




Universitetet
i Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study programme/specialisation: Computer Science	Spring / Autumn semester, 2020 Open/ Confidential
Author: Henrik Lessø Mjaaland	 (signature of author)
Programme coordinator: Vinay Jayarama Setty Supervisor(s): Vinay Jarama Setty	
Title of master's thesis: Detecting Fake News and Rumors in Twitter Using Deep Neural Networks	
Credits: 30 ECTS	
Keywords: Deep Learning Fake News Detection Natural Language Processing	Number of pages: 55..... + 5 Stavanger, 15 June 2020



Faculty of Science and Technology
Department of Electrical Engineering and Computer Science

Detecting Fake News and Rumours in Twitter Using Deep Neural Networks

Master's Thesis in Computer Science
by

Henrik Lessø Mjaaland

Internal Supervisors

Vinay Jayarama Setty

Reviewers

Vinay Jayarama Setty

Avishek Anand

April 23, 2020

*“If you can’t fly,
then run, if you can’t run,
then walk,
if you can’t walk,
then crawl,
but whatever you do,
you have to keep moving forward.”*

Martin Luther King Jr.

Abstract

The scope of this paper is to detect fake news by classifying them as either real or fake based on article content, tweets and retweets of news articles from the Politifact data set using graph neural networks. Fake news generally spread more rapid than real news. This is most likely because fake news are usually more novel and contain more superlatives than real news. Tweets of fake news articles also tend to spread exponentially and have more rumour cascade hops than real news, meaning tweets of fake news are retweeted more than real news. Tweets of real news articles on the other hand, tend to have a more constant and slow spread, and does not reach as many people as fake news. There are generally two characteristics that are used for detecting fake news: article content and rumour path propagation. Most existing works have presented models based solely on one of these characteristics, which has its advantages (i.e. reduced training time), but is also reflected by their prediction accuracies. This thesis proposes a hybrid model based on article content, rumour path propagation in the form of a temporal pattern, and also metadata using the bidirectional LSTM with the Keras Sequential model. Before combining article content, continuous attributes and metadata to a single combined vector, the article content is word embedded using pre-trained GloVe vectors and the continuous attributes are normalized and discretized. CNN, Multimodal NB and SVM is also implemented for comparison. Experimental results demonstrates that the proposed model performs better than state-of-the-art models.

Acknowledgements

This thesis is the culmination of my master's degree in Computer Science and the result of my fascination for computers and social media. I took an interest in computers at a young age, and what first sparked my interest in computers is how small they make the world. At an age of eight, I was already chatting with people from all around the world. As social media emerged, the world was made even smaller. Information was made more accessible to the public as social media can be used to share information, but they can also be used to share fake news. Recently, I have taken an interest in deep neural networks. What interests me about deep neural networks is that they can perform very well when the data size is large enough, they can combine features without being explicitly told to do so, thus, simplifying the process of feature engineering, and they can be used to solve complex problems such as speech recognition, image classification, and natural language processing which can be used for identifying fake news as is done in this thesis. I would like to express my deepest gratitude to Vinay Jayarama Setty for introducing me to deep neural networks and giving me the necessary tools to write this thesis, and for guiding me throughout the semester by giving valuable feedback. I would also like to thank my friends and family for motivating and supporting me.

Contents

Abstract	vi
Acknowledgements	viii
Abbreviations	xi
Symbols	xiii
1 Introduction	1
1.1 Motivation	2
1.2 Problem Definition	3
1.3 Challenges	3
1.4 Contributions	4
1.5 Outline	5
2 Background	7
2.1 Neural Networks	7
2.1.1 Feed-Forward (Forward-Propagation)	8
2.1.2 Back-Propagation	8
2.1.3 Gradient-Descent	9
2.2 Neural Networks for NLP	9
2.2.1 Word Embedding	10
2.2.2 Recurrent Neural Networks	10
2.2.3 Vanishing Loss Gradient	11
2.2.4 Bidirectional LSTM	13
2.3 Related Work	13
2.4 Existing Approaches/Baselines	16
2.5 Analysis	18
3 Solution Approach	19
3.1 Proposed Solution	19
3.2 NLP	20
3.3 Continuous Attributes	20
3.4 Rumour Propagation	20

3.5	Layers	20
4	Experimental Setup and Data Set	23
4.1	Technologies	23
4.1.1	Natural Language ToolKit (NLTK)	23
4.1.2	Wordcloud	24
4.1.3	Keras	24
4.1.4	RE (Regular Expressions)	24
4.1.5	Datetime	24
4.2	Dataset	24
4.2.1	Preprocessing	25
4.2.2	Synthetic Minority Over-sampling Technique (SMOTE)	25
4.2.3	Gridsearch	25
4.3	Experimental Results	25
4.3.1	CDF	25
4.3.2	Users	26
4.3.3	TFIDF	27
4.3.4	Rumour Propagation	27
4.4	Accuracy Measures	29
4.5	Results	32
4.5.1	SVM	32
4.5.2	NB	32
4.5.3	CNN	33
4.5.4	LSTM	33
4.5.5	Optimization Techniques	33
4.5.6	Training Times	34
4.5.7	Loss Functions	34
4.5.8	Accuracies	34
5	Discussion	37
5.1	Baseline Comparison	37
5.2	Implications of Findings	38
6	Conclusion and Future Directions	39
6.1	Future Directions	39
6.1.1	Data	39
6.1.2	Graphical User Interface (GUI)	40
6.2	Conclusion	40
	List of Figures	40
	List of Tables	43
	Bibliography	45

Abbreviations

NN	Neural Networks
DNN	Deep Neural
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
LSTM	Long Short Term Memory
TF	Term Frequency
IDF	Inverse Document Frequency
NLP	Natural Language Processing
DL	Deep Learning
ML	Machine Learning

Symbols

symbol	name	unit
∞	infinity a	distance
m		
P	power	W (Js^{-1})
ω	angular frequency	rads^{-1}

Chapter 1

Introduction

According to the Cambridge Dictionary, fake news are "stories that appear to be news, spread on the internet or using other media, usually created to influence political views or as a joke". Fake news is a problem of increasing concern in today's world. Fake news today spread like wildfire. They are usually spread by bots, and they spread exponentially per cascade hop in social media. Creating your news by manipulating videos and images is only getting easier and easier. 'Anyone' can forge their own news today. Lately, fake news has become a problem of concern especially in Norway, where sharing fake news through memes on social media has become a trend.

Recently, there have been a lot of fake news regarding the Corona Virus from China. The most widespread ones are that the Chinese were infected from eating bats, and that there is already a vaccine for the virus, which is not true, since there are seven kinds of Corona Viruses, and the one causing the outbreak now is a new kind of Corona Virus. When it comes to fake news during elections, attention has mostly been diverted to the West, leaving the global south unchecked[Chinchilla(2019)]. This has led to an increase of 40 percent spread of fake news in India during election times.

In Brazil, 2019, the vast majority of false information shared on WhatsApp in Brazil during the presidential election favoured the far-right winner, Jair Bolsonaro. Out of 11,957 viral messages shared across 296 group chats on the instant-messaging platform in the campaign period, approximately 42 percent of rightwing items contained information found to be false by factcheckers[Avelar(2019)]. In the global South, the preferred messaging app is WhatsApp. WhatsApp offers encrypted peer-to-peer messaging and is very challenging to monitor, unlike apps like Facebook. Monitoring users' conversations violates the users' privacy, but is useful for detecting fake news and their source.

Multiple initiatives have been taken to battle fake news such as the law against information manipulation. The law prevents influencing of election results, which was done e.g. during Brexit.

The law prevents political influencing by requiring “a ‘transparency obligation for digital platforms’, who must report any sponsored content by publishing the name of the author and the amount paid. Platforms exceeding a certain number of hits a day must have a legal representative in France and publish their algorithms.”[Gouvernement(2018)], and by requiring a judge to qualify fake news based on these three criterias:

- “- the fake news must be manifest,
- be disseminated deliberately on a massive scale,
- and lead to a disturbance of the peace or compromise the outcome of an election.”[].

The law also demands a cooperation between the different digital platforms during election periods. The French Broadcasting Authority is responsible for enhancing law enforcement for this requirement. Furthermore, Emmanuel Hoog, former president of the French Press Agency was entrusted with producing a press ethics body. Measures against fake news have also been taken in Finland, Malaysia and Singapore. Finland launched an anti-fake news initiative that aims to teach how to identify fake news already in 2014. The Malaysian government passed the Anti-Fake News Act in 2018. The Act was however accused of seeking to stifle criticism of the administration. In 2019, another law against fake news was passed in Singapore called the Protection from Online Falsehoods and Manipulation Act. This law received the same allegations as the law against fake news from Malaysia.

1.1 Motivation

Fake news has had a massive impact on the news industry since the dawn of the Internet. They are under increasing pressure to produce more material faster, and today they tend to use social media as a source of information. When using social media as a source of information, verifying the authenticity of the news is a big challenge for reasons such as the pressure to produce more, and when journalists incorrectly verify news sources, their brand’s reputation is tarnished and people lose trust in the news media. A brand’s reputation takes years to build up, but only seconds to tear down. The pressure to produce material fast also leads to more unserious journalism. Fake news is a many-faced and complex issue. They can be spread to fuel conflicts, for economic gains, defamation or for political purposes. Also, fake news lead to people in general being less well informed. [Banjo and Lung(2019)] In 2019, fake news regarding the death of 22-year-old Chinese student Alex Chow circulated, claiming that he had not committed suicide, but was in fact chased and/or pushed off a parking garage by the Hong Kong police force. The news also claimed that officers had blocked off an ambulance from reaching Chow. The purpose of these fake news was to incite anti-government protests. [Cheo(2018)] In 2013, fake news regarding two explosions in the White House that had injured the

earlier US president Barack Obama led to the Dow Jones Industrial Average dropping by 143.5 points, which corresponds to the sum total of 130 billion US dollars in stock value. Donald Trump legislated CNN and Washington Post as "fake news" because they were not supportive of him (they referred to his followers as a cult and claimed that he had "bewitched" the republican party on multiple occasions). [Jardine(2018)] Fake news has always been used in political contexts, such as spreading false poll numbers to discourage people from voting during political elections, e.g. in Mexico in 1988 by PRI (Institutional Revolutionary Party), but never on a scale and magnitude as with today's technology. During the 2016 presidential election in USA, which ended in the election of Donald Trump as the US president, there was a surge of fake news on social media. Investigations after the presidential election points to Russian influence during the campaign. Another case of using fake news as a means of political influence, is back in 2014 when ISIS started sharing propaganda on every social media imaginable. Democracy is built trust in the public's capacity for reasoned communication. Digital technologies have made information more accessible to the public, however as deep fake news is getting more and more advanced, people are getting less and less well informed, thus threatening democracy. Political influencing, which was done by PRI, Russia and Bolsonaro as discussed above, also threatens democracy by centralizing the power and taking it away from the people.

