Fake News Detection Using Deep Learning and Computational Linguistics

MSc Research Project
Data Analytics

NEIL SAHAY
Student ID: x19238061

School of Computing
National College of Ireland

Supervisor: Prof. Martin Alain

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	NEIL SAHAY
Student ID:	x19238061
Programme:	Data Analytics
Year:	2021
Module:	MSc Research Project
Supervisor:	Prof. Martin Alain
Submission Due Date:	31/01/2022
Project Title:	Fake News Detection Using Deep Learning and Computational Linguistics
Word Count:	8596
Page Count:	23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Neil Sahay
Date:	30th January 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Fake News Detection Using Deep Learning and Computational Linguistics

NEIL SAHAY
x19238061

Abstract

Though the growth of the internet in the twenty-first century has resulted in significant advancements in many areas, including education, research, and information dissemination, it has also had several negative consequences. While the popularity of social media skyrocketed in 2007 with the introduction of Facebook, the number of social media users has risen substantially since then, resulting in increased dissemination of fake news, which in turn causes societal instability. Manual tracking is difficult as information is vulnerable to a variety of language used by users, such as hilarious content, which is not always considered fake news. While there has been research done in this sector to apply Artificial Intelligence to minimize this recurring issue, it has mainly backtracked owing to the provision of inadequate data and usage of supervised learning algorithms. By integrating artificial intelligence approach with innovative Natural Language Processing (NLP) techniques such as Recurrent Convolutional Neural Network (R-CNN) and Long Short-Term Memory (LSTM), and this study helped in overcoming the difficulty of correctly classifying fake and real content by achieving 95% accuracy, therefore assisting in the tracking and removal of illegal and misleading information.

Keywords— RCNN, LSTM, Fake news, Real News, Lemmatization

1 Introduction

The cosmic rate at which data is generated globally poses significant challenges due to the negative effects which it embodies on the end-user. Many instances covering this broad aspect are observed each day while using social media, wherein some major instances become highly noticeable. The term “Fake News” took prominence in aftermath of 2016 US presidential elections, when the newly elected government under Donald Trump’s victory prompted pro-trump supporters to form groups on social media to spread biased ramifications against the democrats such as Hillary Clinton and other political party leaders. As a result of the dissemination of misleading narratives, readers suffer prejudiced opinions. According to Pew Research Centre, more than 60% coped with a condition of uncertainty caused by social media facts, highlighting a serious problem that should be handled with optimal resources efficiently and effectively. Furthermore, the influence of false news has been noticed in the commercial industry, like in 2013, rumors spread alleging an explosion that injured US President Barack Obama, caused the US stock market to fall, resulting in substantial losses amounting to $130 billion.

Significantly, identifying the sort of information that requires repercussions is a big challenge, which Stewart (2021) highlights as a fundamental difficulty in eradicating misleading narratives and viewpoints. Building on these concerns, the study depicts two significant obstacles in recognizing verified data, namely the dispute over whether the content is moderated and the validity with which the material may be identified as false. Research by Asr and Taboada (2019) also delves with the existing problem to create a separate repository which the research states that it might still not be sufficient to undertake all the necessary data, but the research provides insights by applying neural network techniques such as LSTM and deep neural networks.

Usage of deep learning techniques along with other machine learning techniques is also discussed in Mouratidis et al. (2021) , which uses distinct linguistic patterns such as differential lengths of documents, adjusted adverbs, and adjectives, as by adjusting such parameters the research is able to comprehend different scenarios to accompany within the model, thus adjusting the accuracy to close proximity levels. Also, the importance of combining different neural network models is highlighted by Nasir et al. (2021), which makes use of conjunction of hybrid model based on CNN and R-CNN based neural frameworks to optimize and bring in novel methods in detecting fake news which in itself is a challenging task given the fact about inconsistent datasets and improper neural models which have been implemented, which this research focuses to resolve.

The importance which this research brings in has great impact on the future perspectives. As due to the various different semantic and linguistic patterns, such as content that addresses humour and cannot be inferred as fake topic, can be incorrectly flagged as fake, if the neural network is not trained to handle such instances, moreover the use of special characters such as @,#,& also pose significant difficulties to handle news content and thus it becomes difficult to flag the content as fake, which this research overcomes by using continuous experiments to handle such instances resulting in a better model. Furthermore, this research adopted the use of four separate datasets that comprise news material not just from social media but also from major news sources such as Fox News, CNN, and others. Thus, by combining dynamic datasets with linguistic techniques and integrating all of these instances with neural networks, this research is able to create a much more diverse hybrid model that can detect false instances of various categories such as the ability to detect humour from the news and analyse the meanings of words which comprises of special characters that previous researchers has not been able to comprehend within them to a large extent.

The following instances leads to answer the research question which our research aims to achieve i.e.

To what degree Recurrent Convolutional Neural Network in conjunction with Long Short Term Memory, can assist in detecting and reducing the prevalence of fake news?

Several experiments are conducted with varying test data sampling to remove biases and reduce the over-fitting of the model. Moreover, the research objectives are presented as below:

- Eliminated false propaganda by the usage of advanced machine learning framework, that resulted in reduced manual resources and thereby reducing time to effectively scrutinize extensive data.

- Integrated neural networks in consonance with computational linguistics such as tokenization and word embedding to develop an optimal hybrid module for detection of fake news.

- Evaluated performance metrics for the 3 different experiments.
- Conducted extensive studies to validate the results of the experiments performed which encapsulated multiple scenarios, so as to increase the efficiency of the implemented model.

2 Related Work

Various academics and intellectuals have contributed to the identification of fake news using powerful machine learning approaches. It is impossible to track and remove such information by manual chores, owing to the present prevalent issue relating to data security and data management, which is created at an enormous rate around the globe. As a result, academics and researchers are constantly researching the prevalence of deep machine learning techniques in order to find an effective and efficient solution to the aforementioned problem.

2.1 Role of fact checkers

Fact-checking organizations such as Politifact, ClaimsKG, Snopes, and many others, form a representational group to dissuade fake news and falsely biased claims over the web. The study by (Alam et al.; 2020)shows how to create a unique dataset that is far more dynamic not only in content but also in terms of linguistic capabilities because it incorporated four different languages, resulting in a more efficient dataset in conjunction with implying BERT modelling and using the Question Answering approach to build the model. While the research emphasizes the use of a multilingual dataset, it does not explain how the Natural Language Processing (NLP) technique would account for multilingual text changes inside the model.

There is also a link between partisan news and false news, as (Vargo et al.; 2017) shows in their study by considering the impacts of the causal relationship between the dissemination of fake news and the spread of partisan news. However, while the study addressed critical aspects of the employment of fact checkers, it failed to consider the numerous challenges that machine learning models might confront when exposed to such a dynamic dataset.

There has also been an increase in the use of fact checkers in automated responses to fake news stories, such as on Twitter, which has been thoroughly investigated by (Vo and Lee; 2019), where the text generation, validation of linguistic features in fact checkers, and creation of a discrete dataset containing tweet information are all encapsulated within the model framework. However, because it is more suited for long-range sequencing, the research may have used the Long short term memory (LSTM) technique to cope with the gradient descent issue in Recurrent Neural Network (RNN) modelling. Identifying the source of fake news is also a crucial part, which is explained by (Giachanou et al.; 2020), wherein the researchers developed a framework i.e. CheckerOrSpreader, where checker depends on fact checkers to identify the spreaders. The model utilizes CNN architecture along with word embedding's and identifying linguistic patterns by employing Linguistic Inquiry and Word count (LIWC) proposed by Neuman and Cohen, which provides leverage to incorporate personality features of the end users. However, usage of RNN over CNN would have resulted in better results if it would have captured temporal information which could have considerable effect within the meaning of sentences.

