COVID-19 Related Fake News Detection Model

by

Sumaiya Islam Shondhy
ID:17101532
Forhad Ahmed Khan
ID: 17301083
Syed Shoaib Ibrahim
ID: 17301144
Shuvajit Barua
ID:17301168

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2021

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Sumaiya Islam

Sumaiya Islam Shondhy
ID:17101532

Ahmed Khan.

Forhad Ahmed Khan
ID: 17301083

SS Ibrahim

Syed Shoaib Ibrahim
ID: 17301144

Barua

Shuvajit Barua
ID:17301168

Approval

The thesis/project titled "COVID-19 Related Fake News Detection Model" submitted by

1. Sumaiya Islam Shondhy(ID:17101532)
2. Forhad Ahmed Khan (ID: 17301083)
3. Syed Shoaib Ibrahim (ID: 17301144)
4. Shuvajit Barua (ID:17301168)

Of Fall, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January, 2021.

Examining Committee:

Supervisor:
(Member)

Zavid Parvez

Digitally signed by Zavid Parvez
DN: cn=Zavid Parvez, o=Brac University,
ou=CSE, email=zavid.parvez@bracu.ac.bd,
c=US
Date: 2021.01.15 15:17:50 +06'00'

Mohammad Zavid Parvez,PhD
Assistant Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Ashad
Kabir

Digitally signed by Ashad Kabir
DN: cn=Ashad Kabir, c=AU,
o=Charles Sturt University, ou=School
of Computing and Mathematics,
email=akabir@csu.edu.au
Date: 2021.01.15 08:26:12 +11'00'

Ashad Kabir,PhD
Senior Lecturer
and Deputy Leader of Data Mining Research Group
Charles Sturt University

Co-Supervisor:
(Member)

Mostafijur
Rahman Akhond

Digitally signed by Mostafijur Rahman Akhond
DN: C=BD, OU=Department of CSE, O=Jashore University
of Science and Technology, CN=Mostafijur Rahman
Akhond, E=mr.akhond@just.edu.bd
Reason: I am approving this document
Location: your signing location here
Date: 2021.01.15 16:17:29+06'00'
Foxit Reader Version: 10.1.1

Mostafijur Rahman Akhond
Lecturer
Department of Computer Science and Engineering
Jashore University of Science and Technology

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam,PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chairperson)

Prof. Mahbub Majumdar
Chairperson
Dept. of Computer Science & Engineering
Brac University

Mahbubul Alam Majumdar, PhD
Professor and Dean
Department of Computer Science and Engineering
Brac University

Abstract

In this era of developed information and technology, any sort of information runs faster than air. The reliability of the information can be tricky at times. Some news publishing sources can publish news that are actually misguiding. The drastic evolution of electronic media over the past couple of decades has fueled the spread of fake news causing confusion and misunderstanding among the mass regarding any topic. The main motive behind producing these fake news is to create an agenda or to spread trepidation among people. People tend to become more panicked during any kind of disaster or pandemic, this it is easier to make them believe these misinformation in these times. Likewise, COVID-19 pandemic is not out of the grasp of misinformation spreading. To tackle this, we have proposed a Fake News Prediction model that will be used to detect fake news regarding COVID-19 that are being circulated in different electronic media.

Keywords: *COVID-19,Fake news detection, Dataset, news article.*

Acknowledgement

Firstly, we would like to thank to Almighty for keeping us in good health and everything.

Secondly, we would like to thank our supervisor Dr. Mohammad Zavid Parvez and co-supervisor Dr. Ashad Kabir for guiding us and for the giving us unconditional support. Without their proper guidance we could not have done this.

Thirdly, to our parents for their full support that they have given us throughout our life.

And last but not the least, immense respect to the COVID-19 frontline workers and COVID warriors for giving us the hope of life and motivating us to do this research .

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

WHO (World Health Organization) announced an official name for corona virus syndrome (SARS-CoV-2), which is COVID-19, on Feb 11, 2020. Later in March, it was declared as a pandemic. COVID-19 has spread from China to all over the globe since December 2019, and the misery has no bounds.[13]. Many of the cities countries had lockdown and the only way they can know what's happening around the world is on the internet. So in this alarming time, giving people the best and trustworthy news is the key. As the internet is the only source of contact with the outside world, the fake news media took the chance of it. They are spreading enormous fake data and news to make people panic. This problem is rapidly rising worldwide and is having a serious social disruption.

Fake news is written in an intentional and in such a way that readers have to believe it. The language of fake news is rather unverifiable which makes readers confuse with the actual circumstances. Fake news is widely spread in our daily life and in this modern era of internet availability, fake news is now at it's prime. People tend to believe anything and everything they see on the internet without even verifying if it is true or not. In a post, for example, during the 2016 US election, FBI investigators accused of Hillary e-mail leaks found dead in an obvious assassination suicide[23]. Often during the presidential campaign in 2016, even more false news of this nature emerged and had a very important effect on the outcome of the election. So fake news then impacted the election now it is impacting a more serious matter.If we offer an example of the effect of false news, we can clearly see it here where 77 cell phone towers were set on fire because of the plot to propagate COVID-19 via 5G mobile networks[4].

Such examples show how the impact of fake news has in our daily life especially in this COVID-19 pandemic period. But collecting fake news is really challenging. Firstly, it is really difficult to identify fake news. The writer intentionally writes the news in such a way that it barely has a way to understand if the news is fake or not. The integrity of information is thus sacrificed. So identifying fake news is quite hard. Furthermore, if the news is posted by a trusted source, then it becomes more difficult to identify.The authenticity of newly emerging news is not easy to test because the application data set is not enough to train[12]. Any useful areas of research and further study need to be established in order to find reliable users, extract useful news features and formulate legitimate methods for dissemination of content.

1.2 Problem Statement

Fake news identifying is with the help of deep learning in the field of research for several years and the volume of work grows spontaneously every day. Many data scientist have worked in fake news detection throughout the years and shown really potential output. Although there has been tremendous work on various fake news approaches, our methodology varies from those works in a sense that we only work with news articles about COVID-19. Like some of the researches worked on deep learning method like us, but their work was focused on the fake news of social media, so their data set is based on the misinformation news posted in those social media[14]. Many more works in social media are focused on false news identification. As we work on news articles rather than social media news, Therefore, our methodology is unique from them. We also need a decent dataset to fulfill our research aims. A data set with news verified by fact checkers and the false news that have been proved to be false or hoax. A moderate number of false news articles are collected from the verified dataset COAID[15] which contains fake news from social media posts and tweets also news articles. Some of the news articles are compiled from the dataset of fakeCovid[22]. Since both the dataset is based on COVID-19, the COAID dataset contains social media posts and the fakeCovid dataset contains multilingual news article, further the ratio of real and fake news is quite low. On the other hand, we made our dataset solely based on news articles and maintained a healthy ratio between false and real news, which is a significant classification element. In addition, we collected most of the articles manually from verified websites such as factchecker, load stories, politifact, etc., which makes our dataset stand out from the other. We have therefore created our own benchmark-labeled dataset and performed the analysis.

1.3 Research Objective

Our research is mainly to detect falsified news and misinformation related to COVID-19 on the Internet. As COVID-19 is a very sensitive issue, We need to figure out the best performing model for which we have been experimenting with various methods and models.We used a labelled data set containing false news, which can be identified by state-of-the-art processing techniques for natural language and advanced deep learning approaches. Our intention relies on comparing the accuracy of simple methods with respect to modern and complex techniques in the deep learning family to figure out the best possible combination for the dataset.

In this paper, We are presenting a labeled COVID-19 news article dataset with a classification model that will predict whether the news is false or true .Our work is different from other existing works for following reasons: I) Our dataset has equal amount of true and false news,which we can not see in other existing papers. II) We have evaluated our dataset using both deep learning and traditional machine learning perspective. This kind of trend can not be found in existing works of COVID-19. III) The dataset is solely dedicated to classify fake news related to COVID-19. Other dataset have contents to detect misinformation as a whole.

1.4 Workflow

The first thing we did is, we collected the news from various sources and we stored them in a dataset accordingly. Then, we did data preprocessing which is text preprocessing and representation. After that, we tokenized the padded sequence and did word embedding. After that, we splitted it into trainning and validation dataset and a test dataset. In training and test dataset, we applied Bidirectional LSTM-RNN, unidirectional LSTM-RNN and CNN to train the model. We keep testing model in train data and tune hyper parameter in validation data until there is no defined optimal loss value or before any overfitting. After that, we will train the dataset for prediction. Finally, using test data and the data we trained, we will predict the label of news article and find out model accuracy.

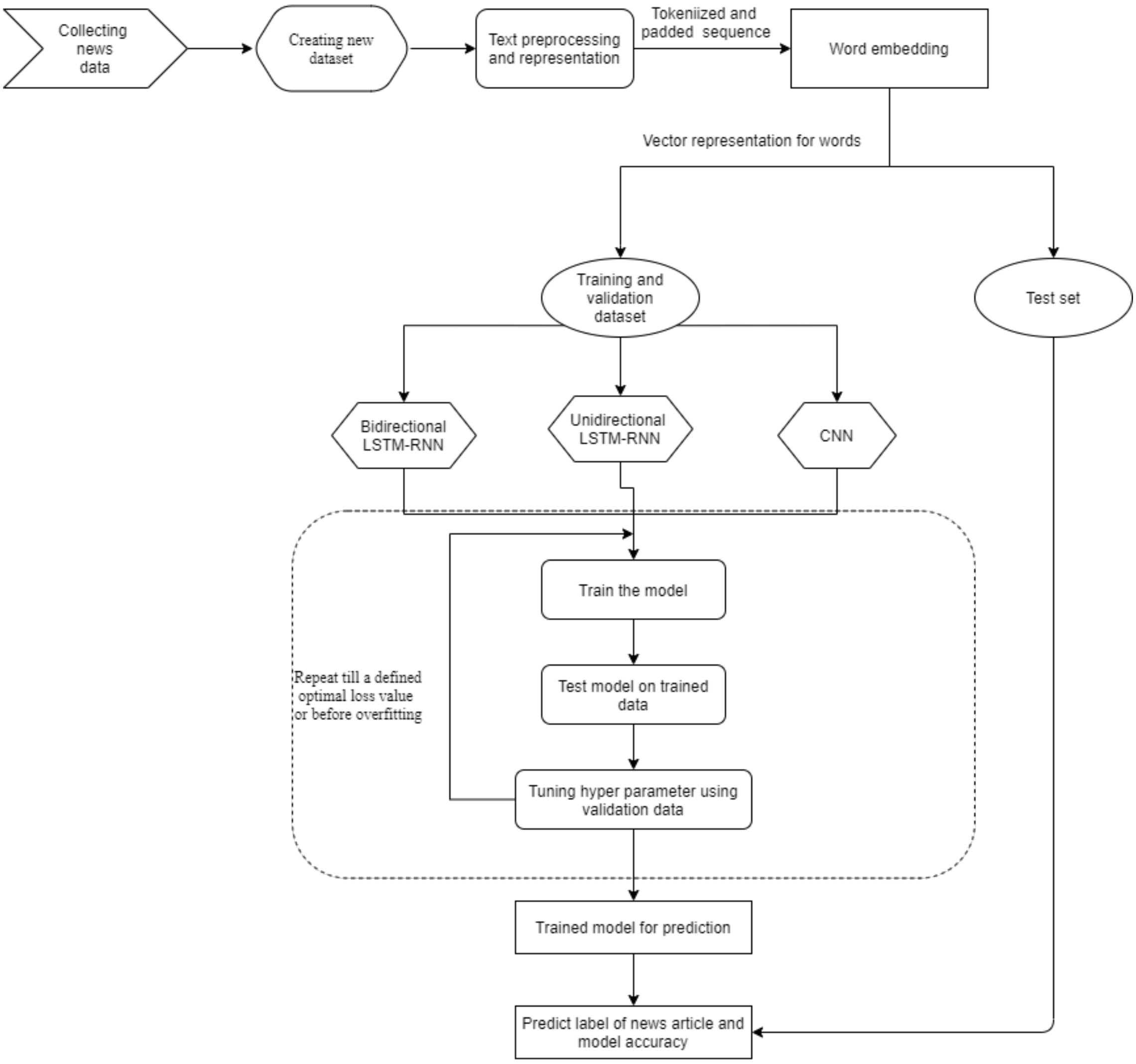

Collecting news data
Creating new dataset
Text preprocessing and representation
Tokeniized and padded sequence
Word embedding
Vector representation for words
Training and validation dataset
Test set
Bidirectional LSTM-RNN
Unidirectional LSTM-RNN
CNN
Train the model
Repeat till a defined optimal loss value or before overfitting
Test model on trained data
Tuning hyper parameter using validation data
Trained model for prediction
Predict label of news article and model accuracy

Figure 1.1: Workflow Diagram For COVID-19 Related Misinformation Detection.

Chapter 2

Literature Review

2.1 Existing Fake News Dataset Related Paper Review

In the world of advance technology statistical misinformation or false news are taken down instantly and as a result there is scarcity of labeled benchmark datasets when it is needed for the development of machine learning models for automatic fake news detection. . The publicly available dataset LIAR is the largest source for fake news detection [8]. 12.8K human classified short statements from poltifact.com are included in the LIAR dataset. The duplicate news were merged and were divided into 6 labels. for authentication of the dataset agreement rate is measured by cohens kappa which is 0.82. Five baselines are being used in this research, a majority baseline, LR, SVM, Bi-LSTM and CNN. A hybrid CNN model is being made to integrate text and metadata.To encode the metadata, a matrix is initialized. To produce the final result, the max pooled text representation is then given to the fully linked layer with a softmax function. Majority baseline have an accuracy of 0.204 and 0.208. SVM and LR acquired improvements. Bi-LSTM did not work well because of over fitting. The CNN outperformed all models with highest level of accuracy.
Fake news has no particular language. English-based fake news detection is very much old in research field but low resource languages like Bangla, there are no fake news detection system [16]. So, for that, this paper reflects on detecting the fake news in Bangla from various news sources from the internet. For that, they have collected satire news from renowned websites, collected misleading news and clickbait. They have a dataset of total 8.5k (approximate) news. They also did a human research where they tested if they can detect the fake news successfully or not. The authors used linguistic features and neural network-based models for this news detection. For linguistic detection, they have used lexical feature, semantic feature, metadata and punctuation. For neural network part, firstly, they have used CNN where they use kernel length from 1 to 4 and 256 kernel and for pooling layer, max pool and average pool is used. Again, ReLU is used as activation function. Moreover, they have also used LSTM. In LSTM, they have used bi-directional LSTM as it shows promising result. Furthermore, pre-trained models like multi-lingual BERT model showed great results. They also use several pre-processing methods like text normalization and Bangla strop words from Stopwords ISO. Again, they also use micro F! scores to evaluate different kind of methods. The data set is split

into 70:30 train-test ratio. For Bi-LSTM, CNN, and BERT based experiments, the hyper-parameters are Optimizer: Adam Optimizer), Learning rate: 0.00002, Batch size: 32. The hidden size for BERT model is 756 while in CNN and Bi-LSTM it is 256. In the result, the authors show us that SVM outperforms RF and LR without the result od news embedding and for most features RF outperforms LR model. F1 scores of MP, word embedding (FastText and news), are better than the POS tags and though it outperforms POS, it falls behind lexical features and neural network model. The linguistic features show 91 percent F1 score which is the best result with SVM. Again, the neural networks can not outperform the linear classifier. Bert scored F1 score of 68 percent, which is the best in neural networks-based results but it falls behind the performance of SVM. Comparing to human baseline dataset, the SVM and all linguistic models shows great results and this shows that linear classifiers can detect the errors that human can not.

2.2 Fake News Detection Method Related Paper Review

Fake news identification is a practical issue in natural language processing to minimize human time and effort to identify and avoid the dissemination of fake news.[9]. This survey offers a systematic overview and compares the implementation of tasks, datasets and NLP solutions that have been developed in this field with a discussion of their possible weaknesses. The aim of the automatic detection of fake news is to minimize human time and effort to identify and help us avoid spreading fake news. Automated fake news detection is analyzed from the point of view of NLP in this article. It highlights the technological challenges of fake news detection and how researchers identify various tasks and devise ML solutions to tackle this problem. In addition, the advantages and drawbacks of each assignment are also presented. With a comprehensive comparison of their role descriptions, datasets, model constructions and performances, an overview of research efforts for fake news identification is also assessed. The contribution can be divided into three sections, such as presenting the first detailed analysis of natural language processing with solutions to automatically identify false news. Secondly, the noteworthy problems and their solutions were also illustrated by a comprehensive review of how fake news identification is compatible with current NLP tasks. Finally, for new researchers interested in this topic, a category and overview is used on datasets, NLP approaches and outcomes, including first-hand experiences and open introductions. The results of classification datasets through different machine-learning models were mainly focused on 3 datasets, namely LIAR, FEVER, FAKENEWSNET. The LSTM model of the LIAR dataset showed greater accuracy than models based on CNN. By replacing the reputation history in LIAR with a broad credibility source, the accuracy can be improved by 21 percent. Attention-LSTM has the highest ranking on both verification and evidence-collection tasks on the FEVER dataset. FAKENEWSNET dataset is based on twitter, Castillo etc. As a result, the HC-CB-3 with over 93 percent accuracy was the highest accuracy obtained by a model from this dataset.