1.2 Problem Definition

This thesis investigates if articles' content and their respective rumour propagation paths can be utilized to improve fake news detection. The goal is to achieve this by modelling a graph neural network in order to classify the articles as either real or fake based on patterns in article content and sharing of tweets. The articles are also classified based on a series of descriptive parameters for both the articles and their respective tweets and retweets such as article id, metadata, tweet and retweet identification, tweet- and retweet content, and follower count.

1.3 Challenges

There are many challenges when identifying fake news. Fake news platforms usually imitate real news platforms by design and content.

Fake news articles are not always one hundred percent fake, they might have some true statements mixed with false statements.

With today's technology, images and videos can be fabricated, making it impossible to

	created_at	text	contributors_enabled_user	...	followers	following	label
0	2015-10-09 21:18:53+00:00	Missed @marcorubio in NH this week? Come to a ...	False	...			real
1	2013-03-22 06:36:15+00:00	2013-HB-4469 Public employees and officers; et...	False	...			real
2	2008-03-03 06:10:37+00:00	Strong Words in Ohio as Obama and Clinton Pres...	False	...			real
...
790	2017-08-29 17:54:38+00:00	Floyd Mayweather Jr. donates a whopping sum of...	False	...			fake
791	2018-02-21 18:47:48+00:00	@TylerHuckabee @LagBeachAntifa9 @AntifaNantuck...	False	...	1140757886706102273, 289419759, 1450422866, 11...	1140757886706102273, 289419759, 1450422866, 11...	fake
792	2018-02-23 18:45:47+00:00	Obama Announces Bid To Become UN Secretary Gen...	False	...			fake

Figure 1.1: Sample Data Set

instantly verify authenticity of news.

It is also easy to create fake entities to share fake news. There are free tools available online for this purpose, and existing photos can be used for face swap[Polyakov(2018)]. Fake news articles often contain more dramatic and intensive language than real news, hence natural language processing (NLP) can be used to identify fake news. NLP is however a complex and time-consuming process as words can have different meanings in different contexts, and each language and dialect requires a separate version of NLP. The amount of data available is limited, and there are more real than fake news available. This affects training of classification models.

1.4 Contributions

In the past, classification and regression models have been used the most for fake news detection. This thesis, however, presents a deep neural network model based on Bidirectional Long Short Term Memory (LSTM). Continuous numeric attributes are normalized in the range $[0,1]$, and then discretized, before word embedding them along with categoric attributes and textual attributes. The word embedded attributes are fed into the model which uses them to classify news articles from the Politifact dataset as either real or fake.

Multimodal NB, SVM and CNN was also implemented for comparison.

High accuracy despite empirical studies showing to ... being ineffective for fake news detection.

1.5 Outline

Chapter 2 introduces theories related to the work done in this thesis including neural networks. It also introduces related works.

Chapter 3 describes the method used in the proposed model in detail, and explains why the given method is used.

Chapter 4 presents the dataset, the experimental setup and the results of the experiments.

Chapter 5 discusses the results, baseline models and deviations from these models, implications of the results, and finally strengths and weaknesses of the proposed model.

Chapter 6 answers the research question, and points to future directions.

Chapter 2

Background

This chapter introduces relevant theories and baseline solutions for detecting fake news using graph neural networks.

2.1 Neural Networks

Neural networks are based on the human brain in the sense that they are algorithms consisting of multiple nodes connected by edges, mimicking neurons. Neural networks with minimum one hidden layer, are referred to as deep neural networks. Hidden layers are the layers between the input layer and the output layer (see Figure 2¹). For now, the required number of hidden layers for a neural network be considered ‘deep’ is more than one, but this number will likely increase soon. Deep neural networks enable computers to artificially learn, hence the term, Artificial Intelligence (AI). Deep neural networks can for example be used to learn features from data such as text, images, sound or they can be used for classification and regression.

As mentioned above, neural networks have many uses, but in this case the target values are either real or fake, which means that this is a binary classification problem. All the nodes in the hidden layers and the output layer, have each their classifier. The classifiers activate the respective nodes based on the inputs from the previous layer and an activation function. There are many activation functions, depending on numerous factors. This thesis uses ReLU (Rectified Linear Unit) followed by Sigmoid. ReLU is the most used activation function.

¹<https://i.stack.imgur.com/Kc50L.jpg>

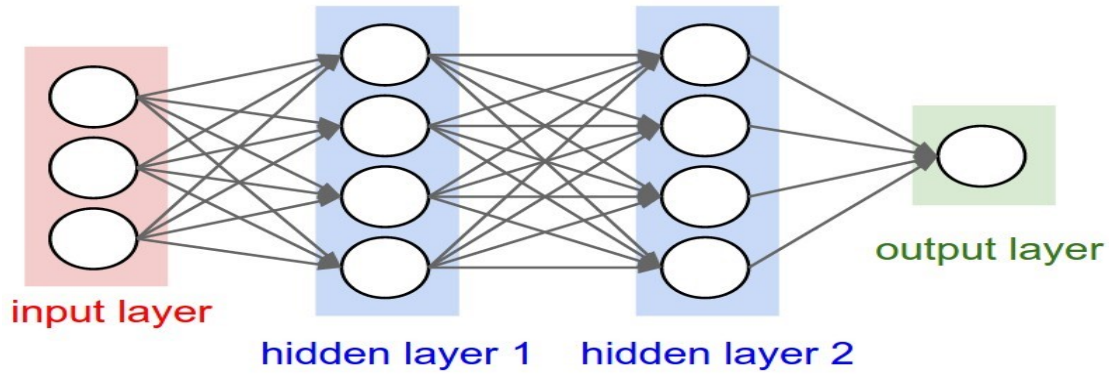


Figure 2.1: Neural Networks Layers

2.1.1 Feed-Forward (Forward-Propagation)

The first hidden layer receives input from the input layer, and then classification scores are passed on from layer to layer. Only activated nodes pass on their classification scores to the next layer. In the output layer, the target values (in this case real or fake) are classified based on the final classification scores, which means that the output is the predicted class. This process is referred to as feed-forward or forward-propagation. Node outputs, $OUTPUT(H)$, are computed by first calculating node nets, $NET(H)$, which is the sum of the input weights (W), multiplied with their respective inputs (I), and biases (b):

$$NET(H_i) = \Sigma(W_{ij} * I + b_i)$$

Then, if the target classes are not linearly separable, the activation function (output), is given by:

$$OUTPUT(H_i) = \frac{1}{1 + E^{-NET(H_i)}}$$

2.1.2 Back-Propagation

Back-propagation is the process of computing the gradients with respect to the weights. While forward-propagation is used for computing outputs, the purpose of back-propagation is to minimize the cost function. This is done by adjusting the weights and biases backwards. The weights are adjusted by first computing the total error (E):

$$E = \Sigma(0.5 * (TARGET(H_i) - OUTPUT(H_i))^2)$$

Next, the learning rate (L) multiplied with derivative of the total error w.r.t weights are subtracted from the given weights:

$$W_i = W_i - (L_i * \frac{dE}{dW_i})$$

$$\begin{aligned} & \frac{dE}{dW_i} \\ &= \\ & \frac{dE}{dOUTPUT(H_{i+1})} * \frac{dOUTPUT(H_{i+1})}{dNET(H_i)} * \frac{dNET(H_{i+1})}{dW_i} \end{aligned}$$

The learning rate that gives the steepest loss function should be picked, or one can gradually reduce it after training epochs.

2.1.3 Gradient-Descent

For an unweighted neural network, a given set of inputs will always result in the same output. In a weighted neural network however, all the nodes have different weights, each resulting in a unique output. The gradient descent algorithm runs forward- and back-propagation multiple rounds or epochs, each time tweaking the weights in order to improve the classification accuracy as explained above. Running forward propagation with a high number of epochs is time-consuming and one should find a balance between accuracy and number of epochs, where accuracy is maximized, and number of epochs is minimized. Too many epochs can also result in overfitting. This can be prevented by plotting training- and testing loss. Overfitting is when the training loss is less than the testing loss, and underfitting is when the training loss is higher than the testing loss. The testing- and training loss should be about the same.

2.2 Neural Networks for NLP

subsection Term Frequency-Inverse Document Frequency (TF-IDF) TF-IDF is the product of TF and IDF. TF is the number of terms across all the documents for each term. IDF is the inverse number of terms per document. TF-IDF says how rare a specific term is; the higher the TF-IDF is, the rarer the given term is. The most common words will thus have a low TF-IDF.