2.2 Implementation of Deep Learning Models

Deep learning models have been widely used in image classification but recent works such as (Bahad et al.; 2019) discusses the use of deep neural network techniques such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN).The researchers used a convolutional network with numerous layers, however after using RNN, it was determined that recurrent approaches were more successful for long-range semantics, but this was owing to the vanishing gradient problem. However, the study overlooked the use of dynamic datasets, which is a key stumbling block in the identification of false news, and instead focused on the application of several machine learning models.

(Balwant; 2019) details about using a Part of speech (POS) strategy to discern the text meaning, such as emotional context, taking into account the role of the meaning of speech inside the text. While the research focuses on implementing dynamic techniques, it fails to mention any specific word embedding technique used, as the reader will find only the generic term ”word embedding” throughout the research document, with no explanation of how to use any specific technique in detail, such as word2vec, Bagofwords, or GloVE. There has been an increase in the spread of fake news on major social media such as Twitter, which (Ajao et al.; 2018) highlights by taking into account data directly through the Twitter API, that will increase the model's efficiency by considering current trends and topics thriving on the web. On analysing the model, the accuracy was greater than 80%, indicating a significant advancement in the use of deep learning techniques. But the prevalence of not detailing word embedding techniques and reliance on a single source of test data i.e. Twitter API may hamper the efficiency of the model in real-time environment.

Research conducted by (Kaliyar; 2018) continues to dig deep in the implementation of machine learning methods to detect fake news, with the introduction of implementation of regular Machine Learning (ML) techniques such as Random Forest, Naïve Bayes, K-nearest neighbor, Decision Tree in conjunction with the hybrid model of CNN-LSTM network and showed an accuracy more than 90%. While although the researcher showcased the increased accuracy attained but the usage of the dataset that was gathered from Kaggle is deprecated and does not contain updated content and thereby forfeits the absolute aim of having an efficient automated system.

The issue of not having sufficient datasets is addressed to a large part in (Goonathilake and Kumaral; 2020), which primarily describes the source of the dataset, which is gathered by web scraping from various governments and non-government organizations. However, the research does not propose an effective word embedding approach, instead opted Word2Vec embedding, which creates substantial issues. In retrospect, the research was able to build effective test data by combining the collective data with fact-checking websites, which, when combined with the installed RCNN-LSTM hybrid model, produced an effective model.

2.3 Eliminating special characters and using supervised learning methods

Special characters such as (@,#,%,_) can readily impact the NLP techniques which (Formento et al.; 2021) elaborates upon by focusing on cumulating token discovery followed by token perturbation under which two steps are performed i.e. Cascading which is also known as aggressive method, followed by Replacement, also termed as a subtle method for token perturbation. Furthermore, when dealing with special characters, this line of approach has a significant influence on the development of the NLP system. The detailed results revealed significant increase in the accuracy of false news detection, demonstrating the impact of deleting unnecessary characters on accuracy. However, despite significant advances in the experimental stage without the use of additional deep neural networks, the study lacked evidence of the influence of extremely sophisticated neural approaches in the identification of fake news.

(de Oliveira et al.; 2021) discusses the use of supervised learning algorithms in addition to deep neural algorithms. The dataset problem is addressed by incorporating data from a variety of sources, including LIAR, Buzzfeed, and others, followed by removal of stop words and special characters, which had an influence on dataset validation, and then tokenization and lemmatization is performed, which is a useful way to get the meaning of the words and quickly train the model. Despite the use of such complicated frameworks, the research went backward in documenting the appropriate outcomes, and instead of focusing on a general description of assessment metrics, it prevented the readers from getting a clear picture of the results obtained. The use of five different Ensemble algorithms, such as Support Vector Machine, Naive Bayes, Decision tree, knn, and Random Forest, is also detailed in the research by Prakash et al. (2019). The paper details the studies done on each of these strategies, including the accuracies and performance acquired on each algorithm. However, the lack of any word embedding techniques might have a significant impact on the veracity of the results produced.

By delving into ensemble techniques (Ahmad et al.; 2020), numerous optimized algorithms such as SVM, knn, Decision Tree, Logistic Regression, LSTM modeling, and CNN are implemented. Although the study claims that neural networks fall short in terms of ensemble approaches, the incorrect inquiry, which ignores dataset inconsistencies and relies on the LSTM approach to emphasize text classification for huge textual methods, worsens the research goal. Research by Setiyaningrum et al. (2019) also makes use of k-Nearest Neighbor along with the implementation of feature selection process i.e. Chi-Square. The researchers made use of a dataset comprising of Arabic Sentiment Analysis, though the process adopted yielded considerable results in showing the implemented architecture but is flawed by not taking into account other ensemble techniques, moreover, knn does not perform better with large datasets and thus in real-time it can impact the efficiency of the system considerable.

2.4 Limitations with test data and linguistic patterns

Relevance of proper datasets, plays a crucial role in the creation of an efficient modelling framework. (Nakamura et al.; 2019), highlights the issues of dataset scarcity by citing relevant works which showed the drawbacks in current datasets availability such as focussing only on text data. Thus the research is concentrated to create an absolute and novel dataset consisting multi-dimensional dataset framework, by utilizing resources from multiple resources such as Reddit, Fakeddit.However, significant meta-data such as emotional analysis of user comments, which is critical in real-world applications, is not taken into consideration. (Shu et al.; 2018) emphasizes the necessity for an efficient dataset, leading to the construction of a novel dataset, "FakeNewsNet," which incorporates data from multiple areas, including spatiotemporal information, which is incorporated in the research foundation. However, the study falls short of articulating how prospective bias will obstruct the applied model's real-time aims. (Oshikawa et al.; 2018), also identifies dataset as the major issue, while implementing the modelling strategies. This research emphasizes notable findings produced by using these procedures in conjunction with non-neural network techniques such as SVM, Naive Bayes algorithms, and a variety of datasets. However, the exclusion of computational linguistics technique such as word embedding's and LSTM, drives the research to non -conclusive results. (Chapman et al.; 2011) also analyzes the roadblocks to NLP development, with a focus on the absence of publicly available data owing to privacy regulations that make the present datasets unsuitable. However, the study only looked at datasets and did not look into the usage of machine learning models, therefore it only gave a partial answer to the question. Research by (Brunsdon and Comber; 2020) explored the effect of big data on data analytics, since extreme amount of data is generated each day, thus researchers define the messy data which can cause issues not only from technical point of reference, but also from the pre-processing, analysis and modelling perspective. The research explored the issues pertained by such big data, due to many diverse factors such as the size of data, data transformation and the complexities involved within the data frame constructed.