As the spreading of fake news resulting in a state of confusion and disorder day by day, the researchers are also contributing in detecting this fake news by adopting several approaches[4]. This study provides the researchers with several types

of methods of deception evaluation on linguistic cue approaches and approaches to network analysis that suggest a fusion between these two approaches to achieve a promising result. This paper provides researchers with a chart of the landscape of methods of assessing misinformation, their main classes and objectives. From separate production sources, these approaches have emerged, using different techniques. Two key types of methods emerge in this research, namely, Linguistic Approaches (data representation, deep syntax, semantic analysis, rhetorical structure and discourse analysis, classifiers) in which fake message content is extracted and analyzed to associate fake language patterns; and Network Approaches in which information from the network, such as message metadata or structured messages, is extracted and analyzed. To fit the study, both types provide machine-learning techniques for training classifiers. These diverse fields need to be considered by researchers. The aim is to provide a survey of the latest research while suggesting a hybrid approach that uses the most powerful methods of deception detection to incorporate a fake news detection tool. In classification tasks, linguistic and network-based methods demonstrated a high accuracy outcome. Therefore, for further review and improvement, a discussion draft for a simple process typology is available, which offers a fundamental build-up for a fake news detection tool. In short, the linguistic processing should be in multiple layers, from lexical to high-level analysis, for optimal efficiency. It is important to make the relationship between method and machine performance clear. A standard dataset in data format should be connected for up-to-date fact testing.
in this age of social media, any news spread all over the world within few seconds whether it is real or fake[13]. It is very helpful if the news is true but it can be very unpleasant if the news is misleading. It is very difficult to detect misleading news, especially as the process of producing misinformation develops. A novel two-path semi-supervised learning approach is then implemented where one path is supervised learning and another path is unsupervised learning, and these paths are implanted with a combined convolutionary neural network to boost output detection. In this age of social media, whether it is true or false, any news spread across the world in a few seconds. It is very good if the news is genuine, but if the news is false, it can be very unpleasant. It is very difficult to detect misleading news, especially as the process of producing misinformation develops. Therefore, where one path is supervised learning and another path is unsupervised learning, a novel two-path semi-supervised learning approach is implemented and these paths are implanted to increase detection efficiency with a combined convolutionary neural network. In the case of restricted labeled data, a semi-supervised two-path convolutional neural network was designed to achieve fake news detection. It is made up of three items, namely CNN shared, CNN supervised, and CNN unsupervised. One path is made up of shared CNN and controlled CNN, while the other is made up of shared CNN and unsupervised CNN. In the learning process, all data will go through both paths and produce the mean squared error loss, while only labeled data will be used to measure the loss from the cross-entropy. By using the weighted sum of the two losses, the model is then optimized. The model was tested by evaluating the PHEME dataset and found that when the training datasets and testing datasets do not share the same distribution, the proposed model performs better than supervised learning models as the supervised learning models would not overfit the proposed model. The findings were compared to the conventional models of machine learning where

the traditional models work worse than the model proposed. Samples created by the TF-IDF, which are too scattered for the traditional model to learn, may be the only explanation for this result. In addition, for certain events in the PHEME dataset that have decreased the Bidirectional Recurrent Neural Network's macro-average values, the class division is not equal. It is, however, noted that with a limited amount of labeled information, the model gives a promising output.

The purpose of this research is to decrease the rapid spread of misinformation[7]. Fake news can break authenticity balance and it changes people's view towards the authentic news. Therefore, to reduce the effects of false news on public developing methods to detect fake news automatically is very much needed. To help with research we are reviewing the survey in two aspects, characterizations and detection.A dataset is being by collecting news from different sources. The dataset contains true news, rumor, hoax, clickbait, spams. Next, extraction of features is performed to represent only the material that is useful. Feature extraction is carried out in two forms: features of news content and features of social content. The model construction comes after feature extraction. Methods based on the key input sources, news content models and social media models have been classified here. The measurement of metrics is used to measure the algorithm's performance. We get to know the accuracy, the recall, the precision. The receiver operating characteristics (ROC) curve offers a way to measure classifier efficiency by looking at the trade-off. Reviews from the perspective of data mining are being carried out in the characterization and identification process of the implementation of fake news media and the review of current detection methods.

In this paper, the authors focus on the TI-CNN which is also known as text and input CNN for detecting the fake news. People all around the world getting effected by the huge level of fake news daily and it had a big impact in US election too [10]. So to detect this fake news more effectively and upgrade the previous works, the authors worked on text and image analysis where they analyze the text and image model by USING CNN. The dataset contains 20,015 news where the fake-true ratio is 11,941:8074. The text analysis is rather is simple as most of the fake news either don't have no specific title, written in capital letters to draw attention. Again, fake news has fewer words than real news. Again, the authors also found that the real news have less question marks than original news. Furthermore, the true news uses more exclusive words and negations but the fake news doesn't as they have fear to get caught. Also, the authors have analyzed parts like lexical diversity, sentiment analysis, image analysis etc. The text branch has two types of features, textual explicit features which are derived from the news text and textual latent feature which is based on CNN. Although CNN is a computer vision thing, it has shown a great work in terms of NLP tasks. Like text branch, image branch has two types feature, visual explicit features where we extract the image and number of faces from a feature vector and visual explicit feature in which they gather information from the image in the news. The max pooling layer is also used in the paper. Again., they have also used rectified linear neural or ReLu activation, regularization, and network training. The textual explicit subbranch and visual explicit subbranch are connected with a dense layer where these subbranches can be learned easily by applying back-propagation algorithm. For textual latent subbranch, the context of word2vec is set to 10 words and for textual explicit subbranch, they add a dense layer and normalization layer. For image analysis, there are 32 filters for each CNN

by a ReLu layer. A max pooling layer is also added with each layer. Among many features tested, the TI-CNN outperforms every other models. They select 100 as word embedding where the precision, recall and f1-measure are balanced, and a model with recall is a really good model. Again, the best batch size is 32 and 64 as it takes less time for each epoch. The best hidden layer dimension is 128, as in 256, the performance drops because of overfitting. For accelerating training process, they set the drop out probabilities from the range of [0.5,0.8].
The topic of fake news detection is a binary classification issue.[12]. The proposed solution defines false news by determining whether the information given in the report is reliable on the basis of quantifying the bias of a published news article and analyzing the association between the news article title and the article body. They also used the Recurrent Neural Network, Long Short Term Memory and Convolutional Neural Network for binary classification. The sentences are first tokenized into terms in order to perform sentence modeling and classification using a simple CNN. Words become the term embedding matrix of d-dimension (input embedding layer). New features are applied to the pooling system, and to form a secret representation, pooled features from separate filters are concatenated with each other. These representations are then followed by one or more fully-connected layers to make the final prediction. In the RNN method, it retains state data over time steps that allow the processing of variable length inputs and outputs. In the form of a reliability analysis of a news story, the whole news articles are of variable length. A word will be viewed as feedback to the present state in order to gain legitimacy in order to explain whether or not a news article is real. RNNs do well for short news articles. An LSTM cell replacements for the secret layer of the basic RNN in the LSTM-RNN. The experiment was carried out across two datasets and the CNN and Vanilla RNN models of Bi Directional RNN-LSTM were found to surpass both.
In many NLP tasks, writers have successfully developed the use of profound learning approaches, such as Recurrent Neural Networks (RNN) and their CNN, which share parallels with counterfeit news and include measuring semantic similarities between sentences and community based surveys[23]. They introduced a feature reduction approach that would involve two layers of the neural network, i.e. CNN and LSTM, with the hybrid deep learning models. To analyze the relationship, four data models have been created. All the features are included in the first model, without pre-processing. In the second model, the unreduced function set is used after preprocessing. Dimensionality decrease[58][59], techniques such as PCA and Chi-square create the third and fourth versions. In this analysis, the most suitable models for use in text data operating in conjunction with the hybrid CNN and LSTM model are explored. Once the characteristics are chosen for each of the four models discussed above, they will feed into the CNN-LSTM architecture. The first layer of the model is the embedding layer, which identifies the headlines and bodies of the input and transforms each word to a vector in size 100. The output of the CNN layer would be fed into LSTM and moved to a fully connected dense layer to create a single position as the final output. PCA and Chi-square are two-dimensional approaches for reducing the scalability of the text classifier used in deep learning models.The last layer of the proposed model is a completely connected dense layer, resulting in one single output. The accuracy without data cleaning and preprocessing is just 78%. After the pre-processing period, however, a very satisfactory precision rate of about 91% was achieved.

Researchers have identified a benchmark analysis to test the efficiency of various approaches for three different datasets[14]. A word classification and counting method is included in the Linguistic Inquiry and Word Count (LIWC) dictionary.In its text analysis module LIWC reads texts from a given dataset and then compares each word in the text to a user-defined dictionary. The dictionary explains the words are related to the socially relevant groups. The percentage of total words in the dictionary is then measured for each group. LIWC can be used to detect deceit in computer linguistics as a basis of functionality. The Hierarchical Attention Network (HAN) is supposed to gather knowledge about the document structure. Since documents are organized in a hierarchical manner (word forms, sentences forms a text), we also establish a document representation by first creating representation of the phrases and then agglomerating them to document representations. The four-stage model is made of. Word encoder, attention word, expression encoding, attention sentence. With its n-gram (Bigram TF-IDF), Naive Bayes has done the best between standard Machine Learning models. In fact, almost 94% of the combined corpus is accurate. CNN, Bi-LSTM, C-LSTM and Conv-HAN are the most promising of these NN based models. For smaller datasets, separate Neural Network models were proposed for the Naïve Bayes algorithm and for larger datasets.

2.3 Covid-19 News Related Paper Review

COVID-19 situation is getting worse every day and with that misinformation regarding covid-19 is spreading like wildfire in social media. Especially, misinformation that contains emerging disease spreads more than normal misinformation [15]. Also, redundancy of the same fake news on one's social media makes that person believe about the news more easily. Computational techniques have been developed to identify this misinformation automatically. To aid these computational methods, a dataset has been created Named COAID. This dataset contains fact checked fake and true news from different sources. To collect and verify the news several sources are being used. The first way to extract news was to use several keywords to look for COVID-19 related articles. The correctness of the articles is being checked by editors of fact checkers and by comparing them with verified datasets. Next, reliable and unreliable websites is being identified to collect healthcare news. The Google Cloud Natural Language API is then used to delete news that is not relevant to health care. Social media engagements such as, twitter posts and replies, Facebook posts etc. are also being collected. To analyze users' sentiments in the data VADER was being used. Next, the top hashtags associated with COVID-19 articles were being analyzed to figure out the most used hashtags on fake news and true news individually.Some methods are applied for misinformation detection task. The methods are: SVM. LR. RF. CNN. BiGRU, CSI, SAME,̌ HAN. Some metrics used to test the algorithm's performance are: PR-AUC, Precision, Recall, and F1 score. User feelings analyzed using VADER indicate that fake news is more negative and has stronger feelings than real news posts. Keras is used to apply the models. Hashtags used in false and real news posts are also not equivalent. In various subjects such as, false cure and several conspiracies, the hashtags on fake articles are more focused. Whereas, the hashtags on true articles are related to healthcare.Some metrics used to test the algorithm's performance are: PR-AUC, Precision, Recall, and F1 score. User feelings analyzed using VADER indicate that fake news is more negative and

has stronger feelings than real news posts. Keras is used to apply the models. Hashtags used in false and real news posts are also not equivalent. In various subjects such as, false cure and several conspiracies, the hashtags on fake articles are more focused. The state-of-the-art approach performs better than simplistic approaches because it integrates user feedback signals and better captures qualitative
Proper communication in this COVID time is really important[24]. So, this machine was designed so that they could track all the news on the internet. Their framework is based on four models: sentiment analysis,text mining approaches, textual networks and latent subject model. They have textual contrast on a dashboard between Italian and English newspapers, which is the alternate way to read the viewpoint on tragic incidents from the mass media. The device is called CO.ME.T.A. CO.ME.T.A. is even for those users who do not trust data analysis, it is optimized to make friendly use. The user interface's intuitive layout is divided between the left control panel, the right plotting space, and the upper-side menu bar with the methods. In the first step, the authors describe the procedure for the preprocessing of multilingual sources, using the work done within the European project as a guide. The dashboard produces the final Document-Term Matrix and cuts sparse words after the pre-treatment stage. In addition, to evaluate the semantic relations, the DTM can be interpreted as an affiliation matrix. They developed a co-occurrence network using a textual network approach and suggested measuring the measure of centrality between words. By scrapping online quest, they created the dataset and gathered 10328 news from different sources. Since the first case was announced in Europe, the authors find that the negative sentiment is illustrated a lot by research. In this pandemic news, the topics are extracted into 5 parts where words are included. The authors believe that it is possible to detect how terms can be linked to the referred topic through the word-topic network, and this network consists of latent topics that are identified with the highest probability through LDA and words.
Misinformation is circulating quickly in this COVID-19 pandemic. The main goal here is to calculate the extent of this fake news.[18]. Basically, hashtags on twitter posts are being used to review and compare with the verified and peer reviewed news. After comparison the false news and account characteristics were being analyzed.There arise some questions in the research. The ability to figure out the actual magnitude of misinformation just using particular hashtags and keywords is undetermined. Moreover, only English tweets were selected in the process, which is a shortcoming.Twitter archiver add-on were being used to search the tweets. Tweets that contained 11 common hashtags and 3 common keywords were identified by analytical tools. Computer generated random sequences selected 50 tweets from the search terms. Accounts were being classified into few categories, verified accounts is considered to be authentic by Twitter. Also, tweets were separated into few categories. The tweets were then cross matched with articles from WHO, PDC etc. The tweets that could not be compared were marked as unverifiable information. Statistics were performed to analyze the tweets and user accounts. Bar graphs were generated, chi square statistic was performed to calculate the presence of misinformation in the unverifiable tweets.After generating the result, a total of 153 tweets included misinformation and 107 tweets contained unverifiable information. It was being found out that informal accounts post more misinformation that official or verified account. Some user accounts were seemed to be highly associated with

misinformation or unverified news.

In this paper, the authors state that, in social media, there are fraction of people who spread misinformation and they have made dataset of 1970 manually annotated tweets[26]. They have categorized tweets in 4 categories as Irrelevant, Conspiracy, true Information and False Information. They also have performed several language models named RNN (BiLSTM,LSTM), CNN(TextCNN), BERT, ROBERTA and ALBERT. The purpose of this paper is to reduce the spread of hoaxes and disinformation, looking forward to this tough time of COVID-19. The data is collected by using tweepy which is python library from accessing twitter API.They have designed a CNN-RNN model and The embedding layer of dimension 300 is utilized to generate a vector representation of tweet text using GloVe. Furthermore, they have also built aRNN-CNN model which employs a single Bi-LSTM. Again, the authors have used three variants of BERT which are Distil_BERT, BERT base, BERT large . They have also used three variants of RoBERTa which are Distil_RoBERTa, RoBERTabase, RoBERTalarge and two variants of ALBERT which are AlbertbaseV 2, AlbertlargeV 2. Moreover, the authors analyze that theRoBERTa-large outperforms every other model with 76% F1-score and with precision of 73.75% and recall of 73.5%. Distil-RoBERTa has precision of 71%, making it favorable for true informative tweets. ALBERT-large outperformed it's smaller version of model with 2.77% F1-score. In the untrained models, the CNN-RNN outperformed from the other networks. LSTM and bi-LSTM has also a great f1 score of 66.8% and 67.87% respectively.

The authors have worked on COVID-19 misinformation on the internet and claims that online social media is the fertile ground to spread the misinformation[17]. There are people who spread this news for political and economic reasons. They have also made a dataset named COVIDLIES which is a dataset of 6761 expert annotated tweets. They evaluate NLI models on misinformation, by equating the class labels agree, disagree and no stance to entailment, contradiction and neutral. They have also used BERTScore to compute a similarity metric in tweet-misconception pairs. The distribution of Agree, Disagree and No Stance is 9.91%, 5.07%, 85.02%. The misconception of stance was found 100%, The highest misconception of Agree is greater than 60% and disagree class misconception was 51%. The COVID-19 misinformation detection systems are data efficient as tit is trained too little to no supervision and flexible as allow to addition, removal or modification of the known ones. The information retrieval approaches that have been used are TF-IDF vectorization of tweets and misconceptions and NLTK which is used for tokenization and vectorization. Another information retrieval approach is BM25 algorithm that is a bag-of-words retrieval technique which retrieves documents in decreasing probability of relevance of the query term. For semantic similarity, they have used two approaches: Cosine similarity and BERTScore. For non-contextualized word embedding of cosine similarity, they used 300D GloVe trained on 2014 Wikipedia and Gigaword embedding and for contextualized embedding of BERTScore, they sued a pre trained BERT-LARGE model. TF-IDF and BM25 outperforms average embedding techniques and BERTScore captures the similarity accurately as well, which makes BERTScore very much accurate than all other techniques. They have trained linear classifiers on three common NLI datasets, SNLI, MultiNLI and MedNLI. Performance of F1score of No Stance shows high, almost 89%. BERTScore(DA)+ SBERT(DA)(On MultiNLI) acknowledges 41.2% F1 for the agree class, alongside of

highest macrop Precision (55.9%) and F1(50.2%).
According to the authors, we are fighting "infodemic" along with COVID-19 pandemic, which means there is fake news and rumors regarding COVID-19 which is causing a great deal of harm[19]. They have merged down a dataset of 10,700 social media mews regarding social media news and divided the news in two sectors: real and fake. They collected the fake news from the fact checking websites like PolitiFact, Snopes, Boomlive etc and the real news from tweets of WHO, CDE, ICMR etc. Data analysis techniques were conducted and four machine learning baselines were implemented. The vocabulary size of the dataset is 37,505 and 5141 common words in both fake and real news. There is 52.34% of real news and 47.66% of fake news. The have split the dataset into train which is 50%, validation and test which is each of 20%. The ML models ran on validation dataset have shown great results as the best F1 score of 93.45% is achieved by SVM following the next best of 92.75% from the Logistic Regression. The Decision Tree and Gradient Boost have also shown great performance as they have F1-score of 85.25% and 86.82% respectively.
To stop spreading the infodemic of rumors and misinformation during COVID-19 pandemic researchers are trying to create various model and for creating that preparing COVID-19 false news dataset is essential, fakecovid is a datset for COVID misinformation[22]. The data for the dataset were mainly collected from poynter and snopes. The data are separated in different parts title, published date, article, country, language, news url, category etc. 5182 articles from 105 countries from 92 fact-checkers have been collected. The news articles are being annotated and fact checked by annotators and fact checkers respectively. Data cleaning is being done by removing faulty URLs and removing duplicates. NLP preprocessing is being applied to remove unwanted information from data. BERT based classification model without fine-tuning is used as a single view to measure the performance of the algorithm. Each training process continues until the restriction or validation loss is continued and the learning rate is 0.001. The main problem with the fakecovid dataset is it is not labelled properly. Also, as this dataset contains news from various countries, in various languages, some articles contains mixed language news and unwanted characters inside the articles. Though fakecovid has good quantity of fake COVID-19 news but the quality of the data is questionable.

