Ministry of Science and Higher Education of the Republic of Kazakhstan

SDU University

Sumeyra Betul Polat

Enhancing Fake News Classification Through DL Models: Encoder-Decoder Architecture with BLSTM for Improved Accuracy

THESIS

Presented in Partial Fulfilment for the

Degree of Master of Technical Science in Computer Science

(degree code: 7M06102)

Department of Computer Science

Faculty of Engineering and Natural Sciences

Supervisor: **Selcuk Cankurt**

Kaskelen, June 2024

SDU University
Faculty of Engineering and Natural Sciences
Department of Computer Science

Dean of Faculty of Engineering and Natural Sciences

Assistant Professor, PhD Akhmedov Ramis

____

« ______» __________________________________ 2024

Topic of the thesis

Enhancing Fake News Classification Through DL Models: Encoder-Decoder Architecture with BLSTM for Improved Accuracy

---

Thesis submitted as part of the requirements for the award of the MSc in "7M06102 - Computer Science" , SDU University

Head of Department - Zhanar Mukash

Academic Supervisor - Selcuk Cankurt

Master student - Sumeyra Betul Polat

Kaskelen, 2024

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Sumeyra Betul Polat

June 2024

Acknowledgements

I want to sincerely thank Prof. Selcuk Cankurt, my supervisor, for all of his help and patience during this journey. His encouragement and support have been crucial in helping to shape this thesis. I am appreciative of his constant faith in my ability and the chance to develop under his guidance.

Dedication

This thesis is dedicated to:

My parents, whose constant support, direction, and encouragement have been the driving force behind my academic path, are the recipients of my thesis dedication. I would also like to express my gratitude to those who have assisted and provided insightful comments on this work. This research would not have been possible without your help, and I am incredibly appreciative of you in my life.

Abstract

In today's world, the subject of fake news is crucial. It discusses how social media and traditional media spread false stories or misinformation. This effort aims to use deep learning models to increase the accuracy of fake news classification. We investigate the utilization of bidirectional long short-term memory (BLSTM) networks and attention processes in conjunction with encoder-decoder design to boost accuracy.

Абстракт

В современном мире тема фейковых новостей крайне важна. Обсуждается, как социальные медиа и традиционные медиа распространяют ложные истории или дезинформацию. Это усилие направлено на использование моделей глубокого обучения для повышения точности классификации фейковых новостей. Мы исследуем специально использование двунаправленных сетей долгой краткосрочной памяти (BLSTM) и процессов внимания в сочетании с дизайном кодировщика-декодера для повышения точности.

Аңдатпа

Қазіргі әлемде жалған жаңалықтар мәселесі өте маңызды. Бұл мәселе әлеуметтік медиа мен дәстүрлі бұқаралық ақпарат құралдарының жалған оқиғаларды немесе жалған ақпаратты қалай таратылатыны жаппай талқыланады. Осындай шаралардың мақсаты - жалған жаңалықтарды жіктеудің дәлдігін арттыру үшін терең оқыту модельдерін қолдану. Біз дәлдікті арттыру үшін екі бағытты ұзақ қысқа мерзімді жад желілерін (BLSTM) және зейін процестерін кодтаушы-декодер дизайнымен бірге пайдалануды арнайы зерттеп жатырмыз.

Abbreviations

LSTM -Long Short-Term Memory
BLSTM -Bidirectional Long Short-Term Memory
DL -Deep Learning
NLP -Natural Language Processing
ED -Encoder Decoder
Att -Attention.
GloVe - Global Vectors for Word Representation
GlobalMaxPool1D - Global Max Pooling 1D
Seq2Seq - Sequence to Sequence
Tf-idf - Term Frequency-Inverse Document Frequency
CBOW - Continuous Bag of Words
Skip-gram - A type of Word2Vec model that predicts context words based on a central word
TP - True Positive
FP - False Positive
TN - True Negative
FN - False Negative
GRU - Gated Recurrent Unit
UML - Unsupervised Machine Learning

Table of Contents

Chapter 1

Introduction

1.1 Introduction

An increasing number of people in our digital age are concerned about fake news, often known as misleading or purposefully incorrect information which is presented as news to trick the reader into believing it to be true. Anticipating whether a statement, tweet, or news item has been purposefully produced to mislead is a good way to characterize the challenge of spotting unreal news [1].

In addition to offering speedy news delivery compared to more conventional sources like newspapers and television, social media makes information easy to obtain and reasonably priced. Several people would rather use social media for news than traditional media sources because of these benefits. Social media is thus gradually replacing traditional news sources. Despite social media's benefits, it's crucial to remember that news content on these sites isn't necessarily reliable [2].

The fake news ecosystem depends on fundamental psychological characteristics in people, including a) Confirmation Bias, or an impulse to believe anything that confirms existing opinions, and b) Naive Realism, or the belief that one's perspective is the only one that matters and that others are wrong. In recent years, concerns about fake news have risen globally, and these worries continue to grow as individuals use social media more and more as their primary source of news [3].

There has been a major increase in research dedicated to detecting false news in recent years, resulting in a significant amount of scholarly work in this area. Traditionally, studies focused on detecting fake news have mostly concentrated on two approaches (a) the categorization of false material within a single modality, whether text or images, and (b) the investigation of multi-modal methods that take both textual and visual components into account. Figure 1 in the report provides a full understanding the utilization of deep learning [4].

Fake news can be identified using a variety of information manipulation tactics. Deceptive content can take numerous forms: it might be entirely made up with the objective of misleading consumers, or it can be dishonest content that adjusts facts to solve specific issues. Furthermore, phoney news might mirror trustworthy sources while being incorrect. Other deceptive elements of fake news content include the use of improved parts, such as misleading headlines and visuals that do not match the actual content or the framing of misinformation within its context alongside real aspects within a false setting[5].

Detecting fake news automatically involves examining the accuracy of claims in news content, which is a contemporary and crucial issue within natural language processing. This significance arises from the substantial societal and political influence of conventional news outlets and social media platforms. Fake news exposure can result in disillusionment, disconnection, and doubt toward certain political figures. Identifying fake news represents a significant use case for natural language processing (NLP), with far-reaching implications for how technology can aid in verifying the truthfulness of statements while also enlightening the general public [6].

The encoder-decoder model, with integrated networks with bidirectional long short-term memory (BLSTM) and an Attention mechanism, is employed to distinguish news [7]. The encoder processes textual data, capturing context using BLSTMs, while the Attention mechanism helps the model focus on critical information. This paper is designed to enhance accuracy in separating fake news from genuine news, providing an effective approach to news seperation. In the process of recognising fake news, the encoder-decoder model is like a smart system that reads and understands news articles. It helps decide whether a news piece is real or fake. The "encoder" part figures out the important details in the text, considering how different parts of the text relate to each other. Then, the "decoder" part interprets this information to decide whether the news is true or not. By using this model, along with BLSTM networks and an Attention mechanism, we can do a better job of telling real news from fake news pay particular focus to the tricky language and details often found in misleading news stories.

Fake news is a term that's used alongside other words like misinformation, disinformation, rumor, and propaganda. These words are often used in similar ways. The key difference, however, is why people are sharing this information. For example, a rumor is news that might be true or false, but we're not sure when we first hear it. We might find out later if it's true or not [8].

1.2 Relevance

The term "fake news" describes misleading information that is promoted as reliable news. 64% of US citizens, according to the New York Times, stated that they have always been confused by "made-up news" [9]. The development of unclear The identification of credible news sources from information in articles from daily access media, such as blogs and online newspapers, has become more difficult. Consequently, there is an increased need for computational techniques that can offer insight into the legitimacy of online material. Following the 2016 electoral process in the United States "fake news" gained significance. Fake news was significantly biased in favor of Donald Trump throughout the election and was widely disseminated. Since then, the study of fake news identification has gained popularity and is of interest to all.

We have chosen this topic as our goal since it is a socially pertinent realm of research and because figuring out how to identify true news would enable everyone to receive knowledge appropriately. We attempt to identify the best machine learning and deep learning model using accessible resources (datasets, literature publications) that offer high accuracy in predicting and classifying the given combination set of real and fake news datasets into factual and false.

1.3 Problem Statement

In recent years, the emergence of fake news on social media and other internet platforms has raised serious concerns. Social networking is becoming increasingly popular and the broad use of the internet have made it easier for people to transmit incorrect information, which has increased the spread of fake news. False information can have detrimental effects on people or organizations, as well as cause panic and other major issues. Fake news can also undermine the authority of news sources and erode public confidence in the media.

Our paper's problem statement is to evaluate efficient deep-learning models for categorizing false news. The deep learning algorithms used in our approach include "Encoder-Decoder Architectures Keras Embedding, Encoder Decoder With Pretrained Embeddings Glove", "Encoder Decoder With Pretrained Embeddings Glove GlobalMaxPool1D() and LSTM", "Encoder Decoder With Pretrained Embeddings Glove BLSTM", and "Encoder Decoder With Pretrained Embeddings Glove GlobalMaxPool1D() and BLSTM".

1.4 Objectives

Our paper aims to determine whether these models can reliably distinguish between authentic and fraudulent news and display positive outcomes in the former. Deep learning techniques and approaches for natural language processing are used in tandem to achieve this.

1.5 Research questions

Detecting fake news is a significant challenge in computer-based language analysis. The extensive utilization of social media platforms has not only made a wealth of information easily accessible but has also accelerated the rapid dissemination of false news stories. Consequently, the impact of fake news is growing, sometimes even spilling over into real-world situations, posing risks to public safety. Given the enormous volume of content available on the internet, the automation of fake news detection becomes a vital and practical problem within the domain of the processing of natural language. It is a valuable tool for online content providers, as it significantly reduces the human resources and time required to identify and combat the spread of fake news, safeguarding the accuracy and integrity of online information. Given such obstacles and the urgent need for practical answers, the following questions explore particular techniques and their possible influence on enhancing the detection of false information.

Question 1: How does the integration of an Encoder-Decoder architecture with Bidirectional Long Short-Term Memory (BLSTM) networks and an Attention mechanism impact the accuracy of fake news classification in comparison to conventional deep learning models?

Contribution 1: The main question we're investigating is whether our new model, which combines the Encoder-Decoder structure with BLSTM and Attention, is better at spotting fake news compared to regular models. We want to see if it does a better job and provides proof that it's a more effective way to identify fake news. This will help improve the methods we use to detect fake news.

Question 2: Can our method effectively handle the tricky language and context often seen in fake news, making it better at identifying fake news with higher accuracy?

Contribution 2: Our method is effective at handling the tricky language and context in fake news. Fake news often uses tricky language and context, which makes it hard to tell if it's true or fake. But our method is built to identify these evaluations, so we can find fake news more accurately. This not only helps in the fight against false information but also paves the way for better ways to spot fake news, making information more reliable in our digital world. This helps us do a better job of spotting fake news with higher accuracy.

As we're aware, computers lack the inherent ability to comprehend human languages, and since users predominantly communicate in human language, a specific area of artificial intelligence, referred to as Natural Language Processing (NLP), has emerged to bridge this gap. NLP empowers machines to grasp and interpret human languages effectively. Consequently, preprocessing the data is imperative before employing machine learning (ML) or deep learning (DL) techniques with NLP. This involves the segmentation of the dataset into training and testing subsets (Step 2). Following this preprocessing stage, various deep learning models such as Bidirectional LSTM (BLSTM), Encoder-Decoder, and Attention models are constructed using the training dataset (Step 4), and their performance is evaluated using the testing dataset (Step 5). Ultimately, in Step 6, news articles are classified as either true or false.

This study examines how to tell apart real and fake news using models like Encoder-Decoder, BLSTM, and Attention. With so much misinformation around, we need better ways to spot the truth. The research dives into the world of fake news identification to find a strong solution. By using advanced Encoder-Decoder models along with smart word representations, the study employs classification methods to achieve higher accuracy and determine the superior model. By using classification techniques, the study aims to enhance performance in detecting fake news and identify the most effective model for this purpose.

Our research employs classification techniques to differentiate between genuine and misleading content. People often try to spot fake news by saying it's either true or fake (binary classification). But this can be tricky because sometimes news is a mix of truth and lies. To handle this, we use two categories. For example, class 0 and class 1. We employ these characteristics to train models across diverse datasets. Our approach underwent evaluation using three openly accessible datasets, and its effectiveness was validated through the assessment of four widely employed performance metrics: accuracy, precision, recall, and F-1 score.

Chapter 2

Literature Review

2.1 Literature Review

[9] The rise of fake news detection has become critical, impacting democratic processes and public opinion. To tackle this, various machine learning models have been developed, including the "Automatic Fake News Classification Through Self-Attention (ACT)" framework, which leverages helpful articles to assess claim credibility. It uses self-attention on bidirectional LSTM networks to represent articles, yielding significant performance improvements in real-world datasets such as Snopes and PolitiFact. These models rely on publicly available datasets, including Snopes and PolitiFact, where claims are categorized as true or false, and external evidence may be retrieved to enhance context. These datasets are vital for training and evaluating fake news detection models, helping distinguish between deceptive and genuine claims in today's digital information landscape. The research assesses model performance using various metrics like accuracy, AUC, precision, recall, and F1 score. Control claims are correctly identified in 492 out of 1140 cases, while 2747 out of 3201 false claims are accurately detected. Using confidence scores can help determine whether to accept or reject system decisions and may enhance operational efficiency. Additionally, more supporting articles for a claim improve model accuracy, increasing from 66.67% to 85.69% for claims with sufficient supporting material.

[10] Identifying false information on social media is crucial due to the spread of negative and dishonest content. This research introduces the AC-BLSTM model, a way to find fake news and sort it into categories. It performs better than other models, with an accuracy of 35.1%. Prior studies have used deep learning techniques like CNN, LSTM, and BLSTM, which achieved accuracy between 21.5% and 44.87%. These approaches outperformed traditional methods, and using attention mechanisms improved accuracy. While the AC-BLSTM is a step forward, there's more to learn in identifying fake news, like analyzing different types of data and understanding how fake news spreads using graph networks.