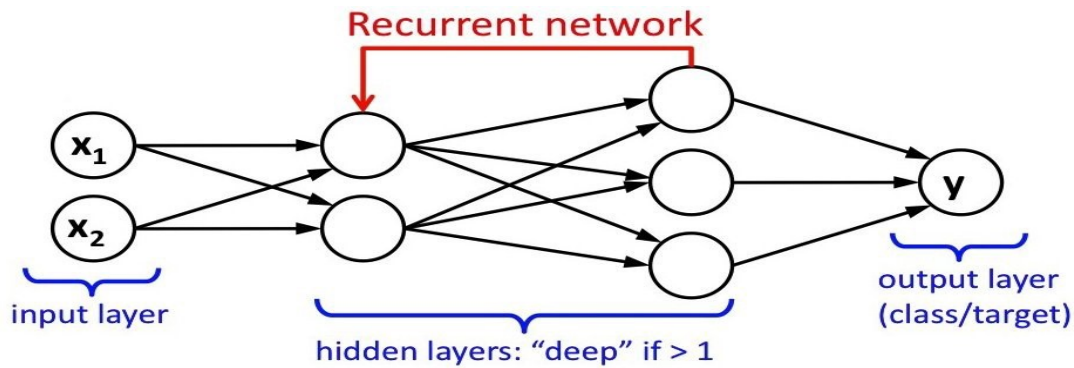


Figure 2.2: Recurrent Neural Network with Loops

2.2.1 Word Embedding

Word embedding is the process of converting terms to vectors of float numbers. This is done to improve the accuracy of Artificial Neural Networks (ANN) because ANNs are built for continuous numeric inputs, and similarities between continuous vectors with more ease than similarities between terms. A common technique for word embedding is GloVe (Global Vectors). GloVe is an algorithm for converting unlabelled data such as words to continuous vectors which reduces dimensionality. GloVe vectors are pre-trained on Wikipedia and Gigaword 5, and are good at capturing semantics

2.2.2 Recurrent Neural Networks

Recurrent neural networks consist of multiple feedforward neural networks (FNNs), and each FNN is considered a time-step. Recurrent neural networks use the gradient descent algorithm to minimize error. However, since there are many time-steps that all share the same parameters now, backpropagation-through-time (BPTT) is used instead of normal backpropagation. There are loss between time steps, thus, there is an extra error gradient compared to normal backpropagation.

Traditional neural networks are not good for predicting based on sequential data such as speech, time series, or text, because they have no memory, and sequential data is ordered. Recurrent neural networks introduce memory by using loops between the hidden layers. The output from the loops are stored in the internal state (see figure 3²).

²<https://i.stack.imgur.com/Kc50L.jpg>

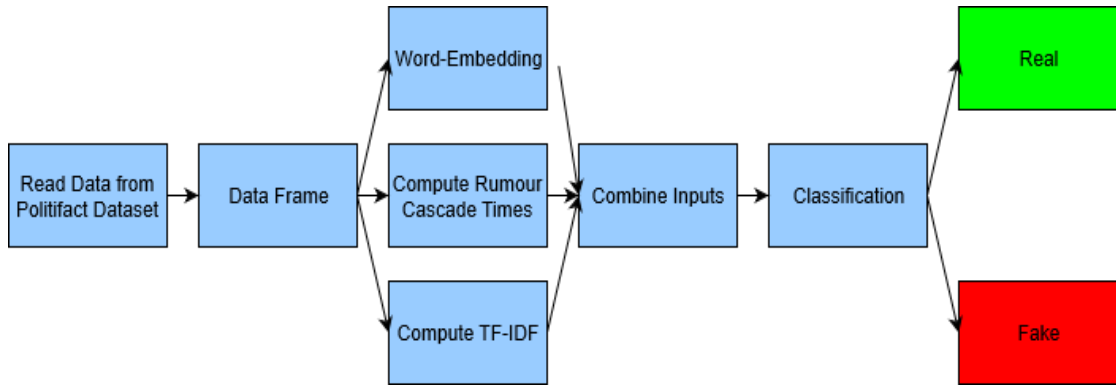


Figure 2.3: Recurrent Neural Network with loops

2.2.3 Vanishing Loss Gradient

When training RNNs with multiple layers a problem called the ‘Vanishing Loss Gradient’ can arise. The problem is as the name suggests, when the gradient ‘vanishes’, meaning that it’s close to zero. The problem arises if many gradients are close to zero, because the gradients are products of previous gradients. The problem usually occurs in the earlier layers (gradients are computed using backpropagation). This is a problem because the weight updates are the subtraction of the gradients multiplied with the learning rate, from the weights themselves. This means that if the gradients are close to zero, the weight updates will have little to no effect at all. The vanishing loss gradient is not a problem for the ReLU activation function, hence the proposed model applies ReLU in the earlier layers and Sigmoid in later layers (see chapter 2.1). ReLU converts values above zero to the original value, and negative numbers to zero, while Sigmoid converts values above zero to one, and negative numbers to zero, and is thus more computationally taxing than ReLU.

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a type of recurrent neural network that solves the vanishing gradient problem by introducing long-term memory. As discussed in 2.2.3, recurrent neural networks have multiple time-steps and a cell which contains one network layer (tanh) for each time-step. LSTM networks on the other hand have a memory cell which contains four network layers for each time-step (see figure 2.4³). Three of the layers are Sigmoid, namely, the input-, output-, and forget gate. There is also one tanh layer.

³https://www.researchgate.net/figure/A-diagram-of-a-basic-RNN-cell-left-and-an-LSTM-memory-cell-right-used-in-this-paper_fig1319770438

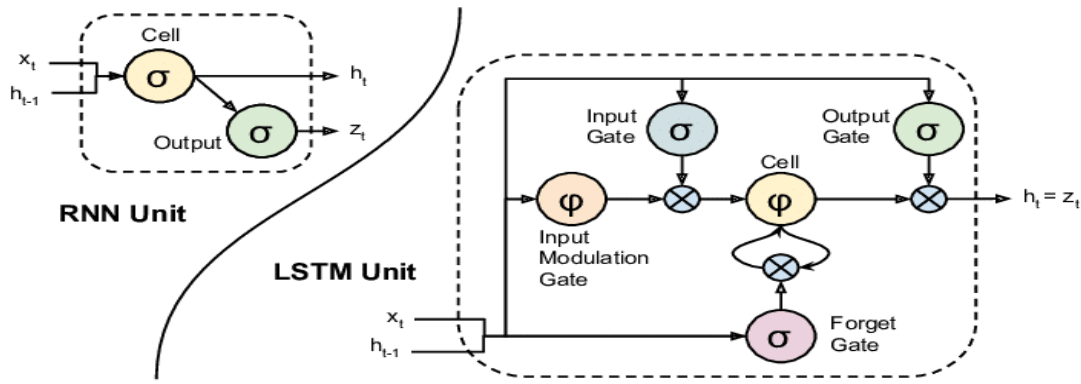


Figure 2.4: RNN Cell and LSTM Gates

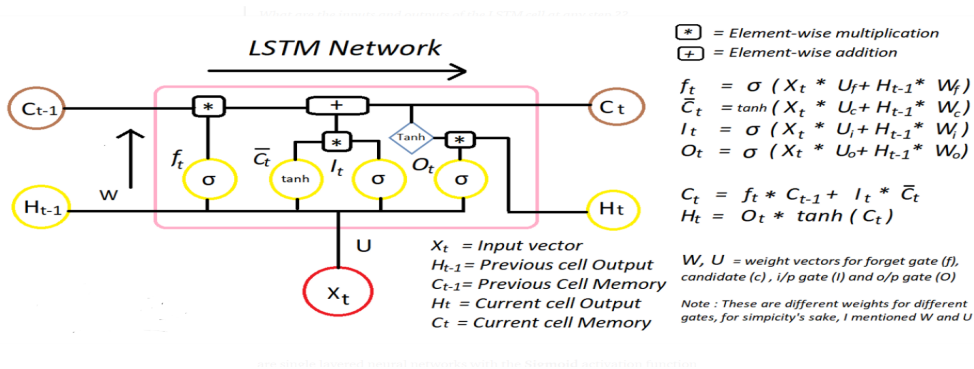


Figure 2.5: LSTM Cell Equations

As depicted in figure 2.4⁴, the memory cell contains an input gate, an output gate, and a forget gate. The input gate chooses which information to remember, and the output gate decides which information to output. LSTM memory cells have two states, namely, hidden- and memory state. The hidden state is the same as the output. The memory state, C , is where memory input is stored. The equations of the different gates and states are explained in figure 2.5⁵ below.

The key to long-term memory is the forget gate. In traditional RNNs, all inputs are accepted. In this case, previously accepted relevant information might be substituted with new irrelevant information. In LSTMs however, the forget gate prevents this problem by filtering irrelevant information from the memory state, thus providing long-term memory. This is done by multiplying the previous memory state with the forget gate, f . If f is zero, the memory state is completely forgotten, and only the current input is used:

$$C_t = C_{t-1} * f_t + C'_t * I_t$$

⁵<https://medium.com/deep-math-machine-learning-ai/chapter-10-1-deepnlp-lstm-long-short-term-memory-networks-with-math-21477f8e4235>

Memory cells also have an input modulation gate (this gate is often considered a sub-gate of the input gate). This gate normalizes the input information to increase convergence speed.

RNNs (including LSTM) use a tanh function to compute the output, H (see figure 2.5), which ensures that all the values stay between -1 and 1:

$$H_t = \text{Tanh}(C_t)$$

This alleviates the exploding loss gradient problem, which is the opposite of the vanishing loss gradient problem, when the gradient reaches astronomically high values, because the gradient can never be higher than 1.

A con with LSTM is that it can cause redundant overhead because some predictions requires less context than others. An example is “the sun is a star”, when predicting the last word, ‘star’, no more information is needed. An example where more context, and thus LSTM, is needed, is “I like to watch soap operas... My favourite TV-show is ‘Friends’”. In this case the first statement is useful when predicting the last word of the second statement.

2.2.4 Bidirectional LSTM

Forward LSTMs only have backward memory and remember the past, while backward LSTMs are trained on reversed inputs, and thus have forward memory and remember the future. Bidirectional LSTMs combine forward and backward LSTMs, which means it has both forward and backward memory. This can improve performance by providing more context and reducing training time. Consider the following scenario: a forward LSTM’s memory contains ‘The boys went to...’, while a backward LSTM’s memory contains ‘...and then they caught two fish’. Predicting the next word (‘fish’) is easier when combining the memories, than it is when only using the forward LSTM’s memory.

