Fake News Detection using Deep Learning and Natural Language Processing

B. Tech. Project Report

Submitted by

Vishal Sharma 111903159

Under the guidance of

Prof. Archana B. Patil

College of Engineering, Pune-5

DEPARTMENT OF COMPUTER ENGINEERING

AND

INFORMATION TECHNOLOGY,

COLLEGE OF ENGINEERING, PUNE-5

May 2023

DEPARTMENT OF COMPUTER ENGINEERING

AND

INFORMATION TECHNOLOGY,

COLLEGE OF ENGINEERING, PUNE

CERTIFICATE

Certified that this project, titled "Fake News Detection using Deep Learning and Natural Language Processing" has been successfully completed by

Vishal Sharma 111903159

and is approved for the partial fulfillment of the requirements for the degree of "B.Tech. Computer Engineering".

SIGNATURE	SIGNATURE
Prof. Archana B. Patil	**Dr. Pradeep K. Deshmukh**
Project Guide	**Head**
Department of Computer Engineering	**Department of Computer Engineering**
and Information Technology,	**and Information Technology,**
College of Engineering Pune,	**College of Engineering Pune,**
Shivajinagar, Pune - 5.	**Shivajinagar, Pune - 5.**

Plagiarism report

ORIGINALITY REPORT

18%	13%	13%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.mdpi.com Internet Source	1%
2	github.com Internet Source	1%
3	aniketbhadane.github.io Internet Source	1%
4	"Combating Fake News with Computational Intelligence Techniques", Springer Science and Business Media LLC, 2022 Publication	1%
5	www.researchgate.net Internet Source	1%
6	www.adaface.com Internet Source	1%
7	medinform.jmir.org Internet Source	1%
8	Ursula Challita, Aidin Ferdowsi, Mingzhe Chen, Walid Saad. "Machine Learning for Wireless Connectivity and Security of Cellular-	<1%

ABP
15/4/23
A.B. PATIL

Abstract

The proliferation of false information on social media platforms has become a major concern in modern society, necessitating the development of effective strategies for detecting and preventing the dissemination of fake news. This study proposes the application of advanced deep learning techniques, including LSTM (Long Short-Term Memory), Bi-LSTM (Bidirectional LSTM), and CNN-Bi-LSTM (Convolutional Neural Network), for the purpose of identifying fabricated news.The dataset we are using for training and testing the model consists of a combination of real and fake news articles from various sources. We will split data into training,validation and testing.The evaluation metrics used include accuracy.Furthermore, the model will have the ability to generalize well to unseen data, which is critical in real-world scenarios. We will apply these models on both balanced and imbalanced datasets of fake news detection.Additionally, the system will employ GloVe word embedding on text data to enhance the accuracy of the proposed models.We will use this technique to represent the text data in a meaningful way that can be used as input to our deep learning models. Additionally, an evaluation of the effectiveness of each of the mentioned architectures will be performed through a comparative analysis to determine their ability to detect fabricated news.The contemporary society is confronted with a significant issue of fake news spread on social media platforms, and there is an increasing requirement for efficient approaches to detect and prevent the propagation of misleading information. To address this challenge, this research recommends the utilization of deep learning techniques such as LSTM (Long Short-Term Memory), Bi-LSTM (Bidirectional LSTM), and CNN-Bi-LSTM (Convolutional Neural Network) to identify false news.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The prevalence of social media platforms and the ease of sharing information has made it difficult to combat the dissemination of fabricated news, which refers to deliberately misleading false information. Furthermore, the problem is compounded by the prevalent habit of sharing news without verifying its authenticity, leading to the widespread propagation of fake news. As a result, identifying false news has become a major issue for both social media platforms and society as a whole. To address this problem, there is a growing need for effective techniques to combat the spread of fake news. Traditional methods rely on human experts to identify fake news, which can be slow and prone to error. Therefore, there is increasing interest in developing automated techniques to detect fake news. Deep learning is a category of machine learning that enables machines to learn and make decisions autonomously without human intervention. Its popularity stems from its ability to extract significant features from raw data and learn from large amounts of information. Deep learning has shown considerable potential in several natural language processing (NLP) tasks, such as sentiment analysis, text classification, and machine translation. Consequently, it is logical to explore the possibilities of using deep learning for detecting fabricated news.

In the realm of natural language processing, deep learning has shown significant advancements, particularly through the utilization of LSTM (Long Short-Term Memory) and Bi-LSTM(Bidirectional LSTM) neural network architectures, which have delivered remarkable outcomes in processing sequential data such as text. Meanwhile, CNNs(Convolutional Neural Networks) have demonstrated great efficacy in handling image data. The fusion of CNN and Bi-LSTM models, known as CNN-Bi-LSTM, has shown exceptional performance, particularly in tasks involving text classification.

Our project aims to investigate the potential of deep learning methods for detecting fake news. We will employ LSTM, Bi-LSTM, and CNN-Bi-LSTM models and apply them to datasets, both balanced and imbalanced. Our objective is to develop accurate models for classifying news articles as real or fake by utilizing the GloVe word embedding technique to represent the text data. GloVe is an unsupervised learning approach that maps words to a vector space based on their co-occurrence with other words in a corpus, which has shown to be successful in capturing word meaning and relationships. We will compare the models' accuracy and evaluate the impact of data balancing on their performance. Our study aims to demonstrate the effectiveness of deep learning methods in detecting fake news and how the utilization of GloVe word embeddings can improve the models' performance.