2.5 Significance of satirical content

Fake news can be classified into several categories, including satirical content, which is researched by (Golbeck et al.; 2018) in defining using a variety of criteria, including scathing criticism, numerous conspiracy theories, dramatic crimes, bigotry, as well as paranormal. As a result, a relevant dataset is produced and visualized by taking all of these elements into account. However, because the results do not take into consideration the many NLP methodologies, it cannot be regarded as a real identification of the problem. (Rubin et al.; n.d.), digs deeper into data collection describing satire material, humour, and ludicrous nature contained within the text. The findings are presented in a such that the Base model outperformed the N-gram model. However, the dataset should also focus also on meta-data which this study neglects to address. Research by (Horne and Adali; 2017) incorporates several sorts of datasets within the framework, with satirical material included under various criteria such as political data and electoral data. However, this study has not taken into consideration other aspects apart from political data, that may have aided in the development of an effective model.

Thus in conclusion, after reviewing the illustrious literature reviews majored by different researchers, it is observed that more research could have been done in utilizing advanced NLP techniques and machine learning techniques involved in dealing with the problem of fake news, which this research will undertake to provide a quantitative and substantial solution. Moreover section 2.4 detailed about the importance of relevant test data which till now has been a major hurdle in conceptualizing the framework to detect fake news, also section 2.3 covered the importance of removing unwanted characters from the data, as data obtained from several sources can be messy which can withheld from attaining the desired results. Thus, this research will help in attain a novel approach to address the dataset issue by taking into account the datasets obtained from various sources and will employ advance deep machine learning techniques such as Long Short Term Memory (LSTM) and Recurrent Convolutional Neural Network (R-CNN) along with implementing word embedding's and variable neurons within LSTM layer, which will not only increase the accuracy of the model, but also have an impact on the efficiency as well.

3 Methodology

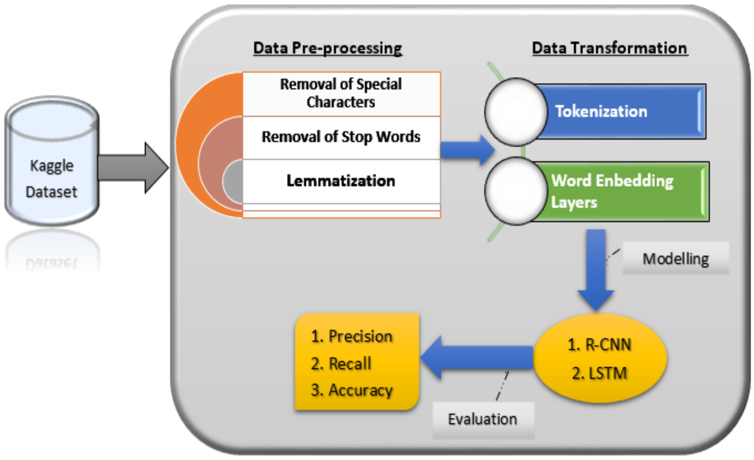

Figure 1: Research Methodology

The research methodology that has been adopted in this research is Knowledge Discovery in Databases (KDD). As the implemented model concern a large amount of data processing and computation. Thus after executing different phases in the methodology section as shown in Figure 1, this research will be able to classify fake and real news content which requires knowledge discovery from the dataset within. Furthermore, the sub-sections below provides insights on the different modules addressed in this scope.

3.1 Data Gathering

The 4 distinct dataset has been acquired from Kaggle through Public Domain availability, wherein the entire corpus is combined to form a cumulative dataset with entire collection of fake and real news data. Moreover, the dataset has been cumulatively selected in such a way as to gain and train the model with efficient knowledge. The data reflects not only social media headlines but also includes major sources news such as Fox, New York Times and many other. The 4 distinct datasets together will form an imperial part within the modelling framework which will enhance the performance and efficiency of the implemented model.

3.2 Data Pre-Processing

Data pre-processing is an integral framework within the specified model, as through cleaned data an appropriate model can be accomplished. Thus, diverse data cleaning techniques are applied to improve the efficiency and scalability of the model. Techniques such as removing unwanted characters, HTML tags, stop words are used. Moreover, lemmatization is also performed to group the conjugated words into a single word which will increase the performance. The below sections details different techniques applied to the dataset.

1. **Removal of special characters:**
   Unwanted characters such as HTML tags, semicolon(:), punctuation marks and other special characters which the dataset consists, are insignificant for processing

and does not yield any results, thus such elements are removed swiftly from the dataset. Research by Sun et al. (2018), also mentions the relevance to eliminate the special characters and noise within the data and thus removing inconsistent data. So, to achieve this task this research took use of python methods, wherein special characters such as **',?,@,#'** are substituted within the entire corpus of text and the text is converted to lower case so as to achieve normalization within the data. By implementing these methodologies, unwanted characters are successfully removed and the resultant test data is then proceeded for further processing.

2. **Removal of Stop words**:
   Stop words are generally in the form of words such as 'a, an, the, etc.', which holds little purpose in analysing the text, thus it becomes eminent to remove such stop words within the text corpus so as to yield better results. To remove the stop words, this research took into consideration the use of spacy library in python which has incredible computational resources and libraries which can help to decode text classification and other NLP tasks. By importing the spacy library and converting the sentence within the document into tokens, we are able to extract and remove the stop words.

3. **Lemmatization**:
   When the above tasks such as removal of unwanted characters and removal of stop words is successfully completed. Then finally Lemmatization is performed as part of pre-processing step which is defined as reducing the words in the form of lemma. For example the lemma form of organizing and organized in it's reduced form will be the word "organize". As, if this task is not performed on the corpus of words , then words such as explaining, explained will be given separate tokens, even though both corresponds to the same word i.e. explain. Thus by performing this task, we will be able to standardize the text within the corpus of words and also will be effective while constructing the model.

3.3 Tokenization and Word Embedding of Textual Data

To implement the model efficiently, the below sections provide a detailed description of the 2 different text pre-processing steps adopted.

3.3.1 Tokenization

Tokenization is the phase where the text data is converted into a format that is essential to be included while implementing the model within the framework. As observed in Figure 2, the text "The quick brown fox jumped over the lazy dog", is tokenized separately and allotted a unique token through word indexing, and from this set of tokens, vocabulary is created which is a set of unique tokens, and taking the top unique tokens can help to boost the performance of the neural network.

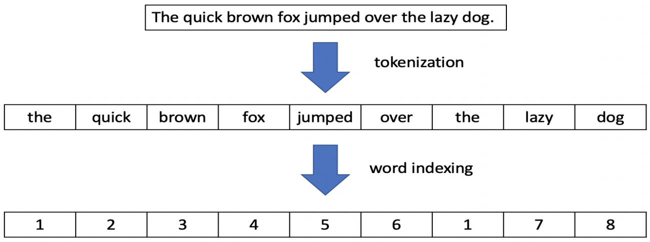

Figure 2: Word Tokenization

3.3.2 Word embedding layers

Identifying the meaning of words with respect to other corpus of words is essential in the model framework. Thus to achieve this, word embedding methodology is used which makes use of vector representation thus helping the neural network to learn efficiently. The algorithm stimulates the neural network architecture by mapping each vector to the mapped word thereby resembling a neural network framework. Figure 3. shows

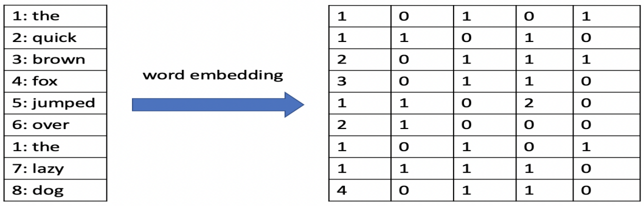

Figure 3: Word Embedding

a sentence with 5 dimensional vectors for each word within the corpus of domain. To attain this implementation, this research used Embedding layers as the deep learning model which will be able to learn from similar embedding vectors, moreover as the text has already been pre-processed, thus after performing tokenization, embedding layers helped to transform tokens to embedding vectors.