2.3 Related Work

Support Vector Machine (SVM) – 75.5, DT – 82.7 percent, Kaggle fake news challenge -1, The last few years, there has been a lot of research on detecting fake news in social media. Many of these works have used machine learning techniques like decision trees and linear Support Vector Machine (SVM) as [[A.Lakshmanarao\(2019\)](#)] and [[Junaed Younus Khan and Afroz\(2019\)](#)]. The works achieved accuracies of 82.7 percent

and 65 percent respectively using decision trees. When using SVM, accuracies of 75.5 percent and 66 percent was achieved. Some of these works focus on article content like [Iglesias(2019)], which achieved an accuracy of 62.4 percent using Fakebox (a machine learning model) and McIntire’s fake-real-news-dataset. Lately, research has also been conducted using deep learning. These works focus mostly on rumour propagation paths, and not so much on content, i.e. [Liu and Wu(2019)]. [Liu and Wu(2019)] claims to be able to detect fake news with accuracies of 85 percent and 92 percent on Twitter and Sina Weibo respectively in 5 minutes after they begin to spread. A graph neural networks-based work focusing on image data, obtained an accuracy of 94 percent, using a Convolutional Neural Networks (CNN) based model [Iglesias(2019)]. [Iglesias(2019)] also obtained an accuracy of 91 percent using LSTM based on content and title of articles. The work used fake news articles from the dataset, ‘Getting Real About Fake News’, and real news articles from sources such as ‘The New York Times’ and ‘The Washington Post’. Some works have also applied TF-IDF for fake news detection such as [Junaed Younus Khan and Afroz(2019)], where an accuracy of 95 percent was achieved using unigram Naïve Bayes on a combination of the ‘Liar Liar’- and ‘Fake or Real News’ datasets. The related works are described further in table 2.1 below. The descriptions in table 2.1 shows that the best results for detecting fake news are generally achieved using deep learning (especially LSTM) as compared to machine learning algorithms, but also that machine learning algorithms can be better for small data sets, and CNN is faster to train than LSTM.

Title	Tags	Description
An Effecient Fake News Detection System Using Machine Learning (A.Lakshmanarao, Y.Swathi, T. Srinivasa Ravi Kiran, 2019)[]	ML, TF-IDF, NLP	Detects fake news using the following machine learning classifiers: SVM, K-Nearest Neighbors (KNN), Decision tree, Random forest (RF). The best accuracy was obtained using NLP algorithms and a RF Classifier on the data set from the Kaggle Fake News Challenge. The reason machine learning algorithms was even better than deep learning models in this case, might be because deep learning models requires padding or shrinking the data, or maybe because the ‘Liar liar’ data set contains statements, and not full articles. The data set contains a number of articles as well as article id, title, author and label.

<p>A Benchmark Study on Machine Learning Methods for Fake News Detection (Junaed Younus Khan , Md. Tawkat Islam Khondaker , Anindya Iqbal and Sadia Afroz, 2019[])</p>	<p>ML, NLP, TF-IDF</p>	<p>Detects fake news using a the following machine learning classifiers: SVM, LR, Decision Tree, Adaboost, Naive Bayes and K-Nearest Neighbours. A combination of the 'Liar liar pants on fire' dataset and the Kaggle Fake News competition data set was used. The 'Liar liar' data set contains statements from Politifact.com. The best accuracy achieved was ninety-five percent, using a Multinomial Naive Bayes (NB) classifier with unigram TF-IDF features. The dissertation recommends using n-gram TF-IDF features with NB for small data sets and unigram for large data sets and claims that LSTM classifiers are better at overcoming overfitting than NB classifiers.</p>
<p>Fake News Detection via NLP is Vulnerable to Adversarial Attacks (Zhixuan Zhou¹, Huankang Guan, Meghana Moorthy Bhat and Justin Hsu)[]</p>	<p>ML, NLP</p>	<p>Evaluates the fake news detector, FakeBox, which uses NLP on article title and content. McIntire's 'Fake or Real News' data set was used. The data used in this work contains news articles from the 2016 USA election cycle. An accuracy of only 65 percent was achieved. The low accuracy led to the conclusion that accurate fake news detection on a general basis requires more than just linguistic characteristics.</p>
<p>Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks (Yang Liu, Yi-Fang Brook Wu, 2019[])</p>	<p>DL, RP, NLP</p>	<p>Detects fake news early on Twitter using CNN, RNN, SVM, Difficult-to-Treat Resistance (DTR), Gated Recurrent Unit (GRU), RF, Propagation Tree Kernel (PTK) for propagation path classification. Three datasets containing social media data were used: Sina-Weibo, Twitter 2015 and Twitter 2016. The proposed model in the dissertation is a combination of RNN (GRU) and CNN using max pooling. RNN alone achieved slightly better results than CNN on it's own, but the difference is negligible.</p>

Fake news detection using Deep Learning (Álvaro Ibrain, Lara Lloret)[]	DL, Image Classification, NLP	Detects fake news using LSTM,CNN and Bidirectional Encoder Representations from Transformers (BERT). BERT, using Google's 'BASE' architecture and 'Wordpiece' word tokenization, achieved the best results, and CNN had slightly better accuracy than LSTM. BERT is a network-based technique for NLP. The fake news articles are gathered from the dataset 'Getting Real About FakeNews', and the real articles are gathered from esteemed sources such as 'The New York Times' or 'The Washington Post'.
A Benchmark Study on Machine Learning Methods for Fake News Detection (Junaed Younus Khan, Md. Tawkat Islam Khondaker, Anindya Iqbal and Sadia Afroz)[]	ML, NLP, TF-IDF	Implements and assesses various machine learning- and deep learning methods for fake news detection. McIntire's 'Fake or Real News' dataset, Wang's 'Liar liar, pants on fire' data set, and a combination of the two were used. The work concluded that even though deep learning models (C-LSTM and bidirectional LSTM) produce the best results, machine learning algorithms (linear SVM) might be best for small data sets with the right feature selection considering training speed.

Table 2.1: Related Works

2.4 Existing Approaches/Baselines

As mentioned, the model combines the RNN (GRU) and CNN architectures (see figure 2.6). Both networks take a time series vector of user characteristics created using propagation paths as input.

GRU is a RNN architecture that controls flow of information by using the reset and update gates as depicted in figure 2.7⁶. The reset gate decides which information to use and which information to throw away, and thus, it can be considered a mix of the LSTM forget and input gates. GRUs are computationally efficient and simple because they don't need a memory unit like LSTM. GRUs also train faster and perform better on small data sets than LSTM. This thesis' proposed solution, however, uses LSTM and not

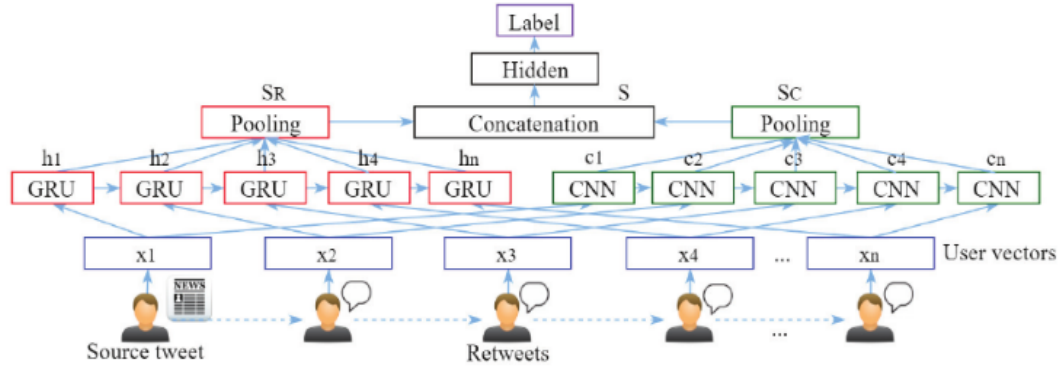


Figure 2.6: GRU-CNN Architecture Diagram

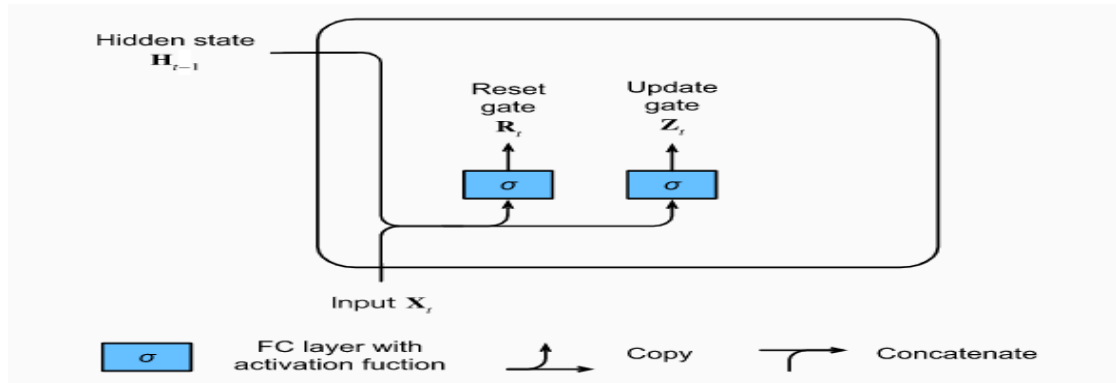


Figure 2.7: GRU Cell

GRU, because LSTMs are better at remembering long sequences and modeling long-distance relations than GRUs, which is good when using article content as input. Mean pooling is applied to the GRU output in order to reduce dimensionality.

CNN applied to the time series vector using a 1D convolutional layer with the ReLU activation function. The output is max pooled just as the output from the GRU network. The outputs from the two networks are concatenated and fed into a multi-layer feedforward neural network that predicts class labels for their respective propagation paths using ReLU and Softmax. Softmax is a simple, but effective classifier for linearly separable problems.