[11] This research introduces a novel approach to detecting fake news by combining text and image features, addressing the challenge of multimedia content on social media. The model utilizes Twitter and Weibo datasets. The Twitter dataset, comprising 14,483 labeled tweets, and the Weibo dataset, a Chinese dataset with 7,531 samples in the training set and 1,996 in the test set. These datasets enable a comprehensive evaluation of

the proposed fake news detection model. On the Twitter dataset, the model attained an accuracy of 85%, surpassing existing models by approximately 10 points. Meanwhile, the Weibo dataset exhibited an overall accuracy of 90%, showcasing balanced F1 scores between fake and real classes. This work contributes to early fake news detection in the era of multimedia social media content.

[12] The research conducted within the framework of the Fake News Challenge 1 (FNC-1) focuses on the task of fake news identification through stance detection, classifying headlines into "discusses," "agrees," "disagrees," or "unrelated" categories concerning a given article. The paper leverages a combination of SVM with TF-IDF cosine similarity features for related/unrelated pairing and different neural network topologies built on Long Short Term Memory Models (LSTMs) for fine-grained classification. Notable models include the Bag of Words (BOW) Multilayer Perceptron (MLP), LSTMs, Stance Detection with Bidirectional Conditional Encoding, and the Bilateral Multi-Perspective Matching Model. To tackle the issue of imbalanced classes, the study splits the problem into two subproblems: detecting related/unrelated pairs and then classifying the related ones as "agree," "disagree," or "discuss." The SVM demonstrates strong performance in the first subproblem, and neural networks, particularly the Bidirectional Conditionally Encoded LSTMs with Bidirectional Global Attention, excel in the second subproblem. These models collectively achieved a score of .8658 on the FNC-1 challenge, outperforming earlier reported results.

[13] The study introduces Word2vec weighted by TF-IDF method for document representation and classification, and the results in Table III shed light on its efficacy in comparison with the TF-IDF and Word2vec methods. The experimental phase is carried out using a dataset comprising 5,581 news items, including 2,878 real news and 2,703 fake news, demonstrating the algorithm's performance. Word2vec with TF-IDF weighting exhibits superior performance, as evidenced by the precision, recall, and F1 scores. In particular, the algorithm achieves remarkable precision rates of 95.16% and 94.65% for real and fake news, respectively, resulting in an impressive average F1 score of 92.02%. This outperforms the traditional TF-IDF and Word2vec approaches, which yield lower precision and F1 scores. Compared to TF-IDF, Word2vec Weighted By TF-IDF demonstrates a significant improvement in both precision and F1 score for fake news classification, underscoring its effectiveness in identifying fake news. These results underscore the potential of the Word2vec Weighted By TF-IDF algorithm as a valuable tool in the realm of fake news detection, making a substantial contribution to the field of document representation and classification.

[14] In the context of text classification, this study introduces the novel BLSTM-2DPooling framework, which incorporates bidirectional long short-term memory networks (BLSTM) with 2D max pooling operations for feature extraction. Additionally, 2D convolution (BLSTM-2DCNN) is integrated to enhance feature representation. The research's contributions include pioneering the application of 2D convolution and 2D max pooling in natural language processing (NLP) tasks and introducing two effective models, BLSTM-2DPooling and BLSTM-2DCNN. These models are evaluated across various text classification tasks and datasets, demonstrating superior performance. Particularly, BLSTM-2DCNN excels in four out of six tasks, achieving impressive test accuracies, including 52.4% and 89.5% based on categorization tasks. Comparative analysis with existing NLP models, such as ReNN, CNN, RNTN, and others, showcases the effectiveness of 2D operations for sentence and document modeling. The results underscore the potential of these approaches for advancing NLP and text classification tasks.

[15] The paper presents a systematic evaluation of the proposed Category-controlled Encoder-Decoder (CED) model. Specifically, the PHEME dataset—a collection of Twitter chat threads connected to significant events—is used to focus on binary classification tasks about true and fake news. Separated into subsets for testing, validation, and training, this dataset contains 1,123 examples of both fake and true news. The CED model exhibits higher performance across a range of criteria when thoroughly compared against baseline techniques. CED outperforms other well-known techniques like DT-Rank (0.562), DTC (0.581), and GAN-ED (0.781) in terms of accuracy, achieving 0.803. With values of 0.795, 0.814, and 0.788, respectively, the Precision, Recall, and F1-score metrics highlight the model's effectiveness even further.

[16] The authors highlight a serious problem of false news spreading on social media and offer Unsupervised false News Detection based on the Autoencoder (UFNDA) technique as a possible solution. Remarkably, the authors used linear SVM as the classifier and TF-IDF as the feature extraction technique to reach a 92% accuracy rate in detecting fake news. Comparably, a method based on a CNN model for text and images produced a 92.2% accuracy rate for identifying fake news. A Deep Convolutional Neural Network model reached an amazing accuracy of 98.36%, whereas the use of Bi-LSTM produced a detection accuracy of 93.1%. In addition, the study showed that pre-trained deep learning models (BERT, XLNet, and RoBERTa) outperformed conventional machine learning models in terms of accuracy, reaching up to 98%. These findings highlight the variety of approaches and developments in the sector and highlight how crucial accurate detection techniques are in the fight against the problems posed by false information on social media.

[17] This review of the literature underlines the significant impact of false news in the age of technology and shows the transition from conventional machine learning to deep learning methods for improved detection accuracy. The impact of false information on significant events most particularly, the 2016 U.S. presidential election highlights how crucial it is to address this global problem. With a particular focus on benchmark datasets such as Fake News, Twitter15, and Liar, the review sorts through a variety of datasets according to their modality, labels, and size. A discussion is held regarding the significance of appropriately dividing datasets for training, validation, and test sets, typical ratios like as 60:20:20, 70:30, and 80:20 are addressed. Some accuracy results, like the remarkable 93.50% highlight the usefulness of models and datasets in advancing the field of false news detection research. This comprehensive summary is an invaluable tool for comprehending the state of the field now and guiding investigations in the future.

[18] The advancement of information and communication technology has dramatically increased internet access, affecting the dynamics of information consumption worldwide. However, this transition has created a crucial difficulty in the widespread propagation of fake news, as evidenced by events such as the United States political campaign. Recognizing the potential threat posed by false news to societal stability, this work investigates the use of deep learning approaches, such as a BERT-based architecture, to detect fake news based only on textual content. The study uses a dataset of 20,015 labeled news items, with a focus on attributes such as titles and contents, and defends the dataset's selection based on its previous successful application in similar circumstances. Additionally, the authors use the Fake News Corpus to test and refine their algorithms. The paper discusses several neural network architectures, with a focus on a Long Short-Term Memory (LSTM)-based model, and offers impressive results, such as an LSTM model obtaining 91% accuracy and a Convolutional model achieving 93.7%. These findings highlight the

intriguing potential of deep learning in efficiently addressing the persistent problem of fake news on the internet.

[19] A lot of study has been conducted and published on the topic of detecting false news in other languages in recent years. Studies and models now in circulation recommend using standard machine learning algorithms for the task of identifying bogus news in foreign languages [20]. Additionally, some writers have presented BERT models [21], citing a 98.90% accuracy rate for FakeBert. The literature offers a wide range of deep neural network models [22]. Some studies [23], combine convolutional neural networks and recurrent neural networks (RNN), achieving an accuracy of 82%. Additionally, several studies utilizing neural learning systems to detect fake news have been published [24] [25] underscoring the intricacy of this field.

[26] Three datasets were compared in a separate study: the LIAR datasets, the dataset compiled from actual and fake news on the Internet, and the dataset made up of both. In that order, the research compared several traditional machine learning models, including SVM, LR, DT, AdaBoost (AB), Naive Bayes (NB), and KNN. It did this by employing bigram, lexical, sentiment, and unigram techniques, term frequency, and inverse document frequency (TF-IDF). Some CNN models were also used to train the model, including NN, CNN, LSTM, BLSTM, hierarchical attention network (HAN), convolutional HAN, and character-level CLSTM. Glove embedding and character embedding were also used. As a consequence, the best result with an accuracy of around 0.94 was obtained by NB using n-gram (biagram TF-IDF) features.

[27] Moreover, the research showed that the CNN model performed better than the LIAR dataset. Nonetheless, the research [28] demonstrated that, across all datasets, the CNN model is the second-best. With 0.60 accuracy and 0.59 F1-score, the NB model performed the best on the LIAR dataset. The dataset Char-level C-LSTM demonstrated the best performance for both fake and true news, with 0.95 accuracy and 0.95 F1-score. With the merged corpus dataset, LSTM-based models performed best, with both Bi-LSTM and C-LSTM achieving an accuracy of 0.95 and an F1-score of 0.95.

[29] A new n-gram model was created by Ahmed et al. with a focus on fraudulent comments and fake news to detect false information automatically. Six techniques were used to classify machine learning, and TFIDF was used to extract features: decision tree (DT), KNN, SVM, linear support vector machine (LSVM), and KNN two methods: LR and stochastic gradient descent (SGD). Compared to conventional methods, the outcomes of their experiments show great promise. 90% accuracy for LSVM is the best outcome.

[30] Various news story features, such as the source and social media posts, were given by Reis et al. To automatically identify false news, they present a novel set of features and evaluate the predictive abilities of KNN, NB, RF, SVM, and XGBoost (XGB). XGB is the best model, with an accuracy of 86%.

[31] The authors describe an ensemble classification model that achieves higher accuracy than the state-of-the-art techniques for identifying false news. The suggested method gathers salient characteristics from datasets including fake news, which are subsequently recognized using an ensemble model comprising three widely used machine learning models: decision tree, random forest, and extra tree classifier. They obtained testing and training accuracies of 44.15% and 99.8% on the Liar dataset, respectively. With the ISOT dataset, they were also able to achieve 100% training and testing accuracy.

[32] The authors automate analysis and increase accuracy by combining ML with NLP techniques like tokenization and TF-IDF. NumPy, Pandas, matplotlib, and Scikit-learn are used to implement classic machine learning and neural network models. NLTK To-

kenization preprocesses data from scraped articles and Kaggle. TF-IDF for machine learning and Word2vec for neural networks are used in training, along with CNN, LSTM, logistic regression, naive Bayes, SVM, and random forests. Strong performance is demonstrated by the accuracy rates of 88.87% for Logistic Regression, 85.81% for Naive Bayes, 89.18% for SVM, and 99.70% for Random Forest. These results demonstrate how well NLP and ML may be integrated for precise and effective analysis.

[33] The urgent problem of false news propagation in the digital age is addressed in the work "Classification of Fake News by Fine-tuning Deep Bidirectional Transformers based Language Model" by Akshay Aggarwal et al. It's getting harder and harder to tell the difference between fake and real news because of how quickly information is absorbed and distributed. Because manually verifying news stories takes a lot of time and labor, automated computational tools are required. The authors of this study suggest using Natural Language Processing to categorize news stories as authentic or fraudulent. On the Bidirectional Encoder Representations from Transformers (BERT) language model, they use transfer learning for this purpose. With only limited text pre-processing, the study shows the resilience of refined BERT models, as evidenced by its astounding accuracy of 97.021% on the NewsFN dataset.

Chapter 3

Methodology

3.1 Methodology

The theories that were applied in this work are covered in this section. The subsection contains a list of the comprehensive connected works. This prior work focused on fake news classification and introduced the Bidirectional Long Short-Term Memory (BLSTM) model, among other models. Furthermore, we'll start by giving a quick overview of the basic idea behind Bidirectional Long Short-Term Memory (BLSTM). After that, we'll explain the Encoder-Decoder and Attention model.

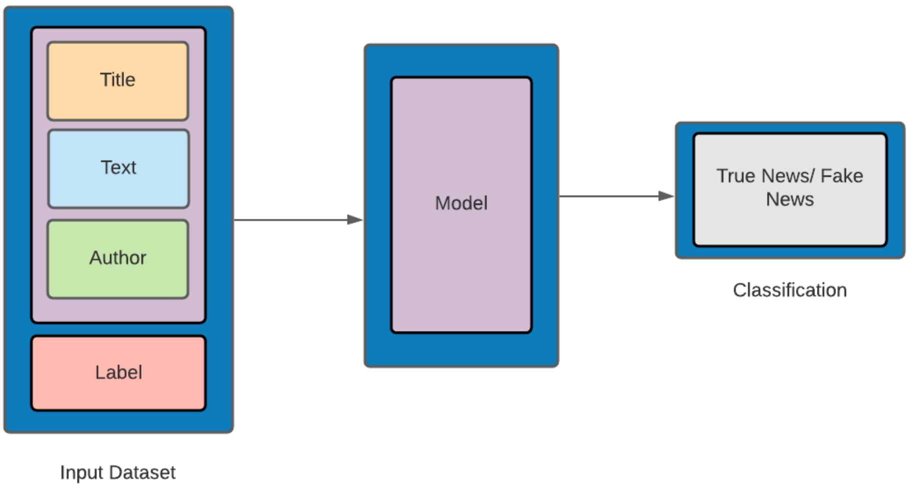


Figure 3.1 - An outline of the approach [34]

3.1.1 Encoder Decoder with Keras Embeddings

The Keras machine learning package provides a word embedding layer that is already constructed for neural network models. Typically, input values from a neural network model are encoded integer values. Consequently, Keras provides the Tokenizer API to get the input ready to feed the neural network. The adaptable layer, the Embedding layer, employs randomly initialized weights to learn how to embed each word in the training dataset. [35].

The encoder and decoder are the two main parts of the design. After processing the input text, the encoder extracts the most important semantic properties and encodes them into a vector representation with a defined length. At the input layer, Keras Embeddings transform unprocessed text inputs into dense vector representations that capture the context-specific meaning of vocabulary terms. After that, the encoded representations are transmitted to the decoder, which produces output sequences that match categorization labels like "true" or "fake."