3.4 Modelling

After completion of all the pre-defined steps, the final step involved is the implementation of the conceptualized model i.e. consonance of Recurrent - Convolutional Neural Network (R-CNN) and Long-short Term Memory (LSTM) approach. The primary rationale for implementing this model is because Recurrent networks can work efficiently with text data and can help identify patterns within the collection of documents moreover by integrating it with CNN, the model will be able to efficiently extract the features and store them to identify patterns, and finally, LSTM integration will be able to withhold the information loss and reduce it to critically low levels, thus resulting in better classification accuracy. This research has not taken into consideration supervised learning techniques such as Support vector machines, K-Nearest Neighbor due to the lack of finding patterns in data which unsupervised algorithms leverage, moreover it cannot detect patterns on its own within the data, which neural networks are able to process efficiently.

The conjunction of the model is achieved by adding convolutional layers i.e. 1D convolution layer followed by pooling layers that form a base to create recursion within the model. The model performs recursion through a 1D convolution layer which uses a series of kernels that lie in low dimension and concurrently paves through the input vector to produce dot products which are then computed to obtain an output vector.

Similarly, Max-pooling is added in a recursive manner which reduces the size of the input vector, by selecting the maximum values. Finally I made use of the LSTM layer, which helped to reduce the information loss as it makes use of forgot gate that assisted the model to retain the information and thus helped to avoid vanishing gradient problem which is the major cause of concern while implementing RCNN model. Moreover, as this is a classification model, with the aim to identify correctly fake news and real news separately, so by adding LSTM layers the model will be able to counter the back propagation effect resulting in better classification of content.

3.5 Results and Evaluation Metrics

The implemented model which is a conjunction of Convolutional network and Long short term memory is evaluated based on 3 different metrics each having it's own perspective which will assist in answering the research question i.e. **"To what degree Recurrent Convolutional Neural Network in conjunction with Long Short Term Memory, can assist in detecting and reducing the prevalence of fake news?"**. To evaluate the model, this research uses three evaluation metrics, detailed below:

1. **Precision:** Precision takes the proportion of all the positive scenarios (True Positive) to the sum of False positives (FP) and True positives (TP) as shown in equation 1. The rationale for using this metric is that it helps to take into account the relevant data points in consideration, moreover, it will also help to avoid scenarios wherein the label tagged as fake news will be classified as real news that will reduce the model reliability. This metric has also been implemented by Clark et al. (2010), wherein Precision is used as an evaluation metric for grammar induction.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

2. **Recall:** The second metric that is been used is Recall that will help to identify all specific data points within the entire corpus of the dataset. This will aid in analyzing the instances at which the model is able to accurately predict the real news instances and not mark them as fake news, since it is critical that the algorithm does not label true instances as false. Recall is the proportion of all positive observations (TP) to the sum of False Negatives (FN) and True Positives (TP) as shown in equation 2. Moreover, this metric is also used in research by Derczynski (2016), wherein Recall is established as a metric to measure the effectiveness of the implemented model.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

3. **Accuracy:** Accuracy score helps to take into account all the true positive and negative predictions which were judged by the model with respect to all the scenarios, thus it provides the ability to analyze and assess the model reliability in judging the correct instances of fake and real news scenarios. Accuracy as depicted in equation 3, is the ratio of the sum of True positive (TP) and True negative (TN) to all the

positive, negative and errors (False positive (FP) and False negative (FN)) that the model predicted on the test set. As this metric takes into consideration the errors within the model, thus it aids to evaluate the reliability of the model as the aim is to classify correctly the real and fake narratives.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

Thus by taking the leverage of these 3 metrics, this research will aim to adequately measure the effectiveness of the implemented model.

4 Design Specification

This section describes the two neural network approaches adopted i.e. Recurrent Convolutional Neural Network and Long Short Term Memory, as well as the functionality used by these algorithms used to classify text appropriately.

4.1 Recurrent Convolutional Neural Network

Convolutional Neural Networks (CNN's) have been widely employed for image classification across the domain, but they have lately been used in the NLP domain as well, with CNN's delivering critical and efficient performance in text categorization. This study uses a recurrent network to modify the CNN model, which helps to sustain information exchange within the network by sliding over the input sequence as shown in Figure 4 and processing the values accordingly. Recurrent networks also perform well for extended data sequences. When a specific pattern is spotted, the model will process the results of each convolution and activate when a specific pattern is detected. Moreover by changing the size of the kernels and concatenating their outputs into 4, 5, or 8 unique words, the model will be able to identify patterns of various sizes. Patterns might be terms such as "I despise," "extremely fantastic," and so on, and CNN's can recognize them in the sentence independent of their position.

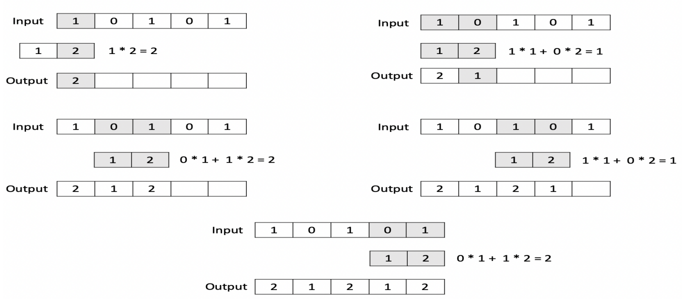

Figure 4: 1D Convolution Operation

In addition, CNN is used in combination with Max pooling to limit the amount of features and to make use of critical components that Abedalla et al. (2019) also mentioned in it's research, that the use of max pooling helped to decrease the clutter of features and optimize the model accordingly.

4.2 Long Short Term Memory (LSTM)

LSTM forms the recurrent structure of a CNN model as it helps to store information for longer periods of time. LSTM involves 3 different types of gates as shown in Figure 5, i.e. input gate, forget gate, and output gate. The input gate function is use to receive the values, the forget gate function is to retain the information that comes through the input gate, and finally the output gate function gives the end values that are obtained after calculations.

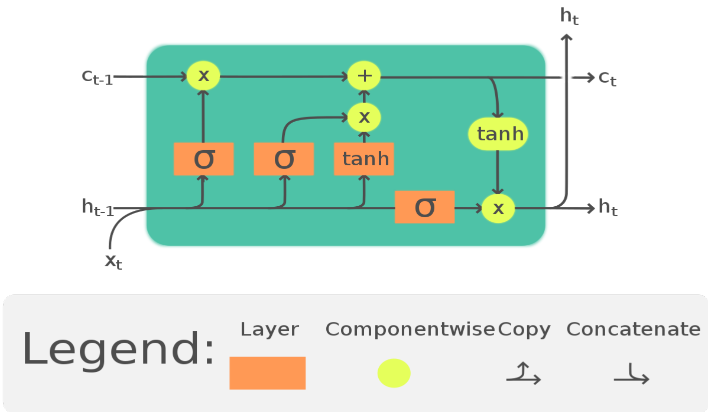

Figure 5: LSTM Architecture

As a result, LSTM is a vital step in an RCNN structure since it works as an interface to not only process information, but also to retrieve essential components of the data, and because this research covers a huge corpus of textual data, significant attributes must be stored appropriately. The implementation continues with CNN being combined with various convolutional layers, and when combined with max-pooling layers, the input size is significantly reduced, while the remaining information is sent to LSTM for computational storage, and finally, the fully connected layer takes the input from LSTM and computes a weighted sum of values using the sigmoid function. Umer et al. (2020) also considers the leverage that LSTM gives inside the architecture, implying that the complete framework, when used together, will considerably increase model efficiency and scalability.