⁶https://www.researchgate.net/publication/329590584_Early_Detection_of_Fake_News_on_Social_Media_Through_Propagation_Path_Classification_with_Recurrent_and_Convolutional_Networks

⁶https://d2l.ai/chapter_recurrent-modern/gru.html

2.5 Analysis

Yang Liu and Yi-Fang Brook Wu's proposed model solved their thesis' problem, which was to detect fake news as early as possible, better than state-of-the-art models, both in speed and accuracy. The model detects fake news fast because it only takes time series of user characteristics as input, and not complex features such as linguistic features.

However, lower training time comes at the expense of lower accuracy. The accuracy is not bad considering how fast the model is trained, but it could be even better using LSTM instead of GRU and by adding linguistic features as input. If one has access to a GPU, time is not of the essence.

Chapter 3

Solution Approach

This chapter presents the proposed solution and explains the solution's preprocessing steps and model layers in detail.

3.1 Proposed Solution

Content, continuous attributes and rumour path propagation have been successfully used in previous works, and this thesis combines these features (see figure 7). Article content and rumour path propagation related attributes are first read from the Politifact dataset and merged into a data frame ordered by articles as shown in figure 1.

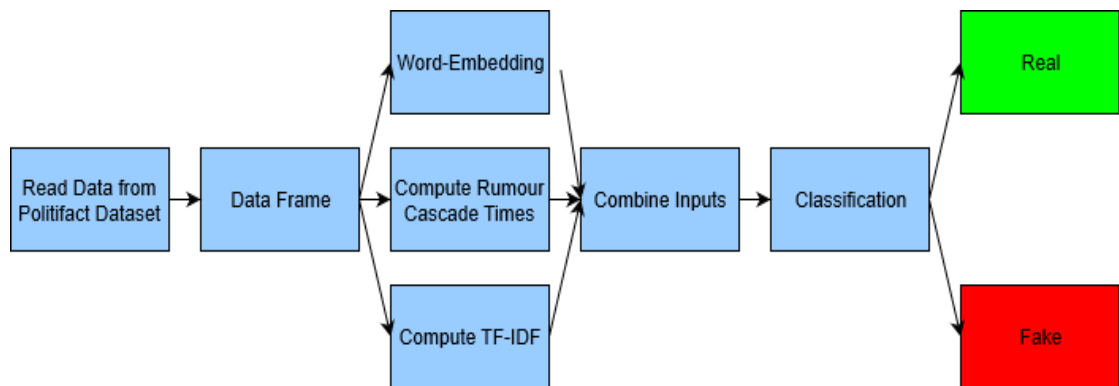


Figure 3.1: Proposed Solution

3.2 NLP

Article content is word-embedded (see 2.2.2). The content of each article corresponds to a vector with a length of one thousand (including padding). The weights for each of the vectors are generated using GloVe vectors. The vectors are fed into an embedding layer (see figure 8), and output as a matrix. Embedding layers are good at capturing sentiments. TF-IDF is computed based on the articles' content (see 2.2.1). TF-IDF vectors are computed for all the terms in each article. Fake articles tend to have higher TF-IDF values, because they usually contain more unique words than real articles. This is most likely because fake article writers have a habit of using pretentious or even vulgar language in order to captivate readers and spread the article as fast as possible.

3.3 Continuous Attributes

Continuous attributes are min-max normalized, and discretized into three bins using the cut function from the Pandas library. Min-max normalization scales all the attributes to the same range ([0,1] in this case), using the formula given below, but is however sensitive to outliers.

$$Y_i = [X_i - \min(X)] / [\max(X) - \min(X)]$$

Normalizing data ensures that all the attributes have the same influence. For neural networks, this decreases gradient descent convergence times, and thus, reduces training time. Discretizing denoises the data. Each bin smoothes out noise in sections of the data.

3.4 Rumour Propagation

Time series containing time in seconds passed per rumour cascade is computed for each article. A rumour cascade is simply put either a tweet or a retweet of an article. Studies have proven that fake articles are spread faster than real articles, meaning that fake articles should have less time passed per rumour cascade than real articles.

3.5 Layers

After combining article contents, time series, and TF-IDF vectors, these combined features are passed as input for the input layer. The input layer is followed by the

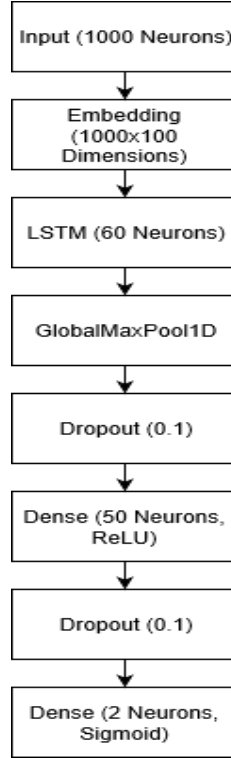


Figure 3.2: LSTM Graph

embedding layer (see 3.1.1). The next layer is a bidirectional LSTM layer with sixty neurons. There is no fixed rules for setting the number of neurons in the hidden layers in the LSTM cell, but in order to prevent overfitting, the number should be less than

$$Nh = Ns / (\alpha * (Ni + No))$$

where Ni is the number of inputs neurons, No is the number of output neurons, Ns is the number of samples in the training data set, α is an arbitrary scaling factor, usually between two and ten. Next up, is a 'GlobalMaxPool1D' layer, which downsamples by computing the most present features (maximum values) for each feature map. After the Max-Pooling layer is a dropout layer with a dropout of '0.1', meaning that ten percent of the neurons or inputs are dropped. Dropout is a regularization technique that prevents overfitting by dropping a random set of hidden units or nodes at each update during training. Nodes are dropped by setting them to zero. Finally there are two dense layers with another dropout layer between them for further regularization that classifies the articles as either fake or real. The first of these dense layers 50 neurons and apply the ReLU activation function. The last dense layer has 2 neurons and applies the Sigmoid activation function. The ReLU activation function is applied before Sigmoid to prevent the vanishing loss gradient (see chapter 2.2.4).

Chapter 4

Experimental Setup and Data Set

This chapter presents the technologies and the data set used in this thesis, as well as the experimental results.

4.1 Technologies

The code that produced the experimental results in this thesis was written in Python, because it has simple syntax rules which makes code easy to read, and it has many libraries that are handy for data science in general such as Pandas, which is good for data representation and can handle large amounts of data efficiently, Pickle, which can be used to save data structures as backups when running time demanding code (TQDM can offer progress bars for such cases), Numpy for scalable and efficient computations, and Pyplot, which offers a large variety of plots for visualization. Python also has good libraries for deep learning specifically, such as Keras.

4.1.1 Natural Language ToolKit (NLTK)

NLTK is a package for natural language processing. It contains for example stopwords from many languages. Stopwords are useless words (they do not give any context or meaning to a given sentence), which are commonly used (such as "the", "a", "and"), and should be removed.

4.1.2 Wordcloud

The Wordcloud package is used to visualize text, where the text size corresponds to significance or frequency (in this thesis TF-IDF, thus, significance). Matplotlib is needed to visualize word clouds.

4.1.3 Keras

Keras has many perks. The library is efficient on both CPU and GPU, it has preprocessing features such as tokenizing and padding, and it also has many models, such as max pooling, dropout, embedding, and bidirectional LSTM. These models can be combined, as they were in this thesis. Keras even have regularizers that can be used to avoid overfitting.

4.1.4 RE (Regular Expressions)

RE allows one to check if a given string matches a regular expression and can be used to remove numerals from strings.

4.1.5 Datetime

The Datetime package contains classes for manipulating dates and time by for example subtracting dates in order to compute time passed.

4.2 Dataset

All the data, including articles, tweets and retweets used for this thesis are retrieved from the Politifact dataset. Politifact was first started by the Tampa Bay Times in St. Petersburg, Florida in 2007. It is run by the Poynter Institute. Politifact is a fact-checking website that rates accuracy of claims by elected officials and others on its Truth-O-Meter [Pri(2019)]. The Truth-O-Meter is a scale ranging from "True" (a hundred percent real) to "Pants on Fire" (a hundred percent fake). The labels in this thesis however, have a binary distribution, either "real" or "fake". The dataset has a total of 1056 articles, of which 432 is labelled fake and 624 is labelled real. A total of 19981 records (11528 fake and 8453 real) and 16 attributes (15 continuous including the time series and article content) was used for training. 80 percent of the data set was used for training, and the rest for testing.

4.2.1 Preprocessing

The data undergoes multiple preprocessing steps in this thesis. First off, columns and rows with mostly NAN (Not A Number) values are dropped. Next, the remaining NAN values are estimated. NAN values for continuous features are estimated to the mean of the column, while Nan values for categorical features are estimated to the most common value. Then, symbols, punctuation and stopwords are removed. Finally, pre-trained 6B 100d GloVe vectors are used for word embedding. Before training the LSTM model and classifying, the dataset is split into a training set consisting of eighty percent of the data and a validation set consisting of the remaining twenty percents.

4.2.2 Synthetic Minority Over-sampling Technique (SMOTE)

The data is imbalanced with a majority of fake articles. Estimators tend to have worse performance on the minority class, hence, real records are oversampled to even the distribution. This is done using Keras' SMOTE. SMOTE generates new records based on existing records. A record is generated using a convex combination of the given record and one of the k nearest neighbours. After applying SMOTE, the distribution is X .

4.2.3 Gridsearch

Gridsearch is an SKLearn tool for tuning hyperparameters of an estimator.

Hyperparameters are constructor arguments that are not directly learnt within a given estimator. In this thesis, Gridsearch was used to estimate the number of epochs and neurons for LSTM and CNN. Gridsearch exhaustively searches over a grid of hyperparameter values for an estimator, and retains the combination of values with the best cross-validation score.

4.3 Experimental Results

This subsection discusses and analyzes the data visualizations generated from the articles and statistics in the Politifact dataset.