This project addresses a crucial societal concern, which is identifying fake news. By utilizing deep learning techniques such as LSTM, Bi-LSTM, and CNN-Bi-LSTM, along with the GloVe word embedding technique, we can potentially offer a solution to the issue of detecting fake news. The creation of reliable models for detecting fake news can play a vital role in combating the spread of false information and preserving the integrity of democratic institutions.

Chapter 2

Literature Review

The main objectives of this survey are to present diverse deep learning, machine learning, and natural language processing techniques utilized in developing systems for detecting fabricated news. To identify the most effective method, I have reviewed the following research papers.

The paper[**1**] "Detecting Fake News with Capsule Neural Networks" by Goldani, Momtazi, and Safabakhsh proposes a novel approach to detect fake news using Capsule Neural Networks (CapsNets), a type of deep learning architecture. According to the authors, CapsNets are capable of capturing the hierarchical links between various aspects of news items, which is helpful for spotting false news.The study compares the performance of CapsNets to other machine learning models, such as SVM (Support Vector Machines) and CNNs (Convolutional Neural Networks), using a dataset of actual and fraudulent news items. The findings demonstrate that CapsNets beat the other models in terms of F1 score, accuracy, precision, and recall.The model was trained and tested on a dataset of news articles collected from various sources and achieved an accuracy of over 95% in detecting fake news.

The paper[**2**] "An ensemble machine learning approach through effective feature extraction to classify fake news" by Hakak, Alazab, Khan, Gadekallu, Maddikunta, and Khan proposes an ensemble machine learning approach to classify fake news by extracting effective features. According to the study's authors, selecting appropriate features is a crucial step in detecting fabricated news, as it can significantly impact the performance of machine learning models. To evaluate their proposed feature selection method, the authors utilized a dataset containing genuine and fake news stories and tested various machine learning models, including Random Forest, Gradient Boosting, and Support Vector Machines (SVM). Their approach was compared to other feature selection methods, such as Information Gain and Chi-Squared. The dataset used in the research consisted of news articles sourced from different outlets, and the proposed model achieved a detection accuracy of over 96% after training and testing.

The paper[**3**] "Sentiment Analysis for Fake News Detection by Means of Neural Networks" by Kula, Choraś, Kozik, Ksieniewicz, and Woźniak proposes a method for detecting fake news by analyzing the sentiment of the text. According to the authors, exaggerated or misleading language is a common feature of fake news articles, which can be identified through sentiment analysis. The authors of the study employed a dataset that comprised both authentic and fake news articles and assessed the performance of various neural network models, including Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), to verify their proposed method. They also compared their approach with other machine learning models such as Support Vector Machines (SVM) and Random Forest. The dataset consisted of news articles sourced from diverse channels, and after training and testing the model, the authors attained a detection accuracy of more than

90%.

The paper[4] by Jamal Abdul Nasir, Osama Subhani Khan, and Iraklis Varlamis proposes a hybrid deep learning approach for detecting fake news using a combination of CNNs (convolutional neural networks) and RNNs (recurrent neural networks). After evaluating their approach on two benchmark datasets, the authors were able to achieve significantly improved accuracies in comparison to existing methods. Their findings indicate that the hybrid CNN-RNN model they proposed can successfully identify fake news and may prove to be a valuable tool in addressing the spread of misinformation on the internet. The paper ultimately adds to the increasing amount of research dedicated to detecting fake news and provides a hopeful approach for further studies in this area.

The paper[5] by Sastrawan, Bayupati, and Arsa presents a deep learning approach to the detection of fake news using a combination of CNNs (Convolutional Neural Networks) and RNN (Recurrent Neural Networks).The authors approach demonstrated enhanced accuracy in identifying fake news compared to current methods, highlighting its potential to aid in the battle against the dissemination of false information. The results of the study are particularly significant in today's context, where fake news has become a major worry due to its capacity to cause harm to individuals, communities, and society as a whole. This research presents a significant contribution to the area of deep learning and its potential use in combating fake news.

Based on their survey paper[6], Kaliyar and Singh have highlighted the various techniques and approaches that have been employed in the detection of misinformation on online social media. They have examined the pros and cons of different algorithms and tools used for misinformation detection and concluded that there is no single approach that can guarantee 100% accuracy in detecting misinformation.However, the authors have highlighted that recent advancements in machine learning techniques have improved the accuracy of detecting misinformation on social media platforms. They have presented some of the state-of-the-art techniques that have been shown to produce promising results in detecting misinformation. These include neural networks, deep learning, and natural language processing techniques.

The paper[**7**] by Hitesh Narayan Soneji and Sughosh Sudhanvan proposes a hybrid approach for detecting false news using CNNs and random forest. The study indicates that the authors' proposed approach surpasses current methods in accuracy, precision, and recall metrics. By utilizing two models, one concentrating on image-based news and the other on text-based news, the approach provides a more comprehensive system for fake news detection. The results suggest that a hybrid approach has potential for future enhancements in detecting fake news.