5 Implementation

The implementation phase enables the scalable and efficient execution of the hypothesized paradigm as shown in Figure 6. The sub-sections explains in detail the execution of the model. In addition, the sections also cover the fundamental libraries required to implement the proposed model, as well as information on the data and the process through which it is cleaned and pre-processed. The following phases go through the proposed algorithms as well as training the dataset.

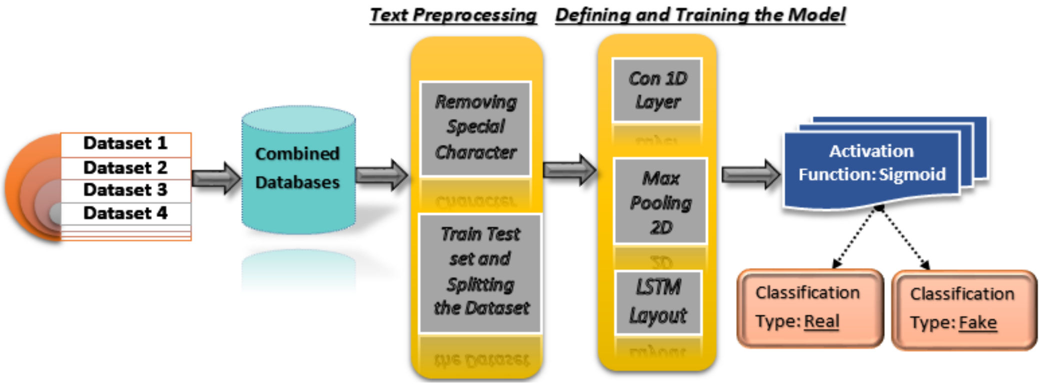

Figure 6: Phases in Implementation of Model

5.1 Materials Utilized

Since deep neural networks require a substantial amount of memory, this study did not use standard system configurations; instead, it used Google Collaboratory with an integrated Graphical Processing Unit (GPU) with approximately 80 GB of data allocation and 13 GB of RAM. Furthermore, the base machine is an HPE laptop with an integrated i5 core CPU and 8 GB of RAM, allowing for a rapid and scalable environment. Additionally, other libraries such as scikit-learn, NumPy, and spacy are extensively utilized for Natural Language Processing since they help to perform certain operations such as Tokenization, Lemmatization, etc. To execute the model Tensorflow is also utilized, which aids in the execution and training of deep neural networks. It also includes libraries for metrics like accuracy, precision, and recall.

5.2 Data Handling

The 4 distinct datasets obtained are combined to from a single dataset containing all the different traits present. Moreover, the dataset is read and converted into a dataframe, and the NA values are dropped. The next step involves analysing and exploring the data and by plotting the Seaborn chart, the dataset is found to be evenly distributed with equal distribution of real and fake news data.

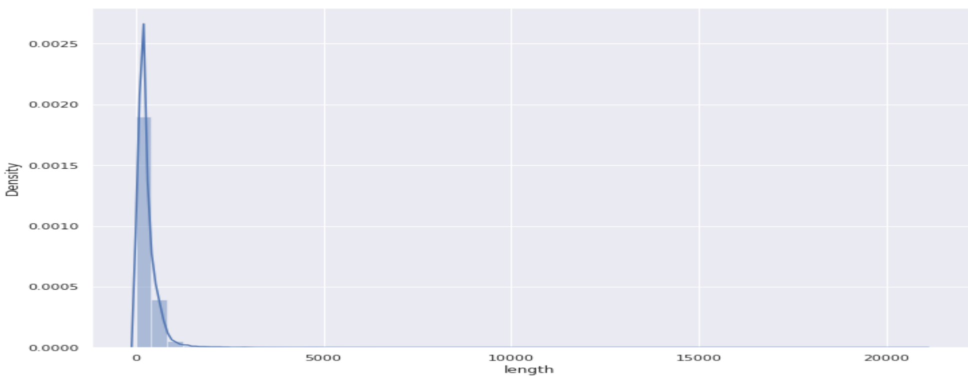

Figure 7: Data Distribution of Length of the articles

On plotting the distribution of word count, figure 7 shows that about 550 words is the average length of the articles while the median is about 404 words, also the figure distribution is observed to be right-skewed with more than 70 percent of articles showing word count of more than 600 words while the longest article is an outlier with more than 20,000 word count. Following the completion of the exploratory analysis, data preparation includes cleaning the data by removing special characters, eliminating stop words, by utilizing the spacy library that contains library to remove stop words and emoticons. Moreover, word tokenization is attained with the usage of Natural Language Tool kit (NLTK) by extending the (word_tokenizer) library, I was able to tokenize the corpus of words, and finally performed lemmatization by using "token.lemma_" function, which transformed the words to their lemma forms. Then finally word embedding layers are used to help the model grasp the text relation between the words in the dataset. The output obtained is then split into training and test samples, where 70% of the data will be used for training while the rest 30% will be used for validating the model.

5.3 Model Implementation

The model architecture implemented in this research follows the conjunction of Recurrent Convolutional Neural Network (RCNN) and Long Short Term Memory (LSTM) approach. As LSTM forms the recurrent layers within the designed module, thus the entire architecture comprises distinct layers which are as detailed below:

5.3.1 1D Convolutional and Max-Pooling Layers

The model starts with the inclusion of 1D convolution layer, while CNN supports various layers such as 1D, 2D, and 3D. The 2D layer is primarily concerned with image-based input, whereas the 3D layer is concerned with 3D image data such as MRI scans and video.

The rationale for implementing 1D convolution layers is majorly because the kernel will slide across a single-dimensional space and as the model will be handling text data which is 1 dimensional, thus to save computational costs and efficiency of the system, 1D convolution layer greatly assist in the architecture of the model. Also, the model makes use of 128 convolutional filters with kernel size of 5 and the activation function in the convolutional layers has been set to Rectified Linear Unit (ReLU) function, because it produce the output if the input sequence is positive, otherwise it will return 0. Also ReLU performs better while developing convolutional layers and is able to handle vanishing gradient problems efficiently.[1] In the next stage Max-pooling layers are also added, and the reason for embedding max-pooling layers is the advantage of reducing the input size of the text while also preserving the essential traits of the data. Thus in conjunction, the convolutional 1D layers along with max-pooling layers, provide the base for the implemented model.