Chapter 3

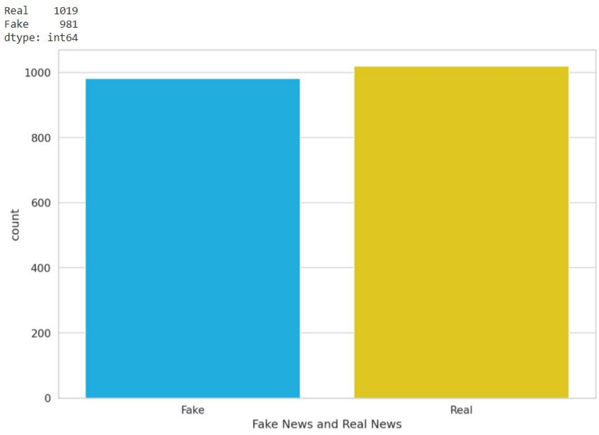
Data Collection And Preparation

3.1 Data Collection

Fake news detection have been addressed as an important topic in research field. There is no vital research about fake news detection model that only focus on COVID-19 related news till now. So basically our target is to make a COVID-19 related fake new model. In order to conduct our research we needed a dataset for training and testing the models and to make the dataset we collected data from various sources. For real news we gathered news from trusted news sources like BBC News, CBC News, CNN, Al Jazeera, CNBC, The New York Times, The Guardian, The Daily Star etc.

As for fake news we had to check different fact checking websites like factcheck.org, poltifact.com and look for the COVID-19 related news in the archive. Also due to advance technology many fake news were taken down immediately and to reduce the insufficiency of the false news we used two COVID-19 misinformation dataset-CoAID[11] and Fakecovid[17].

Then we separated the collected news in two sections, one that is used for training and another used for testing. In the training part there are 2000 news; 1019 fake news and 981 real news. For the test part we separated 300 news; 152 real news and 148 fake news.



```
Real    1019
Fake     981
dtype: int64
```

Figure 3.1: Ratio Of Test News

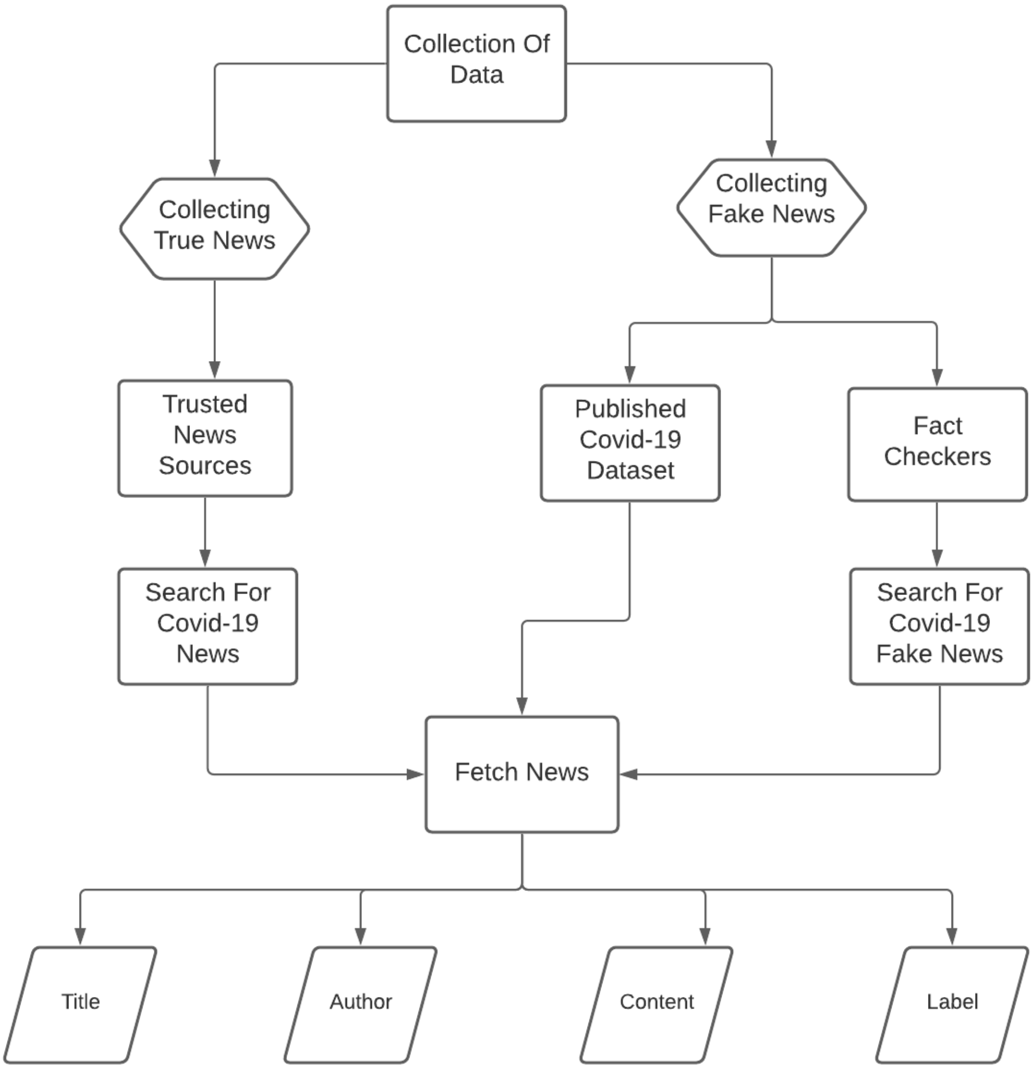
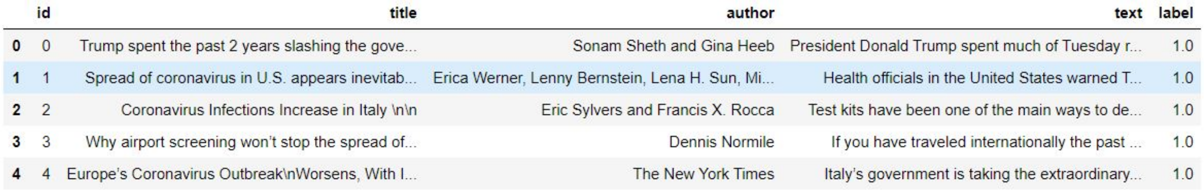


Figure 3.2: Workflow Of Collecting And Creating The Dataset

3.2 Dataset Preparation

As we need a dataset for training and another one for testing the model,we had to make two different dataset that has entirely unidentical news from each other. In both of the dataset, we divided the information regarding the news in several different columns. The first column indicates the number of the news, the second column is the headline of the news which is specified as title, then there is the author column which consists author's name. For the cases where no particular author is mentioned, we have kept the name of the publisher as the author under the author column. The next column consists the main part of the dataset and that is the news. In the last column, the label column points out if the news is true or false. The number '1' in the label column stands for real news and '0' is for fake news.
The training dataset and the testing dataset have quite similar columns. The only difference is that the testing dataset does not have any label column in it. Other than that both part of the dataset consists same type of columns.



	id	title	author	text	label
0	0	Trump spent the past 2 years slashing the gove...	Sonam Sheth and Gina Heeb	President Donald Trump spent much of Tuesday r...	1.0
1	1	Spread of coronavirus in U.S. appears inevitab...	Erica Werner, Lenny Bernstein, Lena H. Sun, Mi...	Health officials in the United States warned T...	1.0
2	2	Coronavirus Infections Increase in Italy \n\n	Eric Sylvers and Francis X. Rocca	Test kits have been one of the main ways to de...	1.0
3	3	Why airport screening won't stop the spread of...	Dennis Normile	If you have traveled internationally the past ...	1.0
4	4	Europe's Coronavirus Outbreak\nWorsens, With I...	The New York Times	Italy's government is taking the extraordinary...	1.0

Figure 3.3: Outlook of dataset

3.3 Data Pre-Processing

Pre-processing is a approach of data mining which converts raw data that is imperfect and inconsistent into a comprehensible format for the computer. To make our COVID-19 dataset machine readable,NLP methods, like converting the characters of the texts into lowercase letters, stopwords removal, stemming, and tokenization was applied, with the implementation of available algorithms in Keras's library.
Before representing the data in vector-based and Machine Learning models and consequently training, the data must be subjected to some filtration, such as stopwords removal, lower casing all the letters, removing punctuation, tokenization and stemming. To avoid punctuation and non-letter characters for each document, we created a standardized operational flow; then the letter cases was lowered for each input.

3.3.1 Stopwords Removal

Stopwords are very generic words that appear in the text which are very small in terms of characteristics and are unrelated in the task. These are the words that are often used in phrases to better link thinking or to help in the form of the language. Stop words are basically articles, prepositions and conjunctions and certain pronouns, for example, 'on', 'a', 'the', 'but', and etc.We decrease the processing time and save space otherwise taken by useless words by eliminating the stopwords.

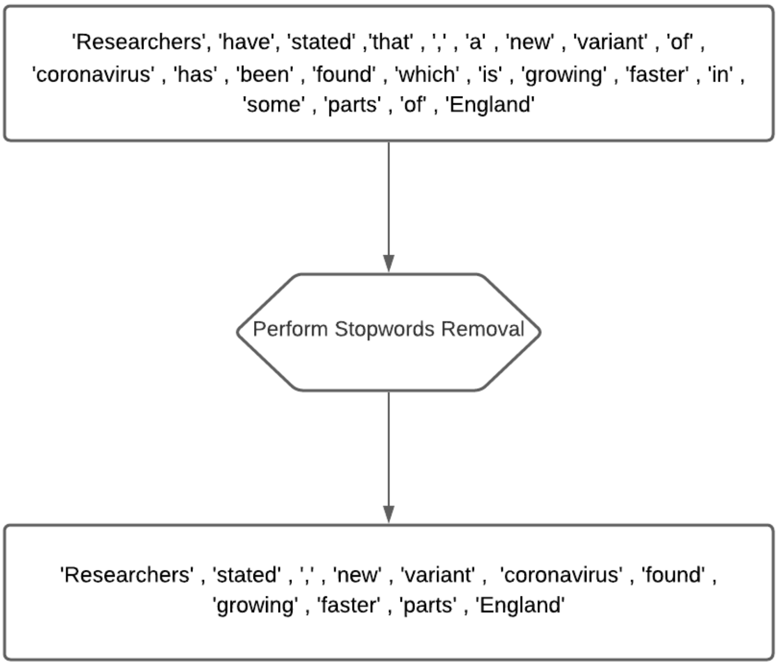

'Researchers', 'have', 'stated' ,'that' , ',' , 'a' , 'new' , 'variant' , 'of' , 'coronavirus' , 'has' , 'been' , 'found' , 'which' , 'is' , 'growing' , 'faster' , 'in' , 'some' , 'parts' , 'of' , 'England'
Perform Stopwords Removal
'Researchers' , 'stated' , ',' , 'new' , 'variant' , 'coronavirus' , 'found' , 'growing' , 'faster' , 'parts' , 'England'

Figure 3.4: Example Of Stopwords Removal

3.3.2 Tokenizer

Tokenizer splits sentences into individual tokens,. In natural language processing exercises, this is a prerequisite where each word needs to be captured and exposed to further study. In addition, we omitted the punctuation from the text. For example, the sentence ”Coronavirus infecting people worldwide” will become ”Coronavirus”, ”infecting”, ”people”, ”worldwide” after tokenization.

3.3.3 Stemming

The next move is to convert the tokens into a regular form. Stemming basically transforms the words to their main form. For instance, the words “Infecting” and “Infected” will become ”Infected” after stemming. As our model needs stemming algorithm which is swifter and more accurate, we have used Porter Stemmer here for that purpose.

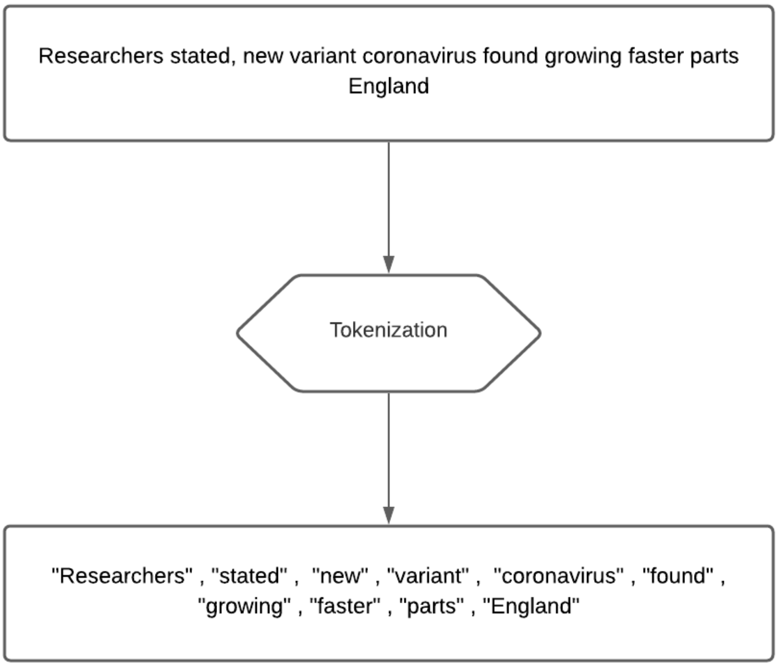

Researchers stated, new variant coronavirus found growing faster parts England
Tokenization
"Researchers" , "stated" , "new" , "variant" , "coronavirus" , "found" , "growing" , "faster" , "parts" , "England"

Figure 3.5: Example Of Tokenization

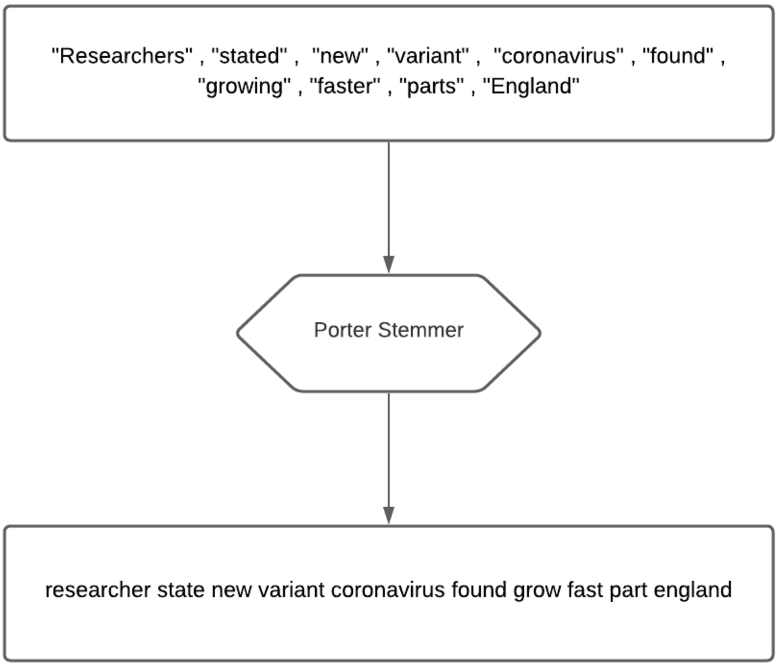

"Researchers" , "stated" , "new" , "variant" , "coronavirus" , "found" , "growing" , "faster" , "parts" , "England"
Porter Stemmer
researcher state new variant coronavirus found grow fast part england

Figure 3.6: Example Of Stemming

Chapter 4

Feature Extraction

4.1 Feature Extraction

Following the execution of the pre-processing steps, we have used several word embedding techniques to extract features which will be fed to our model for learning. In order to be able to read texts by the system, the texts have to be encoded as a continuous vector of numeric values. We have used several state-of-the-art method of word embedding techniques like Word2Vec, GloVe, TF-IDF, BERT and One Hot Representation.

4.1.1 GloVe

An alternative method to create Word embedding is GloVe, also known as Global Vector of word representation. In a textual corpus, it is based on word occurrences. GloVe embeddings have been used extensively for many text mining and natural language processing tasks with great success[20] due to their high quality as textual features. GloVe is based on techniques for matrix factorization on the matrix of the word meaning. A large matrix of data on co-occurrence is constructed and each "word" (the rows) is counted and how often in a large corpus we see this word in "context" (the column).