Chapter 3

Research Gaps and Problem Statement

3.1 Research Gaps

Fake news is a growing problem in today's society. The spread of misinformation can have serious consequences, ranging from political polarization to public health issues. Consequently, there has been significant research in developing automatic fake news detection systems. However, these systems still face several gaps and limitations. In this response, I will discuss the research gaps and limitations in the current fake news detection systems and explain how GloVe embedding and CNN-BiLSTM can help overcome these limitations.

1 Lack of Training Data: Fake news detection systems are trained on large datasets of labeled data. However, collecting labeled data is a time-consuming and expensive process. As a result, the amount of available training data is limited.

2 Sensitivity to the Source: Fake news detection systems are often trained on data from a specific source or set of sources. This makes them less effective at detecting fake news from sources they haven't been trained on.

3 Limited Contextual Understanding: Current fake news detection systems often rely on simple features like the headline, content, and metadata of a news article. However, this approach does not consider the broader context of the article or the news cycle in which it appears.

Using GloVe embedding and CNN-BiLSTM, we can improve the fake news detection system in the following ways:

1 Better Use of Limited Data: By using CNN-BiLSTM, we can effectively capture both the local and global context of news articles, even when training data is limited. This enables us to improve the performance of fake news detection systems even when we have a limited amount of labeled data.

2 Improved Sensitivity to the Source: By using GloVe embedding, we can better understand the meaning of words in news articles, making it easier for fake news detection systems to detect fake news from sources they haven't been trained on.

3 Better Contextual Understanding: By using CNN-BiLSTM, we can capture the broader context of news articles and the news cycle in which they appear. This enables us to better understand the context of a news article, making it easier to detect whether it is fake or not. For example, by understanding the timing of an article in relation to a major news event, the system can better detect whether the article is part of a coordinated disinformation campaign.

In conclusion, fake news detection systems still face several gaps and limitations, including the lack of multilingual capability, limited training data, sensitivity to the source, and limited contextual understanding. However, by using GloVe embedding and CNN-BiLSTM, we can overcome many of these limitations and may improve the accuracy of fake news detection systems. These techniques enable the system to better understand the meaning of words in news articles, capture the local and global context of news articles, and detect fake news more accurately.

3.2 Problem Statement

The proliferation of false information online and the surge of social media have caused an increase in the apprehension regarding fake news in recent times. The spread of false information can have serious consequences, from damaging reputations to influencing elections and inciting violence. Therefore, the problem statement for fake news detection is to develop reliable and efficient methods to identify and classify news articles as either real or fake. This involves analyzing the content and context of the article, as well as the sources and references cited within it, to determine its credibility and accuracy. The goal is to create algorithms and models that can accurately and efficiently differentiate between genuine news articles and those that are intentionally misleading or false, in order to prevent the spread of misinformation and promote the dissemination of accurate and reliable information.

Chapter 4

Proposed Methodology/ Solution

Fake news detection consists of several stages, including data collection, data preprocessing, data visualization, word embedding, model training, and model evaluation.

4.1 Data Collection

To conduct our study, we utilized the ISOT Fake News dataset provided by the ISOT Research Lab at the University of Victoria in Canada [11]. This dataset consists of a combination of authentic news stories and false news articles collected from various credible news sources, as well as untrustworthy websites rated by Politifact.com. The dataset originally came in two separate CSV files - one for genuine news and the other for fake news. We merged these files and applied a shuffling technique to generate the training, validation, and test sets at a ratio of 64

Table 4.1: Distribution of Data.

Training	**validation**	**Testing**
28734	7184	8980
64%	16%	20%

4.2 Data Visualisation

To gain insights from the dataset, we have created a word cloud visualization of the textual data. The word clouds revealed that the words "Trump," "say," "Russia," "House," "North," and "Korea" were prominent in the actual news word cloud, while the words "VIDEO," "Trump," "Obama," "WATCH," and "Hillary" appeared frequently in the fake news word cloud. Interestingly, the word "say" was found to be a common feature in legitimate news, but not in fake news.

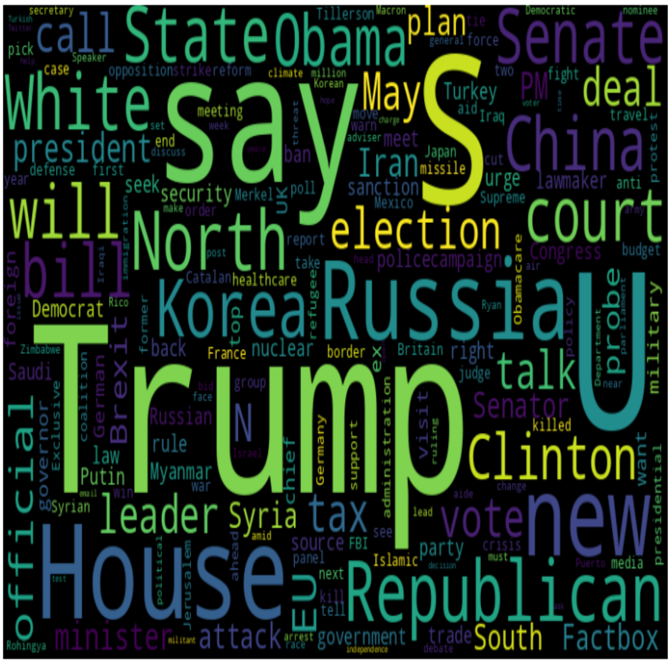

Figure 4.1: Real News Word Cloud

In fake news, the words "VIDEO" and "WATCH" are commonly used, but not in actual news. We can learn some crucial information to distinguish the two kinds of data from these 2 word clouds. Figure 4.1 shows the real news word cloud and Figure 4.2 shows the word cloud for fake news.