5.3.2 LSTM Layer

After the Conv1D and max-pooling layers have been incorporated, the LSTM layer is introduced, which will add depth to the system architecture. The input gate, forget gate, and output gate are the three gates that make up the LSTM layer. To avoid over-fitting in

[1] https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/

the model inside the multi-layered architecture, the next stage is embedding the sequence with initially 100 neurons and a dropout rate of 10%, i.e. 0.1. As can be analysed in figure 8, LSTM layers have been inserted after the initial construction of CNN layers and max-pooling, hence increasing the complexity and scalability of the given model. The number of neurons in the LSTM layer will be tweaked in the experiments at regular basis to increase the complexity of the model and attain greater accuracy.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 512, 80)	16480640
conv1d (Conv1D)	(None, 512, 256)	102656
max_pooling1d (MaxPooling1D)	(None, 256, 256)	0
conv1d_1 (Conv1D)	(None, 256, 512)	655872
max_pooling1d_1 (MaxPooling1D)	(None, 128, 512)	0
conv1d_2 (Conv1D)	(None, 128, 768)	1966848
max_pooling1d_2 (MaxPooling1D)	(None, 64, 768)	0
conv1d_3 (Conv1D)	(None, 64, 1024)	3933184
max_pooling1d_3 (MaxPooling1D)	(None, 32, 1024)	0
conv1d_4 (Conv1D)	(None, 32, 1280)	6554880
max_pooling1d_4 (MaxPooling1D)	(None, 16, 1280)	0
lstm (LSTM)	(None, 16, 80)	435520
dropout (Dropout)	(None, 16, 80)	0
lstm_1 (LSTM)	(None, 80)	51520
dropout_1 (Dropout)	(None, 80)	0
dense (Dense)	(None, 1)	81

Figure 8: RCNN-LSTM layered model architecture

5.3.3 Fully Connected Layer

After adding all of the initial layers in the model by using several CNN layers and Max-pooling layers, the output of these layers is transferred to a fully connected layer that employs a sigmoid activation that returns Boolean value i.e. True or False which is also the aim of the model to classify whether the content is fake or real. Furthermore, the output is sub-divided into two classes, i.e. 0 and 1, that will be used to categorize the validity of news articles. The LSTM_text_Classifier function incorporates hyper-parameters such as pool size, batch size, and convolutional parameters such as number of layers, kernel size, and filters, which will be tweaked on regular basis during the experimentation stage to obtain classification results.

5.4 Training

The model is trained on the training set which is 70% of the entire dataset. The RCNN and LSTM model is executed on the training data to train the model efficiently and achieve optimal results on the test data. Also the hyper-parameters are tweaked in the 3 experiments such as the number of epochs, number of CNN Layers, dropout level, and LSTM layers are some of the parameters that are changed regularly. As a result, this combination is critical for training data on the deployed framework.

The model is compiled using various parameters as listed below:

1. **Binary Cross entropy:**
The loss function used is Binary cross entropy as it compare the probabilities of predicted values to actual output as shown in equation 4. and thus helps to classify the error between actual and predicted values. [2]

$$\text{Loss} = \sum_j (t_i) log(p_i) \tag{4}$$

2. **Adam Optimizer:**
To optimize the learning rate in the model, adam optimizer has been utilized, which is a combination of Adagrad and RMSprop. Adagrad optimizer will help to find minima in convex function while RMSprop is able to perform better on non-convex function. Thus this combination of both in adam optimizer will help the model to adjust its learning parameters effectively and sustain an optimal efficiency.

3. **Callbacks**:
Callbacks are generally used to monitor the execution of the defined model in the execution state. I have used two types of Callbacks while training the model which are as listed below:

   (a) **Early stopping:** This callback helps to stop the execution of the model when the model is no longer achieving an increase in its metrics. While training the model Early Stopping monitored the accuracy and stopped the execution as soon as accuracy becomes stagnant. This will help in avoiding added epochs and also avoid over-training the model.

   (b) **Model Checkpoint:** This callback helps to store the best model obtained in the training stage of the experimentation. While training the model, as the maximum value of accuracy is achieved, that model gets saved as the best model obtained during the training stage. This helps to analyse the model definition for the best set obtained.

6 Evaluation

This section will go over the three experiments that were executed by tweaking the hyper-parameters in the consonance of Recurrent Convolutional Neural Network (RCNN) and Long Short-Term Memory (LSTM) model. Furthermore, parameters such as dropout and callbacks are modified during testing, and the results are analysed for each test, with finally the best metric outcomes reported.

6.1 Experiment 1

The first experiment is carried out by tweaking the hyper-parameters, such as setting the LSTM layers to 100 and the kernel size to 5, which will aid in the extraction of features and their transmission to CNN layers. Furthermore, the number of Convolutional 1D layers is set to 3 by default, since the number of CNN will be changed in subsequent phases

[2]https://www.analyticsvidhya.com/blog/2021/03/binary-cross-entropy-log-loss-for-binary-classification/#:~:text=Binary%20cross%20entropy%20compares%20each,far%20from%20the%20actual%20value.

to test the model's efficacy. The maximum vector length defined in the hidden layers is also set to 128 because the vocabulary length and input length of text are large and contain complex textual data, and because the embedding layer is faster in performance, this experiment will be able to cover broader ranges by setting a larger vector length.

Table 1: Result set for Experiment 1.

Epochs	Recall	Accuracy	Precision
3	0.5087	0.5173	0.5161
5	0.5112	0.5286	0.5154
10	0.5254	0.5364	0.5178

The results in table 1. shows that after 3 epochs, 5 epochs, and 10 epochs, the maximum precision observed is 0.5178, demonstrating that the model is able to partially predict fake and real news instances. Furthermore, the recall of 0.5254 and accuracy of 0.5364 demonstrates that the model is unable to classify fake news effectively.
Moreover, when model is implemented on the test data which is used for validation, the accuracy on test set is observed to be 50%, thus revealing that the model has been over-fitted, and therefore in the next experiment hyper-parameters will be tweaked effectively to reduce such occurrence and thus also increasing the model performance.

6.2 Experiment 2

In the second experiment, several hyper-parameters are modified in order to minimize overfitting while simultaneously boosting the model complexity in order to build an efficient model. The dropout rate has been adjusted to 0.25 to avoid over-fitting. Furthermore, the number of neurons in LSTM layers and number of CNN layers has been increased to 128 and 5 respectively, so as to increase the computational efficiency of the model.

Table 2: Result set for Experiment 2.

Epochs	Recall	Accuracy	Precision
3	0.988	0.982	0.989
5	0.976	0.985	0.988
10	0.979	0.987	0.998

Table 2. shows several epoch executions as well as the metrics gathered when assessing the model's performance. The execution is repeated in three distinct epoch ranges, ranging from 3 to 10 epochs. As can be analysed from table 2, accuracy is best at 10 epochs with 0.987, indicating that the model is able to accurately forecast false news instances, but recall is highest at 3 epochs with 0.988, indicating that the model is able to better categorize genuine articles effectively.
Thus, by increasing the number of neurons in LSTM layer and by increasing the dropout rate, a significant increase in accuracy is observed, so in the third model the hyper-parameters will then be tweaked again further so as to improve the model performance.

6.3 Experiment 3

The final experiment involves pushing the hyper-parameters to their limits in order to obtain an ideal model. While analyzing and training the model, this experiment takes into account a variety of characteristics such as setting the dropout value to 0.5, which is also considered an ideal value across the neural network. By using this number, the built model will avoid using the extra parameters and, as a consequence, will help to avoid over-fitting the dataset. Furthermore, the number of neurons in the LSTM layers have been increased to 228, while the number of CNN layers has been set to 5, resulting in a more computationally efficient model. Hence, the LSTM layers with added neurons will help to counter information loss across the network, thus increasing the efficiency.

Table 3: Result set for Experiment 3.