4.3.1 CDF

A Cumulative Distribution Function (CDF) is used to find the probability that a random variable is less than or equal to a given value. A probability, y , is computed by

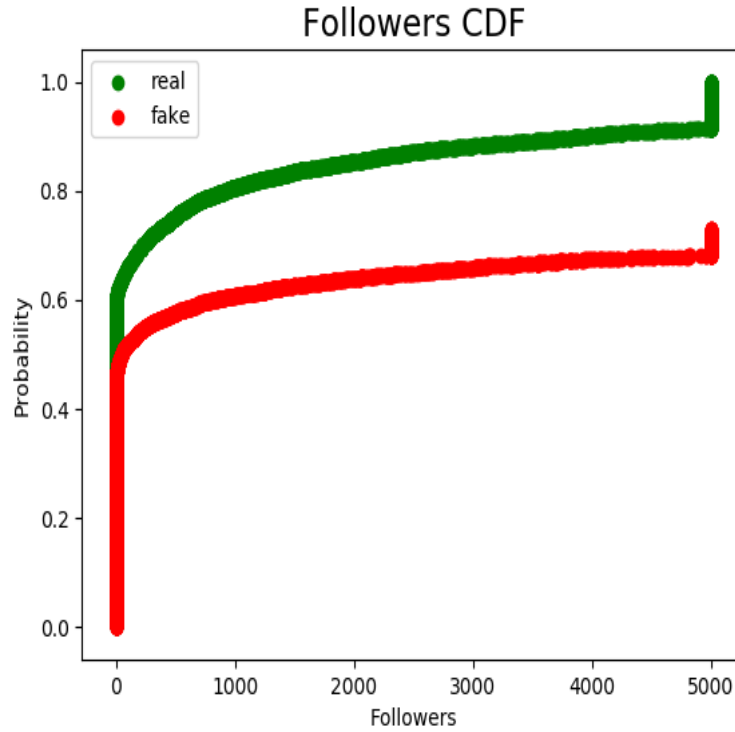


Figure 4.1: Followers CDF (Cumulative Distribution Function)

dividing the number of events that are less than or equal to y by the number of possible outcomes (all values less than or equal to y).

4.3.2 Users

Figure 4.1 indicate that users that tweet fake articles have less followers than users that tweet real articles (users that tweet real articles have a higher probability of having many followers than users that tweet fake articles). Figure 4.2 shows that most users that tweet real articles have approximately 850 tweets, while most users that tweet fake articles have less than 800 tweets. Real articles also have a higher range of tweet frequencies than fake articles. These results indicate that users that tweet real articles tweet more than users that tweet fake articles. A possible explanation for the results might be that the users that tweet fake articles create new accounts as people learn that their tweets are fake. This implies that metadata such as follower count per user is viable for classification.

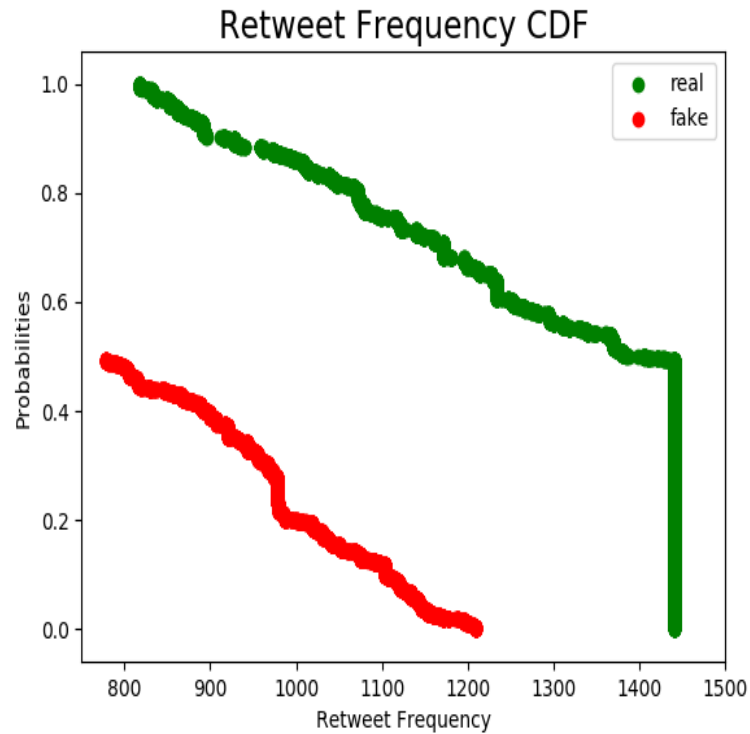


Figure 4.2: Tweet Frequency CDF

4.3.3 TFIDF

Figure 4.3 shows that terms from real articles tend to have higher TF-IDF than terms from fake articles. This means that fake articles have a smaller vocabulary than real articles and that they tend to reuse words. For example, a lot of fake articles revolve around presidential elections, which can be seen in figure 4.4 (Trump, Obama, and Clinton are among the most common words in fake articles). Figure 4.4 also shows that fake articles contain lots of superlatives (i.e. supreme), the language in fake news tend to be negative in regards to sentiments (i.e. suspended), and that the plots are usually either celebrity gossip (i.e. Dogg) or serious matters like criminal cases or events that can affect people's daily lives (i.e. killed). Figure 4.5 displays the most common terms in real articles for comparison.

4.3.4 Rumour Propagation

According to figure 4.6 and 4.7, tweets of fake news articles reach much more followers per retweet, and are also retweeted faster. The differences are significant, especially for the time cascade. Fake news spread a lot quicker than fake news. Fake news also start spreading much earlier than real news. Figure 4.7 also tells us that even though fake

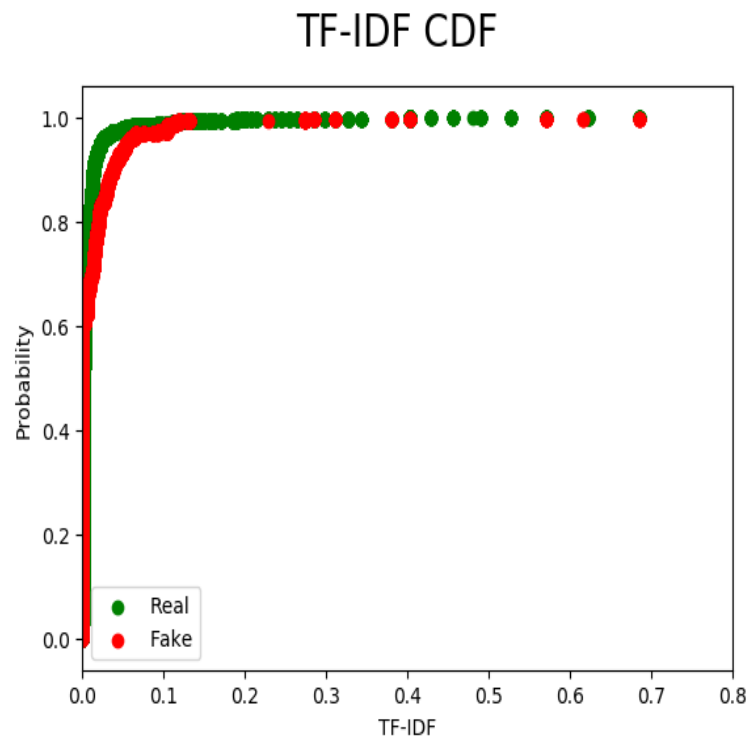


Figure 4.3: TF-IDF CDF

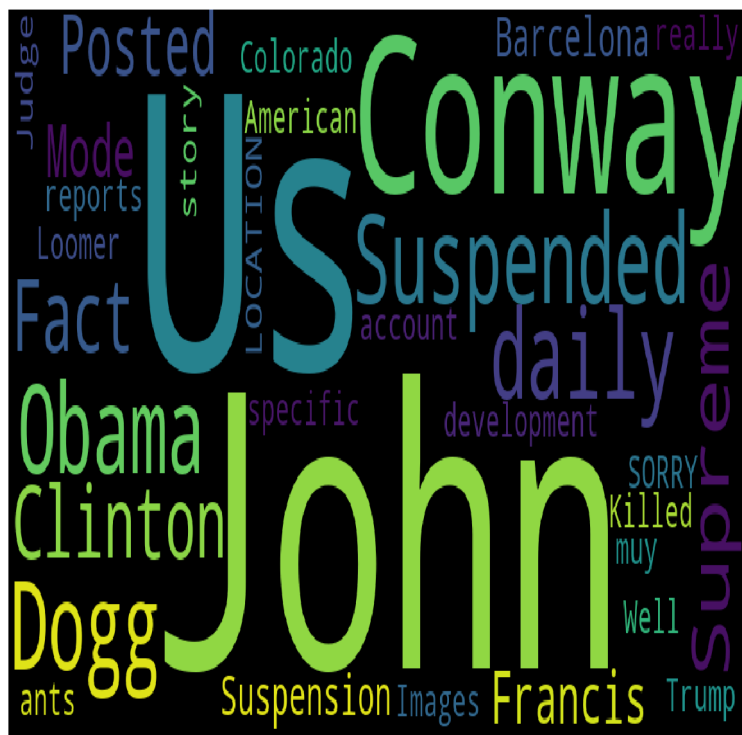


Figure 4.4: fake TFIDF Common Wordcloud



Figure 4.5: real TFIDF Common Wordcloud

news spread faster than real news, real news actually spread quickly, when they first start to spread. This might be because followers of authors who often tweet real tweets tend to be a lot more loyal than followers of fake tweet authors, and that they receive notifications when the first tweet is tweeted, thus, they all retweet said tweet in the same time span.

4.4 Accuracy Measures

A confusion matrix is an accuracy measure for labelled data sets. Confusion matrices can be computed for multiple classes, but for a two-class problem, it will have four cells containing True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN), see figure 4¹.

TP is the number of correctly predicted positive (real) records (articles). FP is the number of wrongly predicted positive records. FN is the number of wrongly predicted negative (fake) records. TN is the number of correctly predicted negative records.