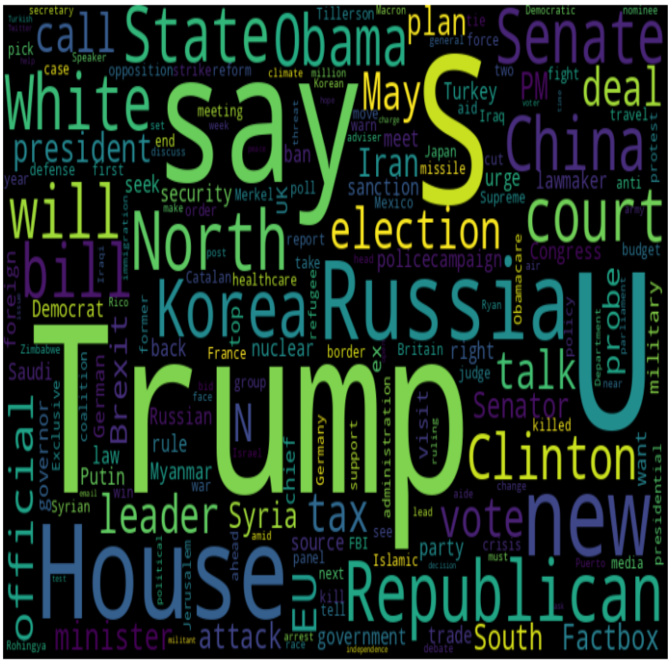

Figure 4.2: Fake News Word Cloud

We have generated a graph to analyze the length of news titles, which enabled us to identify appropriate sentence lengths for training our model on the dataset. It was crucial to exclude titles with extremely long lengths, as this could lead to the model being fitted on unbalanced data. By selecting titles with reasonable lengths, we were able to ensure that the model was trained effectively and could generalize well to new data.

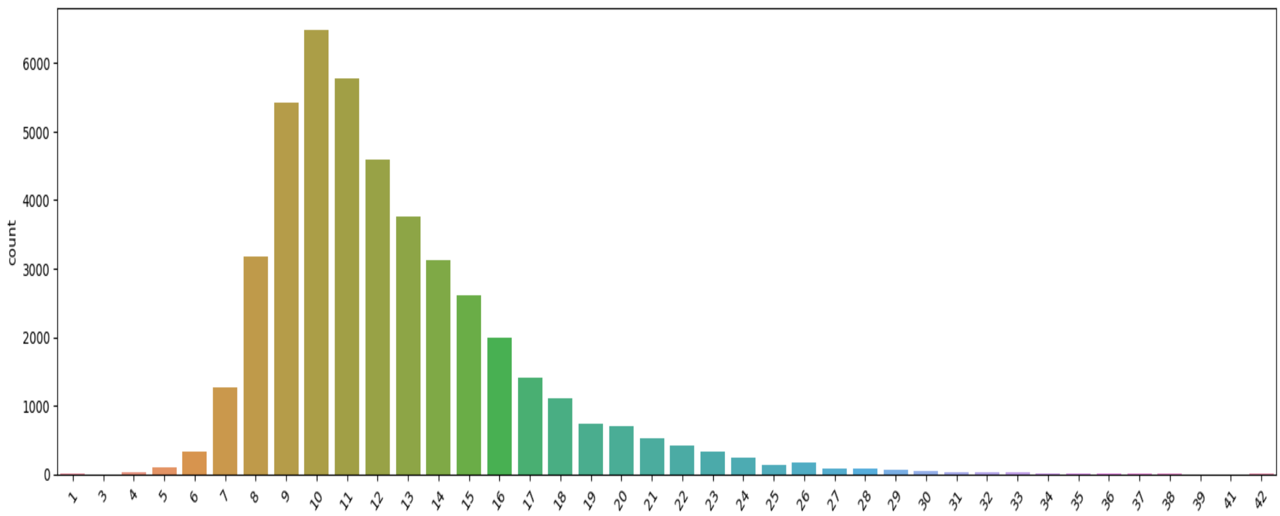

Figure 4.3: Inspection of News Titles

4.3 Data Preprocessing

The major objective of this section is to preprocess the input data using NLP techniques in order to set up the subsequent stage of feature extraction. News headlines and content are included in the data we utilise. The average length of news titles in our dataset was found to be 12.45 words, while the average word count of each content was 405.28. Due to the lengthy nature of the content, it was not feasible to use the entire text for training purposes. Instead, we focused solely on the titles for fake news identification. Moreover, the content of the news articles contained a vast amount of specific details and facts, which could potentially confuse the models during training. To tackle this problem, we created a preprocessing pipeline for every statement to remove any irrelevant information from the fake news dataset.