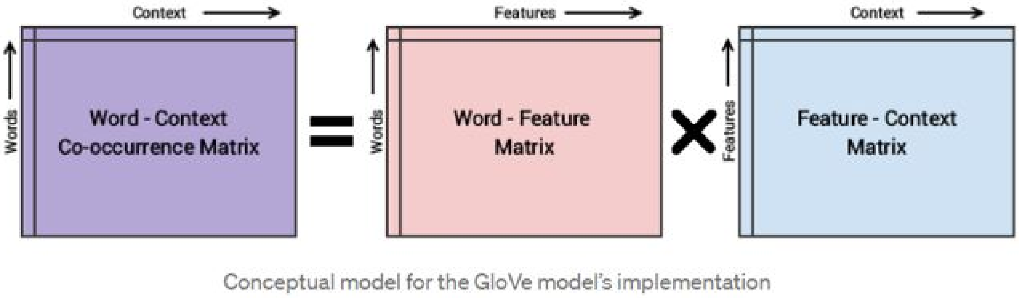


Figure 4.1: Conceptual model for GloVe's implementation

4.1.2 Word2Vec

Word2Vec is a two-layer neural network that is configured in linguistic contexts to reconstruct terms. It takes as its input a large corpus of words and generates a vector space. Word vectors are arranged in vector space in such a way that terms which share similar contexts in the corpus are in close proximity in space to each other. Word2Vec is a specifically computationally powerful predictive model for learning word embeddings from raw text. It comes in two flavors, the Continuous Bag-of-Words (CBOW) model and the Skip-Gram model. Algorithmically, these models are similar. Word2Vec is a simple hidden layer neural network with weights, like all neural networks, and its purpose is to adjust those weights during training to reduce a loss function. However, for the mission on which it was taught, Word2Vec will not be used, but we will only take its hidden weights, use them as our word embeddings, and throw the rest of the model away.

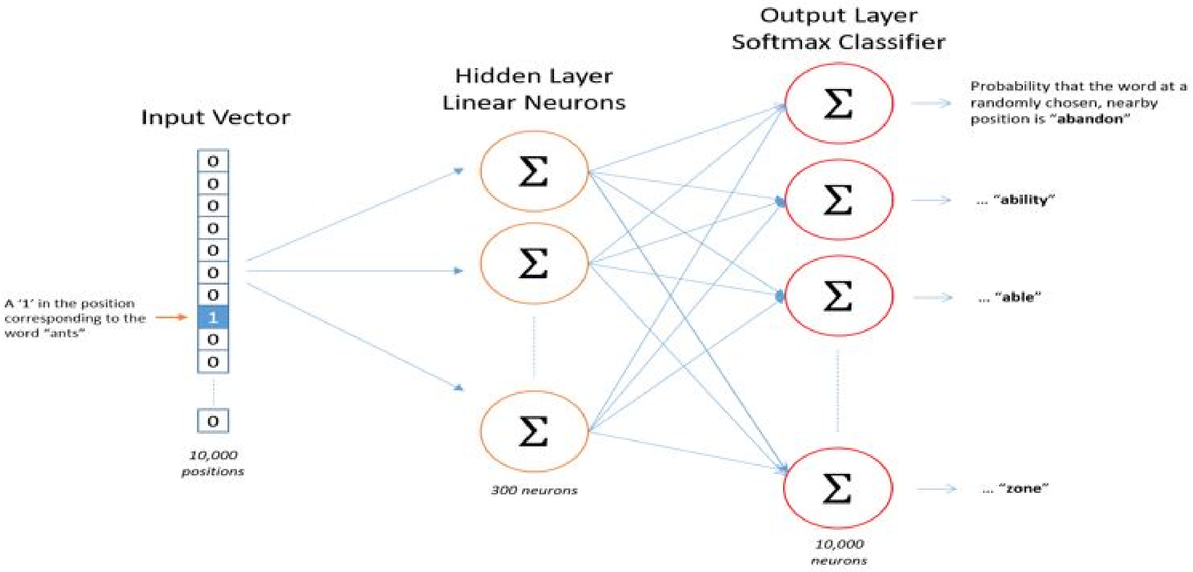


Figure 4.2: Conceptual model for GloVe's implementation

The input and label are one-hot vector but the output is nothing but a collection of target words.

4.1.3 TF-IDF

TF-IDF originated from the IDF suggested by Sparck Jones. (1972, 2004) with heuristic intuition that a question word that appears in many documents is not and should not be a strong discriminator. Less weight than one that occurs in few documents should be given. The classical formula of TF-IDF used for term weighting is Figure 8.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

Figure 4.3: TF-IDF representation

The basic concept of TF-IDF is that the terms in a given text can be divided into two groups from the theory of language modeling: those words with eliteness and those words without Eliteness, i.e., whether or not a word is appropriate (Roberston, 2004), With the subject matter of a specific text. In addition, the eliteness of a word TF and IDF can be evaluated for a given document and in TF-IDF Formulation is used to calculate the meaning of a word in the sense of Document gathering.

4.1.4 BERT

Transformer Representations of BERT or Bidirectional Encoder is a pre-training natural language processing that is essentially a neural network-based technique. The Bert model is based on the architecture of the Transformers. With a large set of unlabeled texts, it is pre-trained. To understand the meaning of each word, the BERT model will consider the whole text passage, so it is bi-directional. This is BERT's breakthrough ability, which is bi-directional and not like the conventional way of training on the sequence of words ordered. Basically, BERT relies on deep network transformers. A fundamental transformer consists of an encoder to read the text input and a decoder to predict the operation. But as the aim of the BERT model is to produce a model of language representation, it only needs the encoder component. Input to the encoder for BERT is a series of tokens, which are first changed over to vectors and then prepared in the neural network. But BERT requires the input to be prepared with certain metadata prior to processing:
1. Token embedding: At the beginning of the first sentence, a [CLS] token is attached to the input word tokens, and a [SEP] token is embedded at the end of each sentence. 2. Segment embedding: Attached to each token is a marker indicating Sentence A or Sentence B. The encoder is therefore allowed to differentiate between sentences. 3. Positional embedding: To determine its place in the sentence, a positional embedding is added to each token.

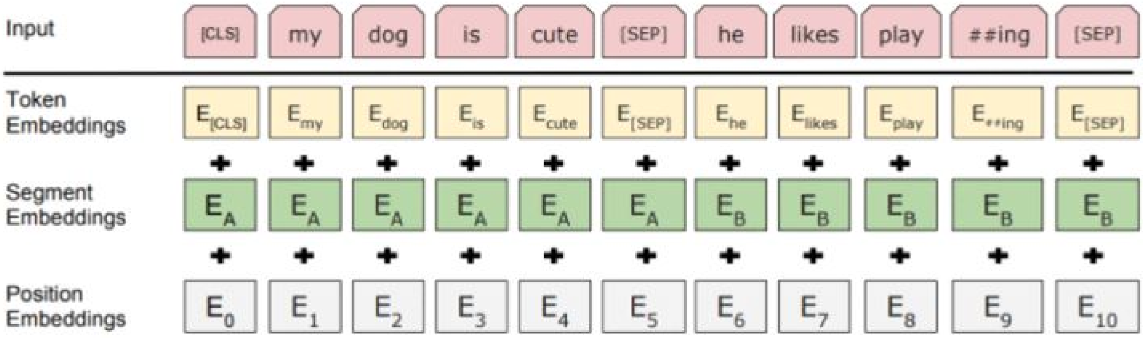
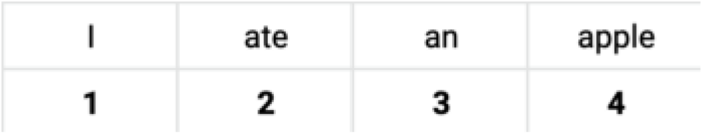

Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} E_{dog} E_{is} E_{cute} $E_{[SEP]}$ E_{he} E_{likes} E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Segment Embeddings
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Position Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Figure 4.4: Visual Representation of BERT model

The Transformer basically stacks a layer that maps sequences to sequences, so a sequence of vectors is the output.

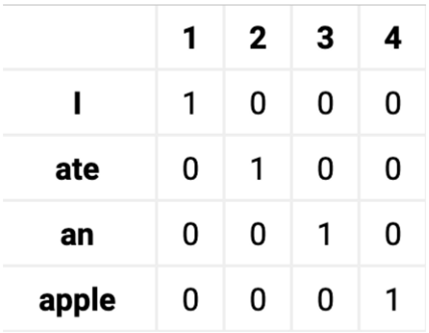
4.1.5 One Hot Representation

The simplest form called one hot embedding or ”1 of N” has been used here. One-hot encoding is a sparse way in which data can be interpreted in a binary string in which only one bit can be 1, while all others are 0. First of all, the vocabulary of sentences is indexed in an array in this system and it is interpreted in the vector according to their index position. ”For example, if the phrase ”I ate an apple” is taken into account, here we can start the vocabulary set indexing, Here, since its location is in index 1, one hot vector representation of the term ’I’ will be [1, 0, 0, 0]. Similarly, the vector representations will be [0, 0, 0, 1] for the word ”apple.” In the source vocabulary, the number of words present sets the dimensions of the vocabulary collection of one hot vector representation. So the vector representation will look like this for the whole sentence. one hot encoding ensures that the system does not take superior numbers for granted. This methodology is also followed by the model while dealing with texts. In addition, illegal states can be easily introduced, updated, and defined. This process makes the implementation quicker.



I	ate	an	apple
1	**2**	**3**	**4**

Figure 4.5: Position of each word in the vocabulary.



	1	**2**	**3**	**4**
I	1	0	0	0
ate	0	1	0	0
an	0	0	1	0
apple	0	0	0	1

Figure 4.6: One hot vector representation of each word in the vocabulary.

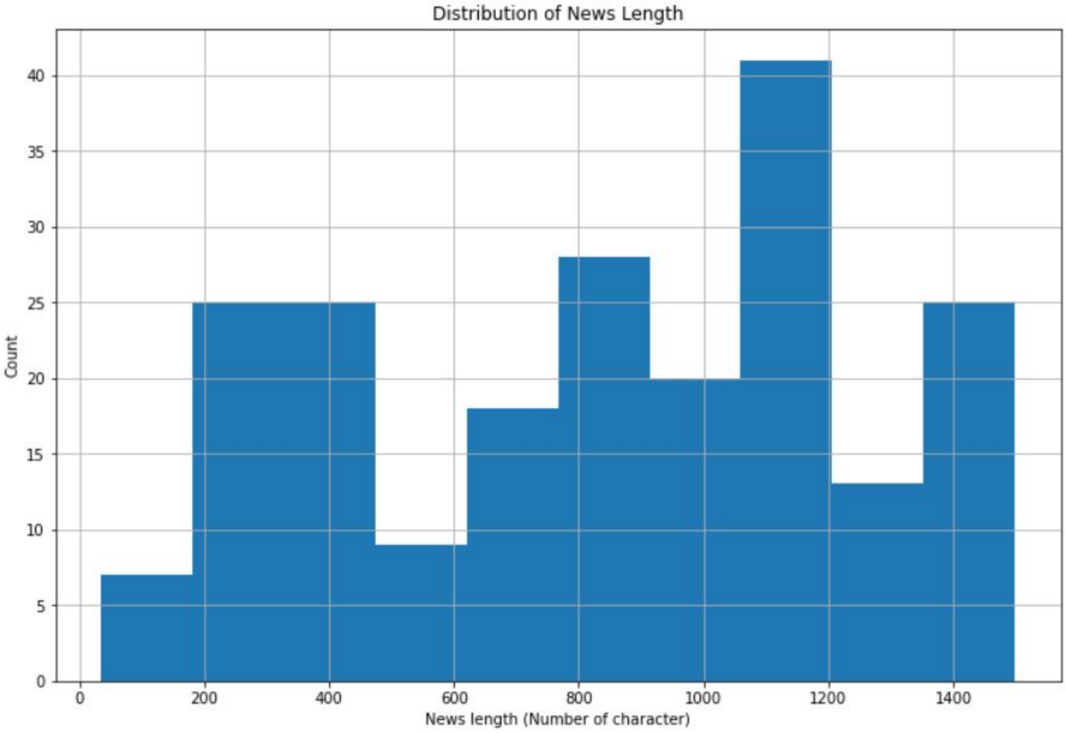


Figure 4.7: Distribution Of News Length

Chapter 5

Deep Learning Models

Fake news identification relevant to COVID-19 is a binary classification problem where the news is either true (1) or fake (0). To solve this natural language processing problem, there are many machine learning methods, but we will mainly concentrate on Convolutional Neural Network (CNN), Long Short Term Memory (LSTM) and Bi Directional RNN-LSTM.

5.1 Convolutional Neural Network (CNN)

In a neural network where links between nodes do not form a rotation is a Convolutionar Neural Network. Even though CNN is used in computer vision but they have shown promising results when applied to various NLP tasks. The phrases are tokenized into words by performing classification on the basis of CNN and sentence modeling first. By converting words, known as the input embedding layer, a word embedding matrix of the d dimension is formed. We can present this input embedding layer as a function, y=f(x), where x is input, y is output, and both are tensors. For translating sentences into sentence matrices, there are different word embedding models available, namely, word2Vec, GloVe, FastText, etc. By applying convolutional filters of various window sizes, a new feature representation is created from the input embedding layer. The Max pooling method is used to establish a hidden representation. These secret representations are accompanied by one or more completely related layers to make the final prediction.

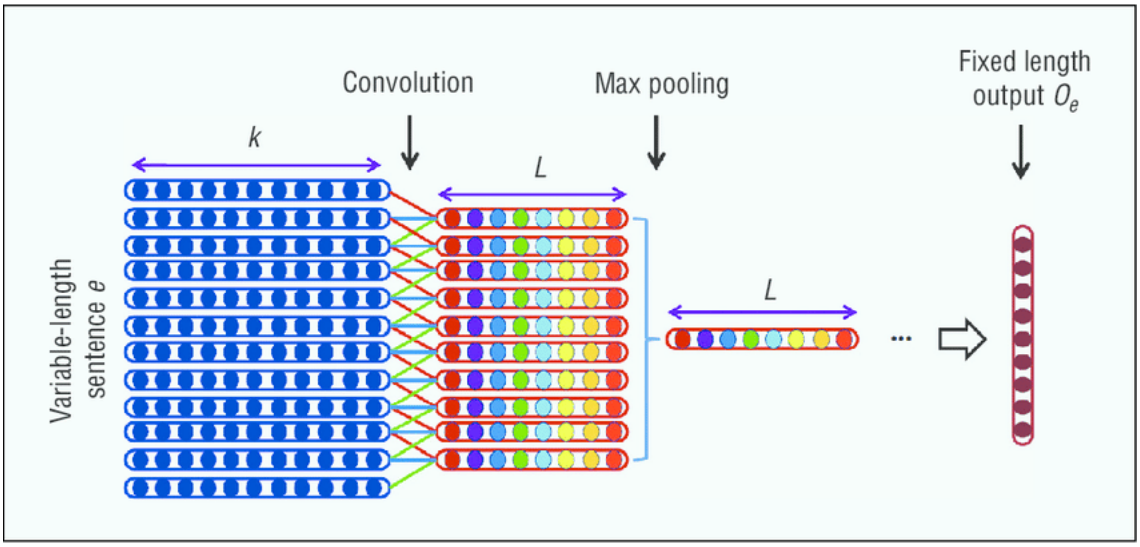


Figure 5.1: General Architecture CNN.

When the article is lightweight, CNN performs better and quickly, but its efficiency and precision is very low in the event of a long sequence of words.

5.2 Long Short-Term Memory - Recurrent Neural Network (LSTM)

It is an updated recurrent neural network (RNN) version that makes it easier to memorize past memory data. RNN has a problem of gradient vanishing and bursting, which in this model is solved. For training the model, it uses back propagation. Three gates are present in an LSTM network, such as the input gate, output gate, and forget gate.

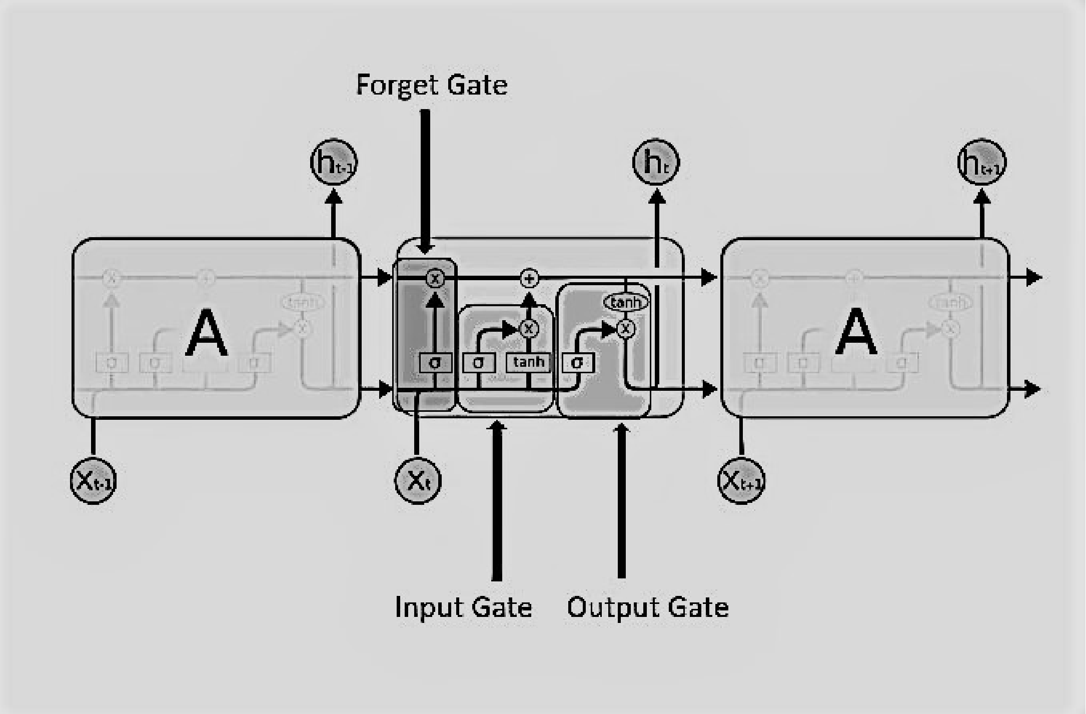


Figure 5.2: General Architecture LSTM.