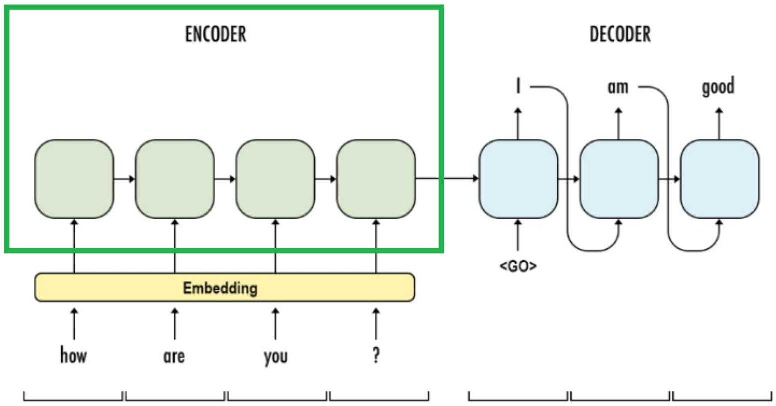


Figure 3.2 - Simple encoder-decoder with Keras Embedding [36]

The layers and parameters of the architecture must be configured within the Keras deep learning framework to use the Encoder-Decoder model with Keras Embeddings. Keras Embeddings are utilized at the input layer to convert immediate word representations into dense vector representations. The encoder is usually composed of layers of a recurrent neural network (RNN), such as the Gated Recurrent Unit (GRU) or Long Short-Term Memory (LSTM), which processes the input sequence and records sequential dependencies. Similarly, the Decoder uses the encoded representations to create output sequences using RNN layers.

3.1.2 Encoder Decoder With Pretrained Embeddings Glove

Three situations exist in which pre-trained embeddings can be used to train an encoder-decoder architecture to classify fake news: on the encoder side, or both the encoding and decoding components. These embeddings, which capture semantic links discovered from extensive text collections, provide words with their first dense vector representations.

In this case, the values of the pre-trained embeddings stay stable because they are not modified during training. The model successfully classifies news articles as true or false, with labels 0 and 1, respectively, by utilizing the rich semantic information included in the pre-trained embeddings. By concentrating on tuning other model parameters for better classification performance, our technique guarantees that the model gains from the semantic richness of the pre-trained embeddings.

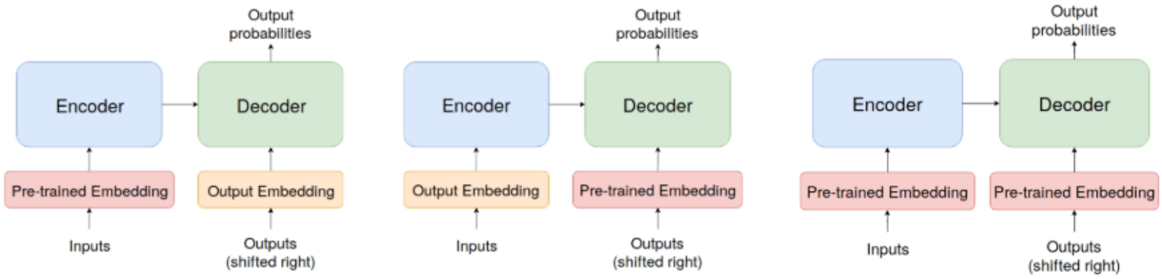


Figure 3.3 - Word embeddings in the encoder that have been pre-trained (left). Word embeddings that have been pre-trained in the decoder (Center). The encoder and decoder both use pre-trained word embeddings (right). [37]

3.1.3 Encoder Decoder With Pretrained Embeddings Glove GlobalMaxPool1D() and LSTM

In this study, we use pretrained GloVe embeddings in our Encoder-Decoder architecture, which incorporates GlobalMaxPool1D and LSTM layers, for the classification of fake information. Transfer learning is the use of a model was trained on a huge set of general data to improve performance on a specific task [38].

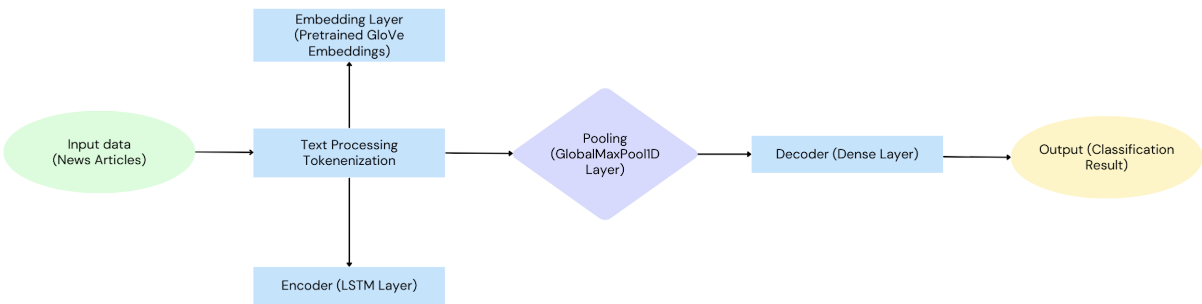


Figure 3.4 - With pre-trained embeddings Glove GlobalMaxPool1D() and LSTM, a simple architecture for encoder/decoder

GloVe embeddings are trained on large text corpora to capture semantic associations between words, and these are then utilized to initialize our model's embedding layer. This method allows us to apply the deep semantic understanding embedded in these embeddings to the specific task of categorizing news stories as true or untrue.

We still use pre-trained embeddings, which is a type of transfer learning. The Encoder processes the input sequences through an LSTM layer to capture sequential dependencies and then utilizes a GlobalMaxPool1D layer to extract the most important features. The dimensionality reduction of the LSTM output is simplified by the GlobalMaxPool1D layer, which maintains the most important features [39].

3.1.4 BLSTM and Attention Mechanism

Attention is a method for identifying and giving words significance the detection of false information. The most important aspect of an actual fake news detection system is the terminology and words used in the news, each of which has an important meaning and relevance. Therefore, the attention mechanism which is the idea of assigning weights based on relevant data and location embedding must be integrated into the encoder model.

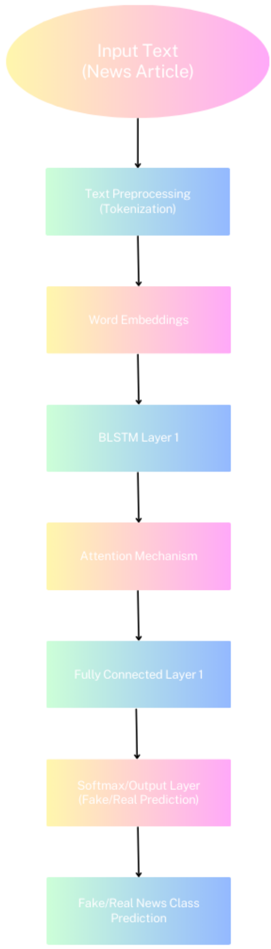


Figure 3.5 - BLSTM architecture

The LSTM network [40], has proven to be highly effective for tasks involving sequential learning, such as named entity tagging [41] and machine translation.

In essence, text classification involves working with information in a step-by-step manner. However, the feature sequences acquired simultaneously from the convolutional layer lack sequential information. BLSTM specializes in handling sequences and can additionally extract contextual information extracted from the feature sequences produced by the convolutional layer. The purpose of BLSTM is to construct word vector representations at the text level. Given that each word contributes differently to the context's sentiment, it's common to assign varying weights to words as a solution. The attention mechanism serves to allocate distinct weights to words, enhancing our comprehension of the overall sentiment in the text. Consequently, the combination of BLSTM and the attention mechanism can enhance the efficiency of text classification.

BLSTM acquires word annotations by aggregating data from both the forward and backward directions, effectively encompassing contextual information. Within BLSTM, there's the forward $\overrightarrow{\text{LSTM}}$, responsible for processing feature sequences from Lc_1 to Lc_{100} in the forward direction, and the backward $\overleftarrow{\text{LSTM}}$, which processes the sequences in reverse, from Lc_{100} to Lc_1. To formalize this, the outputs of BLSTM can be described as follows:

$$hf = \text{LSTM}(L_{cn}), \quad n \in [1, 100] \tag{3.1}$$

$$\overleftarrow{h_f} = \text{LSTM}(Lc_n), \quad n \in [100, 1] \tag{3.2}$$

The combination of the forward hidden state $\overrightarrow{h_f}$ and the backward hidden state $\overleftarrow{h_b}$ yields an annotation for a particular feature sequence Lc_n. These states perform word encoding and condense all of the information related to Lc_n found in the entire text.

The mechanism that controls attention is to reduce the impact of non-keywords on text sentiment by concentrating on the characteristics of keywords. It is accomplished using a function known as softmax in conjunction with a fully integrated layer.

A single-layer neural network is used as an underlying representation of h_f to process the word annotation u_f to produce u_f. The following is the mathematical equation for u_f:

$$\overrightarrow{u_f} = \tanh(wh_f + b) \tag{3.3}$$

of this case, $\tanh(\cdot)$ is the hyperbolic tangent function, and w and b stand for the weight and bias of the neuron, respectively. The model calculates each word's relevance by comparing its similarity with a word-level context vector $\overrightarrow{u_f}$. Afterward, the normalized weight $\overrightarrow{u_f}$ for every word is obtained by utilizing the softmax function. $\overrightarrow{u_f}$ can be expressed as follows:

$$\overrightarrow{a_f} = \frac{P(\exp(\overrightarrow{u_f} * \overrightarrow{v_f}))}{\sum_{i=1}^{M}(\exp(\overrightarrow{u_f} * \overrightarrow{v_f}))} \tag{3.4}$$

In this case, M stands for the word count in the text, and $\exp(\cdot)$ represents the number of words in the text, and $\exp(\cdot)$ denotes the exponential function. The word-specific context vector $\overrightarrow{v_f}$ can be interpreted as a high-level representation of the informative words within the text. It is initially randomized and undergoes joint learning during the training process.

Following this, weighted aggregation of the forward word annotations, determined by the assigned weight $\overrightarrow{a_f}$, is calculated to obtain the forward context representation F_c. F_c constitutes a portion of the output from the attention mechanism, which can be represented as:

$$\overrightarrow{F_c} = \left(\overrightarrow{a_f} * \overrightarrow{h_f}\right) \tag{3.5}$$

Similar to the calculation of a_f, the value of a_b can be determined using the backward hidden state h_b. Much like F_c, the reverse context representation H_c is a component of the output produced by the attention mechanism, and this can be formulated as:

$$Hc = \sum(\overleftarrow{ab} * \overleftarrow{hb}) \tag{3.6}$$

3.1.5 Encoder-Decoder Model

Considering a source sentence X, denoted as $(x_1, x_2, \ldots, x_l)$, and a corresponding target sentence Y, simplified as $(y_1, y_2, \ldots, y_{l'})$, where x_i and y_i belong to the same vocabulary, and l and l' represent the lengths of the respective sentences. Our objective is to construct a neural network that can capture the conditional probability $p(Y|X)$ and subsequently train this model to optimize and maximize the probability.

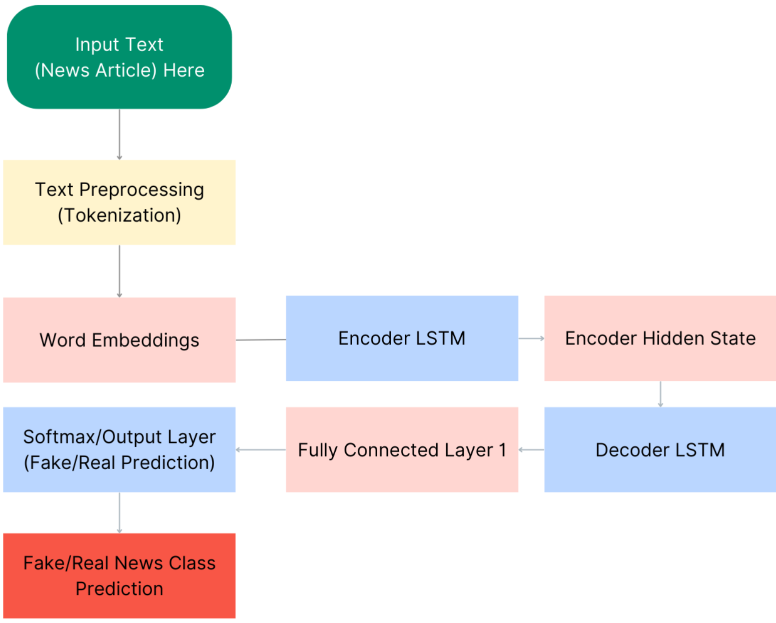


Figure 3.6 - Encoder-Decoder Model

LSTM Encoder-Decoder model is presented in Figure 1 [42]. This model employs a one-hot representation of words in the input sequence at the input layer, which is then converted into a 300-dimensional domain subsequent embedding layer. The incorporation of an embedding layer can lead to a significant improvement in performance, particularly when dealing with a large vocabulary. The word embeddings are then passed through two LSTM layers, resulting in a vector representation of the input sequence subsequent to processing all the words. Finally, it decodes this vector to generate the output sequence by passing it through two LSTM layers followed by an output embedding layer.

As an illustrative example, let us consider the input sentence "Man with high intelligence" as a complex sentence and the simplified sentence "a very smart man" as the output. We represent this pair of sentences as a pair of word indices (arbitrary indices are used here) [43]:

$$[\text{Man, with, high, intelligence}] \Rightarrow [\text{A, very, smart, man}]$$
$$[15, 27, 6, 18] \Rightarrow [1, 2, 12, 15]$$

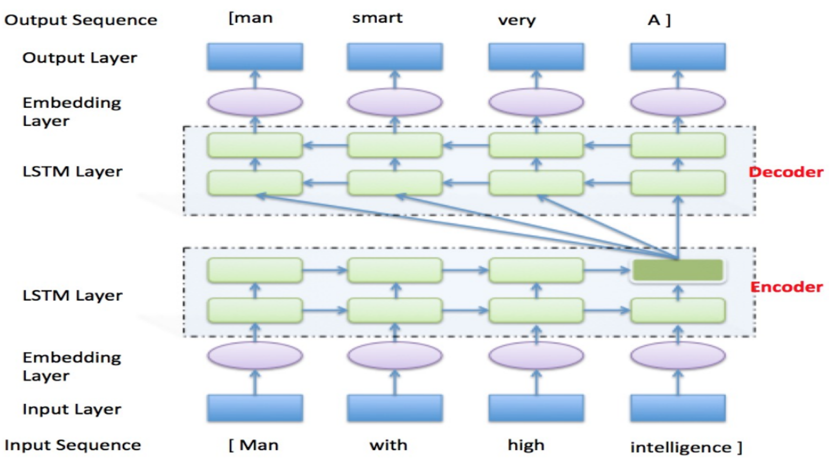


Figure 3.7 - LSTM Encoder-Decoder Model

3.1.6 Word2vec

Word2vec is now extensively utilized in both commercial and scientific contexts. The underlying theory is that, given the idea that a word's meaning can be deduced from its company, it is possible to estimate a word based on its context. Word2vec can generate a distributed word representation using two different architectures: Skip-gram (which attempts to forecast the context words based on a central word) and continuous bag-of-words, or CBOW (which predicts the present word by analyzing a grouping of neighboring context words).

