OULUN AMMATTIKORKEAKOULU

Lara Aslan

AUTOMATIC FALSE NEWS DETECTION USING MACHINE LEARNING

AUTOMATIC FALSE NEWS DETECTION USING MACHINE LEARNING

Lara Aslan
Bachelor's Thesis
Autumn 2023
Degree Programme in Information
Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Bachelor of Engineering, Degree Programme in Information Technology

Author: Lara Aslan
Title: Automatic False News Detection Using Machine Learning
Supervisors: Jukka Jauhiainen, Michal Ptaszynski
Term and year: Autumn 2023
Number of pages: 48

False news refers to false or misleading information presented as real news. In recent years, we have seen a growing trend on false news, especially on the Internet. The goal for this thesis was to study automatic false news detection using machine learning and natural language processing techniques, and to determine which of the techniques works the most effectively.

This thesis first studies what exactly is false news, how it differs from other types of misleading information, and the results achieved by other researchers about the same topic. After building a foundation to understand false news, and the various ways of automatically detecting it, this thesis provides its own experiments. These experiments were done on four different datasets, one that was made just for this thesis, and using 10 different machine learning methods.

The results for this thesis were good, and the results answered the original research questions set up in the beginning of this thesis. From the experiments, this thesis could determine that passive-aggressive algorithms, support vector machines, and random forests are the most efficient methods to automatically detect false news. This thesis also concluded that more complex experiments, such as using multiple levels of identifying false news, or detecting computer-generated false news, require more complex machine learning models.

Keywords: False News Detection, Machine Learning, Natural Language Processing, ChatGPT

PREFACE

I would like to thank Professor Michal Ptaszynski, who helped me through the thesis by meeting with me regularly, discussing my work and providing me with good feedback. With his guidance, I was able to understand difficult concepts related to my thesis, as well as giving me interesting new ideas regarding the thesis and its experiments.

I would also like to thank my thesis examiner, Principal Lecturer Jukka Jauhiainen. He gave me encouraging feedback about my thesis, as well guiding me through the technical process regarding my thesis and graduation.

CONTENTS

1 INTRODUCTION

False news can be described as false or misleading information, often with the idea of damaging the reputation of the entity being targeted (Wikipedia 2023a). Recent years have seen the spread of false news, especially on the Internet. This issue needs to be addressed, as the growing spread of false news poses a threat to journalism, can cause political turmoil, and can negatively impact people's daily life (Wang 2017).

While the internet already has many websites that professionally perform fact-checking on news articles, such as PolitiFact (PolitiFact 2023), FactCheck.org (FactCheck.org 2023), FactChecker (The Washington Post 2023), Snopes (Snopes.com 2023), The Reporters' Lab (Duke Reporter's Lab 2023) or FaktaBaari (Faktabaari 2023), in today's world, websites like these managed by human volunteers are not enough. False news or false information is getting harder to detect as the ways to create it become more advanced and accessible (Reis, Correia, Murai, Veloso & Benevenuto, 2019). For example, ChatGPT (ChatGPT 2023) can just in a few seconds generate an article, a review, or a statement that can be false but still look believable (Hsu & Thompson, 2023). With the help of social media, these pieces of false information can spread like wildfire to millions of people in just a few seconds. This is why, it is important to include at least some level of automation in the process of fact-checking.

The goal of this paper is to study automated false news detection. I review previous work related to this subject, as well as conduct analysis of such data and classification experiments using machine learning algorithms. Specifically, several classification models using machine learning and natural language processing techniques are tested to find the best way to automatically detect when a news article or a piece of information is true or false, and analyse the language commonly used in news articles that could be considered false. An experiment on how the developed classifiers perform when given articles written by a human compared to those generated automatically is also conducted.

The usage of generative Large Language Models, including the recently popular ChatGPT makes it easier to write false content in large quantities. Thus, the inclusion of data generated with ChatGPT is important when it comes to false news detection.

Most fact-checking services or previous research conducted in automated false news detection focuses mainly on political issues. False news, however, can be spread about any topic, like how we saw a rise in false information being spread about COVID-19 (Apuke and Omar, 2021). Therefore, it is important to include other topics in the scope of analysis as well.

The two main research questions posed by this research were:

1. What are the best computational methods to use when detecting false news?
2. Will there be a difference in results when using human generated text vs automatically generated text?