The input gate chooses which input value should be used to adjust the memory. The sigmoid function defines the values to be allowed to pass through 0,1 and the tanh function determines the value level by giving the weighting value from -1 to 1.

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

The Forget gate, which is determined by the sigmoid feature, discards the unnecessary information from the block. By looking at the previous state and material input, it determines and returns a number between 0 and 1.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

The output gate gives the output chosen by the block of input and memory. Sigmoid function and tanh values multiplied along with sigmoid output create the primary output.

$$o_t = \sigma \left(W_o \ [h_{t-1}, x_t] \ + \ b_o \right)$$
$$h_t = o_t * \tanh \left(C_t \right)$$

LSTMs help to maintain the error that a deep network will back-propagate over time and in lower layers. LSTM works well for long data strings, but when the article is mixed with real and false news, it can often miscalculate the outcome.

5.3 Unidirectional Long Short Term Memory

Unidirectional LSTM is an architecture of RNN with one secret LSTM cell layer. An embedded vector that contains the sequence of sensor events forms the input layer of this model. N LSTM cells are then completely linked to these inputs and have recurring ties to all LSTM cells. The classification task is carried out by a thick output sheet. For all LSTM dependent approaches selected in the validation process, the number of cells (n) and the learning rate are the common hyperparameters. The single cell layer is presented at t, where the input and output states are Xt and Yt, respectively.

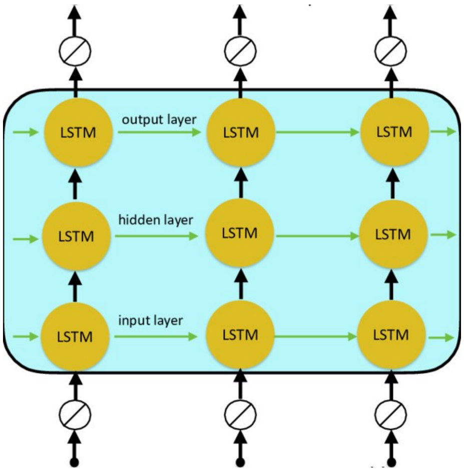


Figure 5.3: General Architecture Unidirectional LSTM.

5.4 Bi-directional Long Short-Term Memory - Recurrent Neural Network

The right model is Bi-directional LSTM-RNN or simply Bi-directional, predicting large sequences of text and text classification. With two different hidden layers, the bi-directional model steps through the input sequence in both forward and backward directions, which allows the device to better understand the context. This ties the two different hidden layers to the same layer of output. The news articles are first pre-processed and a binary mark is set as 1 for fake news and 0 for real news for each COVID-19 news story. By using the regular LSTM updating equations, both the forward and backward layer outputs are measured. The BDLSTM layer produces an output vector, *yt*, in which the following equation is used to calculate

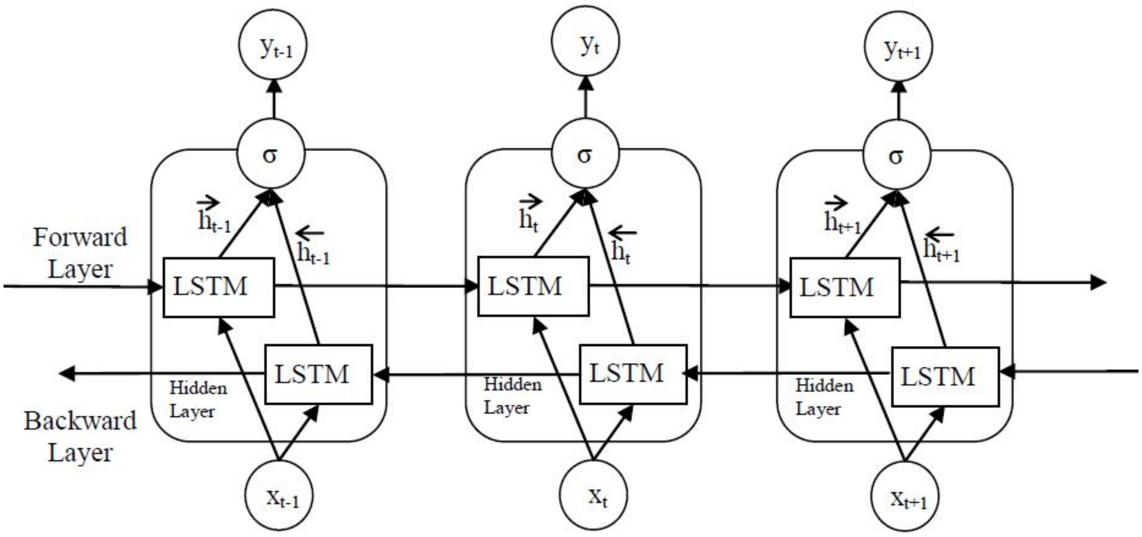


Figure 5.4: General Architecture Bi-Directional LSTM-RNN.

each element: $y_t = \sigma(h_t^{\rightarrow}, h_t^{\leftarrow})$ Where, σ function is used to combine the two output sequences. After that, using the global peak pooling layer, the maximum values are deduced from each filter. The performance further passes through multiple hidden dense dropout layers. Finally, the prediction of the article being false or true is determined by the softmax sheet. To increase accuracy and decrease the loss function, the model is iteratively educated. We considered the cross entropy loss to classify the fake news post. BDLSTM is better at understanding the meaning of the news. There are several papers that are a mixture of true and false data that can confuse the method in forecasting the outcome. BDLSTM overcomes this issue and understands the entire premise of the article in order to foresee the best outcome for it. It also deals substantially with the long data series.

Chapter 6

Machine Learning Algorithm

6.1 Support Vector Machine (SVM)

Support-vector machines (SVMs, also support-vector networks) on machine learning is supervised models of learning with related learning algorithms which analyze information from classification and regression analysis. SVM- Universal learners [1] are support vector machines. SVM's remarkable function is its ability to learn independently despite the dimensionality of space for functions. The Hypothesis's complexity is calculated by SVM based on the margin separating the hypothesis Not the number of features of the plane [8]. SVM is a fast and precise service algorithm for classification with a small amount of data to be assessed that works well. Within the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features you have) with the value of each function being the value of a selected co-ordinate. Then we carry out classification by identifying the hyperplane that separates the two classes. Imagine two tags: red and green. There are two characteristics for our data: x and y. If it's red or green, given a pair of (x,y) coordinates, we want an output classifier. A vector for assistance The machine takes these data points and produces the best hyperplane differentiates the tags (which is simply a line in two dimensions). This side, this line The boundaries of the decision are: We're going to classify something that comes down to One side is orange, and whatever falls on the other side is red. But how is it possible to choose the right hyperplane? It's the one in the SVM, the one that Maximizes the margins for the two tags. In other words: the hyperplane with the hyperplane with The largest distance to each tag's nearest feature.

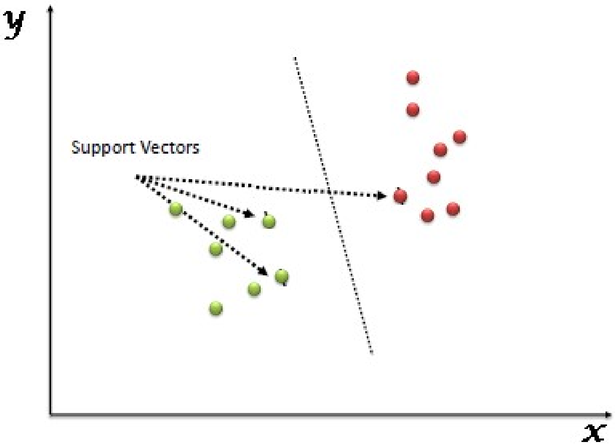


Figure 6.1: SVM Graph

We won't always get linear hyper-planes, however. Occasionally, the limits of decisions can be non-linear. We're not going to have a linear hyper-plane, In the example below, between the two groups, so how does SVM define Both of these classes?

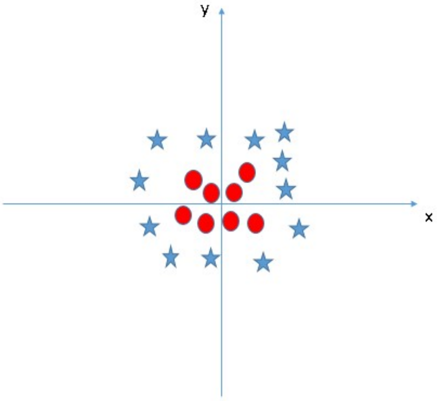


Figure 6.2: SVM Decision boundary xy

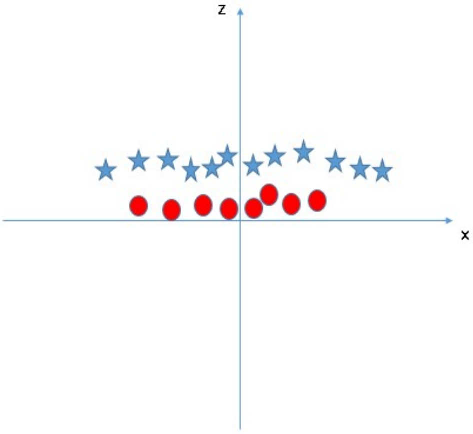


Figure 6.3: SVM Decision boundary xz

By adding extra features, this issue is solved. We are going to be introducing here is the new z=x^2+ y^2 function (Figure 6.4). A system the kernel trick call can be found in the SVM algorithm. The SVM The kernel adopts a low-dimensional space input feature that converts it to a larger dimensional space, i.e. converting a non-separable problem into a higher dimensional space Separable Problem. It is primarily advantageous for the non-linear problem split separation. Simply put, it does some extremely complex conversions of data, then works out the method of separating the data-based information on the labels or outputs defined by us. When we take a look at the hyper-plane looks like a circle in the original input space:

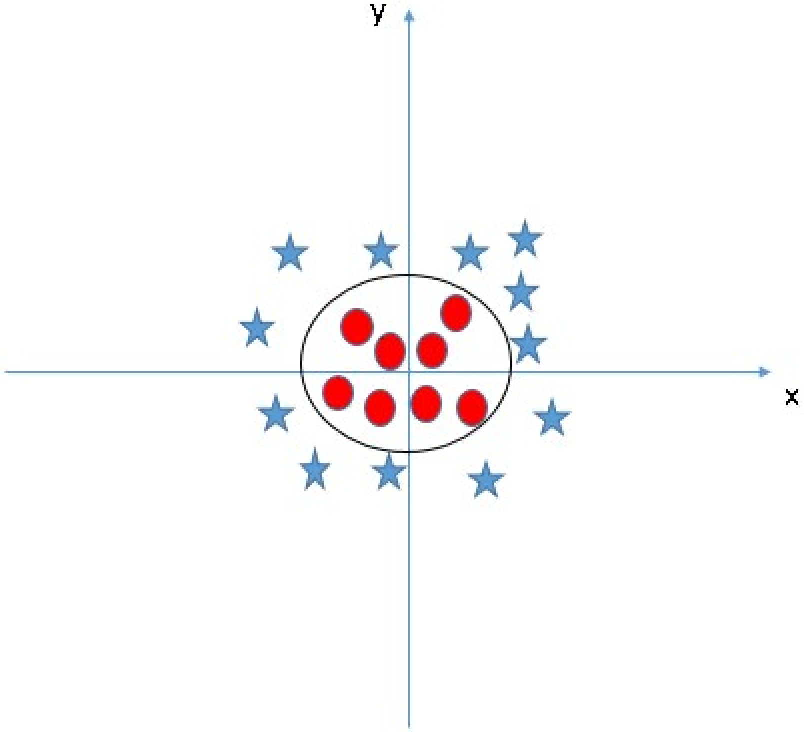


Figure 6.4: SVM decision boundary circle

There is a kernel trick that allows us to expensively sub-step just calculations. Typically, the kernel is linear, and we'll get a linear classifier. However, by using a nonlinear kernel, we can get a nonlinear classifier Without transforming the data at all, the nonlinear kernel: we are only changing the dot product to the room we like, and SVM will chug along happily. Instead, that's what we do. Let's imagine the new room we want.
z= x^2+y^2, Find out what the dot product looks like in that space:

$$a \cdot b = xa \cdot xb \ + \ ya \cdot yb \ + \ za \cdot zb$$

$$a \cdot b = xa \cdot xb \ + \ ya \cdot yb + \ (xa^2 + ya^2) \cdot (xb^2 + yb^2)$$

We call this a kernel feature that is going to be used by SVM. We can use the kernel trick easily to discover the decision boundary from our Data mark, non-linear.

Now we need to first apply SVM in Natural Language classification. Before that we need to preprocess our data such that the SVM input is a number vector. This means we are treating a text as a bag of words, and we have a text A feature for any word in that bag that appears. How widespread the phrase is the value of that feature will be in the text. This method comes down to this technique only to count how many times each word appears in a text and to divide that's about the total number of terms. The term monkeys, therefore, has a frequency of 2/10 = 0.2 in the phrase "All monkeys are primates, but not all monkeys are primates" Primates are primates, and the term is 1/10 = 0.1 in frequency. We chose an appropriate word vectorizer and applied the state-of-the-art model to the In our Fake News linked to COVID-19, SVM named the dataset and evaluated the Outcomes.

6.2 Multinomial Naive Bayes (MNB)

Naive Bayes, which is very popular and quick to implement in Computational terms are a learning algorithm often used in problems with description of text. The classifiers of The Naive Bayes (NB) are a family established, based on the common probability theorem of bayes, of classifiers, To create simple, powerful models, especially in the field of documentation,[11] Classification and Prediction of Diseases. Two models are used, they are,

i) Multivariate Bernoulli Model of Event, ii) Multivariate Model of Event
The Multivariate Event model is referred to as the Multinomial Naive Bayes. Second, we need to understand how Naive Bayes operates in order to understand The notion of Bayes' law. This was suggested by Thomas Bayes (1701-1761), Model of probability and it can be written as:

$$Posterior\ Probability = \frac{Conditional\ Probability * Prior\ Probability}{Predictor\ Prior\ Probability}$$

$$P\left(\frac{A}{B}\right) = \left(\frac{P(A \cap B)}{P(B)}\right) = \frac{P(A) * P\left(\frac{B}{A}\right)}{P(B)}$$

Where,
PA = the previous likelihood of A
PBA = B's probability of condition provided that A occurs
PAB = the likelihood of A condition provided that B occurs
PB= The chance of B occurring

The subsequent probability can be translated as: What is the revised probability? Probability of occurrence of an incident after new information has been taken into account Mindfulness?

Conditional Probability is the chance that one event A will occur when Another case B that has already occurred with some relation to A is named probability conditional.

$$P\left(\frac{B}{A}\right) = \left(\frac{P(A \cap B)}{P(A)}\right)$$

Only when P(A) is greater than zero is this expression valid.
The probability that can be defined as the previous probability is the prior probability. Awareness or belief, i.e. the probability of an occurrence measured prior to the collection of fresh knowledge. This possibility is updated as new data It is available to achieve more specific outcomes. If the preceding Observations are used to evaluate the likelihood, which we call earlier probability.

Now, first, for our COVID-19 fake news classified dataset, the fraction of the data set there is a measure of the documents in each class:

$$\pi_c = \frac{class_c}{\sum_{n=1}^{N} class_n}$$

Then, for a given class, we can find the average of each term for the Our probability estimation, For class c and term w,

$$P(w|c) = \frac{word_{wc}}{word_c}$$

However, because some words have 0 counts, we can perform a low α Laplace smoothing with

$$P(w|c) = \frac{word_{wc}+\alpha}{word_c+|V|+1}, \alpha = 0.001$$

Where V is an array of the vocabulary of all words.
Combining the distribution of probability of P with the fraction of documents belonging to each individual class,

$$\Pr(c)\ \alpha\ \pi_c \prod_{w=1}^{|V|} \Pr(w|c)^{f_w}$$

We have used the sum of logs in order to prevent underflow,

$$\Pr(c)\ \alpha\ \log(\pi_c \prod_{w=1}^{|V|} \Pr(w|c)^{f_w})$$

$$\Pr(c) = \log \pi_c + \prod_{w=1}^{|V|} f_w \log(\Pr(w|c))$$

One issue is that the risk of it appearing again rises if a The term reappears. To smooth this, we take the log of the frequency:

$$\Pr(c) = \log \pi_c + \prod_{w=1}^{|V|} \log(1\ + f_w) \log(\Pr(w|c))$$

Also, we can add an Inverse to take into account stop terms. Weight of Document Frequency (IDF) on every word:

$$t_w = \log(\frac{\sum_{n=1}^{N} doc_n}{doc_w})$$

$$\Pr(c) = \log \pi_c + \prod_{w=1}^{|V|} f_w \log(t_w \Pr(w|c))$$

Although the stop words have been set to 0 for this specific word already, To generalize the function, IDF implementation is applied to the use case. As we can see, the IDF has little, as we have excluded the stop words effect. However, it makes the model more precise for smoothing our optimal model, therefore, is,

$$\Pr(c) = \log \pi_c + \prod_{w=1}^{|V|} \log(1 + f_w) \log(\Pr(w|c))$$

To categorize fake news based on public opinion (Fake/Real), this was the algorithm we used. With a label like False or Real, for each one, we're using the Bag in which word order is disregarded and based instead on the amount of each word's occurrences. Each document is known as a "bag" of words that consist of multiple sets of word expressions to train the data On the MNB classifier and evaluate with the model's output.