The following 3 components make up the preprocessing pipeline:

1. Whitespace was used in place of characters that are not in the range of a to z or A to Z.

2. Changed all of the characters to lowercase.

3. Putting forth Lemmatization: Each word is reduced to its lemma using the part-of-speech information. Lemmatization is the process of organising various word forms into a single unit that can be studied as a whole. For instance, the lemma "run" can be formed from the words "run," "running," and "ran."

To ensure that our model was trained on a balanced dataset with reasonable sentence lengths, we implemented a strategy to clip the titles to ensure that they were no longer than 42 characters. This approach enabled us to exclude titles with extreme lengths, which could have led to unbalanced data and model overfitting. By limiting the sentence length, we were able to train our model more effectively and ensure that it could generalize well to new data.

4.4 Word Embedding

Tokenization is a crucial step in preparing the dataset for use in machine learning models. In our study, we developed a tokenizer to divide the text into individual words and generate token sequences that can be fed into LSTM, Bidirectional LSTM, and CNN-Bi LSTM models.

4.4.1 GloVE Word Embedding

GloVe (Global Vectors for Word Representation), a popular word embedding method that maps words to high-dimensional vector representations that capture both semantic and syntactic relationships. Semantic relationships pertain to the meaning-based associations between words, such as the similarity between "cat" and "dog," while syntactic relationships involve the way in which words are utilized in language, such as the relationship between a verb and its object. The GloVe algorithm builds a co-occurrence matrix to count the number of times each word appears in the context of every other word within a corpus. A fixed window of words surrounding a given word determines its context. This matrix is then factorized using singular value decomposition (SVD) to acquire a low-dimensional representation of the co-occurrence matrix. The resulting embeddings capture the relationships between words based on their co-occurrence patterns.

4.4.2 FastText Word Embedding

FastText is a popular method for generating word embeddings, which are numerical representations of words that capture their semantic meaning. To create these embeddings, FastText first breaks down each word in a given corpus into its constituent subwords, which are typically character n-grams. For example, the word "cat" might be broken down into the subwords "c", "a", "t", "ca", "at". Next, each subword is represented as a vector, which is randomly initialized. These subword vectors are then combined to create a vector representation for each word in the corpus. This is done by adding up the subword vectors for each subword in the word, and normalizing the result. The resulting vector representation for each word is essentially a weighted average of the subword vectors, where the weights correspond to the frequency of each subword in the corpus. Once the vector representations for each word have been created, a neural network is trained to predict the likelihood of a word appearing in a given context, based on the vector representation of the word and the context words. During training, the subword vectors and the neural network weights are adjusted to minimize the prediction error.

4.5 Training and Testing Models

Deep learning models are a class of artificial neural networks (ANN) that use multiple interconnected layers of nodes to detect and learn patterns in data. The models are trained with a large dataset containing labeled information and an algorithm that fine-tunes the weights and biases of the nodes to reduce the difference between predicted and actual outputs. The neuron is the fundamental unit of a deep learning model, which mimics the biological neuron in the human brain. Neurons take one or more inputs, conduct mathematical operations, and produce an output. The output of one neuron becomes the input of another neuron in the following layer. In the training process, the model learns from labeled data by repeatedly adjusting the weights and biases of its nodes until a desirable level of accuracy is attained. Once trained, the model can apply this knowledge to make predictions on new, previously unseen data by passing it through the input and hidden layers, and producing an output from the output layer.