¹https://www.google.no/search?q=confusion+matrix&tbm=isch&ved=2ahUKEwj9uISIk5LoAhWHapoKHcSIB04Q2-cCegQIABAA&oq=confusion+matrix&gs_l=img.3.35i3912j0i3018.96895.98917..99196...0.0..0.120.1171.17j1.....0....1..gws-wiz-img...0i19.q

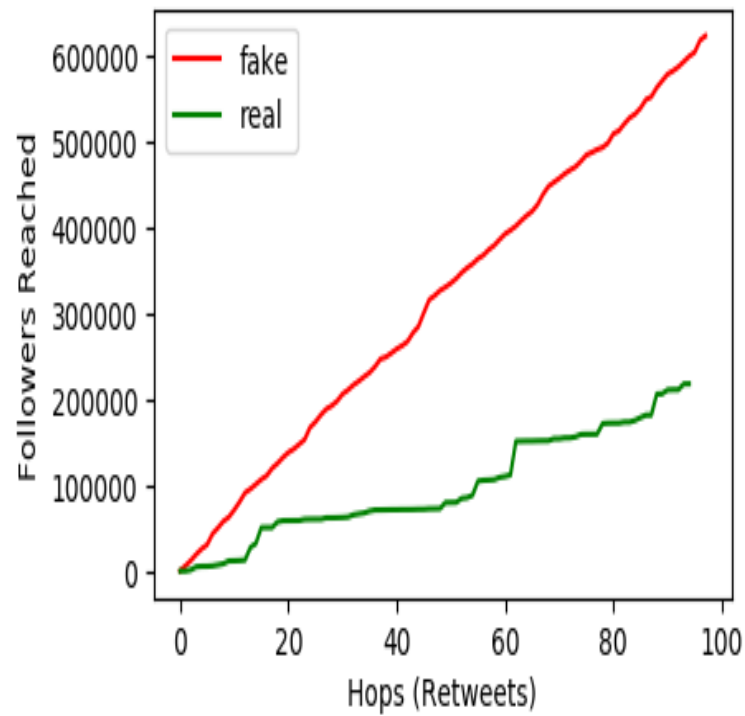


Figure 4.6: Followers Reached per Hop

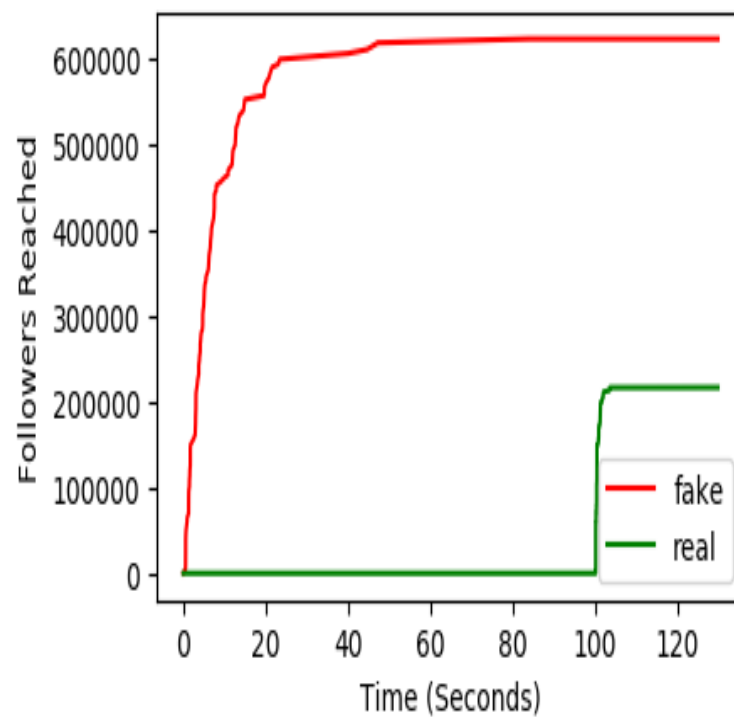


Figure 4.7: Time Cascade

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 4.8: Confusion Matrix

Precision is the number of correctly predicted positive records divided by all the records predicted as positive:

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

Recall is the number of correctly predicted positive records divided by all the records that are actually positive:

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

Micro accuracy is computed as

$$Micro = \frac{TP + TN}{TP + TN + FP + FN}$$

and is the percentage of correctly predicted records for both classes. Macro accuracy is the mean of the recalls:

$$Macro = \frac{\frac{TP}{TP+FN} + \frac{TN}{TN+FP}}{2}$$

F1-Score is a useful accuracy measure if you want a balance between precision and recall, and the class distribution is uneven. The formula for F1-Score is given below:

$$F1 = \frac{2 * (Precision * Recall)}{Precision + Recall}$$

Area Under Curve (AUC) accuracy measures how well a model distinguishes between two classes. The 'Curve' in AUC is an ROC curve, which is a probability curve plotted with TP on the y-axis and FP on the x-axis. It is a good measure for binary classification problems with a skewed distribution (as is the case in this thesis). The

formula for AUC is the same as for Micro accuracy, but AUC takes prediction probabilities as input instead of predicted labels. The formula for AUC is

$$AUC = \int_0^1 TPR(FPR^{-1}(x))dx$$

where

$$TPR(T) = \int_T^{\infty} f_1(x)dx$$

and

$$FPR(T) = \int_T^{\infty} f_0(x)dx$$

F_0 is the probability density function for the negative class and f_1 is the probability density function for the positive class.

4.5 Results

The proposed models takes time series, article content and various metadata as input and classifies news articles as either real or fake using bidirectional LSTM. The results was compared to models based on CNN, SVM and NB. Graphs, descriptions and results are given below. <https://www.diagrameditor.com/>

4.5.1 SVM

SVM classifies multi-dimensional data by reducing dimensionality, making the problem linearly-separable, and then drawing a separator as a hyperplane. SVM is effective when there are many features, but is best used on small data sets due to training time. The SVM model was trained using 10000 iterations and generated using SciKit-Learn (SK-Learn) with the 'c' parameter (classification margin) set to 1, a linear kernel, 'degree' set to 3, and 'gamma' set to auto. The 'degree' parameter is the degree of the polynomial kernel function and is ignored if a polynomial kernel is not used. Gamma is the kernel coefficient for the non-linear kernels. When set to auto, $1/n$, where n is the number of features, will be used instead.

4.5.2 NB

Naive Bayes classifiers are simple and fast probabilistic classifiers that classify labels based on probabilities using frequencies. For NLP, non-naive Bayes classifiers compute zero probability for sentences not in the training set. Naive Bayes classifiers, however, treat terms independent of each other and compute per-term-probabilities. NB is best

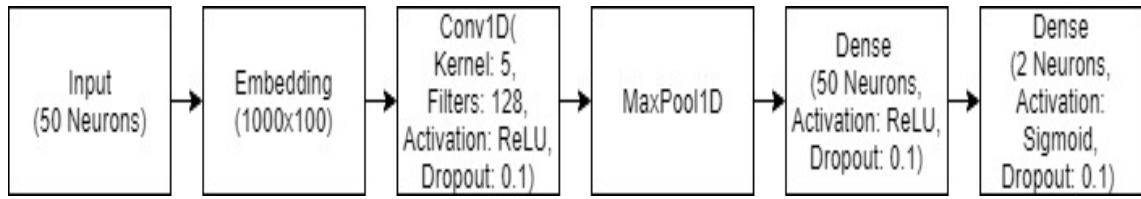


Figure 4.9: CNN Graph

applied on TF-IDF vectors, because TF-IDF reflects term importance per document. Smoothing is also important when using NB, to avoid zero values. Multinomial Naive Bayes is used because term frequencies are discrete values, meaning that the distribution is multinomial.

4.5.3 CNN

CNN is a type of neural networks that convolutes a kernel filter to estimate data points. For each convolution, the filter is used to estimate the value of the center point. CNN is generally best used to estimate pixels for computer vision, i.e. image classification. However, it can also be used for NLP. CNNs are fast to train and can have decent accuracy, even though they don't have memory like LSTM. CNNs can be used for NLP by sliding the sliding window or kernel filter over patches of the embedding matrix or a matrix of one-hot vectors. In this model, the kernel filter is slid over the embedding matrix consisting of GloVe vectors, which is of the dimensions, 1000x100. On a general basis, LSTM outperforms CNN on NLP problems, but CNNs can perform better than LSTM on feature detection problems with short texts. LSTM performs best on problems that involve long sequences and require some sort of memory such as translations.

4.5.4 LSTM

See chapter 2.2 for information on LSTM. The layers are depicted in figure 3.4.

4.5.5 Optimization Techniques

Optimization techniques are used to update the weights in a neural network, in order to minimize the loss function. Stochastic Gradient Descent is the most common optimizer. However, multiple other optimizers were tested in this thesis. The highest achieved accuracy using RMS Prop and AdaGrad respectively, was X and Y.

Root Mean Square Propagation (RMSProp) uses different learning rates for each parameter. The learning rates are adjusted using weight gradient averages, hence, the technique performs well on noisy data.

Adaptive Gradient (ADAGRAD) also uses per-parameter learning rates. ADAGRAD learning rates are low for frequent data and high for infrequent data, and thus improve performance on problems with sparse data (i.e. NLP problems).

The best result was obtained using the Adaptive Moment (AdaM) optimizer which yielded an accuracy of X. Like RMSProp, Adam also uses weight gradient averages, and are thus effective on noisy data. However, Adam also uses variance, thus, combining the perks of RMSProp and ADAGRAD. Adam is also easy to implement and computationally efficient.

RBF

Polynomial

Sigmoid

4.5.6 Training Times

Training times: LSTM 5 epochs, SVM 10k iterations

4.5.7 Loss Functions

Loss function figure LSTM and CNN

4.5.8 Accuracies

Accuracy tables (F1-score, micro-averaging and macro averaging. simple weight - none, micro globally, macro per class)

The accuracy measures explained in 4.4 are listed in table 4.1 for all the implemented estimators.