6.3 Random Forest Classifier (RFC)

A random forest is an ensemble classifier that estimates on the base of the combination of various decision-trees. Effectively, with variety of decision tree classifiers, it fits different subsamples of the dataset. Also, each tree within the forest was founded on a random subset of the most effective characteristics. Finally, of all the random subsets of features, the most effective subset of features is provided to us by the act of activating these trees. to come up with each individual decision tree, an attribute selection indicator like the info gain, gain ratio, and Gini index of every attribute is employed. Instead of using just one classifier to predict the goal, we use multiple classifiers in the community to predict the goal rather than only one classifier to predict the goal[25]. based on each tree, the independent random sample is predicated. Each tree votes, and also the commonest class is chosen as the final outcome, during a classification problem. It operates in four steps: I) Pick random samples from a given dataset. II) Create a decision tree for every sample and find a prediction result from each decision tree. III) Execute a vote on each expected result. IV) because the final prediction, pick the anticipated result with the foremost votes.

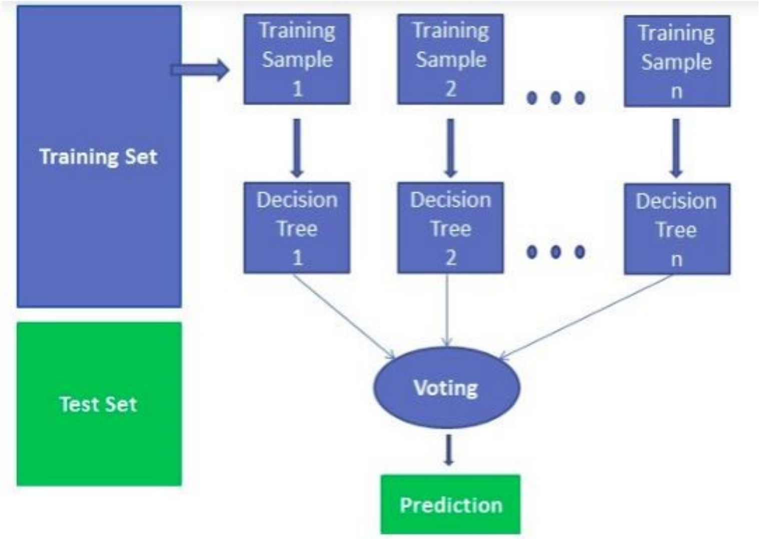
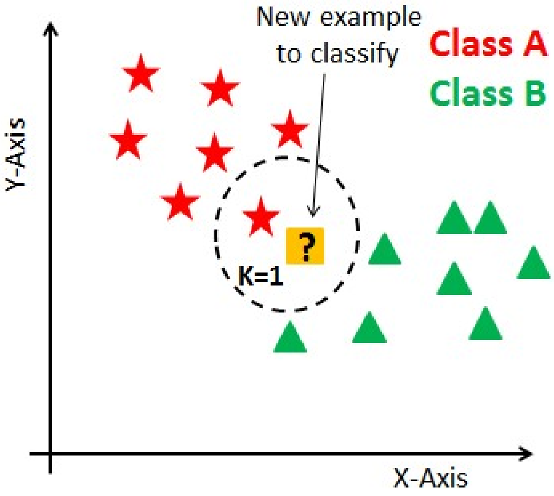


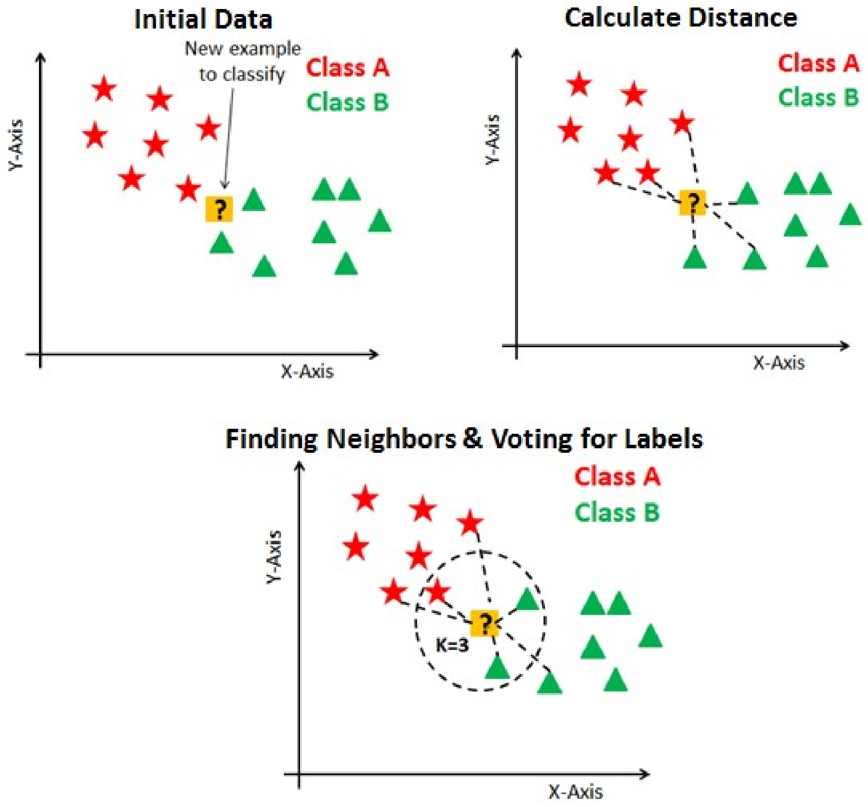
Figure 6.5: Random Forest Classifier Algorithm

6.4 K-Nearest Neighbor (KNN)

A simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems is the K-nearest Neighbors (KNN) algorithm. The KNN algorithm in close proximity implies that there are similar things happening. In other terms, related items are close to each other. The system considers the k closest neighbors of the training documents to describe a new text and uses the categories of the k closest neighbors to weight the candidate category[2].The KNN classifier determines the class of a data point via the principle of majority voting. The nearest 5-point groups are checked when k is set to 5. Estimation is completed according to the dominant class. Similarly, kNN regression is used to take the mean value of the 5 closest points. Now, let's see what KNN does when k=1. Assume P1 is the point that the mark has to estimate. Next, you find the nearest point to P1 and then you find the closest point mark assigned to P1. This is what the mark expects.



Let's see what KNN does now, if k>1. Suppose P1 is the point where the label wants to forecast. The k nearest point to P1 is first identified and then points are listed by a plurality vote of its k neighbors. Where each object votes for its class and the class with the most votes, the prediction is taken.



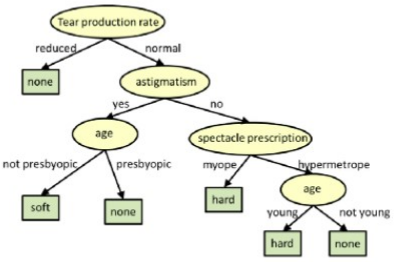
Using distance dimensions such as Euclidean distance, Hamming distance, Manhattan distance, and Minkowski distance to find the nearest connected points, we will find the distance between points.

6.5 Decision Trees (DT)

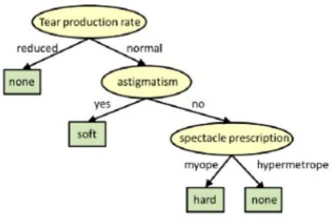
Decision Trees is a non-parametric supervised learning technique used for classification and regression (DTs). With a group of if-then-else decision rules to approximate a sinus curve, decision trees learn from data. The deeper the tree, the more intricate the laws of decision and therefore the more complicated the model would fitter. The goal of employing a decision tree is to construct a training model that may be accustomed to predict the category or meaning of the target variable by studying simple decision rules obtained from previous data (training data). a decision tree could be a classifier defined because the recursive partition of an instance space. the decision tree consists of nodes that form a rooted tree, meaning that it's a root node directed tree with no incoming edges[3].In Decision Trees, we start from the root of the tree to predict a category label for a text. We compare the values of a root attribute with the attributes of the record. We obey the branch similar to that value on the premise of similarity and leap to the subsequent node. sorts of decision trees are supported the kind of target variable we've. Categorical variable Decision Tree and Continuous Decision Tree Variable are two types of DTs. Both categorical and numerical data can be treated by decision trees.

If the dataset consists of N attributes, then evaluating which attribute to place as an interior node at the root or at different tree levels is also a sophisticated step. By just randomly selecting some node to be the root, we are not able to

solve the matter. With low accuracy, it can give us bad outcomes if we pursue a random approach. to resolve this attribute selection problem, researchers have worked and invented several solutions. They suggested using such parameters as Entropy, Gaining Information, Gini Index, Gain Ratio, Variance Reduction, Chi-Square, etc. The values are determined by these criteria for each attribute. The values are sorted and so the attributes are arranged within the tree in line with the order during which the attribute with a high value is placed at the root (in the case of information gain).
In order to avoid overfitting, we use pruning. When pruning, the tree branches are cut off, i.e., the decision nodes ranging from the leaf node are eliminated so the full precision isn't affected. this is often accomplished by segregating the whole testing set into two sets: test data set, D and validation data set, V. Using the separated training data set, D, plan the decision tree. To optimize the precision of the validation data set, then begin trimming the tree appropriately.



Original Tree



Pruned Tree

Here, since it has more value on the right side of the tree, the 'Age' attribute was pruned on the left side of the tree, thereby removing overfitting. In various classification problems, we can use decision trees by using the measures of attribute selection and handling the overfitting by pruning.

6.6 Stochastic Gradient Descent (SGD)

In Stochastic Gradient Descent, instead of the entire data set for each iteration, a few samples are randomly chosen. There is a term called "batch" in Gradient Descent that describes the total number of samples from a dataset that is used for each iteration to measure the gradient. The batch is understood to be the entire dataset for conventional Gradient Descent optimization, such as Batch Gradient Descent.Although it is very beneficial to use the whole dataset to get to the minimum in a less noisy and less random way, when our datasets get massive, the problem arises. Stochastic Gradient Descent solves this issue. The stochastic gradient descent is a little different, as the coefficient update only takes place while the process of training is underway. [5]. In SGD, a single sample, i.e. one batch size, is the only way to carry out each iteration. The sample is uniformly mixed for the iteration and picked. SGD in terms of math,

$$\begin{gathered} for\ i\ in\ range\,(m): \\ \theta_j = \theta_j - \alpha\,(\hat{y}^i - y^i)\,x_j^i \end{gathered}$$

Therefore, for each iteration, we detect in SGD the gradient of the cost function, instead of the sum of the cost function of all instances. The path taken by the algorithm in SGD to find the minimum is typically more noisy than the regular Gradient Descent algorithm, since only one sample is taken from the dataset randomly for each iteration. All this doesn't matter, though, as it doesn't matter how much the algorithm takes as long as we meet the minimums and with slightly shorter training time.

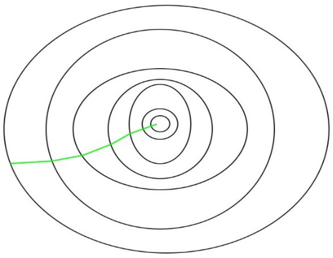


Figure 6.6: Path taken by Batch Gradient Descent

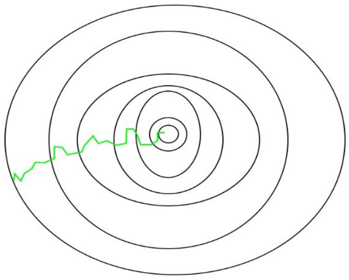


Figure 6.7: Path taken by Stochastic Gradient Descent

One point to remember is that, since SGD is normally more noisy than a standard gradient descent, due to its randomness, a larger number of iterations are generally required to get to the lowest. While the minimum value requires a larger number of iterations than traditional gradient descents, it is also much less computational than typical gradient descents.

6.7 Gradient Boosting

Gradient boosting redefines boosting as an optimization for numbers problems where the aim is to mitigate the loss of the model by adding weak learners by the use of Gradient descent. One kind of ensemble learning is gradient boosting. Ensemble learning integrates a variety of poor learners in contrast to classical approaches to learning to create an efficient and strong learner[6]. Sequence is used to produce the models in the ensemble booster technique, which reduces the error of previously trained models to a minimum, unlike the bagging technique where the models are produced separately. The M additive tree model is combined to learn a predictive model (f0, f1, f2, ., . - f$_M$) to predict the outcomes,

$$f(x) = \sum_{m=0}^{M} f_m(x)$$

The tree ensemble model is optimized by reducing the predicted generalization error L,

L is a loss function which calculates the delta loss of a data point between the target y i and the prediction yi.

$$L = \sum_{i}^{n}(y_i - \hat{y}_i)^2$$

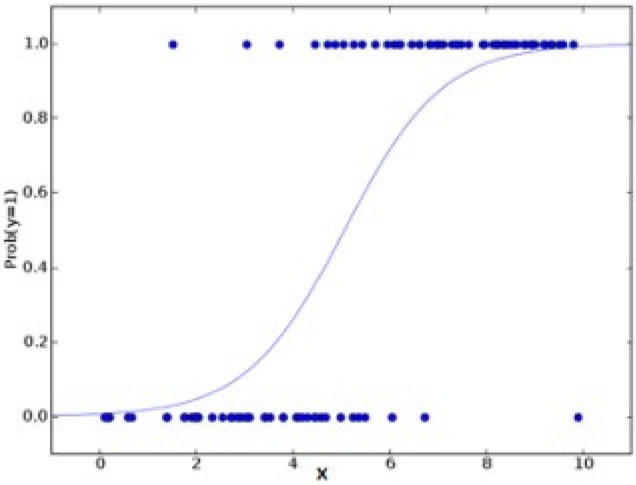
The use of ensemble gradient boosting makes it possible for classifiers to improve their power while reducing their variances and biases. By that the loss of each classifier while simultaneously optimizing its benefits, the nature of the boosting technique could reduce errors.

6.8 Logistic Regression

One of the simple and popular classification resolution algorithms is Logistic Regression. It's named 'Logistic Regression' since it is quite the same because of its basic methodology as Linear Regression. The Logistic Regression condenses the output of a linear function in the range of 0 to 1. It is defined as,

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

And if we plot it, the graph will be S curve,



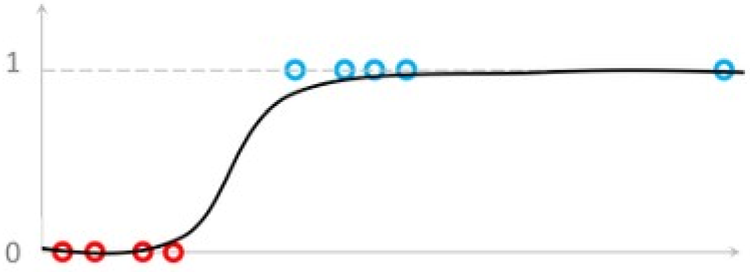
A mathematical function that is responsible for this S shaped curve is the Sigmoid function [21]. In a univariate regression model, let's take t as a linear function.

$t = \beta_0 + \beta_1 x$

Therefore, we can write the Logistic Equation as,

$p(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$

In the formula of the logistic model,
when $\beta 0 + \beta 1x == 0$ then the p will be 0.5,
similarly,$\beta 0 + \beta 1x > 0$, then the p will be going towards 1 and
$\beta 0 + \beta 1x < 0$, then the p will be going towards 0.
Now, when the model of logistic regression detects an outliner, it will take care of it.



But sometimes, depending on outlier positions, it can change its y axis to left or right.

Chapter 7

Experimental Setup

7.1 Creating the Model

We have tried several combinations of word embedding techniques and Deep Learning layers with a view to finding the best possible combination for our Fake News Classification purpose. It is vital to conduct rigorous experiment with several combinations as it will lead us to get the best performing model. However, there are some common approach for all combinations.

First of all, we have defined the number of embedding vector features as 40. Embedding vector takes the input, vectorizes it and passes it onto the next layer. The first layer is the sequential layer which is appropriate for 1 tensor input and 1 tensor output. The next layer is the Embedding layer whose first parameter is the vocabulary size, second parameter is the embedding vector features which we defined as 40. The next parameter is the length of the sentence which we have set as 1250 because from figure 11, we found out the max length to be over 1200.

The next step is the drop-out layer. Dropout is implemented on a neural network per layer. Any of the layers, such as dense completely linked layers, convolutionary and recurrent layers, may be used, such as a long-term network memory layer. The dropout value of 0.2 has persisted, which makes it more likely that a node's output is maintained in a hidden layer.

The output then enters our next layer, which is the specified LSTM layer with 100 neurons. Our output enters the Dense layer after that.The neural network layer is closely connected such that any neuron in the dense layer gets inserts from all the neurons in the previous layer. Figure 22 displays the visual perspective of the above model.

We kept a CNN layer and a Maxpooling layer in a hybrid environment between the layers of Dropout and LSTM. For a 1D convolutional sheet, the input to Keras must be three dimensional. Each input sample refers to the first dimension; we only have 64 samples in this situation, the same as our batch size. The activation function is Relu, which applies the activation function to the rectified linear unit. This returns the regular ReLU activation with default values: max(x, 0), the element-wise limit of 0 and the input tensor. Figure 23 provides a generalized description of the hybrid model.