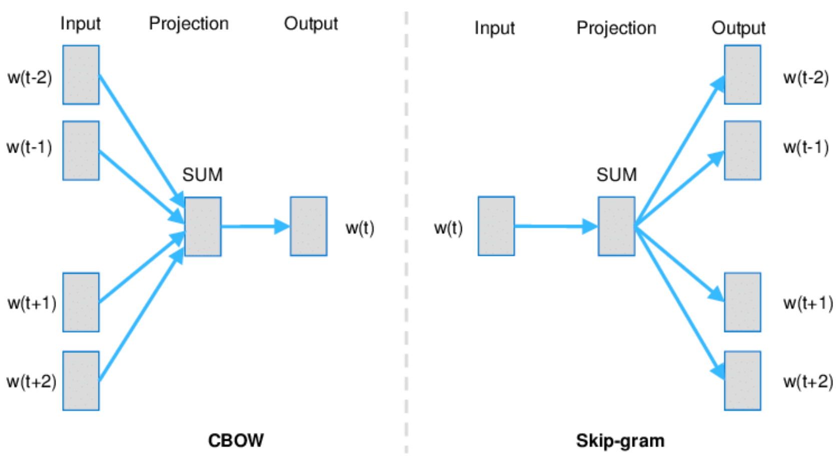


Figure 3.8 - Skip-gram and CBOW are both implemented by the Word2vec algorithm. The fundamental notion of the two training models is that a word may be used to forecast its context (Skip-gram) or the other way around, where a current word can be predicted using its context (CBOW). The exercise is repeated throughout the corpus, word by word [44].

3.1.7 Tokenization

For data preprocessing, tokenization should be done first. When a phrase, paragraph, sentence, or even an entire text is tokenized, it efficiently divides it into smaller components, like a single word, prefix, or term. The term tokens refers to each subgroup. Processing a natural language requires tokenization to first identify the words that make up a string of letters. For NLP (text data), tokenization is therefore a necessary first step. That's important since a quick translation of the document's context could be achieved by interpreting the language's vocabulary [45].

3.1.8 Word Embedding

Compared to real news, fake news spreads six times faster on Twitter and reaches a much larger audience. And of all the phony news out there, false political news spreads more quickly, widely, deeply, and farther than any other kind.

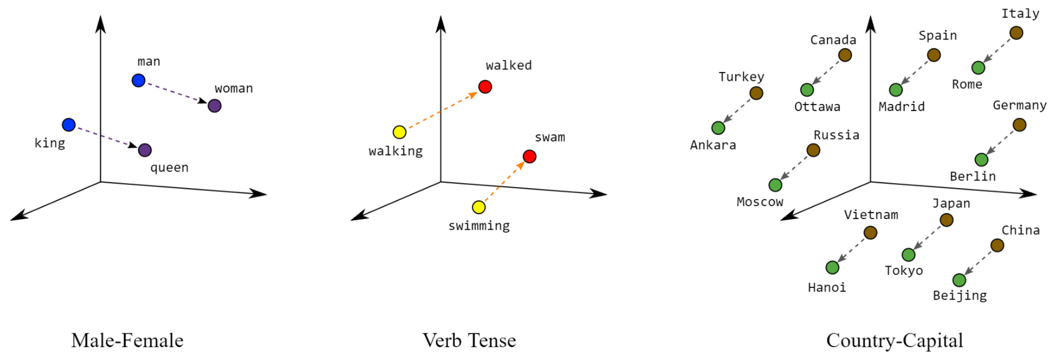


Figure 3.9 - Semantics in a suitable embedding can be encoded via position in the vector space. The following real embedding visualizations, for instance, illustrate geometrical relationships that represent semantic relations such as the relationship between a nation and its capital. Your machine learning system will have the possibility to identify patterns in this kind of meaningful area that could aid in the learning work.

Features and phrases present in fake paperwork and publications can be utilized to identify patterns. It is possible to distinguish fake news by looking at textual elements such as author, context, and writing style.

Nonetheless, it must first transform the input data into a numerical representation that the model we're creating can understand before we can apply any machine learning technique to text. This is the initial point of concern. As the number of features increases, our model's inaccuracy increases due to the Curse of Dimensionality and other negative consequences caused by high-dimensional representations of linguistic information produced by traditional methods such as Term frequency-inverse document frequency (TF-IDF).

Using word and phrase embeddings to obtain low-dimensional, spread representations of the data is one approach to overcoming these issues. Words and phrases are represented as embeddings in multidimensional spaces, where words and sentences with comparable meanings have similar embeddings. It indicates that a vector of real numbers is used to map each word or sentence to its corresponding word or sentence [46].

Text categorization and neural networks require input text in a vector or matrix format for processing. Word vectors are vector representations of text, with each word having its vector. These word vectors are known as word embeddings. Word embeddings are trained on a large corpus, typically language or domain-specific, to capture statistical relationships between words in the corpus. Using publically available pre-trained word embeddings is a more viable option than training them.

3.1.9 LSTM based on feature embedding

Specialized recurrent neural networks that can learn long-term reliance are called long short-term memory networks. Figure 3.8 depicts its structure. In our previous research, we employed a special model known as "LSTM based on feature embedding." LSTM, which stands for Long Short-Term Memory, is a clever technique that helps computers understand data sequences, like text. In this model, we skillfully embedded features to assist the computer in recognizing patterns within the data. This process enabled us to gain a deeper understanding and classify the news articles. We explored different variations of this model, including Static Embeddings and Dynamic Embeddings, as well as other models like LSTM and B-LSTM, to gauge their effectiveness in identifying fake news.

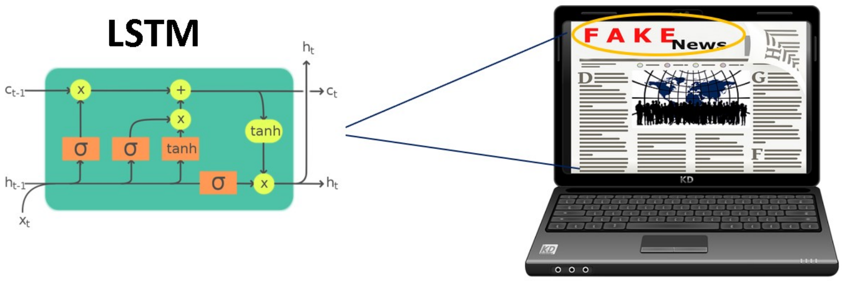


Figure 3.10 - Data preprocessing pipeline

To do this, a model known as the Long Short-Term Memory (LSTM) model can selectively allow information to move through the door mechanism and add or remove data from cells. The forget gate, input gate, and output gate make up an LSTM. The input gate determines what data should be updated to the cell state after the forget gate determines which data should be removed from the cell state. The cell state can be modified once these two points have been established. The output gate ultimately selects the network's final output [47].

The Python library Keras provided the embedding layer and dense layers that we utilized to train the LSTM model. The embedding layer that preprocessed each word was employed before the LSTM layer. Thus, the layers in our model are as follows: the dense layer, the LSTM layer, and the embedding layer [48].

3.1.10 BLSTM

The hidden vector is computed in two directions by the two LSTM layers that make up the BLSTM model: from front to back and from rear to front. When these two layers are combined, the BLSTM's output is produced. The usual feed-forward element that other neural networks rely on is not a feature of this neural network, making it distinct. In a BLSTM, there is no connectivity between the inner nodes. Using the contextual information found in co-occurrence probabilities between words, the GloVe approach is useful for assessing the surrounding context of each word. This method is akin to computer vision's transfer learning, which uses a trained model to enhance subsequent models.

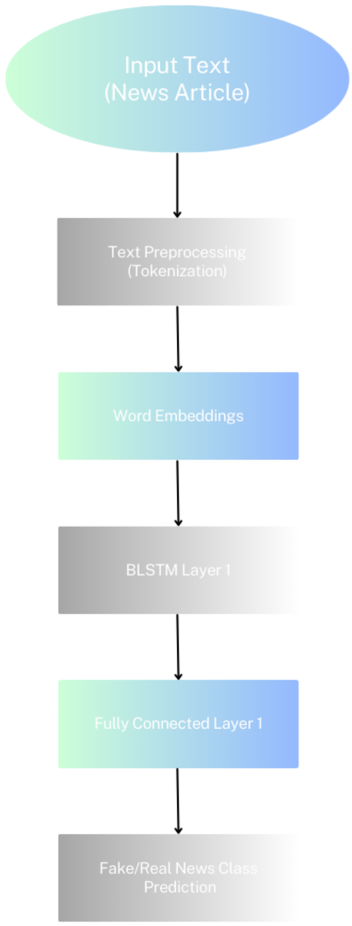


Figure 3.11 - Simple BLSTM Architecture

Each layer's inner nodes of BLSTM are not connected. By using co-occurrence probabilities among words, which carry contextual information, the GloVe approach is useful for assessing each word's context. This method is comparable to using a pre-trained model to enhance subsequent models in computer vision, or transfer learning. In NLP, fine-tuning is a standard procedure when implementing a strategy on a novel job [49].

The error that can propagate backward in time and to deeper levels of a deep network is preserved by LSTM. An obvious method for text classification and large-scale text sequence prediction is bi-directional processing. A Bi-Directional LSTM network steps across the input sequence simultaneously in both directions [50]. The methodology is summarized in Table 3.1.

3.1.11 Model Evaluation

Four metrics have been employed for the evaluation of the results. These metrics are based on the number of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) in the binary classifier predictions:

Additionally, Table 3.1 provides an overview of the technique used in this thesis. It contains text categorization methods and deep learning models.

7. **Accuracy**: $\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.7)$

8. **Recall**: $\text{Recall} = \frac{TP}{TP+FN} \quad (3.8)$

9. **Precision**: $\text{Precision} = \frac{TP}{TP+FP} \quad (3.9)$

10. **F1 Score**: $\text{F1} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \quad (3.10)$

Table 3.1 - Summary of Methodology

Model/Section	**Methodology**
BLSTM and Attention Mechanism	BLSTM (Bidirectional Long Short-Term Memory) are used to construct word vector representations at the text level. The attention mechanism assigns varying weights to words based on relevant data and location embedding, enhancing the efficiency of text classification.
Encoder-Decoder Model	LSTM Encoder-Decoder model is employed to capture conditional probability $p(Y\|X)$ by processing input sequences through LSTM layers and generating output sequences.
Tokenization	Tokenization divides text into smaller components, such as words or terms, for natural language processing tasks.
Word Embedding	Word embeddings are used to transform text data into numerical representations, overcoming high-dimensional representations produced by traditional techniques like TF-IDF.
LSTM is based on feature embedding	LSTM model embeds features to recognize patterns within data sequences, aiding in classification tasks.
BLSTM	BLSTM (Bidirectional Long Short-Term Memory) computes hidden vectors in both directions to capture sequential dependencies and contextual information. GloVe embeddings are utilized for context assessment.
Model Evaluation	Four metrics - Accuracy, Recall, Precision, and F1 Score - are employed to evaluate the performance of binary classifier predictions.

Chapter 4

Dataset Description

4.1 Dataset Description

In our study, we used three sets of data to explore different aspects from different reliable news outlets identified as untrustworthy by Politifact.com, and the true news articles were sourced from trustworthy outlets like CNN, BBC, Reuters, the New York Times, and other reputable publications, whereas the false news articles were gathered from online news platforms. We started using the "Fake News Dataset," which had 8019 statements. The second dataset, collected from Kaggle.com, included 20,799 articles. The final dataset, "ISOT Fake News," contains 44898 K of confirmed articles and false news from several trustworthy news sources, as well as those that Politifact.com has labeled untrustworthy. The study made use of a dataset with true or false binary labels and took into account the title, text, subject, and date that the articles were posted. The data was preprocessed to exclude features like emotions, symbols, pictographs, maps, flags, punctuations, HTML syntaxes, URL data, and emojis to assess the performance of standard models. Figure 4.1 displays a chart that illustrates the quantity of false and true news samples.

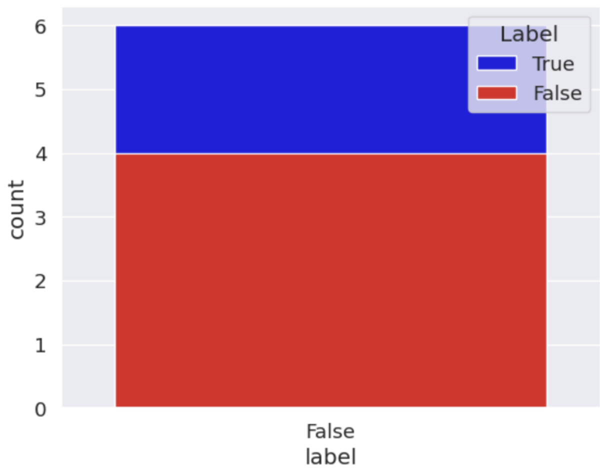


Figure 4.1 - The amount of true and false articles on news

4.1.1 Pre-processing of Data

A thorough preparation workflow was used in this study to get textual data ready for further analysis. The raw text was first cleaned via several phases to guarantee consistency and quality. Emojis, URLs, punctuation, and HTML syntax were all meticulously eliminated using special methods that made use of regular expressions.