Epochs	**Recall**	**Accuracy**	**Precision**
3	0.988	0.9908	0.9927
5	0.996	0.9966	0.9971
10	0.9983	0.9986	0.9988

As shown in table 3, the peak values of accuracy and precision at 10 epochs are 0.9986 and 0.9988 confirming the model efficiency, thus by expanding the layers and neurons inside the framework, the final model is able to demonstrate efficient outcomes.

6.4 Discussion

This section offers thorough information on the outcomes acquired after the three distinct experiments were completed successfully. This research is able to illustrate the ability of the implemented model, i.e. the combination of Recurrent Convolutional Neural Network (RCNN) and Long Short Term Memory (LSTM). So, after analyzing the results of the experiments, the research question for this study, "To what degree Recurrent Convolutional Neural Network in conjunction with Long Short Term Memory, can assist in detecting and reducing the prevalence of fake news?" has been thoroughly answered with conclusive evidence.

In the first experiment, the number of layers in Long Short Term Memory (LSTM) is set to 100, number of Conv1D layers are set to 3 while other key parameters such as the kernel size is set to 5, which will aid the model in retrieving features and passing them to Convolutional layers. Furthermore, in order to train the model for processing vast amounts of textual input, the maximum vector length is set to 128. Thus, after initializing the parameters, the model is trained for different epochs ranging from 3, 5 to 10 at which the respective accuracies and precision are obtained, with the highest accuracy discovered at 10 epochs being 0.5364, implying that the model has not been trained to locate real and fake news effectively.

Moreover, since the model is overtrained as test set results show accuracy of 50%, hence a second experiment is performed with the dropout rate set to 0.25 and other hyper-parameters such as the number of CNN layers consistently increased to 5, in order to make the model more complex in terms of learning rate. This is because in real-world scenarios, a large amount of data will be generated, and if this model is used, it will be able to handle this data in a scalable and efficient manner. In order to offset the increase in CNN layers, the number of LSTM layers has been raised to 128. As a result of adjusting these values, the model is run at three distinct epoch levels: 3, 5, and 10. Since the model uses callbacks and EarlyStopping, the model execution is interrupted at epoch level 10, aiding in determining the maximum number of epochs at which maximum accuracy, precision, and Recall are observed. The results clearly reveal that at epoch level 10, maximum accuracy and precision of 0.987 is recorded. When comparing these findings to the previous experiment, an improvement in accuracy and precision can be seen, showing that modifying the hyper parameters influenced the model's metrics and aided in consistent performance improvement. As a result, the last experiment is carried out in order to determine the highest accuracy that this model may attain by altering the dropout rate and other specified parameters.

Following the successful completion of the first two experiments, the third and final

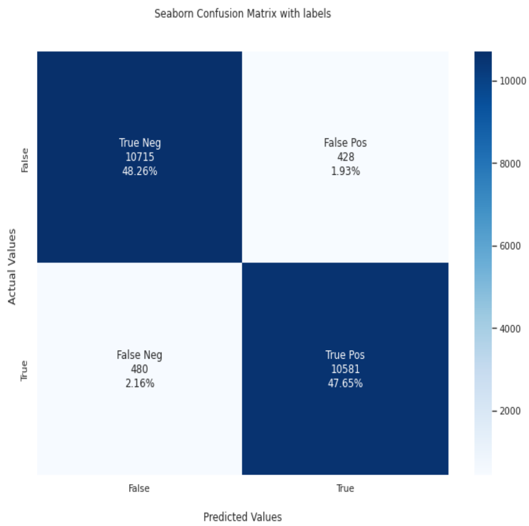

Figure 9: Test Results Confusion Matrix

experiment is carried out by analyzing the outcomes of the previous two experiments and adjusting the hyper-parameters to get the best possible metrics performance. In the final experiment the number of neurons in the Long Short Term Memory (LSTM) is increased to 228, while the number of layers in the convolutional layer is still set to 5, moreover the dropout has been increased to 0.5. Thus this combination makes the model computationally efficient, and it resulted in obtaining the optimal results as shown in figure 9. So, the model is analyzed on epoch levels again, with the highest accuracy and precision obtained at the 10th epoch level, with accuracy approaching to one, demonstrating that the model is capable of correctly classifying Real and Fake news. Furthermore, the precision is also analyzed to be the highest with 0.99. When the confusion matrix for the last experiment is plotted as shown in Figure 9, it can be seen that the percentage of the model correctly predicting True positive and True negative is greater than 48%, implying that the model had an overall accuracy of 95.91%. And with an overall accuracy of 95.91% on the test set, this model stands out for its best of ability to achieve high resilience, scalability, and reliability. Furthermore, it is concluded that by combining RCNN and LSTM has a significant impact on how deep learning models are designed to classify fake news.

7 Conclusion and Future Work

After conducting the experiments and analysing the results, it can be concluded that the hybrid approach of combining Recurrent Convolutional Neural Networks (RCNN) and Long Short Term Memory (LSTM) yielded detailed results and conclusive evidence that the model performed reasonably well and with proven efficiency in detecting fake news across the domain. The novel approach of combining these neural architecture with a complex dataset, which is a combination of four different datasets containing data not only from social media websites but also from major news publications, has yielded comprehensive results, with an overall accuracy of over 95% on the test dataset.
This study aims to develop an efficient and scalable model that can be used in real-time scenarios to handle large amounts of data. Furthermore, by adjusting hyper-parameters such as the number of Convolution layers and Long short term memory (LSTM) layers, it was discovered that the accuracy, precision, and recall was highest in the last experiment and that can be used in the classification of real and fake news. Also, the use of LSTM architecture has been shown to improve learning rate and assist the CNN model in retaining information for longer periods of time, thereby reducing the vanishing gradient issue. The model also avoids over-fitting by employing a dropout layer with varied levels for each experiment, offering a leverage to avoid biased outcomes. Additionally, various pre-processing algorithms such as tokenization, lemmatization, and word embedding have greatly aided in the development of an efficient and robust model.
Hence, since the model involves the identification of fake news, it is critical to maximize the model's dependability and efficacy. In the future, as the topic of fake news detection grows increasingly difficult to handle, thus by implying certain architectures such as Resnet-50, GoogleNet, which have more complex and deep layers within their architecture, can be applied to achieve a more comprehensive model. Also, since test data is a major source of concern in fake news detection due to its scarcity, more comprehensive and up-to-date data repositories can be established in future studies, which can also leverage API from prominent social media cites such as Twitter, Facebook, and others, allowing the neural architecture to stay updated while avoiding content misclassification.

8 Acknowledgement

I would like to express my gratitude to Professor Martin Alain for his consistent valuable assistance and supervision across the entire project duration. Also I am grateful for his detailed explanation of the project structure, and regular feed backs on the project report, which helped me to formulate the report properly. I'd also want to express my gratitude to my family and friends for their unwavering support during the study effort.

References

Abedalla, A., Al-Sadi, A. and Abdullah, M. (2019). A closer look at fake news detection: A deep learning perspective, *PervasiveHealth: Pervasive Computing Technologies for Healthcare* pp. 24–28.

Ahmad, I., Yousaf, M., Yousaf, S. and Ahmad, M. O. (2020). Fake news detection using machine learning ensemble methods, *Complexity* **2020**.