4.5.1 LSTM

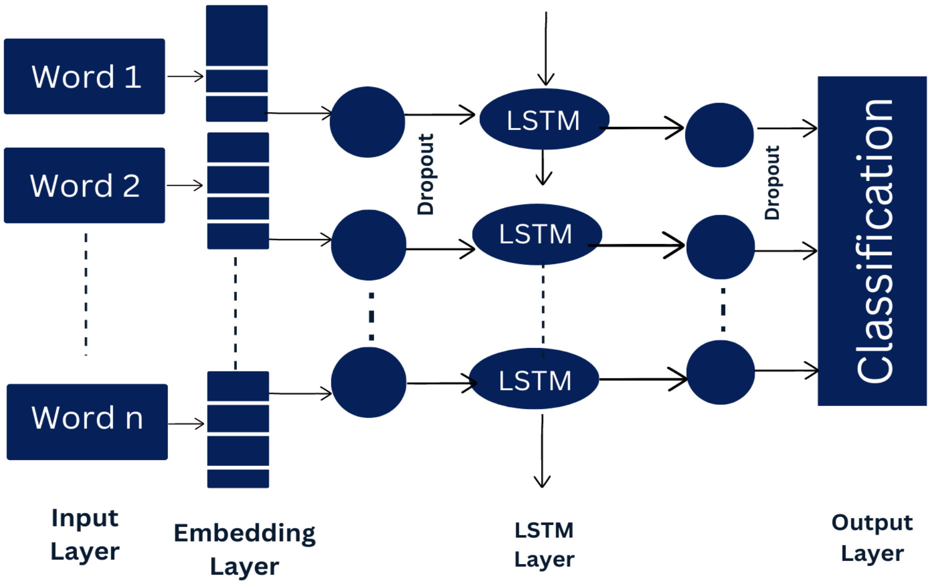

Figure 4.4: LSTM Model Architecture

LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) that addresses the vanishing gradient issue commonly found in conventional RNNs. The core concept of LSTM is to enable the network to selectively reserve or discard the past information, based on the current input. To regulate the flow of information through the network, specialized "memory cells" and gating mechanisms are employed. The LSTM cell comprises three primary components: the input gate, forget gate, and output gate. These gates use a sigmoid function that considers the present input and the output from the previous timestep to determine how much information to allow through. The input gate is responsible for deciding how much of the new input to allow through and adjusts the memory cell accordingly. In contrast, the forget gate controls how much of the previous memory to forget and updates the memory cell accordingly. The output gate determines how much of the current memory state to output as the prediction. The memory cell retains information over time and can be modified or reset by the input gate and forget gate. The combination of these components enables the LSTM to selectively remember or disregard past information based on the current input, making it a potent tool for processing sequential data.

4.5.2 Bidirectional LSTM

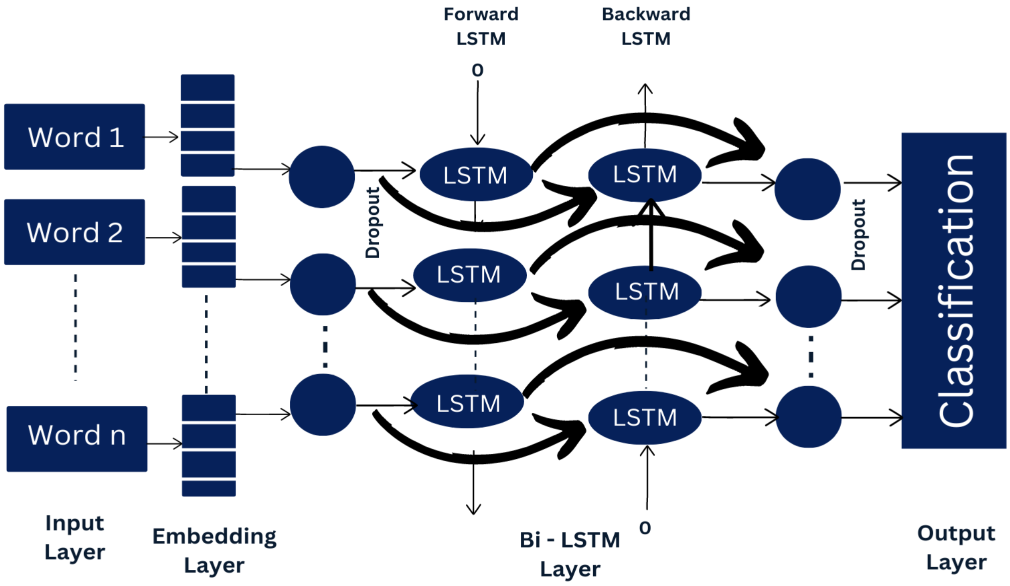

Figure 4.5: Bi-LSTM Model Architecture

A bidirectional LSTM network is a type of neural network that processes input data in both the forward and backward directions. The network consists of two separate LSTM layers: one that processes the input sequence in the forward direction, and another that processes it in the backward direction.

When processing the input sequence in the forward direction, the LSTM layer starts with the first input and processes each input in turn, updating its hidden state at each time step. When processing the input sequence in the backward direction, the LSTM layer starts with the last input and processes each input in reverse order, also updating its hidden state at each time step. The outputs from both LSTM layers are then combined using an appropriate merging function, such as concatenation. The concatenated output is then passed through a dense layer to produce the final output. By processing the input sequence in both directions, bidirectional LSTMs can capture both past and future context, which can be useful for many natural language processing tasks, such as sentiment analysis, named entity recognition, and machine translation.