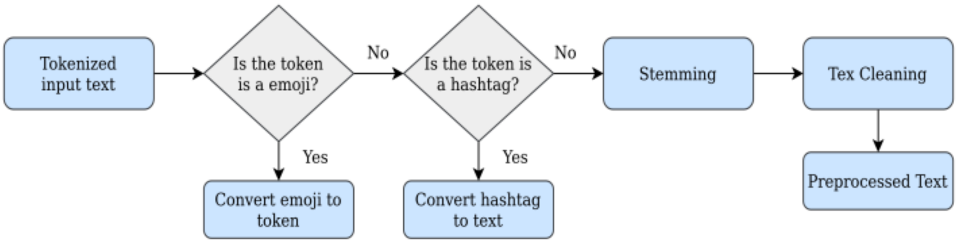


Figure 4.2 - Data preprocessing pipeline

Additionally, lemmatization was applied to the corpus to standardize word forms and improve the coherence of textual representations. The processed data was then put through TF-IDF vectorization, which is an essential step in converting the text into numerical vectors so that quantitative analysis may be performed. Together, these preprocessing procedures helped reduce noise and extraneous data, which made it easier to conduct more precise and perceptive analysis throughout the study's latter stages.

4.1.2 Data Investigation

Finding patterns and insights from both fake and real news can be accomplished by analyzing and visualizing the data during the data exploration stage. We used the Python packages Matplotlib [51] and Seaborn [52] to plot different kinds of charts.

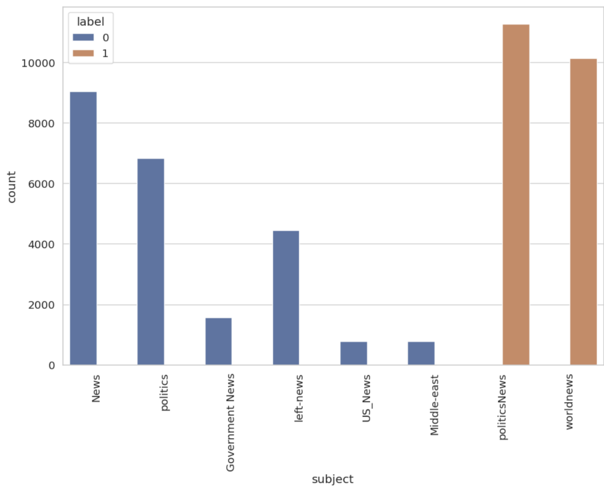


Figure 4.3 - Real and fake news counts by subject.

For the real and false news samples, we first created word clouds. Every key keyword in the datasets was displayed in word clouds. The actual news keywords for the terms in the title are displayed in Figure 4.3a's word clouds, together with remarks about Trump, the US, the Prime Minister, the State, the White House, percentage, election, stating, plan, and China. Figure 4.4b displays the word cloud sample for fake news, which consists of comments from the selections' titles, including Trump, people, video, American, president, country, vote, Republican, and President Obama.

(a)

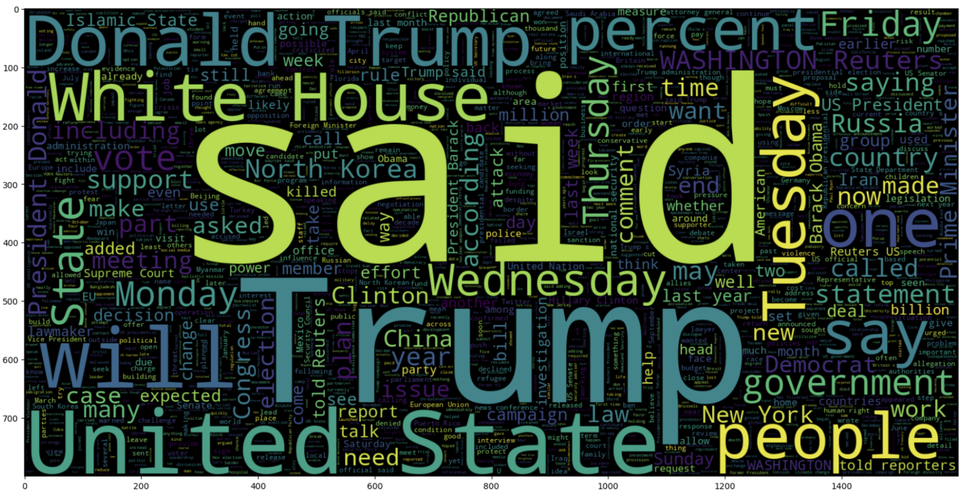


Figure 4.4 - Word clouds made up of title terms from both the false and true news samples (a) word cloud of frequently appearing title words from authentic news samples.

In Figure 4.5, we displayed a bar chart with numbers of different news topics. After the dataset was cleaned, the highest counts were found in political and foreign news.

(b)

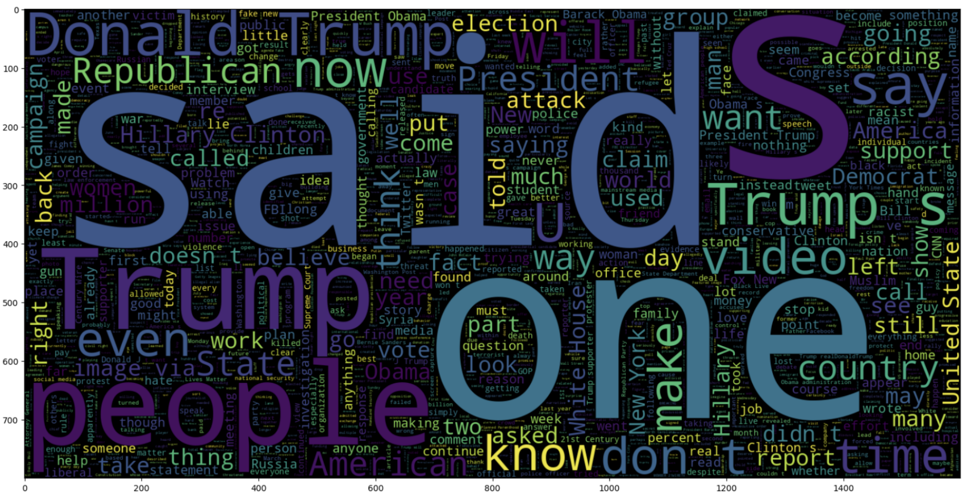


Figure 4.5 - (b) word cloud displaying commonly appearing title words from false news samples.

Figure 4.6a examines the characters in real and fake news content, while Figure 4.7b investigates the words in real and fake news text. The longest character in real news is

15000, whereas it is approximately 24000 in fake news. The longest word in text count is around 15900, but in fake news, it is over 15000.

(a)

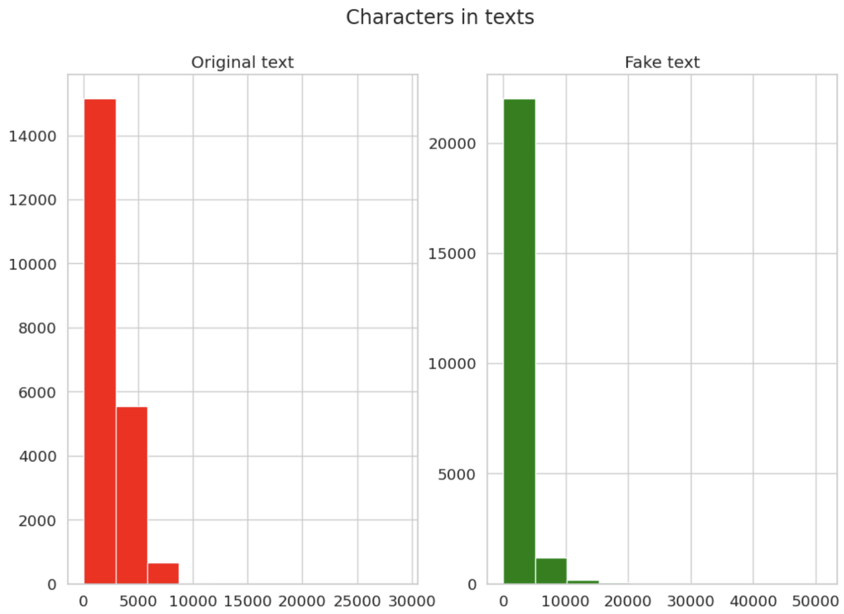


Figure 4.6 - Characters in texts that include counts of fake and real news; (a) text character counts for fake news and real news.

(b)

Figure 4.7 - Words in texts that include counts of fake and real news; (b) words in texts count for fake news and real news.

Figure 4.8 displays the average word length count for both real and fake news units.

As basic units of text analysis, unigrams contain single words that are independent of their context. By examining the contextual relationship between adjacent keywords,

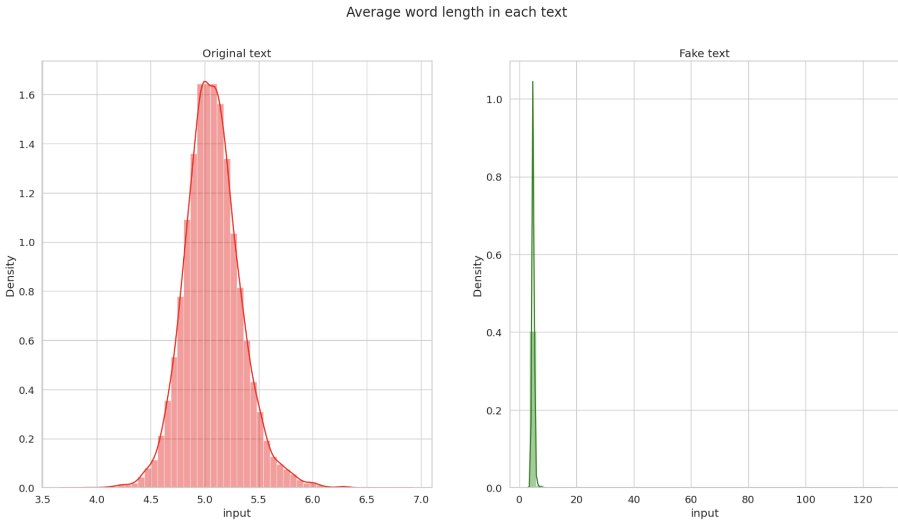


Figure 4.8 - The average word length in each text

bigrams, on the other hand, provide a more nuanced perspective by representing consecutive pairings of words.

Using the Matplotlib [53] and Seaborn [54] Python tools, a Unigram analysis was performed shown in Figure 4.9. To examine the most popular unigrams in the dataset, the visualization was created.

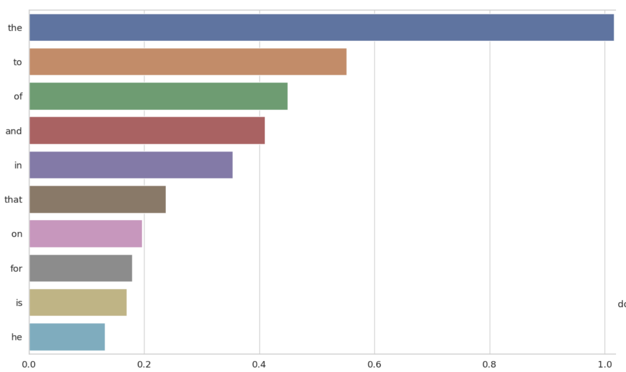


Figure 4.9 - Top Unigrams Bar Plot

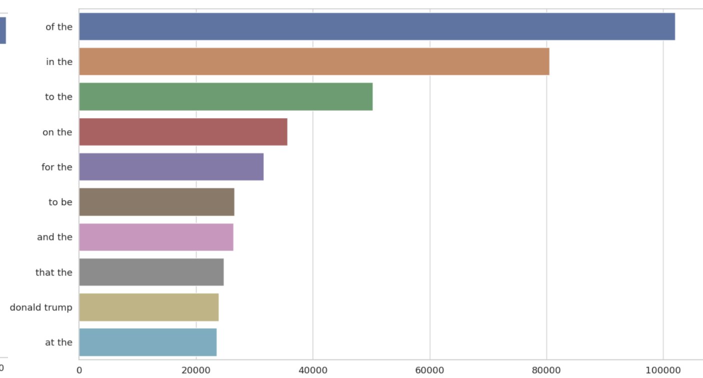


Figure 4.10 - Top Biagram Analysis

Additionally, word pairs adjacent to one another in the dataset were examined by the biagram analysis. We created a graph to display the top 10 pairs that frequently emerged like in the unigram analysis. Each pair's frequency of appearance is displayed on the graph by presenting the pairs on one line and the count on the other, as shown in Figure 4.10.

4.1.3 Divide and pre-process the dataset

The datasets are read as Pandas DataFrame objects, and sci-kit-learn's LabelEncoder is used to encode the class labels. Next, each dataset is divided into 80–20 percent test and training subsets. All datasets have undergone pre-processing to convert the raw texts into the proper format for each model, which is necessary to validate the classification models. For this task, a Python script has been specifically written. Using the re python package for regular expressions, texts are first cleaned of IP and URL addresses. Next, the exam is divided into terms and sentences. After removing English stopwords, the NLTK package is used to stem the remaining phrases.

The first Kaggle dataset includes both "Fake" and "True" news stories from 2016 to 2017. The collection contains two sorts of articles: fake and true news. This dataset was compiled from real-world sources; the accurate articles were gathered by Reuters.com (news website). The fake news came from several sources. Fake news items were gathered from unreliable websites highlighted by Politifact (a fact-checking organization in the United States) and Wikipedia.

The second dataset "ISOT Fake News" contains various articles on various themes; however, most articles are about politics and world events. There are two CSV files in the dataset. It included a vast collection of 44,898 articles, comprising both real and fake news. This dataset was carefully displayed, with 21,417 articles considered truthful and 23,481 containing false information. The dataset includes the text, article title, kind, and publication date in addition to the entire body of each article. To evaluate the performance of traditional models utilized with pre-trained embedding models BLSTM, along with a basic LSTM model for a neural network. More than 12,600 items from reuters.com are contained in the first file, "True.csv." The second file, "Fake.csv," has around 12,600 items from various sources of fake news outlets. The following details are included in every article, the kind, date of publication, article title, and text. The majority of the stories we collected were from 2016 to 2017, to match the false news data gathered for kaggle.com.