The remainder of this paper is arranged as follows. In section 2 I talk about the different types of deceptive information and lay out the specific background of what the present research focuses on. In section 3 I present previous research done in the field of false news detection. In Section 4 I introduce the datasets used in my research. In section 5 I introduce the applied methods used in my own research. In section 6 I go through the experiments that I performed on the datasets that I used. In section 7 I discuss the results of my experiments. In section 8 I conclude my research.

2 BACKGROUND

The scope of deceptive information especially in today's age is large, and often different definitions of different types of deceptive information are used indiscriminately with each other. This creates confusion and uncertainty when trying to differentiate one piece of deceptive information from another.

Different types of deceptive information include rumours, hoaxes, fake news, fake reviews, satires, urban legends, and propaganda, among many more. It is important to take into consideration that sometimes these different types of deceptive information can intertwine with each other. For example, a false news article can be satirical. To understand better the differences between these different types of deceptive information a summary of all known types of deceptive information so far with their short definitions is provided in Table 1.

In this research, as previously pointed out, the main focus is detecting false news. However, even false news can be divided into sub-categories where each different sub-category has a specific purpose and a different impact level. This is why various guides and taxonomies have been created to differentiate and understand different types of false news (Svärd and Rumman, 2017; Wardle, 2017; Boungeru, Gray, Venturini and Mauri, 2018).

Wardle (Wardle 2017) proposed a taxonomy of misinformation and disinformation that divided false news into seven different types of mis- and disinformation, and provided a description of the intended harm that these different types of false news are causing. This research mainly focuses on misleading content and fabricated content. This taxonomy is shown in table 2.

To understand false news, we also need to understand the definition of news that can be classified as real. For news to be considered legitimate or real, it needs to fill certain journalistic standards. This standard usually means that the news is neutral, it uses the right sources, and is factual based on the information available at the time. Chong, et al. (2020) in their research on misinformation considered news to be legitimate or real if it followed these following characteristics:

1. Presented in a neutral, balanced, and non-inciting manner.
2. Verifiable by an independent source or party within reasonable limits.

3. Accurate and factual, based on the information available or as provided by the source.
4. Comprehensive - with no malicious censorship, modification, or manipulation.

For this research, a look out is kept for these characteristics proposed by Chong, et al. (2020) when differentiating real news from the false ones.

TABLE 1. Descriptions of different types of deceptive information types.

Type	**Description**
Rumour	Quickly spreading story or news that can be true or invented.
Hoax	A deceptive piece of information used to trick people into believing in it.
False news	False or misleading information presented as news. Often used as a tool to do harm.
False reviews	A review that is not an actual consumer's opinion or doesn't reflect the actual opinion of a consumer. Often used to manipulate a consumer not to buy a certain product.
Satires	A type of a parody where a content is presented with irony or humour. Often used to criticize events, peo-ple etc.
Urban legends	A false story that is circulated between people as true. Usually humorous, horrifying, or cautionary in its na-ture.
Propaganda	Information that is usually biased or misleading. It's used to promote a political cause or a point of view.

TABLE 2. “7 types of mis- and disinformation”. A taxonomy created by Wardle (Wardle, 2017).

Type	**Description**
Satire or parody	No intention to cause harm but has potential to fool.
Misleading Content	Misleading use of information to frame an issue or individual.
Imposter Content	When genuine sources are impersonated.
Fabricated Content	New content is 100 per cent fake, designed to deceive and do harm.
False Connection	When headlines, visuals or captions don’t support the content. Also known as clickbait.
False Context	When genuine content is shared with false contextual information.
Manipulated Content	When genuine information or imagery is manipulated to deceive.

3 PREVIOUS RESEARCH

This section goes through previous research that has been conducted in the field of false news detection and related areas.

3.1 False news detection

Rubin, et al. (2016) studied satirical news, and how to expose them as false news. The contrast between satirical news and false news is worth noting. Satirical news leave cues on purpose in its text to reveal the false nature of the news, whereas false news tries to convince the reader to believe in it. In their research, Rubin, et al. (2016) proposed an algorithm based on Support Vector Machines (SVM) with five features, namely, absurdity, humour, grammar, negative affect, and punctuation to predict satirical news. Their research was very successful, as they were able to achieve a 90% precision and 84% recall in satirical news detection.