	Accuracy		Fake Tweets			Real Tweets			
Estimator	Micro	Macro	Precision	Recall	F1	Precision	Recall	F1	AUC
LSTM	y	z	i	j	k	l	j	k	l
CNN	y	z	i	j	k	l	j	k	l
SVM	y	z	i	j	k	l	j	k	l

Multimodal NB	y	z	i	j	k	l	j	k	l
------------------	---	---	---	---	---	---	---	---	---

Table 4.1: Accuracies

See table 4.2 for results with and without tuning using Gridsearch on the DNN estimators.

	Accuracy		Fake Tweets			Real Tweets			
Tuning	Micro	Macro	Precision	Recall	F1	Precision	Recall	F1	AU
Gridsearch	y	z	i	j	k	l	j	k	l
None	y	z	i	j	k	l	j	k	l

Table 4.2: Tuning

As mentioned in 4.2.2, the class distribution is skewed, and SMOTE is used to oversample the minority class (real). The majority class (fake) is also undersampled. The results of undersampling and oversampling separately using the DNN estimators is given in table 4.3.

	Accuracy		Fake Tweets			Real Tweets			
Sampling	Micro	Macro	Precision	Recall	F1	Precision	Recall	F1	AU
SMOTE	y	z	i	j	k	l	j	k	l
Undersampling	y	z	i	j	k	l	j	k	l

Table 4.3: Sampling

The confusion matrices for all the implemented estimators is given in table 4.4-7.

		Actual	
		Positive	Negative
Predicted	Positive	pp	pn
	Negative	np	nn

Table 4.4: LSTM Confusion Matrix

		Actual	
Predicted	Positive	Positive	Negative
	Negative	pp	pn
		np	nn

Table 4.5: CNN Confusion Matrix

		Actual	
Predicted	Positive	Positive	Negative
	Negative	pp	pn
		np	nn

Table 4.6: SVM Confusion Matrix

		Actual	
Predicted	Positive	Positive	Negative
	Negative	pp	pn
		np	nn

Table 4.7: Multimodal NB Confusion Matrix

Chapter 5

Discussion

This chapter compares the results of the proposed solution with baseline solutions and discusses the implications of the experimental results.

5.1 Baseline Comparison

The proposed model demonstrated better prediction accuracy than all the estimators from the related works. The purpose of [Liu and Wu(2019)] is to detect fake news as fast as possible, and the work claims to be able to detect fake news within five minutes. The high accuracy in this thesis' proposed model comes at the expense of higher training time, but the accuracy for the proposed model is still higher than the proposed model in [Liu and Wu(2019)] when spending only five minutes on reading data and training the model (see comparison in table 5.1).

		Fake Tweets			Real Tweets		
Model	Accuracy	Precision	Recall	F1	Precision	Recall	F1
PPC RNN+CNN [Liu and Wu(2019)]	0.921	0.896	0.962	0.923	0.949	0.889	0.918
LSTM	z	i	j	k	l	j	k

Table 5.1: Early Detection

5.2 Implications of Findings

The inputs besides the metadata are sequential, and as expected, LSTM demonstrated the best results with an accuracy of 98 percent. The CNN performed surprisingly well, it

had almost the same accuracy as LSTM (97 percent). CNN performing well on the given data supports the hypothesis that fake articles contain more superlatives than real articles, because CNNs work well for feature detection (detecting terms in this case).

CNN is much faster at training than LSTM, and since fake news spread much faster than real news, one might argue that the CNN-based model is the better choice for fake news detection. NB demonstrated a decent performance with an accuracy of 71 percent without oversampling and 67 percent after oversampling. The accuracy is most likely reduced after oversampling because the estimator puts more weight on real records after oversampling them, making the estimator bias to them. Even though the overall accuracy is reduced, the micro accuracy is improved (see table 5.2).

	Accuracy		Fake Tweets			Real Tweets			
Sampling	Micro	Macro	Precision	Recall	F1	Precision	Recall	F1	AUC
SMOTE	y	z	i	j	k	l	j	k	l
None	y	z	i	j	k	l	j	k	l

Table 5.2: Sampling

Even though NB does not perform as well as the graph-based models, NB is fast at training and simple to use. SVM got an accuracy of 50 percent, which is barely any better than chance. Precision total percentage of relevant results. high recall - many correctly classified positives (TP) High Precision - many correctly classified (TP and TF) records F1 score harmonic mean of precision and recall. want to maximize f1 score

Chapter 6

Conclusion and Future Directions

6.1 Future Directions

This thesis presents a hybrid model based on article content and rumour propagation, which performs better than state-of-the-art models. That is not to say that other models with better performance can not be developed. For example, the none-sequential metadata can be used with machine learning algorithms. Even if the accuracy does not exceed the accuracy of the proposed model, the training time will most likely be less.

6.1.1 Data

Most of the tweets are not retweeted at all, meaning that there is not much data provided for the rumour propagation time series and some of the metadata. User characteristics can be used to identify naive users who easily fall for fake news and share them. The data can be denoised further (i.e. remove irrelevant text from article content). More data can also be gathered from other social media platforms. The data set also contains many other features that were not used in this work, which can be applied for example in the means described in the beginning of this section. Many of the records contained video URLs instead of articles, and had to be discarded. These videos can be used for video classification using CNN. Images such as user profile pictures can also be added to the data set for image classification using CNN. One can also implement a combination of LSTM and CNN, using LSTM on sequential data, and CNN on image data or metadata.

6.1.2 Graphical User Interface (GUI)

The proposed solution can be further developed by creating a web application with a GUI, that takes article content and/or a number of features as input, and detects if the given article is fake or not using the proposed model.

6.2 Conclusion

This thesis successfully utilizes article content and rumour propagation paths to improve fake news detection by modeling a graph neural network. The proposed solution, using both LSTM and CNN, scores better than state-of-the-art solutions from related works and the SVM- and NB estimator implemented in this thesis, with the best scores being 98 percent and 97 percent respectively. The proposed solution also takes continuous metadata as input, in addition to the inputs mentioned above. CNN is faster at training than LSTM and the accuracy is almost the same, hence, choice of architecture depends on the requirements. Machine learning algorithms good for small data sets.

List of Figures

1.1	Sample Data Set	4
2.1	Neural Networks Layers	8
2.2	Recurrent Neural Network with Loops	10
2.3	Recurrent Neural Network with loops	11
2.4	RNN Cell and LSTM Gates	12
2.5	LSTM Cell Equations	12
2.6	GRU-CNN Architecture Diagram	17
2.7	GRU Cell	17
3.1	Proposed Solution	19
3.2	LSTM Graph	21
4.1	Followers CDF (Cumulative Distribution Function)	26
4.2	Tweet Frequency CDF	27
4.3	TF-IDF CDF	28
4.4	fake TFIDF Common Wordcloud	28
4.5	real TFIDF Common Wordcloud	29
4.6	Followers Reached per Hop	30
4.7	Time Cascade	30
4.8	Confusion Matrix	31
4.9	CNN Graph	33

List of Tables

2.1	Related Works	16
4.1	Accuracies	35
4.2	Tuning	35
4.3	Sampling	35
4.4	LSTM Confusion Matrix	35
4.5	CNN Confusion Matrix	36
4.6	SVM Confusion Matrix	36
4.7	Multimodal NB Confusion Matrix	36
5.1	Early Detection	37
5.2	Sampling	38

Bibliography

- [Chinchilla(2019)] Laura Chinchilla. Post-truth politics afflicts the global south, too, 2019. URL <https://www.kofiannanfoundation.org/supporting-democracy-and-elections-with-integrity/annan-commission/post-truth-politics-afflicts-the-global-south-too/>.
- [Avelar(2019)] Daniel Avelar. Whatsapp fake news during brazil election ‘favoured bolsonaro’, 2019. URL <https://www.theguardian.com/world/2019/oct/30/whatsapp-fake-news-brazil-election-favoured-jair-bolsonaro-analysis-suggests>.
- [Gouvernement(2018)] The French Gouvernement. Against information manipulation, 2018. URL <https://www.gouvernement.fr/en/against-information-manipulation>.
- [Banjo and Lung(2019)] Shelly Banjo and Natalie Lung. How fake news and rumors are stoking division in hong kong, 2019. URL <https://www.bloomberg.com/news/articles/2019-11-11/how-fake-news-is-stoking-violence-and-anger-in-hong-kong>.
- [Cheo(2018)] James Cheo. Fake news can make - or break - stock prices, 2018. URL <https://www.businesstimes.com.sg/opinion/fake-news-can-make-or-break-stock-prices>.
- [Jardine(2018)] Eric Jardine. Beware fake news, 2018. URL https://www.cigionline.org/articles/beware-fake-news?gclid=EAIaIQobChMIl_X4nZn-5wIVkRsYCh11MQueEAAYASAAEgI7LPD_BwE.
- [Polyakov(2018)] Alexander Polyakov. Detecting fake content: One of the biggest challenges for 2020, 2018. URL <https://www.forbes.com/sites/forbestechcouncil/2020/01/02/detecting-fake-content-one-of-the-biggest-challenges-for-2020/#8c82b4e1219d>.
- [A.Lakshmanarao(2019)] T. Srinivasa Ravi Kiran A.Lakshmanarao, Y.Swathi. An efficient fake news detection system using machine learning. 2019. ISSN 2278-3075.

- [Junaed Younus Khan and Afroz(2019)] Anindya Iqbal Junaed Younus Khan, Md. Tawkat Islam Khondaker and Sadia Afroz. A benchmark study on machine learning methods for fake news detection. 2019.
- [Iglesias(2019)] Lara Lloret Iglesias. Fake news detection using deep learning. 2019.
- [Liu and Wu(2019)] Yang Liu and Yi-Fang Brook Wu. Early detection of fake news on social media through propagation path classification with recurrent and convolutional network. 2019.
- [Pri(2019)] The principles of the truth-o-meter: Politifact’s methodology for independent fact-checking, 2019. URL <https://www.politifact.com/article/2018/feb/12/principles-truth-o-meter-politifacts-methodology-i/>.