Table 4.1 presents the distribution according to categories and subjects.

Table 4.1 - An analysis of the ISOT dataset

Category/Type	**Number of Articles**	**Articles Size**
Real News	21,417	–
Fake News	23,481	–
World News	–	10,145
Politics News	–	11,272
Government News	–	1,570
Middle East	–	778
US News	–	783
Left News	–	4,459
Politics	–	6,841
News	–	9,050

We conducted our tests using the third dataset, which is worldwide available in Kaggle called the Kaggle FakeNews dataset, which contains 20800 rows $\times$ 5 columns. Additional author-related information is included in the dataset. Politifact.com editors choose which comments to label based on the majority of stories.

Chapter 5

Results

5.1 Results

5.1.1 Overview of Models

The research includes five separate models: "Encoder-Decoder Architectures Keras Embedding", "Encoder Decoder using pre-trained GloVe embeddings", "Encoder Decoder With Pretrained Embeddings Glove GlobalMaxPool1D() and LSTM", "Encoder Decoder With Pretrained Embeddings Glove BLSTM", "Encoder Decoder With Pretrained Embeddings Glove GlobalMaxPool1D() and BLSTM"

5.1.1.1 Model Performance Evaluation

The desired outcomes were achieved by splitting the data into an 80% training set and a 20% testing set. Eighty percent of the data were used for training, and the remaining twenty percent were used for testing.

5.1.1.2 Encoder-Decoder Models

When examining the accuracy, an overview of the encoder-decoder models has been created through dataset comparison. With Glove Embeddings, the models in Dataset 1 have excellent accuracy levels and sophisticated architectures like GlobalMaxPool1D() with LSTM or BLSTM exhibit the greatest results.

Table 5.1 - Encoder Decoder with Keras Embeddings

Dataset	**Accuracy**	**Precision**	**Recall**	**F-score**
Dataset 1	0.96	0.96	0.96	0.96
Dataset 2	0.88	0.88	0.88	0.88
Dataset 3	1.0	1.0	1.0	1.0

Figure 5.1 provides a visual comparison of encoder-decoder models for 3 datasets. With the highest score of 0.98 for Dataset 1 and the score of 0.97 for Dataset 2, the Encoder-Decoder with pre-trained GloVe and GlobalMaxPool1D and LSTM consistently outper-

formed the evaluation. With scores of 1.0, every model on Dataset 3 performed flawlessly, suggesting that this dataset was simpler to categorize. On Dataset 2, simpler models with accuracy dropping to 0.88 and 0.77, respectively, were the Encoder-Decoder with Keras Embeddings and pretrained GloVe.

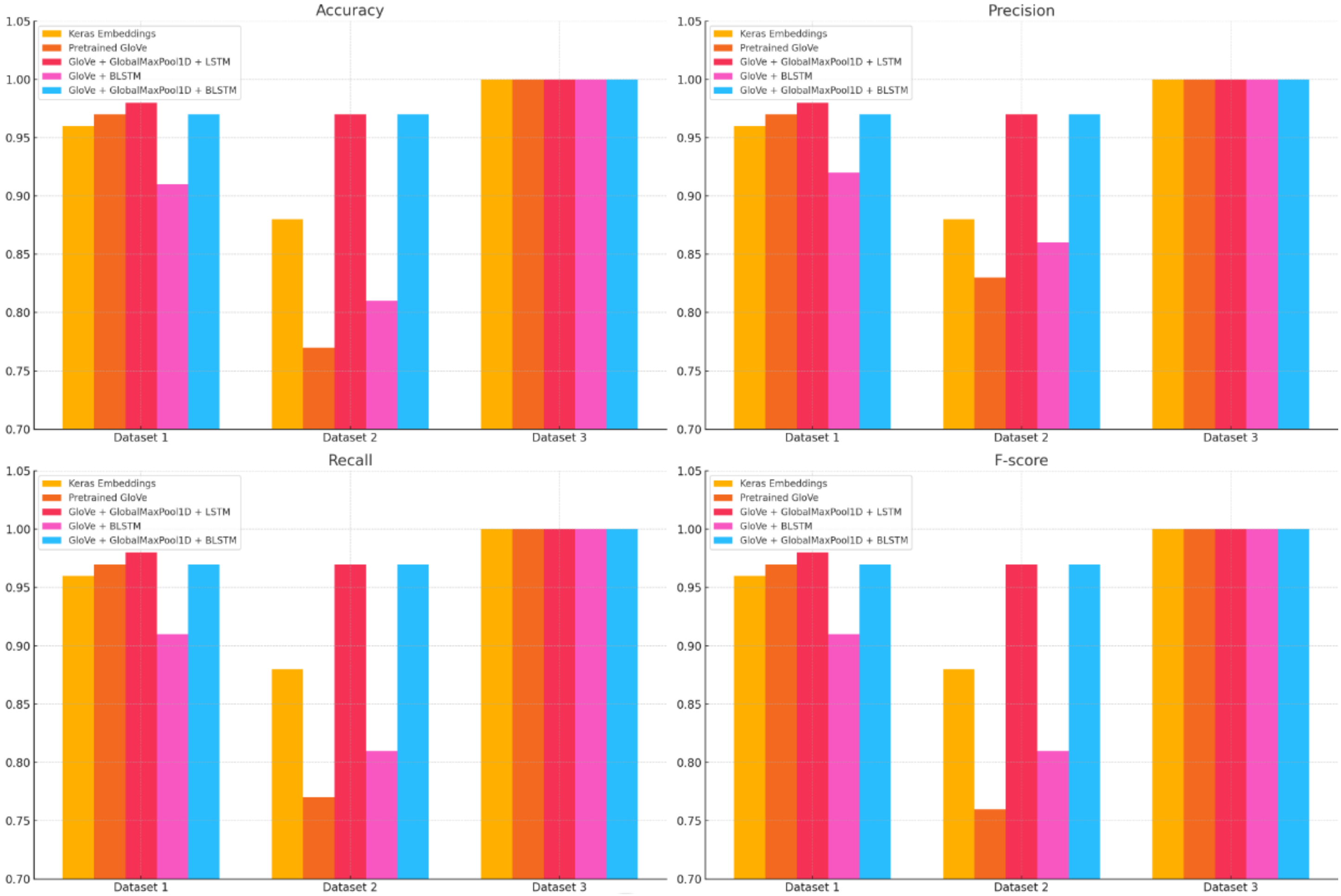


Figure 5.1 - Comparison of Encoder Decoder Models

In particular, models that combined LSTM with GlobalMaxPool1D() produced an accuracy of 0.98%. The precision of the conversion from Keras embeddings to Glove embeddings is seen in Dataset 2, where it decreases from 0.96% to 0.77%. Dataset 3 demonstrates that all models have flawless accuracy ratings, demonstrating their capacity to capture patterns in the dataset properly. GlobalMaxPool1D() and LSTM or BLSTM models reach an accuracy of 0.97%.

Table 5.2 - Encoder Decoder With Pretrained Embeddings Glove GlobalMaxPool1D() and LSTM

Dataset	Accuracy	Precision	Recall	F-score
Dataset 1	0.98	0.98	0.98	0.98
Dataset 2	0.97	0.97	0.97	0.97
Dataset 3	1.0	1.0	1.0	1.0

The performance of our suggested Encoder Decoder models using Keras Embeddings and Pretrained Embeddings is thoroughly analyzed and compared to other benchmark models in this section. Metrics including accuracy, precision, recall, and F-score are considered throughout the evaluation, which is carried out across several datasets.

Table 5.3 - Encoder Decoder With Pretrained Embeddings Glove GlobalMaxPool1D() and BLSTM

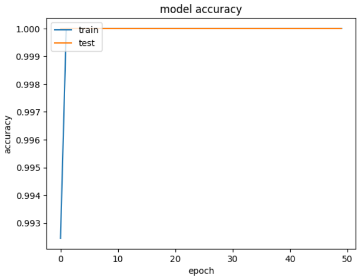
Dataset	**Accuracy**	**Precision**	**Recall**	**F-score**
Dataset 1	0.97	0.97	0.97	0.97
Dataset 2	0.97	0.97	0.97	0.97
Dataset 3	1.0	1.0	1.0	1.0

Table 5.4 - Encoder Decoder With Pretrained Embeddings Glove

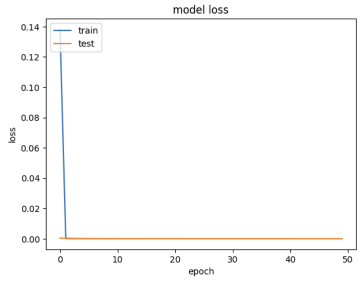
Dataset	**Accuracy**	**Precision**	**Recall**	**F-score**
Dataset 1	0.97	0.97	0.97	0.97
Dataset 2	0.77	0.83	0.77	0.76
Dataset 3	1.0	1.0	1.0	1.0

Table 5.5 - Encoder Decoder With Pretrained Embeddings Glove BLSTM

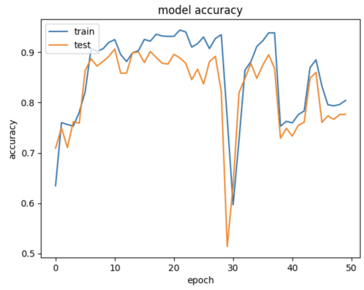
Dataset	**Accuracy**	**Precision**	**Recall**	**F-score**
Dataset 1	0.91	0.92	0.91	0.91
Dataset 2	0.81	0.86	0.81	0.81
Dataset 3	1.0	1.0	1.0	1.0



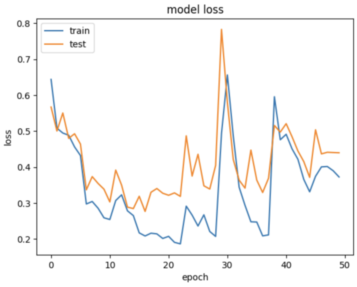
(a) Accuracy (Dataset 3, GloVe)



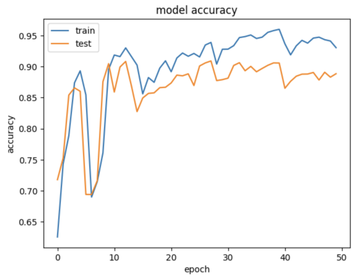
(b) Loss (Dataset 3, GloVe)



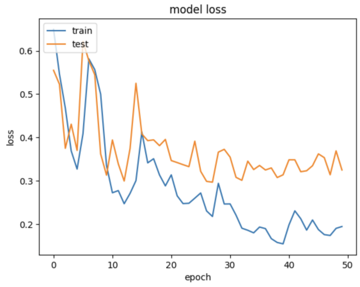
(c) Accuracy (Dataset 2, GloVe)



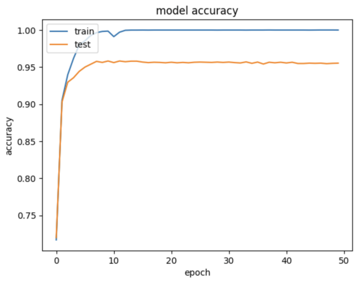
(d) Loss (Dataset 2, GloVe)



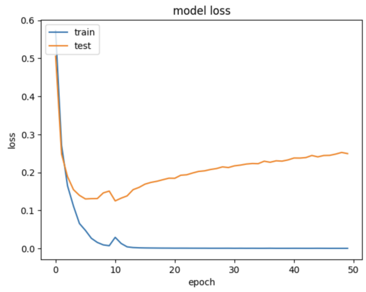
(e) Accuracy (Dataset 1, GloVe)



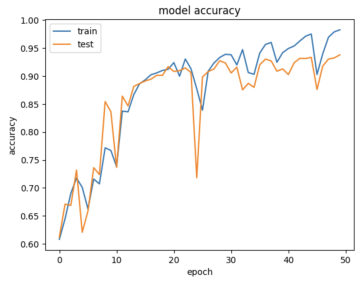
(f) Loss (Dataset 1, GloVe)



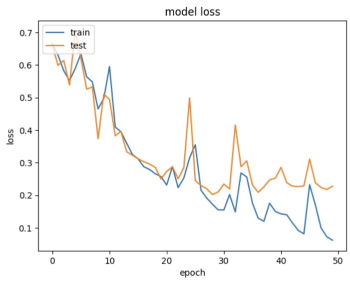
(g) Accuracy (Dataset 2, GloVe, BLSTM)



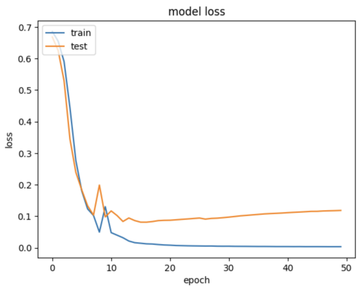
(h) Loss (Dataset 2, GloVe, BLSTM)



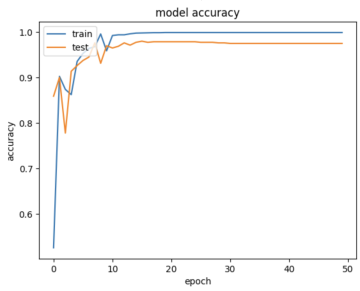
(i) Accuracy (Dataset 2, GloVe, BLSTM)



(j) Loss (Dataset 2, GloVe, BLSTM)

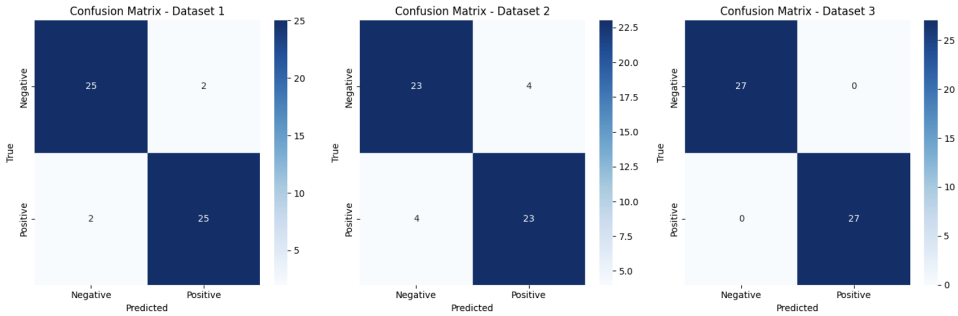


(k) Loss (Dataset 1, GloVe, LSTM)

