aadityakhant/fakeNewsDetection.git

5. Conclusion

In this project we created three DL models using CNN, LSTM and BERT architecture. We cleaned and pre-processed dataset to train models. We trained three created models and evaluated. As per intuition, we show that LSTM model outperformed CNN model, and BERT model took significantly more computational resources compare to CNN and LSTM model because of complex architecture and large number of trainable parameters.

6. Future Work

DL Models used in this project were not parameter costly. Reason behind this is limited availability of computational resources and training time. With more complex architecture and larger dataset all three models can achieve much higher accuracy for detecting fake news. One can also combine these models with other approaches, such as fact checking, and create ensemble method to concretely achieve the objective.

7. References

[1] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).
[2] https://github.com/several27/FakeNewsCorpus
[3] Yang, Yang, et al. "TI-CNN: Convolutional neural networks for fake news detection." *arXiv preprint arXiv:1806.00749* (2018).
[4] https://www.kaggle.com/datasets/mrisdal/fake-news

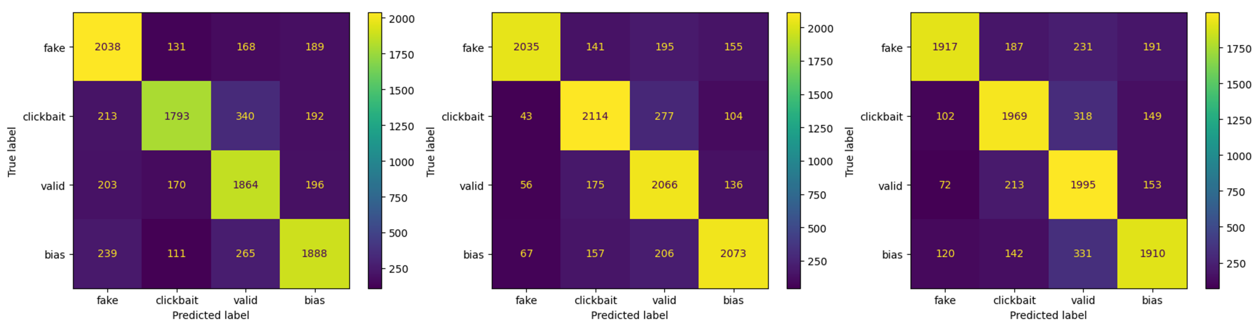

Figure 6 Confusion matrix for CNN (left), LSTM(center) and BERT(right)

	precision	recall	f1-score	support
0	0.76	0.81	0.78	2526
1	0.81	0.71	0.76	2538
2	0.71	0.77	0.74	2433
3	0.77	0.75	0.76	2503
accuracy			0.76	10000
macro avg	0.76	0.76	0.76	10000
weighted avg	0.76	0.76	0.76	10000

	precision	recall	f1-score	support
0	0.92	0.81	0.86	2526
1	0.82	0.83	0.82	2538
2	0.75	0.85	0.80	2433
3	0.84	0.83	0.83	2503
accuracy			0.83	10000
macro avg	0.83	0.83	0.83	10000
weighted avg	0.83	0.83	0.83	10000

	precision	recall	f1-score	support
0	0.87	0.76	0.81	2526
1	0.78	0.78	0.78	2538
2	0.69	0.82	0.75	2433
3	0.79	0.76	0.78	2503
accuracy			0.78	10000
macro avg	0.78	0.78	0.78	10000
weighted avg	0.79	0.78	0.78	10000

Figure 7 Precision, recall and f1-score for CNN (left), LSTM(center) and BERT(right)

[5] https://www.kaggle.com/datasets/jruvika/fake-news-detection

[6] Zhou, Xinyi, and Reza Zafarani. "A survey of fake news: Fundamental theories, detection methods, and opportunities." *ACM Computing Surveys (CSUR)* 53.5 (2020): 1-40.

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

[8] https://fullfact.org

[9] Mitra, Tanushree, and Eric Gilbert. "Credbank: A large-scale social media corpus with associated credibility annotations." *Proceedings of the international AAAI conference on web and social media*. Vol. 9. No. 1. 2015.

[10] Cohen, Sarah, James T. Hamilton, and Fred Turner. "Computational journalism." *Communications of the ACM*54.10 (2011): 66-71.

[11] Zhou, Xinyi, et al. "Fake news early detection: A theory-driven model." *Digital Threats: Research and Practice* 1.2 (2020): 1-25.

[12] Pérez-Rosas, Verónica, et al. "Automatic detection of fake news." *arXiv preprint arXiv:1708.07104* (2017).

[13] Karimi, Hamid, and Jiliang Tang. "Learning hierarchical discourse-level structure for fake news detection." *arXiv preprint arXiv:1903.07389* (2019).

[14] Potthast, Martin, et al. "A stylometric inquiry into hyperpartisan and fake news." *arXiv preprint arXiv:1702.05638* (2017).

[15] Feng, Song, Ritwik Banerjee, and Yejin Choi. "Syntactic stylometry for deception detection." *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2012.

[16] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016.

[17] Zhang, Xiang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification." *Advances in neural information processing systems* 28 (2015).

[18] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[19] Rashkin, Hannah, et al. "Truth of varying shades: Analyzing language in fake news and political fact-checking." *Proceedings of the 2017 conference on empirical methods in natural language processing*. 2017.

[20] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

[21] Shu, Kai, Suhang Wang, and Huan Liu. "Beyond news contents: The role of social context for fake news detection." *Proceedings of the twelfth ACM international conference on web search and data mining*. 2019.

[22] Zhou, Xinyi, and Reza Zafarani. "Network-based fake news detection: A pattern-driven approach." *ACM SIGKDD explorations newsletter* 21.2 (2019): 48-60.

[23] Zhang, Jiawei, et al. "Fake news detection with deep diffusive network model." *arXiv preprint arXiv:1805.08751* (2018).

[24] Nørregaard, Jeppe, Benjamin D. Horne, and Sibel Adalı. "NELA-GT-2018: A large multi-labelled news dataset for the study of misinformation in news articles." *Proceedings of the international AAAI conference on web and social media*. Vol. 13. 2019.

[25] https://www.buzzfeednews.com/article/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook

[26] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.

[27] https://nlp.stanford.edu/projects/glove/

[28] https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3

[29] https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-4_H-512_A-8/1