Thota, et al. (2018) used Deep Learning architectures to detect false news. They highlighted that a problem with majority of false news detectors is that they only use binary classification methods, making them unable to understand the relationship between two pieces of texts. In their research, they tackle this problem through stance detection, in which they use neural network architecture to predict how similar the headline is to a news article. Their model provided to be successful, as they were able to detect when a news article was false with stance detection with 94.21% accuracy, which outperformed existing models at the time by 2.5%.

Karimi, et al. (2018) conducted their research about false news detection with an inclusion of various degrees of "falseness". They propose a Multi-source Multi-class Fake News Detection framework to tackle this problem. This framework combines automated feature extraction, multi-source fusion and automated degrees of fakeness detection into a one single model. Their model could differentiate between the different degrees of fakeness from the news that they used. They also integrated multiple sources to fake news detection, which could help false news detection as multiple sources give a much better context when detecting false news, as opposed to only using the context given by the news article.

Oshikawa, et al. (2018) studied the potentials and limitations of Natural Language Processing (NLP) solutions in false news detection. NLP techniques are perhaps the most common way of analysing false news, and their study prove that NLP techniques are useful in automatic fake news detection. Das, et al. (2023) additionally studied the task of automated fact-checking through the point of NLP techniques. However, their research points out more of the fact that automated fact-checking is less reliable when compared to manual fact-checking. Das, et al. (2023)'s solutions for this limitation is to develop a hybrid system for automatic fact-checking, that would use humans in the process alongside with computers.

Waikhom, et al. (2019) used ensemble machine learning methods, such as XGBoost, Bagging, Random Forest (RF), Extra Trees, and Gradient Boost. The methods they used allowed them to achieve relatively high accuracy scores in classification when using the LIAR dataset (Wang 2017). Ahmad, et al. (2020) as well used ensemble methods with machine learning in their research about false news detection. They as well achieved very good results with ensemble methods. Waikhom, et al. (2019)'s and Ahmad, et al. (2020)'s research and results conclude, that machine learning algorithms well for false news detection when implemented with ensemble learners.

Gundapu, et al. (2021) conducted research on false news detection for COVID-19 related news. They used machine learning models, deep learning models and transform models to conduct their research. They achieved the best results when they developed an ensemble model consisting of all three different transform models that they used (BERT, ALBERT, and XLNET). With this ensemble model, they were able to receive an excellent accuracy score of 98%.

Wu, et al. (2021) conducted their research on multimodal (text and image) false news detection. They proposed a Multimodal Co-Attention Networks-based model to better include text and image together for the purpose of false news detection. They proposed a Multimodal Co-Attention Networks-based model to better include text and images together for the purpose of false news detection. Their model first extracted visual features from images and then textual features from the text, fusing these extracted features together that then can be used to detect false news. Their model was able to achieve good accuracy results on the two datasets that they used for their research. On the first dataset that they used they achieved an accuracy score of 80%, and on the second they achieved an accuracy score of 89%. Nadeem, et al. (2023) recently concluded research on utilizing visual features for false news detection as well. They proposed a multimodal Extreme Fake News Detection (EFND) that gathers context, social context, and visual data to create a multimodal vector. The results they achieved were high, with the accuracy score being 98% and 99% on different datasets.

3.2 Linguistic and textual analysis of false news

Singh, et al. (2017) used linguistic analysis alongside machine learning in their research. Their research provides interesting information about linguistic differences between false and real news. From their research one can learn that in general, false news tends to be shorter, show less expertise or confidence, appear negative in tone, and show less analytical thinking. However, their research shows that the package they used for linguistic analysis LIWC (Linguistic Analysis and Word Count), associates the language found in false news to be more authentic. Since in LIWC, a higher authenticity score is associated when the language is more personal and disclosing. In comparison, a lower authenticity score is associated when language is more guarded and distanced. This could provide reasoning for why people can be tricked into believing false news.

Ahmed, et al. (2018) proposed a detection model that combines text analysis using n-gram features and term frequency with machine learning classification. They also introduce a new n-gram model in their research, that generates various sets of n-gram frequency profiles from their trained data, to differentiate between false and true content. Their research conducted that linear function-based classifiers achieved better results than non-linear classifiers. The research also found that if an n-gram size was increased, the detection accuracy decreased. This would suggest that the language used in false news is not consistent.