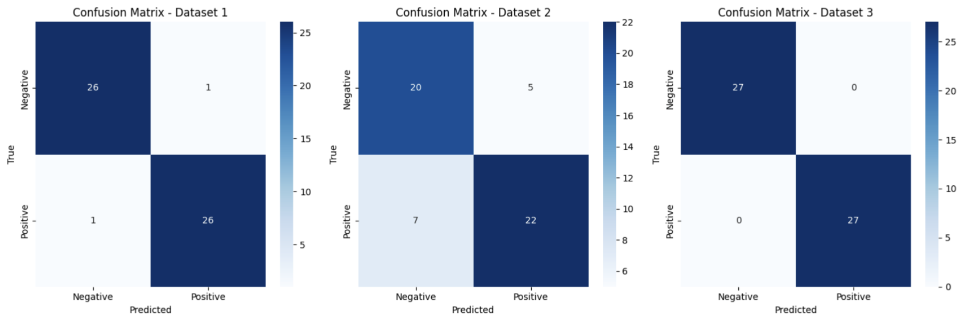


(l) Accuracy (Dataset 1, GloVe, LSTM)

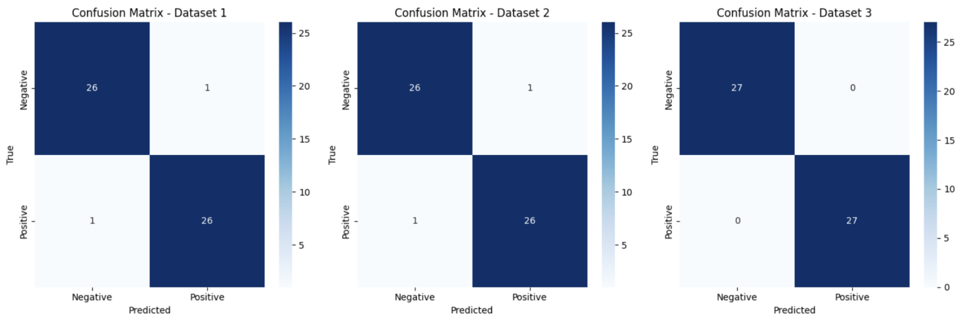
Figure 5.2 - Model performance comparison across different datasets and model configurations.



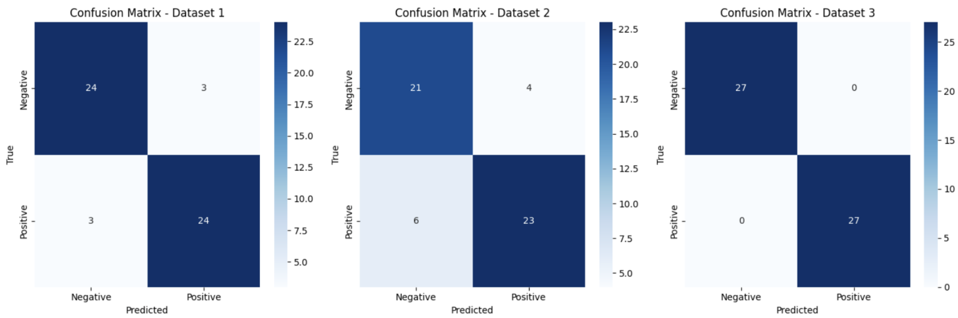
(a) Encoder Decoder with Keras Embeddings



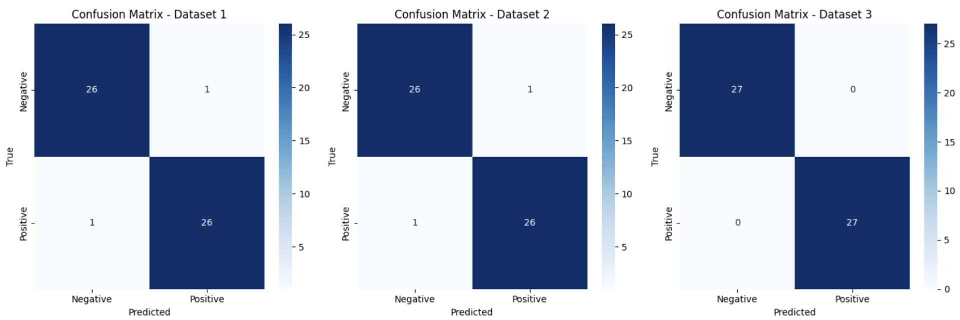
(b) Pretrained Embeddings Glove



(c) Pretrained Embeddings Glove GlobalMaxPool1D() and LSTM



(d) Pretrained Embeddings Glove BLSTM



(e) Pretrained Embeddings Glove GlobalMaxPool1D() and BLSTM

Figure 5.3 - Model performance comparison across different datasets and model configurations.

Chapter 6

Discussions

6.1 Discussions

In this field of our research, we introduce the evaluation of our models for false news concerning the problem statement and research questions outlined in the thesis. Our initial objective was to determine whether deep learning models, especially Encoder Decoder models with LSTM and BLSTM and Pretrained Embedding Glove, can distinguish between true and false news. The results showed that the models BLSTM and Attention processes performed well in differentiating the news, showing effective classification compared to other related works.

Comparing the results to other relevant works, it was clear that the BLSTM and Attention processes accomplished superior performance in identifying true from fake news. It is evident from this that comprehending the nuances of language used in news items requires making use of both forward and backward contextual information, which the BLSTM captures.

6.1.1 Key Contributions

Investigation of advanced deep learning models, including Encoder-Decoder architectures with BLSTM and attention mechanisms, for fake news classification.

1: GloVe embeddings that have been pretrained considerably improved the models comprehension of word semantic links. This was essential in recognising the small indications that frequently set fake news apart from trustworthy sources, such sensationalist wording or particular phrases that are more frequently used in made-up articles.

2: Evaluation of model effectiveness using widely adopted performance indicators like accuracy, precision, recall, and the F1 score. The model's performance was significantly enhanced by the Attention mechanism, which assigned varying weights to words according to their significance within the sentence. As a result, the model was able to concentrate on more pertinent passages, which increased the accuracy of the categorization.

3: Introduction of innovative variations of deep learning models, including combinations of Encoder-Decoder architectures with pre-trained GloVe embeddings, LSTM, and BLSTM, to improve the accuracy of fake news classification. The Encoder-Decoder models with BLSTM and Attention performed better than more conventional machine learning techniques and less complex neural network designs. This emphasises how important deep learning techniques are for addressing difficult text categorization jobs where it's important to capture contextual information and sequential relationships.

6.1.2 Dataset Investigation

Three main datasets were used in this study: the ISOT Fake News dataset, the Kaggle Fake News dataset, and an extra Kaggle dataset including news stories from 2016 to 2017. We processed and examined every dataset to ensure our classification models were reliable and valid.

Initially, the datasets were read as Pandas DataFrame objects, and sci-kit-learn's LabelEncoder was used to encode the class labels. Effective data manipulation and analysis were made possible by the datasets' original import as Pandas DataFrame objects. Using scikit-learn's LabelEncoder function, class labels were encoded to help convert categorical class labels into numerical representation, readying the data for model training.

To make it easier to evaluate the models, each dataset was split into 80–20 percent training and testing subsets. A Python script was used to carry out extensive preprocessing using the re-package for regular expressions. After tokenizing the texts into terms and sentences, the software cleaned them by deleting IP and URL addresses. English stopwords were eliminated, and the NLTK software was used to apply stemming.

6.1.3 The impact on results:

The properties of these datasets had a big influence on how well our deep-learning models worked. For example, the ISOT dataset offers a thorough testing environment for generalizing models.

To improve performance with each iteration, the datasets have been tested several times.

The effectiveness of our deep-learning models was significantly impacted by the datasets we employed.

An extremely useful dataset was ISOT. It contained a wide range of news stories from diverse sources covering a wide range of subjects.

This diversity enabled us to test our models under various conditions. Using these datasets, we conducted numerous tests on our models in an effort to improve them.

Based on our understanding from analysing the datasets, we also modified the way we handled the data and created the models.

6.1.4 Comparing the Performance of Conventional Machine Learning Algorithms

Three classical machine learning algorithms: Naive Bayes, Support Vector Machines (SVM), and Random Forest [38] are compared to see how well our Encoder Decoder models perform.

6.1.4.1 Naive Bayes

We found that, in terms of accuracy, our Encoder Decoder models consistently outperformed Naive Bayes on all datasets. For instance, our models achieved better accuracy levels, ranging from 88% to 100%, compared to Naive Bayes, which achieved accuracy levels ranging from 85.81% to 89.02%. This increased accuracy shows how effective our suggested method is.

6.1.4.2 Support Vector Machines

Comparably, our models outperform SVM on all datasets in terms of performance measures. Our models achieved equal or higher accuracies, ranging from 88% to 100%, while SVM reached accuracy levels ranging from 89.18% to 99.70%. These outcomes highlight the stability and effectiveness of our suggested encoder-decoder models in producing precise categorization results.

6.1.4.3 Random Forest

While Random Forest showed better accuracy overall, particularly with Dataset 3, our models performed competitively with Datasets 1 and 2. Our models obtained similar or higher accuracy, ranging from 88% to 100%, compared to Random Forest's 92.88% to 99.70% accuracy levels. This shows how well our models capture complex patterns in the data.

Table 6.1 - Results of Different Models

Model	Accuracy Train (%)	Accuracy Test (%)
Naive Bayes	85.81	85.39
Support Vector Machines	89.18	89.02
Random Forest	99.70	92.88

6.1.5 Comparing Performance with Neural Network Architectures

We also examined how well our models performed when compared to several neural network architectures, such as the Long Short-Term Memory (LSTM), Bidirectional LSTM (BLSTM), and Convolutional Neural Network (CNN) with GlobalMaxpooling.

Table 6.2 - Neural Networks Training Results

Model	Accuracy Train (%)	Accuracy Test (%)
CNN with GlobalMaxpool	99.55	97.77
CNN with DeepNetwork	94.31	92.62
LSTM	94.10	93.59

The accuracy of our suggested models was better than that of conventional neural networks. The accuracy levels of the traditional designs ranged from 93.59% to 99.55%, but our models consistently attained accuracies between 88% and 100%.

6.1.6 Comparing Other Models for the Identification of Fake News:

6.1.6.1 BERT and LSTM

BERT and BERT + LSTM [40] performed particularly well in a recent study that used the PolitiFact and GossipCop datasets to evaluate the models.

On PolitiFact, BERT's accuracy was 86.25%, and on GossipCop, it was 83.00%. It performed best on PolitiFact.

In comparison, BERT + LSTM's accuracy was the highest of the models that were examined; it was 88.75% on PolitiFact and 84.10% on GossipCop.

When comparing these outcomes to our models, we found that the LSTM model and Glove GlobalMaxPool1D() produced remarkably accurate and precise answers for all datasets, with accuracy rates of 98%, 97%, and 100% for Datasets 1, 2, and 3, respectively 86.25% accuracy on PolitiFact and 83.00% accuracy on GossipCop indicate a great performance, with PolitiFact showing exceptional strength.

Table 6.3 - Comparison of Models for PolitiFact

Model	Accuracy (%)	Precision	Recall	F1 Score
BERT	86.25	0.90	0.87	0.88
BERT + LSTM	88.75	0.91	0.90	0.90

Table 6.4 - Comparison of Models for GossipCop

Model	Accuracy (%)	Precision	Recall	F1 Score
BERT	83.00	0.89	0.89	0.89
BERT + LSTM	84.10	0.89	0.91	0.89

Additionally, the Glove BLSTM model showed competitive performance, with accuracy rates of 91%, 81%, and 100%, particularly on Dataset 3. Furthermore, across all datasets, the Keras Embeddings model consistently produced accuracy ratings of 95%, 88%, and 100%.

These results highlight how well our suggested models perform in fake news detection tests, indicating their potential applicability in real-world situations.

Furthermore, [41] we compare our results with those of a cited study written by Abdullah Marish Ali et al. to assess how effective false news identification techniques are. Results from the other study utilizing the ISOT dataset describe the classification accuracy of their model, as reported in the cited paper.

Their method consists of two stages: numerous binary classifications in the first stage, and multi-class classification in the second step.

The model they use achieves 99.94% accuracy in the first stage for both the real and fake news classifications, which is excellent accuracy. Furthermore, both classes' precision is 99.89%, with 100% recall for fake news and 99.89% recall for real news.

The F1 score, which accounts for recall as well as precision, registers at 99.95% for fake news and 99.94% for news that is true.

As the model moves on to the second stage of multi-class classification, it continues to achieve 100% accuracy, precision, recall, and F1 score for the actual and fake news classes.

For Dataset 3, on the other hand, our models regularly demonstrate perfect accuracy on all measures. Our models specifically show outstanding recall, accuracy, precision, and F1-score, suggesting ideal performance in recognizing false news articles in Dataset 3.

The results mentioned below highlight the strength and effectiveness of both methods in accurately distinguishing false information. Overall, the performance metrics between the two sets of findings show no significant differences.

Finally, the excellent accuracy and precision attained show that our model architectures and pre-processing procedures are appropriate for detecting fake news.

These results offer a strong basis for further investigation and possible practical applications in the field of classifying fake news.

Note: Encoder-Decoder with Keras Embeddings correspond to Tables in the Results section:

Table 6.5 - Overview of the datasets used in this study.