4.5.3 CNN-Bi-LSTM

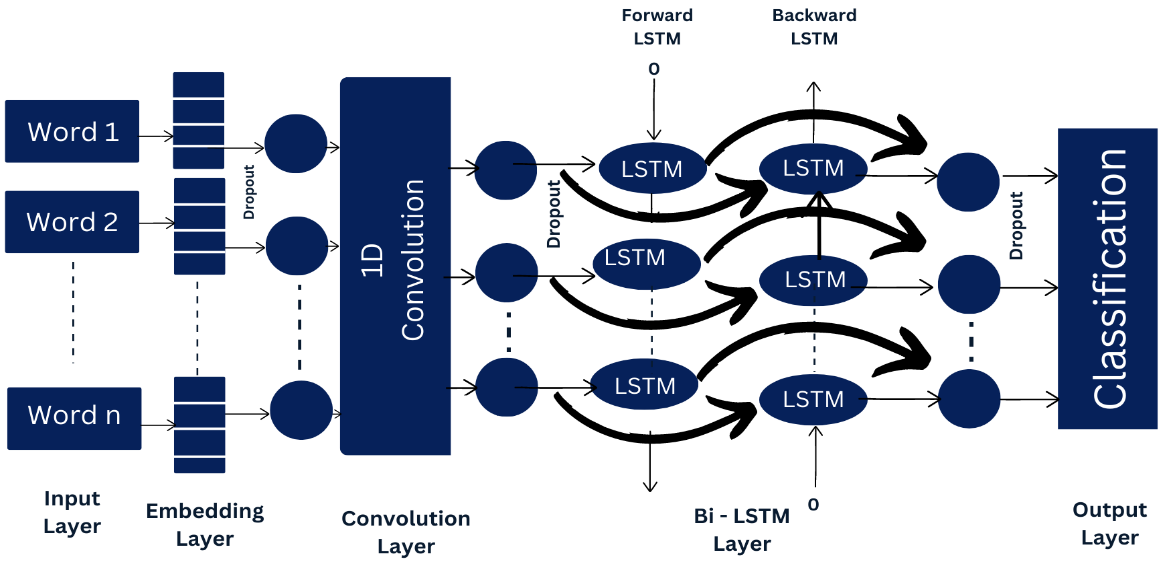

Figure 4.6: CNN-Bi-LSTM Model Architecture

The CNN-Bidirectional LSTM is a neural network architecture designed to process sequential data by combining the strengths of CNNs (Convolutional Neural Networks) and Bi-LSTM (Bidirectional Long Short-Term Memory networks). The CNN component is used to extract spatial features from the input sequence, while the BiLSTM component captures temporal dependencies in the sequence. Here's a high-level overview of how the CNN-BiLSTM works:

1. Convolutional layer: The input sequence is first passed through a convolutional layer, which applies a set of learnable filters to the input sequence to extract spatial features.

2. Pooling layer: The output from the convolutional layer is then passed through a pooling layer, which reduces the spatial dimension of the output.

3. Bidirectional LSTM layer: The output from the pooling layer is fed into a bidirectional LSTM layer. The BiLSTM is a type of recurrent neural network that processes the input sequence in both forward and backward directions, allowing it to capture temporal dependencies in both directions.

4. Output layer: The output from the BiLSTM layer is then passed through a fully connected output layer, which produces the final output of the network.

During training, the weights of the network are adjusted using backpropagation to minimize a specified loss function. The network is typically trained on a large labeled dataset, with the goal of learning to accurately classify new, unseen examples which we can utilize it for fake news detection.

```
Layer (type)                Output Shape              Param #
=================================================================
input_1 (InputLayer)        [(None, 44)]              0

embedding (Embedding)       (None, 44, 100)           2667000

dropout (Dropout)           (None, 44, 100)           0

conv1d (Conv1D)             (None, 44, 16)            4816

dropout_1 (Dropout)         (None, 44, 16)            0

max_pooling1d (MaxPooling1D  (None, 22, 16)           0
)

bidirectional (Bidirectiona  (None, 64)               12544
l)

dropout_2 (Dropout)         (None, 64)                0

dense (Dense)               (None, 1)                 65

=================================================================
```

Figure 4.7: Model Architecture for imbalanced dataset

```
Layer (type)                Output Shape              Param #
=================================================================
input_1 (InputLayer)        [(None, 36)]              0

embedding (Embedding)       (None, 36, 100)           1584500

dropout (Dropout)           (None, 36, 100)           0

conv1d (Conv1D)             (None, 36, 16)            4816

dropout_1 (Dropout)         (None, 36, 16)            0

max_pooling1d (MaxPooling1D  (None, 18, 16)           0
)

bidirectional (Bidirectiona  (None, 64)               12544
l)

dropout_2 (Dropout)         (None, 64)                0

dense (Dense)               (None, 1)                 65

=================================================================
```

Figure 4.8: Model Architecture for balanced dataset

4.6 Model Evaluation

We use accuracy as an evaluation metric to measure the effectiveness of our model on the test dataset. The accuracy metric is commonly used to evaluate classification models, as it indicates the model's ability to make correct predictions. This score is obtained by dividing the number of correct predictions made by the model with the total number of predictions made. The resulting value represents the percentage of correctly classified articles in the test dataset. In particular, the accuracy score is computed by dividing the total number of articles in the test set by the sum of true positives (legitimate news items correctly identified as such) and true negatives (fake news articles correctly identified as such).

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

Figure 4.9: Evaluation metric : Accuracy

Chapter 5

Experimental Setup

5.1 Requirements

5.1.1 Software Requirements

(a) Python3

(b) Google Colab or Jupyter Notebook

(c) Word Cloud

(d) Numpy

(e) Pandas

(f) Seaborn

(g) Matplotlib

(h) Sklearn(Scikit Learn)

(i) Natural Language Toolkit (NLTK)

(j) TensorFlow

(k) Keras

(l) Other Python Libraries

5.1.2 Hardware Requirements

(a) Laptop or computer with CPU having processor i3

(b) Minimum of 4 GB RAM is required.

(c) The Operating system that supports deep learning framework

5.1.3 Dataset

The distribution of genuine and fake data for both Original and Imbalanced datasets in the training,validation and testing sets is depicted in Figure 5.1.