3.3 Automatically generated text detection

Mitrović, et al (2023) conducted research on detecting short texts generated by ChatGPT using a transformer-based model. In their research, they also analysed the language generated by ChatGPT and concluded that ``ChatGPT’s writing is polite, without specific details, using fancy and atypical vocabulary, impersonal, and typically it does not express feelings." (Mitrović, Andreoletti, Ayoub, 2023, p. 1). The research focused on restaurant reviews generated by ChatGPT, and the goal of it was to classify the reviews according to whether it was created by a human or ChatGPT.

The research achieved a good 79% accuracy, even though the research stated that the transformer-based model had problems with differentiating between human and ChatGPT-generated reviews.

3.4 False news detection based on user interaction

Tacchini, et al. (2017) proposed an idea regarding hoax and false news detection, where the nature of a Facebook post could be determined by the users who "like" the posts. The baseline for their research was that a user who "likes" a post determined as hoax, is anticipated to "like" even more hoax posts. They would analyse a post according to the users who "liked" the post, and if there was enough number of users who had previously "liked" several posts determined as hoax, the current post being analysed would be determined as hoax as well. For their experiments, they used two different classification techniques: logistic regression and boolean label crowdsourcing (BLC). Both of their techniques used achieved very high results in detecting whether a Facebook post could be determined as a hoax or non-hoax, suggesting that analysing the users interacting with Facebook posts can accurately determine the nature of the Facebook posts.

Del Tredici, et al. (2020) used linguistic analysis and user detection to detect false news. They proposed a model that would create representations of users on social media based on the language that they use and the news that they spread, and this model would be used to detect false news. The model was built by using Convolutional Neural Networks (CNNs), as it suits well for text classification. In their research, they analysed the language commonly used by people who share false news. The study concluded that the language used by users who spread false news is consistent, which in turn makes it easier to detect news based on just by the people who share them, just like in Tacchini, et al. (2017)'s research.

4 DATASETS

In this section, the datasets used in my research are introduced. These datasets include three datasets created by previous research, namely, LIAR (Wang, 2017) and FakeNewsNet (Shu, Mahudeswaran, Wang, Lee, and Liu, 2019), and Twitter15 (Liu, 2015), as well as a novel dataset that I build by using ChatGPT. The sizes of the datasets, the data elements used in the datasets, and the different values of the data elements are analysed. A summary of all datasets that can be seen on Table 3.

TABLE 3. A summarisation of all the used datasets.

Datasets	**All Samples**	**True Samples**	**False Samples**	**Information Type**
LIAR	12851	7134	5707	News related to politics.
FakeNewsNet	23921	6480	17441	News related to politics and celebrity gossips.
Twitter15	1490	372	370	Rumours spread on Twitter.
Novel ChatGPT	300	100	200	Automatically generated false and real news articles.

4.1 LIAR Dataset

The LIAR dataset (Wang, 2017) is a benchmark dataset created for fake news detection. It contains 12.8 thousand real-world manually labelled short statements that were collected from PolitiFact.com with various contexts. The dates for the statements are primarily from 2007-2016. The dataset also includes an analysis report and links to source documents for each statement, as well as information about the speaker, the speaker's job title, subject, political party affiliation, the credit history of the speaker, and the context for each statement.

The dataset has six different labels to determine the truthfulness ratings. These labels are *pants-fire*, *false, barely-true, half-true, mostly-true,* and *true*. The p*ants-fire* case represents when the statements is completely false, and the *true* label represents when the statement is completely true. The distribution of cases for each label is balanced, except for *pants-fire* that has a significantly less cases compared to other labels. The *pants-fire* label has 1,050 cases whereas the other labels have cases ranging from 2,063 to 2,638 cases.

The statements have 732 different subject types ranging from various topics, where the most frequent subject being health care, and it appears in the dataset 5 times. The average statement is 17.9 tokens long. Most of the speakers of the statements are U.S. politicians, but other speaker types are also included such as journalists, social media posts and Internet newspapers. Overall, there is 2910 unique speakers, and each speaker appears 3.5 times on average in the dataset. The most common speaker in the dataset is Barack Obama, who appears in the dataset 5 times.

TABLE 4. An example of a randomly chosen statement from the LIAR dataset. The label history shows how many times the speaker has made a statement that belongs into one of the six different label cases in the dataset.