Name	Source	Description
Dataset 1	Kaggle	The Kaggle dataset from 2016 to 2017 includes both "fake" and "true" news stories. The collection covers two categories of articles: fake news and real news. The dataset was created from real-world sources, with genuine articles sourced from Reuters.com (news website). Several sources reported false information. Fake news was sourced from untrustworthy websites, as identified by Politifact and Wikipedia.
ISOT Fake News Dataset	Kaggle	The "ISOT Fake News" collection includes articles on a variety of topics, with a focus on politics and world affairs. Two csv files in the dataset. The collection had 44,898 articles, including both true and bogus news. The dataset included 21,417 articles with accurate information and 23,481 with incorrect information. The dataset contains the text, title, kind, and publication date of each article, as well as its whole body. This study compares the performance of classic models such as Static Embeddings, Dynamic Embeddings, LSTM, BLSTM, and a simple LSTM for neural networks. Emotions, symbols, pictographs, maps, and flags were preprocessed as features.
Fake News Dataset	Kaggle	We tested using the Kaggle FakeNews dataset, which has 20800 rows and 5 columns and is available worldwide. The dataset includes additional author-related information. Politifact.com editors select comments to classify depending on the majority of stories.

Table 6.6 - Performance Comparison of Different Models

Model	Dataset	Accuracy Train (%)	Accuracy Test (%)	Other Metrics
Naive Bayes	All	85.81	85.39	-
Support Vector Machines	All	89.18	89.02	-
Random Forest	All	99.70	92.88	-
CNN with GlobalMaxpool	All	99.55	97.77	-
CNN with DeepNetwork	All	94.31	92.62	-
LSTM	All	94.10	93.59	-
BERT	PolitiFact	-	86.25	Precision: 0.90, Recall: 0.87, F1 Score: 0.88
BERT + LSTM	PolitiFact	-	88.75	Precision: 0.91, Recall: 0.90, F1 Score: 0.90
BERT	GossipCop	-	83.00	Precision: 0.89, Recall: 0.89, F1 Score: 0.89
BERT + LSTM	GossipCop	-	84.10	Precision: 0.89, Recall: 0.91, F1 Score: 0.89

Table 6.7 - Performance Comparison of Different Models

Model	Dataset	Accuracy	Precision	Recall	F-score
Naive Bayes	All	85.39	-	-	-
Support Vector Machines	All	89.02	-	-	-
Random Forest	All	92.88	-	-	-
CNN with GlobalMaxpool	All	97.77	-	-	-
CNN with DeepNetwork	All	92.62	-	-	-
LSTM	All	93.59	-	-	-
BERT	PolitiFact	86.25	0.90	0.87	0.88
BERT + LSTM	PolitiFact	88.75	0.91	0.90	0.90
BERT	GossipCop	83.00	0.89	0.89	0.89
BERT + LSTM	GossipCop	84.10	0.89	0.91	0.89
Encoder-Decoder with Keras Embeddings	Dataset 1	0.96	0.96	0.96	0.96
Encoder-Decoder with Keras Embeddings	Dataset 2	0.88	0.88	0.88	0.88
Encoder-Decoder with Keras Embeddings	Dataset 3	1.0	1.0	1.0	1.0
Encoder-Decoder with Pretrained Embeddings Glove	Dataset 1	0.97	0.97	0.97	0.97
Encoder Decoder with Pretrained Embeddings Glove	Dataset 2	0.77	0.83	0.77	0.76
Encoder-Decoder with Pretrained Embeddings Glove	Dataset 3	1.0	1.0	1.0	1.0
Encoder-Decoder with Pretrained Embeddings Glove GlobalMaxPool1D() and LSTM	Dataset 1	0.98	0.98	0.98	0.98
Encoder-Decoder with Pretrained Embeddings Glove GlobalMaxPool1D() and LSTM	Dataset 2	0.97	0.97	0.97	0.97
Encoder-Decoder with Pretrained Embeddings Glove GlobalMaxPool1D() and LSTM	Dataset 3	1.0	1.0	1.0	1.0
Encoder-Decoder with Pretrained Embeddings Glove BLSTM	Dataset 1	0.91	0.92	0.91	0.91
Encoder-Decoder with Pretrained Embeddings Glove BLSTM	Dataset 2	0.81	0.86	0.81	0.81
Encoder-Decoder with Pretrained Embeddings Glove BLSTM	Dataset 3	1.0	1.0	1.0	1.0
Encoder-Decoder with Pretrained Embeddings Glove GlobalMaxPool1D() and BLSTM	Dataset 1	0.97	0.97	0.97	0.97
Encoder-Decoder with Pretrained Embeddings Glove GlobalMaxPool1D() and BLSTM	Dataset 2	0.97	0.97	0.97	0.97
Encoder-Decoder with Pretrained Embeddings Glove GlobalMaxPool1D() and BLSTM	Dataset 3	1.0	1.0	1.0	1.0

Chapter 7

Conclusion

7.1 Conclusion

In this study, we examined the effectiveness of multiple advanced models, including Encoder-Decoder, GlobalMaxPool1D, and Keras Embeddings, based on LSTM and BLSTM architectures with pre-trained GloVe embeddings, on three benchmark datasets: ISOT Fake News Dataset, Fake News, and Fake-News-Detection. Five significant models were tested, each demonstrating varying levels of effectiveness. These models are Encoder-Decoder with Keras Embeddings, Encoder-Decoder with Pretrained GloVe Embeddings, Encoder-Decoder with Pretrained GloVe Embeddings and GlobalMaxPool1D LSTM, Encoder-Decoder with Pretrained GloVe Embeddings BLSTM, and Encoder-Decoder with Pretrained GloVe Embeddings and GlobalMaxPool1D BLSTM. The results highlight the accuracy is increased when pre-trained embeddings are combined with advanced models, particularly for challenging datasets.

Bibliography

[1] Aloshban, Nujud. "ACT: Automatic fake news classification through self-attention." Proceedings of the 12th ACM Conference on Web Science. 2020.

[2] Trueman, T. E., Kumar, A., Narayanasamy, P., & Vidya, J. (2021). Attention-based C-BiLSTM for fake news detection. Applied Soft Computing, 110, 107600.

[3] de Oliveira, N. R., Pisa, P. S., Lopez, M. A., de Medeiros, D. S. V., & Mattos, D. M. (2021). Identifying fake news on social networks based on natural language processing: trends and challenges. Information, 12(1), 38.

[4] Kotsiantis, Sotiris B., Ioannis Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." Emerging artificial intelligence applications in computer engineering 160.1 (2007): 3-24.

[5] Oshikawa, Ray, Jing Qian, and William Yang Wang. "A survey on natural language processing for fake news detection." arXiv preprint arXiv:1811.00770 (2018).

[6] Choo, S., Seo, W., Jeong, D. J., & Cho, N. I. (2018, August). Multi-scale recurrent encoder-decoder network for dense temporal classification. In 2018 24th International Conference on Pattern Recognition (ICPR) (pp. 103-108). IEEE.

[7] Luvembe, A. M., Li, W., Li, S., Liu, F., & Xu, G. (2023). Dual emotion based fake news detection: A deep attention-weight update approach. Information Processing & Management, 60(4), 103354.

[8] Aloshban, Nujud. "ACT: Automatic fake news classification through self-attention." Proceedings of the 12th ACM Conference on Web Science. 2020.

[9] Trueman, T. E., Kumar, A., Narayanasamy, P., & Vidya, J. (2021). Attention-based C-BiLSTM for fake news detection. Applied Soft Computing, 110, 107600.

[10] Kumari, Rina, and Asif Ekbal. "Amfb: Attention-based multimodal factorized bilinear pooling for multimodal fake news detection." Expert Systems with Applications 184 (2021): 115412.

[11] Chopra, Sahil, Saachi Jain, and John Merriman Sholar. "Towards automatic identification of fake news: Headline-article stance detection with LSTM attention models." Proc. Stanford CS224d Deep Learn. NLP Final Project (2017): 1-15.

[12] Polat, Sumeyra Betul, and Selcuk Cankurt. "Fake News Classification using BLSTM with Glove Embedding." 2023 17th International Conference on Electronics Computer and Computation (ICECCO). IEEE, 2023.

[13] Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. Neural Computation.

[14] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidi- directional lstm-crf models for sequence tagging. arXiv:1508.01991

[15] https://github.com/ria18405/fake-news-classifier/blob/deployed_version/Report.pdf

[16] https://encyclopedia.pub/entry/42064

[17] Wang, Tong, et al. "An experimental study of LSTM encoder-decoder model for text simplification." arXiv preprint arXiv:1609.03663 (2016).

[18] Zhang, Shenhao, Yihui Wang, and Chengxiang Tan. "Research on text classification for identifying fake news." 2018 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC). IEEE, 2018.

[19] Zhou, Peng, et al. "Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling." arXiv preprint arXiv:1611.06639 (2016).

[20] Wu, Lianwei, et al. "Category-controlled encoder-decoder for fake news detection." IEEE Transactions on Knowledge and Data Engineering (2021).

[21] Li, D., Guo, H., Wang, Z., & Zheng, Z. (2021). Unsupervised fake news detection based on autoencoder. IEEE access, 9, 29356-29365.

[22] Mridha, Muhammad F., et al. "A comprehensive review on fake news detection with deep learning." IEEE Access 9 (2021): 156151-156170.

[23] Kong, Sheng How, et al. "Fake news detection using deep learning." 2020 IEEE 10th symposium on computer applications & industrial electronics (ISCAIE). IEEE, 2020.

[24] Buzea, Marius Cristian, Stefan Trausan-Matu, and Traian Rebedea. "Automatic fake news detection for romanian online news." Information 13.3 (2022): 151.

[25] Kaliyar, Rohit Kumar, Anurag Goswami, and Pratik Narang. "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach." Multimedia tools and applications 80.8 (2021): 11765-11788.

[26] Wang, William Yang. "" liar, liar pants on fire": A new benchmark dataset for fake news detection." arXiv preprint arXiv:1705.00648 (2017).

[27] Ajao, Oluwaseun, Deepayan Bhowmik, and Shahrzad Zargari. "Fake news identification on twitter with hybrid cnn and rnn models." Proceedings of the 9th international conference on social media and society. 2018.

[28] Jiang, T. A. O., et al. "A novel stacking approach for accurate detection of fake news." IEEE Access 9 (2021): 22626-22639.

[29] Thota, Aswini, et al. "Fake news detection: a deep learning approach." SMU Data Science Review 1.3 (2018): 10.

[30] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

[31] Hunter, John D. "Matplotlib: A 2D graphics environment." Computing in science & engineering 9.03 (2007): 90-95.

[32] Waskom, Michael L. "Seaborn: statistical data visualization." Journal of Open Source Software 6.60 (2021): 3021.

[33] Khan, Junaed Younus, et al. "A benchmark study of machine learning models for online fake news detection." Machine Learning with Applications 4 (2021): 100032.

[34] Wang, William Yang. "liar, liar pants on fire": A new benchmark dataset for fake news detection." arXiv preprint arXiv:1705.00648 (2017).

[35] Khan, Junaed Younus, et al. "A benchmark study of machine learning models for online fake news detection." Machine Learning with Applications 4 (2021): 100032.

[36] Ahmed, Hadeer, Issa Traore, and Sherif Saad. "Detecting opinion spams and fake news using text classification." Security and Privacy 1.1 (2018): e9.

[37] Reis, Julio CS, et al. "Supervised learning for fake news detection." IEEE Intelligent Systems 34.2 (2019): 76-81.

[38] Lai, Chun-Ming, et al. "Fake news classification based on content level features." Applied Sciences 12.3 (2022): 1116.

[39] Lai, Chun-Ming, et al. "Fake news classification based on content level features." Applied Sciences 12.3 (2022): 1116.

[40] Rai, Nishant, et al. "Fake News Classification using transformer based enhanced LSTM and BERT." International Journal of Cognitive Computing in Engineering 3 (2022): 98-105.

[41] Ali, Abdullah Marish, et al. "Deep ensemble fake news detection model using sequential deep learning technique." Sensors 22.18 (2022): 6970.

[42] Mohammed, Hamza Haruna, et al. "Multi-label classification of text documents using deep learning." 2020 IEEE International Conference on Big Data (Big Data). IEEE, 2020.

[43] Ouassil, Mohamed-Amine, et al. "A fake news detection system based on combination of word embedded techniques and hybrid deep learning model." International Journal of Advanced Computer Science and Applications 13.10 (2022).

[44] Youssef, Mellah, et al. "Text to Code Conversion Using Deep Learning for NLP." 2020 International Symposium on Advanced Electrical and Communication Technologies (ISAECT). IEEE, 2020.

[45] Font, Joel Escudé, and Marta R. Costa-Jussa. "Equalizing gender biases in neural machine translation with word embeddings techniques." arXiv preprint arXiv:1901.03116 (2019).

[46] Farokhian, Mahmood, Vahid Rafe, and Hadi Veisi. "Fake news detection using dual BERT deep neural networks." Multimedia Tools and Applications (2023): 1-18.

[47] Zhang, Yunxiang, and Zhuyi Rao. "n-BiLSTM: BiLSTM with n-gram Features for Text Classification." 2020 IEEE 5th information technology and mechatronics engineering conference (ITOEC). IEEE, 2020.

[48] Luan, Yuandong, and Shaofu Lin. "Research on text classification based on CNN and

LSTM." 2019 IEEE international conference on artificial intelligence and computer applications (ICAICA). IEEE, 2019.

[49] Bahad, Pritika, Preeti Saxena, and Raj Kamal. "Fake news detection using bidirectional LSTM-recurrent neural network." Procedia Computer Science 165 (2019): 74-82.

[50] Vyas, Piyush, Jun Liu, and Omar El-Gayar. "Fake news detection on the web: An LSTM-based approach." (2021).

[51] Aggarwal, Akshay, et al. "Classification of fake news by fine-tuning deep bidirectional transformers based language model." EAI Endorsed Transactions on Scalable Information Systems 7.27 (2020): e10-e10.

[52] Kaliyar, Rohit Kumar. "Fake news detection using a deep neural network." 2018 4th international conference on computing communication and automation (ICCCA). IEEE, 2018.

[53] Verma, Pawan Kumar, et al. "WELFake: word embedding over linguistic features for fake news detection." IEEE Transactions on Computational Social Systems 8.4 (2021): 881-893.

[54] Landthaler, Jörg, et al. "Extending Thesauri Using Word Embeddings and the Intersection Method." ASAIL@ ICAIL 8.1 (2017): 112-119.

Bibliography