Ajao, O., Bhowmik, D. and Zargari, S. (2018). Fake news identification on twitter with hybrid cnn and rnn models, *ACM International Conference Proceeding Series* **18**: 226–230.
URL: *https://doi.org/10.1145/3217804.3217917*

Alam, F., Shaar, S., Dalvi, F., Sajjad, H., Nikolov, A., Mubarak, H., Martino, G. D. S., Abdelali, A., Durrani, N., Darwish, K., Al-Homaid, A., Zaghouani, W., Caselli, T., Danoe, G., Stolk, F., Bruntink, B. and Nakov, P. (2020). Fighting the covid-19 infodemic: Modeling the perspective of journalists, fact-checkers, social media platforms, policy makers, and the society.
URL: *https://arxiv.org/abs/2005.00033v5*

Asr, F. T. and Taboada, M. (2019). Big data and quality data for fake news and misinformation detection.
URL: *https://github.com/*

Bahad, P., Saxena, P. and Kamal, R. (2019). Fake news detection using bi-directional lstm-recurrent neural network, *Procedia Computer Science* **165**: 74–82.

Balwant, M. K. (2019). Bidirectional lstm based on pos tags and cnn architecture for fake news detection, *2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2019* .

Brunsdon, C. and Comber, A. (2020). Big issues for big data: challenges for critical spatial data analytics, *Journal of Spatial Information Science* .
URL: *https://arxiv.org/abs/2007.11281v2*

Chapman, W. W., Nadkarni, P. M., Hirschman, L., D'Avolio, L. W., Savova, G. K. and Uzuner, O. (2011). Overcoming barriers to nlp for clinical text: the role of shared tasks and the need for additional creative solutions, *Journal of the American Medical Informatics Association* **18**: 540–543.
URL: *https://academic.oup.com/jamia/article/18/5/540/829390*

Clark, Alexander, Fox, Chris, Lappin and Shalom (2010). The handbook of computational linguistics and natural language processing.

de Oliveira, N. R., Pisa, P. S., Lopez, M. A., de Medeiros, D. S. V. and Mattos, D. M. F. (2021). Identifying fake news on social networks based on natural language processing: Trends and challenges, *Information 2021, Vol. 12, Page 38* **12**: 38.
URL: *https://www.mdpi.com/2078-2489/12/1/38/htm https://www.mdpi.com/2078-2489/12/1/38*

Derczynski, L. (2016). Complementarity, f-score, and nlp evaluation.

Formento, B., Ng, S. K. and Foo, C. S. (2021). Special symbol attacks on nlp systems, *Proceedings of the International Joint Conference on Neural Networks* **2021-July**.

Giachanou, A., Ríssola, E. A., Ghanem, B., Crestani, F. and Rosso, P. (2020). The role of personality and linguistic patterns in discriminating between fake news spreaders and fact checkers, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **12089 LNCS**: 181–192.
URL: *https://link.springer.com/chapter/10.1007/978-3-030-51310-8*$_1$7

Golbeck, J., Mauriello, M., Auxier, B., Bhanushali, K. H., Bonk, C., Bouzaghrane, M. A., Buntain, C., Chanduka, R., Cheakalos, P., Everett, J. B., Falak, W., Gieringer, C., Graney, J., Hoffman, K. M., Huth, L., Ma, Z., Jha, M., Khan, M., Kori, V., Lewis, E., Mirano, G., Mohn, W. T., Mussenden, S., Nelson, T. M., Mcwillie, S., Pant, A., Shetye, P., Shrestha, R., Steinheimer, A., Subramanian, A. and Visnansky, G. (2018). Fake news vs satire: A dataset and analysis, *WebSci 2018 - Proceedings of the 10th ACM Conference on Web Science* pp. 17–21.

Goonathilake, M. D. and Kumaral, P. P. (2020). Cnn, rnn-lstm based hybrid approach to detect state-of-the-art stance-based fake news on social media, *20th International Conference on Advances in ICT for Emerging Regions, ICTer 2020 - Proceedings* pp. 23–28.

Horne, B. D. and Adali, S. (2017). This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news, *ArXiv* **abs/1703.09398**.

Kaliyar, R. K. (2018). Fake news detection using a deep neural network, *2018 4th International Conference on Computing Communication and Automation, ICCCA 2018* .

Mouratidis, D., Nikiforos, M. N. and Kermanidis, K. L. (2021). Deep learning for fake news detection in a pairwise textual input schema, *Computation 2021, Vol. 9, Page 20* **9**: 20.
URL: *https://www.mdpi.com/2079-3197/9/2/20/htm https://www.mdpi.com/2079-3197/9/2/20*

Nakamura, K., Levy, S. and Wang, W. Y. (2019). r/fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection, *LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings* pp. 6149–6157.
URL: *https://arxiv.org/abs/1911.03854v2*

Nasir, J. A., Khan, O. S. and Varlamis, I. (2021). Fake news detection: A hybrid cnn-rnn based deep learning approach, *International Journal of Information Management Data Insights* **1**: 100007.

Oshikawa, R., Qian, J. and Wang, W. Y. (2018). A survey on natural language processing for fake news detection, *LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings* pp. 6086–6093.
URL: *https://arxiv.org/abs/1811.00770v2*

Prakash, P. B., Kumar, M. P., VenkataManaswini, G. and Mehata, K. M. (2019). Fake data analysis and detection using ensembled hybrid algorithm, *Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019* pp. 890–897.

Rubin, V. L., Conroy, N. J., Chen, Y. and Cornwell, S. (n.d.). Fake news or truth? using satirical cues to detect potentially misleading news, pp. 7–17.
URL: *https://youtu.be/2X93u3anTco*

Setiyaningrum, Y. D., Herdajanti, A. F., Supriyanto, C. and Muljono (2019). Classification of twitter contents using chi-square and k-nearest neighbour algorithm, *Proceedings - 2019 International Seminar on Application for Technology of Information and Communication: Industry 4.0: Retrospect, Prospect, and Challenges, iSemantic 2019* pp. 78–81.

Shu, K., Mahudeswaran, D., Wang, S., Lee, D. and Liu, H. (2018). Fakenewsnet: A data repository with news content, social context and spatialtemporal information for studying fake news on social media, *Big Data* **8**: 171–188.
URL: *https://arxiv.org/abs/1809.01286v3*

Stewart, E. (2021). Detecting fake news: Two problems for content moderation, *Philosophy Technology 2021* pp. 1–18.
URL: *https://link.springer.com/article/10.1007/s13347-021-00442-x*

Sun, W., Cai, Z., Li, Y., Liu, F., Fang, S. and Wang, G. (2018). Data processing and text mining technologies on electronic medical records: A review, *Journal of Healthcare Engineering* **2018**.

Umer, M., Imtiaz, Z., Ullah, S., Mehmood, A., Choi, G. S. and On, B. W. (2020). Fake news stance detection using deep learning architecture (cnn-lstm), *IEEE Access* **8**: 156695–156706.

Vargo, C. J., Guo, L. and Amazeen, M. A. (2017). The agenda-setting power of fake news: A big data analysis of the online media landscape from 2014 to 2016:, *https://doi.org/10.1177/1461444817712086* **20**: 2028–2049.
URL: *https://journals.sagepub.com/doi/full/10.1177/1461444817712086*

Vo, N. and Lee, K. (2019). Learning from fact-checkers: Analysis and generation of fact-checking language.
URL: *https://doi.org/10.1145/3331184.3331248*