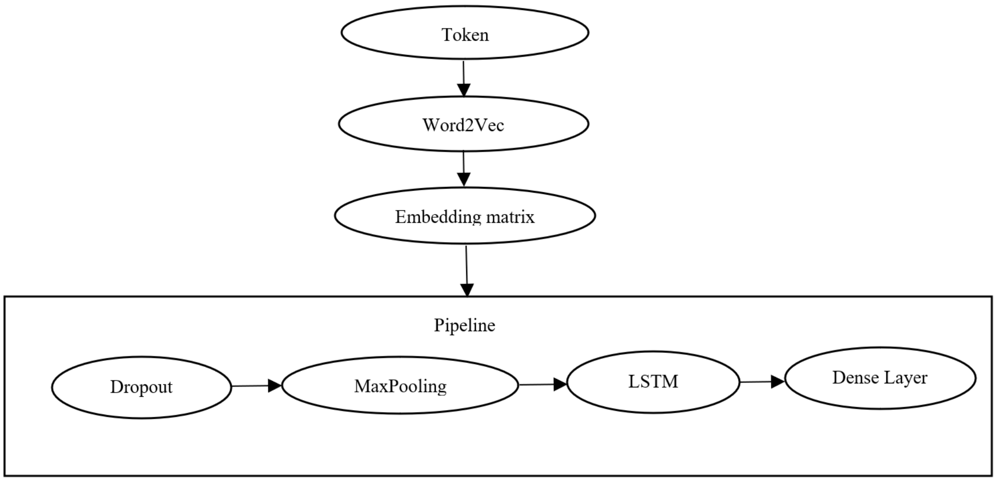


Figure 7.1: Word2Vec and LSTM model

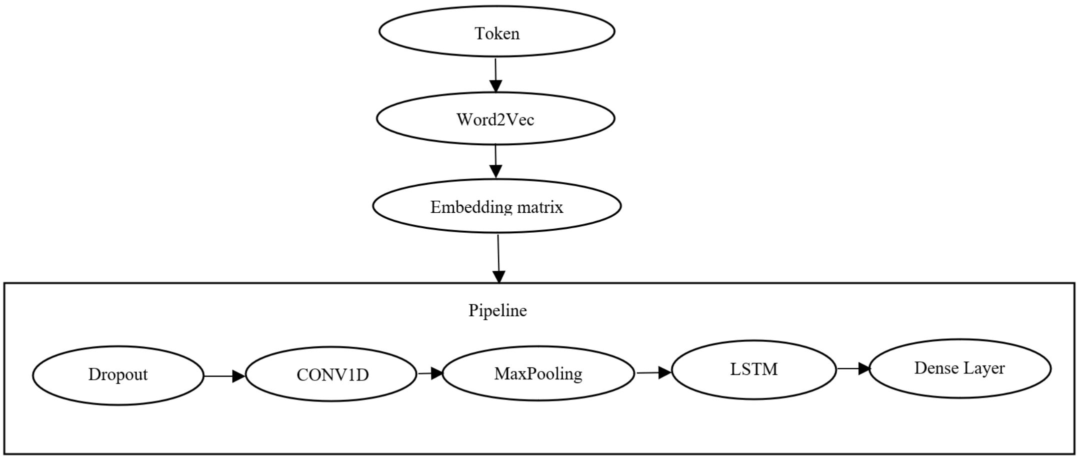


Figure 7.2: Word2Vec and LSTM+CNN model

We have compiled our model with binary_crossentropy as loss, adam as optimizer and accuracy is defined as metrics.

7.2 Training the Model

We have splitted the dataset into training, validation and testing sections. We have kept 15% data for validation purpose and 70% for training with a random state of 97. The rest of the 15% of data is kept for testing our model. We set the epoch as 50 as initial epoch number as it produces consistent accuracy. We have set the batch size as 64. This is the number of samples processed before the model is updated.

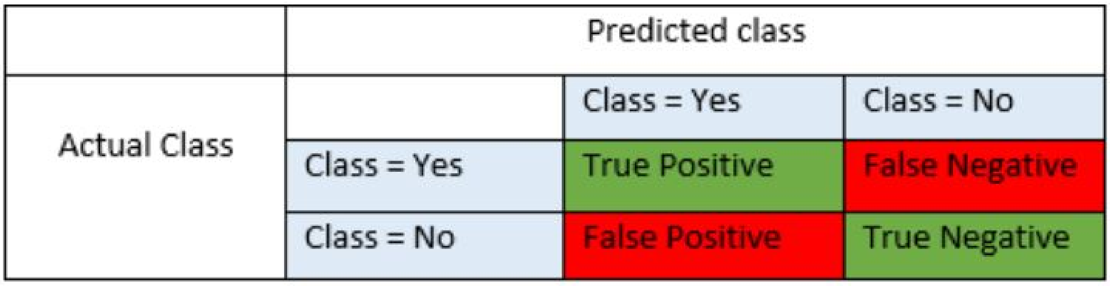
Chapter 8

Performance Evaluation

In order to understand the outcome of the experiment and establishing a notable comparison between different experimental setups, it is vital to understand the performance measures. In this chapter, we will discuss the various performance measures used in our paper.

8.1 Confusion Matrix:

The table form used mostly to report the output of the classification model on the test data set to understand the true values is a matrix of uncertainty.Using the left-most four parameters, all the steps except AUC can be measured.



		Predicted class	
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Figure 8.1: Components of Confusion Matrix

The correctly predicted results are true positive and true negative ones, which is thus shown in green. We want to reduce in our article the false positive and the false negative, and this can be seen in red. It is understandable that it can be very confusing to use these words. So, for a better understanding, we'll explain and fully understand each term.

True Positives (TP) - True positives are effectively predict positive values, which mean that the true class value is yes and that the predicted class value is yes. For instance, the predicted class will say the same thing if the actual class value indicates that this passenger has survived.

True Negatives (TN) - The negative values that are correctly estimated are the real negative values, which means that the actual value is no, and the predicted value

is no as well. For example, if the actual class value says that it did not survive, the predicted class will tell you the same thing.
The value of false positive and false negatives is present when the actual class compares with that predicted.

False Positives (FP) – If the verdict of actual class is no, and the predicted class is yes, false positive results are observed. The actual class says, for example, that the passenger did not survive but would survive, while the class you are predicted to do.

False Negatives (FN) – If the actual class is yes, there would be false positives, so no. For example, the real value of the class means that the passenger has survived, while the class prediction tells you that the passenger will die.

We will continue to measure the precision, accuracy, recall and F1 scores, after learning these four parameters.

Accuracy - Accuracy is established as the most instinctive indicator of success. It is just a relationship between the predicted observation and the findings as a whole. We should conclude that our model is the highest with a high precision. Accuracy is possible, but only if we have symmetrically constructed data bases with almost identical false positive and false negative values. We have to consider other parameters in order to assess the utility of our model.

Accuracy = TP+TN/TP+FP+FN+TN

Precision - Precision is the ratio of positive observations to the overall positive observations predicted correctly. How many passengers have survived marked the question of this metric answer? How many have actually survived? The low false positive rate corresponds to high precision.

Precision = TP/TP+FP

Recall - Recall the proportion of positive observations for all actual class-yes observations is predicted correctly. The answer to the recall question is: How many of all the passengers who actually survived have we labeled?

Recall = TP/TP+FN

F1 score - The F1 score is the weighted precision and recall average. This score also takes into account both false positives and false negatives. It is not so easy to understand intuitively as precision, but F1 is generally more useful than accurate, particularly if your division is uneven. If false positive and false negative costs are equal, accuracy functions very well. It is better to look at accuracy and reminder, because the costs of false positive and false negatives are very different.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Chapter 9

Evaluation

After training, we have observed the Precision, Recall, F1 score and Test accuracy for each combination.It is observed that Bi-directional LSTM with Word2Vec as word embedding has gained a satisfactory test accuracy of 99.3% . The hybrid LSTM+CNN model is the second best performing model which also has Word2Vec as word embedding method. It achieved an accuracy of 98.9% The models where TF-IDF was used as word embedding have yielded the least accuracy rates which, however, is also over 90/The table 1 shows the corresponding Precision, Recall, F1 Score and test accuracy for all the experimented combinations. After that, he accuracy vs epoch and loss vs epoch graphs for each combination is given. Finally the confusion matrix also illustrates that Word2Vec with Bidirectional LSTM is the most effective and feasible model.

9.1 Evaluation of Word2Vec. GloVe, TF-IDF with LSTM and CNN and Bert

Table 9.1: Results

Word Embedding	**Model**	**Label**	**Precision**	**Recall**	**F1 Score**	**Test Accuracy**
Word2vec	LSTM	0	0.99	0.98	0.99	0.986
		1	0.98	0.99	0.99	
Word2vec	LSTM+CNN	0	0.98	0.98	0.99	0.989
		1	0.99	0.98	0.99	
Word2vec	Bi-LSTM	0	1.00	0.99	0.99	0.993
		1	0.99	1.00	0.99	
Glove	LSTM	0	0.99	0.98	0.99	0.986
		1	0.98	0.99	0.99	
Glove	LSTM+CNN	0	0.99	0.98	0.99	0.986
		1	0.98	0.99	0.99	
Glove	Bi-LSTM	0	0.99	0.98	0.98	0.983
		1	0.98	0.99	0.98	
TF-IDF	LSTM	0	0.93	0.91	0.92	0.917
		1	0.91	0.93	0.92	
TF-IDF	LSTM+CNN	0	0.93	0.91	0.93	0.922
		1	0.92	0.93	0.92	
Bert	N/A	0	0.98	0.95	0.96	0.963
		1	0.95	0.98	0.96	

9.2 Graphs

Figure 7.1 to 7.15 illustrate the graphs of accuracy vs epoch and loss vs epoch for each combination respectively. We set the epoch at 50 for experiment and noticed a relatively steadier graph along the way.