Data	Training Set		Validation Set		Testing Set	
	Original	Imbalanced	Original	Imbalanced	Original	Imbalanced
True	13644	13644	3449	3449	4324	4324
Fake	15090	1509	3735	374	4656	466

Figure 5.1: Imbalanced Data Distribution

5.1.4 Hyper-parameter Settings

(a) Embedding Dimension d = 100

(b) Dropout = 0.5

(c) Convolutional filters = 16

(d) Conv1D Activation function - Relu

(e) Conv1D kernel size = 3

(f) LSTM units = 32

(g) Activation function - Sigmoid

(h) Regularizer = 0.01

(i) Optimizer - Adam

(j) Loss function - Binary_crossentropy

(k) Epochs = 15

(l) Batch size = 32

Chapter 6

Results and Discussion

In our sample, both real and fake news are equally represented. Contrary to what the dataset suggests, however, the distribution of true and false news is not equal in reality. To mirror reality, we thought there would be a lot more true news than fake news. To evaluate the effectiveness of each model, we ran two tests on balanced and unbalanced datasets. To imitate the real-world situation, we first produced a false dataset and then reduced it to a tenth of its original size. From there, we created the unbalanced dataset. We are using the "Accuracy" measure, which offers a thorough evaluation of the classification situation and is thought to be the most appropriate representation for this purpose, to gauge the effectiveness of our models developed for the fake news detection task. The accuracy scores for the three models we tested on both balanced and imbalanced datasets are presented in Tables 6.1 and 6.2. Our findings reveal that in the context of detecting fake news, training on datasets which are imbalanced produces slightly better results compared to training on datasets which are balanced.

Model	Accuracy
LSTM	0.9697
Bi-LSTM	0.9689
CNN-Bi-LSTM	0.9748

Table 6.1: Performance on the Balanced Dataset(Using GloVe Embedding)

Model	Accuracy
LSTM	0.9798
Bi-LSTM	0.9790
CNN-Bi-LSTM	0.9823

Table 6.2: Performance on the Imbalanced Dataset(Using GloVe Embedding)

Model	Accuracy
LSTM	0.9636
Bi-LSTM	0.9672
CNN-Bi-LSTM	0.9680

Table 6.3: Performance on the Balanced Dataset(Using FastText Embedding)

Model	Accuracy
LSTM	0.9754
Bi-LSTM	0.9708
CNN-Bi-LSTM	0.9740

Table 6.4: Performance on the Imbalanced Dataset(Using FastText Embedding)

Chapter 7

Conclusion

The objective of our project was to assess the ability of deep learning models, namely LSTM, BiLSTM, and CNN-BiLSTM, to identify fake news by utilizing the ISOT Fake News dataset. To prepare the text data for input into these models, we employed preprocessing and word embedding techniques, which involved cleaning the text, tokenizing it into words, and mapping each word to a high-dimensional vector space. Deep learning models have gained popularity in detecting fake news due to their ability to process large volumes of data and uncover patterns that are not always apparent to humans. The LSTM, BiLSTM, and CNN-BiLSTM models are particularly suitable for processing sequential data like news articles. Our experimental results showed that all three models performed well in detecting fake news, with CNN-BiLSTM outperforming the basic LSTM model due to its ability to combine CNN and LSTM strengths and capture both local and global dependencies in text data. Our findings suggest that deep learning models have potential for improving national security and politics by detecting fake news effectively.

Bibliography

[1] Goldani, M. H., Momtazi, S., Safabakhsh, R. (2021). Detecting fake news with capsule neural networks. Applied Soft Computing, 101, 106991

[2] Hakak, S., Alazab, M., Khan, S., Gadekallu, T. R., Maddikunta, P. K. R., Khan, W. Z. (2021). An ensemble machine learning approach through effective feature extraction to classify fake news. Future Generation Computer Systems, 117, 47-58.

[3] Kula, S., Choraś, M., Kozik, R., Ksieniewicz, P., Woźniak, M. (2020, June). Sentiment analysis for fake news detection by means of neural networks. In International Conference on Computational Science (pp. 653-666). Springer, Cham.

[4] Jamal Abdul Nasir, Osama Subhani Khan , Iraklis Varlamis (2021, April).Fake news detection: A hybrid CNN-RNN based deep learning approach.In International Journal of Information Management Data Insights.Volume 1 issue 1.

[5] I. Kadek Sastrawan, I.P.A. Bayupati, Dewa Made Sri Arsa (2022, sep). Detection of fake news using deep learning CNN–RNN based methods. In ICT express. Volume 8, Issue 3,(pg no. 396-408).

[6] Kaliyar, R. K., Singh, N. (2019, July). Misinformation Detection on Online Social Media-A Survey. In 2019 10th International Conference on

Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[7] Hitesh Narayan Soneji Sughosh Sudhanvan (2021, March). Fake News Detection Using Convolutional Neural Networks and Random Forest—A Hybrid Approach.In Cybernetics, Cognition and Machine Learning Applications pp 349–359.

[8] Lazer, D. M., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., ... Zittrain, J. L. (2018). The science of fake news. Science, 359(6380), 1094-1096.

[9] Ahmed, H., Traore, I., Saad, S. (2018). Detecting opinion spams and fake news using text classification. Security and Privacy, 1(1), e9.

[10] Reis, J. C., Correia, A., Murai, F., Veloso, A., Benevenuto, F. (2019). Supervised learning for fake news detection. IEEE Intelligent Systems, 34(2), 76-81.

[11] https://www.uvic.ca/engineering/ece/isot/datasets/fake-news/index.php

[12] https://scikitlearn.org/stable/modules/generated/sklearn.utils.shuffle.html

[13] https://www.geeksforgeeks.org/pre-trained-word-embedding-using-glove-in-nlp-models/