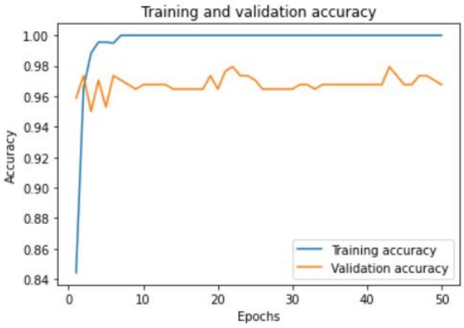


Figure 9.1: BERT accuracy vs epoch

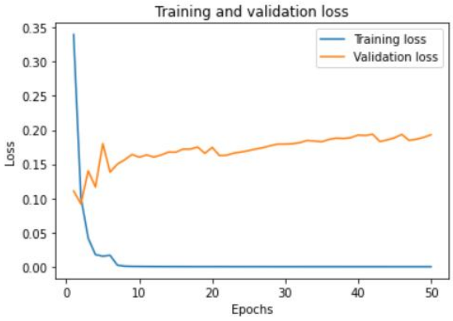


Figure 9.2: BERT loss vs epoch

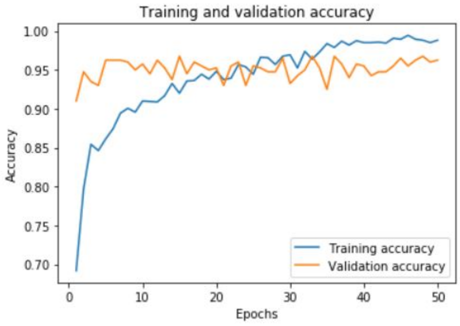


Figure 9.3: GloVe+LSTM accuracy vs epoch

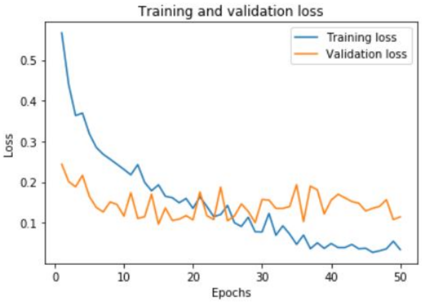


Figure 9.4: GloVe+LSTM loss vs epoch

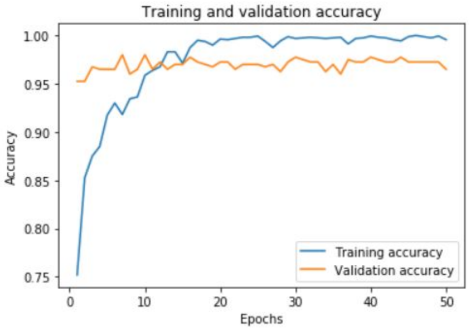


Figure 9.5: GloVe+LSTM+CNN accuracy vs epoch

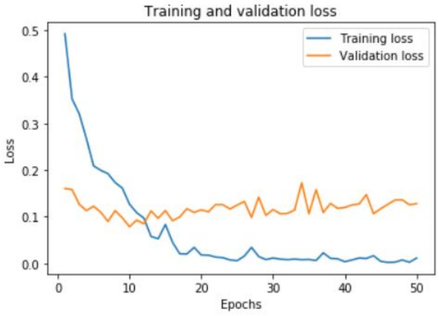


Figure 9.6: GloVe+LSTM+CNN loss vs epoch

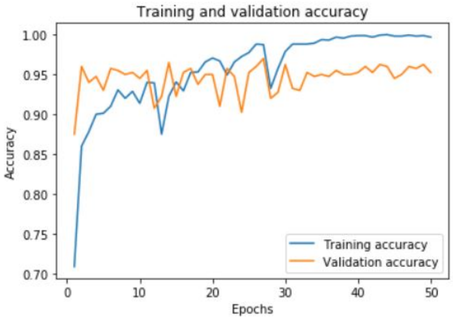


Figure 9.7: GloVe+Bidirectional LSTM accuracy vs epoch

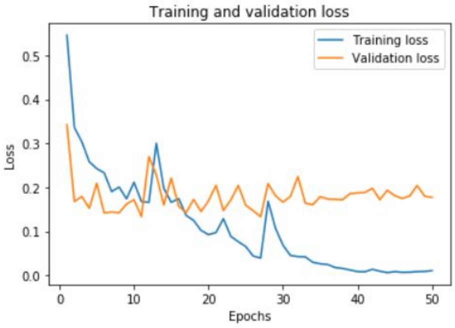


Figure 9.8: GloVe+Bidirectional loss vs epoch

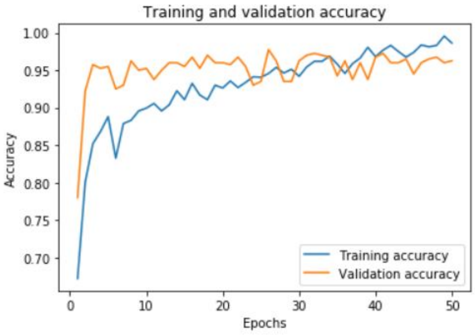


Figure 9.9: w2v+LSTM accuracy vs epoch

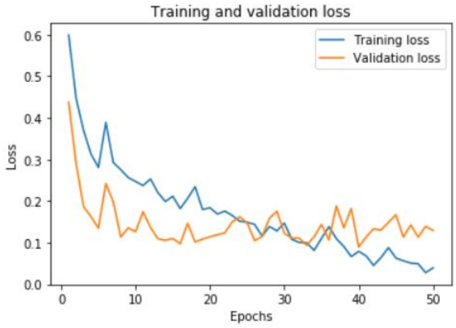


Figure 9.10: w2v+LSTM loss vs epoch

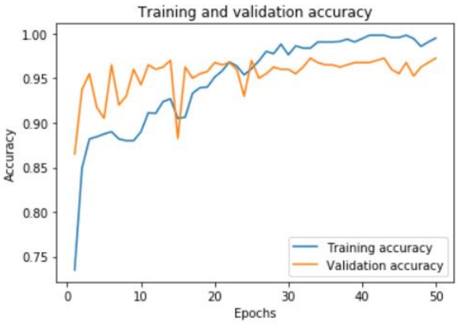


Figure 9.11: w2v+Bidirectional LSTM accuracy vs epoch

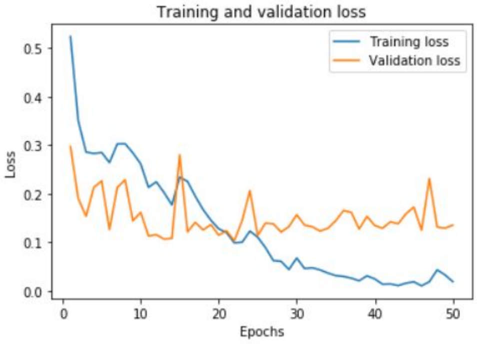


Figure 9.12: w2v+Bidirectional LSTM loss vs epoch

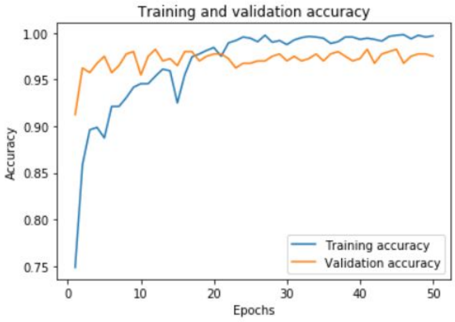


Figure 9.13: w2v+LSTM+CNN accuracy vs epoch

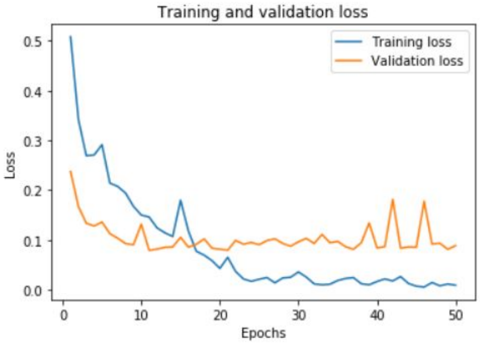


Figure 9.14: w2v+LSTM+CNN loss vs epoch

9.3 Confusion Matrix

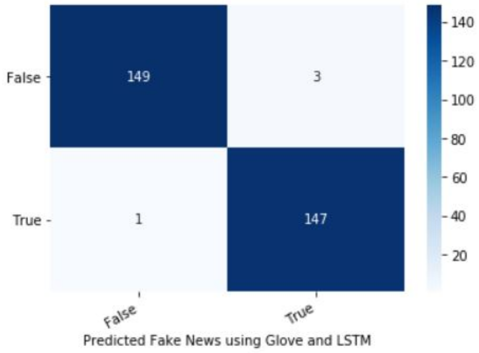


Figure 9.15: Confusion Matrix Of Glove And LSTM

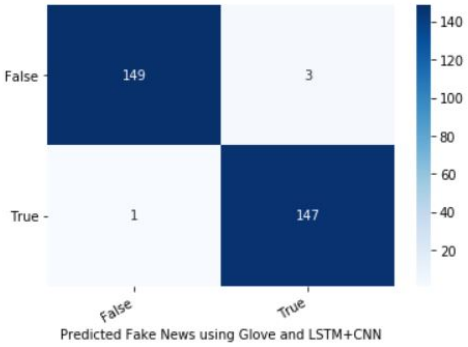


Figure 9.16: Confusion Matrix Of Glove And LSTM+CNN

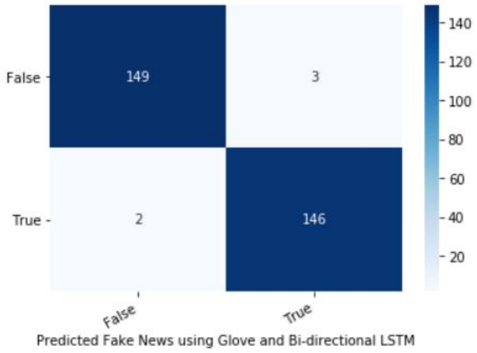


Figure 9.17: Confusion Matrix Of GLOVE And Bi-directional LSTM

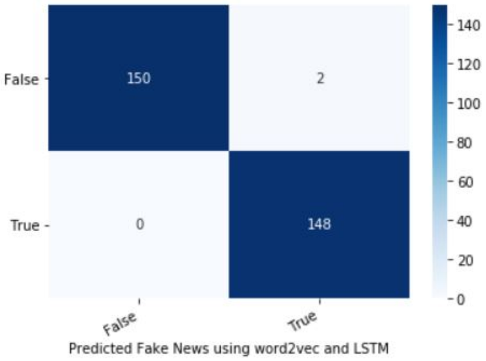


Figure 9.18: Confusion Matrix Of Word2vec And Bi-directional LSTM

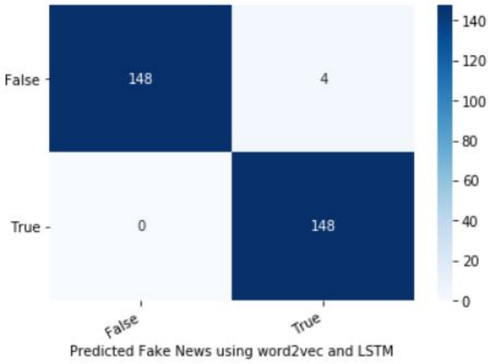


Figure 9.19: Confusion Matrix Of Word2vec And LSTM

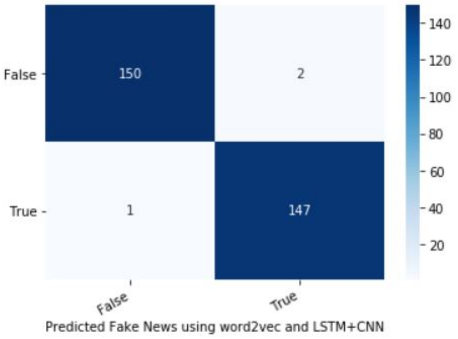


Figure 9.20: Confusion Matrix Of Word2vec And LSTM+CNN

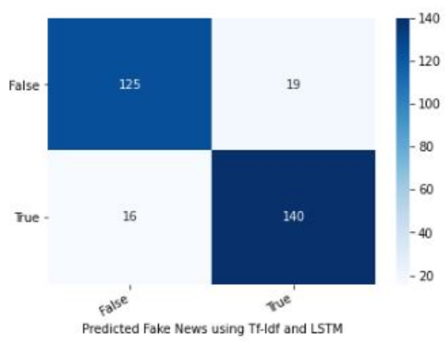


Figure 9.21: Confusion Matrix Of TF-IDF And LSTM

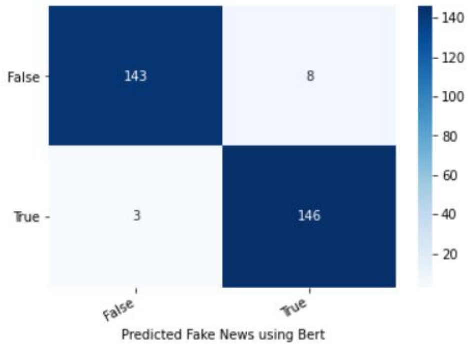


Figure 9.22: Confusion Matrix Of BERT

9.4 Machine Learning Algorithms

For an issue as sensitive as the COVID-19, it is vital to find out the best possible model to classify fake news. Therefore, along with state-of-the-art deep learning architectures, we have also experimented the traditional Machine Learning algorithms for classifying the news text. Here we have used TF-IDF for vectorizing the text. In order to train, we used 8 Machine Learning algorithms and they are: Multinomial Naive Bayes, Support Vector Machine, K-nearest neighbour, Random Forest Classifier, Stochastic Gradient Descent, Decision Tree, Gradient Boosting,Logistic Regression and Gradient Boosting. Table 2 shows the Results yielded from Machine Learning Algorithms from where we can see Stochastic Gradient Descent model generated the best result, a testing accuracy of 97.7% However, Support Vector Machine is also not far off with an impressive accuracy rate of 97.3%

Table 9.2: Results for ML algorithms

Model	Label	Precision	Recall	F1 Score	Test Accuracy
SVM	0	0.98	0.97	0.97	0.973
	1	0.97	0.98	0.97	
Multinomial NB	0	0.98	0.84	0.90	0.91
	1	0.85	0.99	0.92	
Random Forest Classifier	0	0.95	0.95	0.95	0.95
	1	0.95	0.95	0.95	
KNN	0	0.94	0.85	0.89	0.897
	1	0.86	0.95	0.90	
Decision Tree	0	0.89	0.92	0.90	0.9
	1	0.92	0.88	0.90	
SGD	0	0.98	0.97	0.98	0.977
	1	0.97	0.98	0.98	
Gradient Boosting	0	0.95	0.94	0.95	0.947
	1	0.94	0.95	0.95	
Logistic Regression	0	0.96	0.94	0.95	0.95
	1	0.94	0.96	0.95	

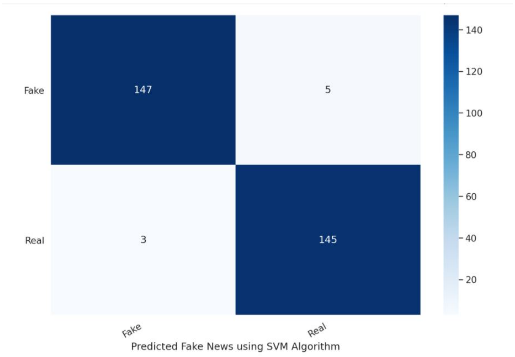


Figure 9.23: Confusion Matrix Of SVM Algorithm

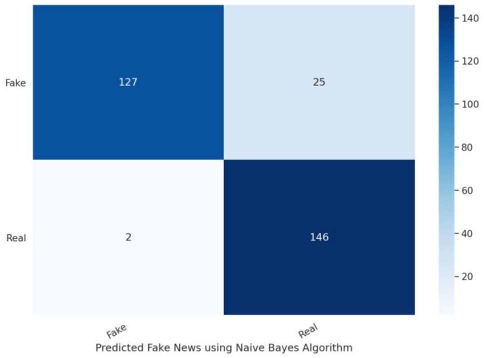


Figure 9.24: Confusion Matrix Of Naive Bayas Algorithm

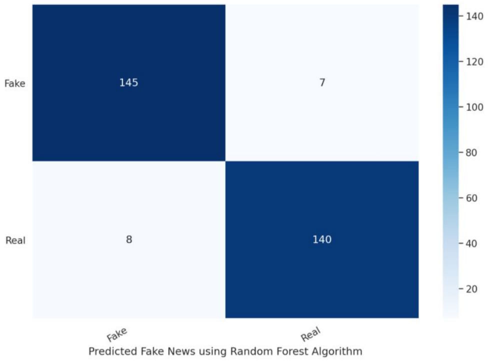


Figure 9.25: Confusion Matrix Of Random Forest Algorithm

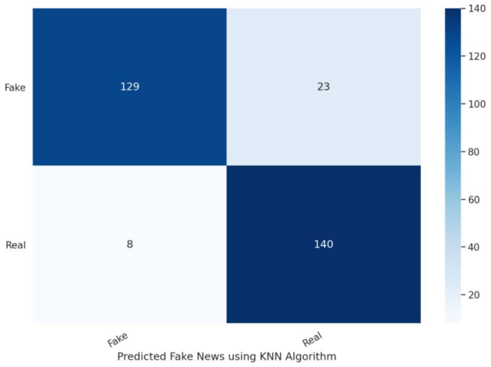


Figure 9.26: Confusion Matrix Of KNN Algorithm

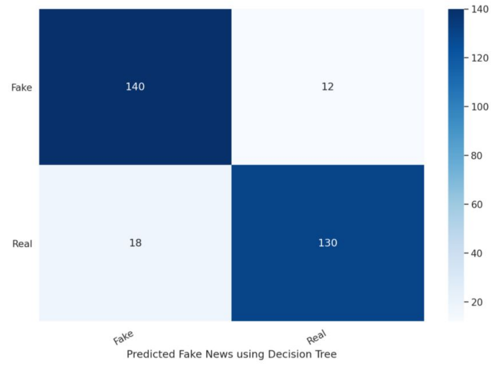


Figure 9.27: Confusion Matrix Of Decision Tree

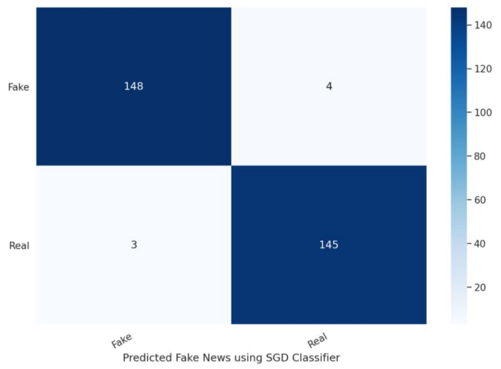


Figure 9.28: Confusion Matrix Of SGD Classifier

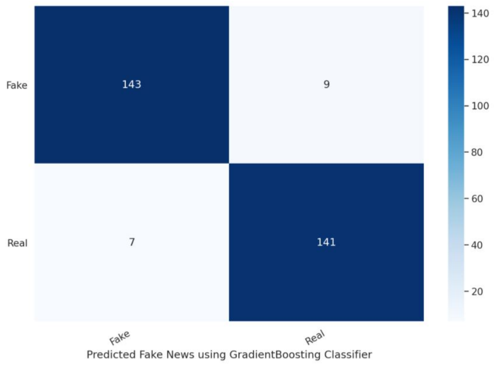


Figure 9.29: Confusion Matrix Of Gradient Boosting Classifier

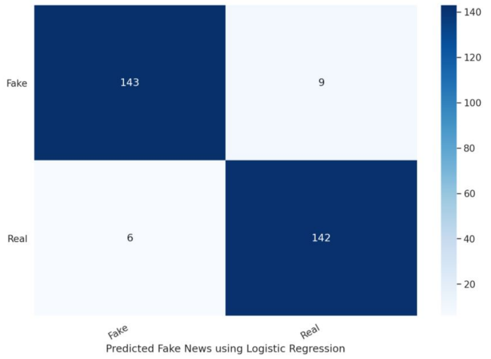


Figure 9.30: Confusion Matrix Of Logistic Regression

Chapter 10

Conclusion

10.1 Discussion and Future Work

To find the best performing model, we tested 17 different combinations of models overall. To classify fake news, it involves Deep Learning models with advanced word embedding techniques as well as traditional machine learning algorithms. Although a few deep learning models have outperformed some of the machine learning models, we observed that the best performance was generated by Bi-directional LSTM with Word2Vec.

The dataset can be further improved and the reliability and usability of our model will increase as more and more news will be added. To detect false news about COVID-19, the model can be used to create a web or mobile based user interface. As per the feedback received during our defense, we can train and test our model for graphical dataset as well. That will allow our model to detect fake news based on graphical data e.g. pictures, videos as well as existing textual data.

10.2 Conclusion

As the phenomenon linked to the vaccine continues to increase, it is possible that false news linked to COVID-19 will continue to spread like wildfire. Some kind of fake news is causing more fear among people around the world about this pandemic. The goal of our detection model is to recognize this incorrect one. The dataset used in this study focuses mainly on COVID-19 world and healthcare news on various websites across the globe. Different features of the news may be the extensible essence of the model used in this analysis. In addition, Bi-directional LSTM-RNN or simply Bi-directional Large sequence can be predicted because LSTM cells have memory that can store information from previous time steps.

Bibliography

[1] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” in *European conference on machine learning*, Springer, 1998, pp. 137–142.

[2] C. Manning and H. Schutze, *Foundations of statistical natural language processing.* MIT press, 1999.

[3] L. Rokach and O. Maimon, “Top-down induction of decision trees classifiers-a survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 35, no. 4, pp. 476–487, 2005.

[4] N. K. Conroy, V. L. Rubin, and Y. Chen, “Automatic deception detection: Methods for finding fake news,” *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015.

[5] J. Brownlee, “Gradient descent for machine learning,” *URL: https://machinelearningmastery.com/gradient-descent-for- machinelearning/gt;(Accessed October 1, 2017)*, 2016.

[6] M. Lango, D. Brzezinski, and J. Stefanowski, “Put at semeval-2016 task 4: The abc of twitter sentiment analysis,” in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 2016, pp. 126–132.

[7] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake news detection on social media: A data mining perspective,” *ACM SIGKDD explorations newsletter*, vol. 19, no. 1, pp. 22–36, 2017.

[8] W. Y. Wang, “” liar, liar pants on fire”: A new benchmark dataset for fake news detection,” *arXiv preprint arXiv:1705.00648*, 2017.

[9] R. Oshikawa, J. Qian, and W. Y. Wang, “A survey on natural language processing for fake news detection,” *arXiv preprint arXiv:1811.00770*, 2018.

[10] Y. Yang, L. Zheng, J. Zhang, Q. Cui, Z. Li, and P. S. Yu, “Ti-cnn: Convolutional neural networks for fake news detection,” *arXiv preprint arXiv:1806.00749*, 2018.

[11] M. Abbas, K. A. Memon, A. A. Jamali, S. Memon, and A. Ahmed, “Multinomial naive bayes classification model for sentiment analysis,” *IJCSNS*, vol. 19, no. 3, p. 62, 2019.

[12] P. Bahad, P. Saxena, and R. Kamal, “Fake news detection using bi-directional lstm-recurrent neural network,” *Procedia Computer Science*, vol. 165, pp. 74–82, 2019.

[13] X. Dong, U. Victor, S. Chowdhury, and L. Qian, “Deep two-path semi-supervised learning for fake news detection,” *arXiv preprint arXiv:1906.05659*, 2019.

[14] J. Y. Khan, M. Khondaker, T. Islam, A. Iqbal, and S. Afroz, "A benchmark study on machine learning methods for fake news detection," *arXiv preprint arXiv:1905.04749*, 2019.

[15] L. Cui and D. Lee, "Coaid: Covid-19 healthcare misinformation dataset," *arXiv preprint arXiv:2006.00885*, 2020.

[16] M. Z. Hossain, M. A. Rahman, M. S. Islam, and S. Kar, "Banfakenews: A dataset for detecting fake news in bangla," *arXiv preprint arXiv:2004.08789*, 2020.

[17] T. Hossain, R. L. Logan IV, A. Ugarte, Y. Matsubara, S. Young, and S. Singh, "Covidlies: Detecting covid-19 misinformation on social media," in *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, 2020.

[18] R. Kouzy, J. Abi Jaoude, A. Kraitem, M. B. El Alam, B. Karam, E. Adib, J. Zarka, C. Traboulsi, E. W. Akl, and K. Baddour, "Coronavirus goes viral: Quantifying the covid-19 misinformation epidemic on twitter," *Cureus*, vol. 12, no. 3, 2020.

[19] P. Patwa, S. Sharma, S. PYKL, V. Guptha, G. Kumari, M. S. Akhtar, A. Ekbal, A. Das, and T. Chakraborty, "Fighting an infodemic: Covid-19 fake news dataset," *arXiv preprint arXiv:2011.03327*, 2020.

[20] F. Sakketou and N. Ampazis, "A constrained optimization algorithm for learning glove embeddings with semantic lexicons," *Knowledge-Based Systems*, p. 105 628, 2020.

[21] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A comparative analysis of logistic regression, random forest and knn models for the text classification," *Augmented Human Research*, vol. 5, no. 1, pp. 1–16, 2020.

[22] G. K. Shahi and D. Nandini, "Fakecovid–a multilingual cross-domain fact check news dataset for covid-19," *arXiv preprint arXiv:2006.11343*, 2020.

[23] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, and B.-W. On, "Fake news stance detection using deep learning architecture (cnn-lstm)," *IEEE Access*, vol. 8, pp. 156 695–156 706, 2020.

[24] E. Zavarrone, M. G. Grassia, M. Marino, R. Cataldo, R. Mazza, and N. Canestrari, "Co. me. ta–covid-19 media textual analysis. a dashboard for media monitoring," *arXiv preprint arXiv:2004.07742*, 2020.

[25] A. Adl and S. Eid, "Detection of social interaction with rumor through social network using nlp and random forest classifier,"

[26] S. Kumar, R. R. Pranesh, and K. M. Carley, "A fine-grained analysis of misinformation in covid-19 tweets,"