Elements	**Values of Elements**
ID	8303
Label	half true
Statement	Tuition at Rutgers has increased 10 percent since Gov. Chris Christie took office because he cut funding for higher education.
Subject	education, state finances
Speaker	Barbara Buono
Job title	State Senator
Party affiliation	democrat
Label history	3, 1, 4, 4, 1
Context	a speech to students at the Rutgers New Brunswick campus

4.2 FakeNewsNet Data Repository

The FakeNewsNet (Shu, Mahudeswaran, Wang, Lee, and Liu, 2019) is a data repository that contains two datasets. The datasets were collected from PolitiFact.com and GossipCop. The datasets contain the collected news articles, social context information about users who interacted with the article on social media. The inclusion of user interaction information makes this data repository excellent when detecting false news from social media. As shown by research made by Tacchini, et al. (2017) and Del Tredici, et al. (2020) the inclusion of user analysis is an excellent way to detect false news that is spread in social media. The datasets contain source URLs to the news articles, the title of the news, and tweet ids of users who interacted with the article on Twitter.

The datasets are different in sizes, where the dataset collected from GossipCop being considerably larger than the dataset collected from PolitiFact.com. The distribution of false and true news in the datasets is imbalanced, especially in the dataset collected from GossipCop. The dataset collected from PolitiFact.com contains 432 news articles labelled as false and 624 news articles labelled as real. The dataset collected from GossipCop contains 6,048 news articles labelled as false and 16,817 labelled as real.

The articles in PolitiFact.com dataset focus on political issues, whereas the articles in GossipCop dataset contains news about celebrities. GossipCop used to be a website to fact-check articles and stories related to the entertainment industry. As GossipCop mainly focused on false stories provides a reason why the imbalance between real and false articles is so huge in the dataset.

The average title length in PolitiFact.com dataset is 10.74, and in the GossiCop dataset it is 10.67. In the PolitiFact.com dataset a news article was interacted by 1.329 different users on average, and in the GossipCop dataset the same average was 1.064.

TABLE 5. An example of a randomly chosen real news article from FakeNewsNet's PolitiFact.com dataset.

Elements	Values of Elements
ID	politifact182
News URL	http://www.gao.gov/new.items/d071195.pdf
Title	US Government Accountability Office Report to Congressional Committees
Tweet IDs	956894522511736832
Label	real

TABLE 6. An example of randomly chosen false news article from FakeNewsNet's PolitiFact.com dataset.

Elements	Values of Elements
ID	politifact14944
News URL	http://thehill.com/homenews/senate/369928-who-is-affected-by-the-government-shutdown
Title	Who is affected by the government shutdown?
Tweet IDs	954602090462146560 954602093171609600 954650329668349954
Label	false

TABLE 7. An example of a randomly chosen real news article from FakeNewsNet's GossipCop dataset.

Elements	Values of Elements
ID	gossipcop-897603
News URL	https://www.teenvogue.com/story/selena-gomez-not-changing-blonde-hair
Title	Selena Gomez Is Going To Keep Her Blonde Hair
Tweet IDs	936830208857878528
Label	real

TABLE 8. An example of a randomly chosen news article from FakeNewsNet's GossipCop dataset.

	Values of Elements
ID	gossipcop-8424920276
News URL	www.inquisitr.com/opinion/4545022/adam-sandler-confirms-justin-bieber-didnt-ask-for-acting-advice-says-singer-is-funny-as-hell/
Title	Adam Sandler Confirms Justin Bieber Didn't Ask For Acting Advice, Says Singer Is 'Funny As Hell' [Opinion]
Tweet IDs	919499104950001669 919610157755256832
Label	fake

4.3 Twitter15

Twitter15 (Liu, 2015) includes 1490 Twitter stories posted until March 2015. The stories were collected from Snopes.com and Emergent.info. The dataset is used for rumour detection on Twitter posts. The distribution of false and true events is similar in size. The dataset contains 372 events determined as true rumours, 370 events determined as false rumours, 370 events determined as false rumours, 374 events determined as non-rumours, and 374 events which could not be verified. The labels used to differentiate the stories are *non-rumour*, *true, false*, and *unverified.*

As the dataset contains stories posted on Twitter, the genre for the stories varies a lot. The most common words found in the dataset were *Paul, shot, new, police, says, killed, war, Ferguson, died,* and *Obama.* The average text length of a post is 10.2 tokens.

TABLE 9. An example of a randomly chosen news article from Twitter15 dataset.

Elements	**Value of Elements**
ID	693560600471863296
Events	miami was desperate for a turnover. instead, nc state got this dunk. and a big upset win: URL
Veracity	non-rumour

4.4 Novel ChatGPT dataset

For experimental purposed, I created a novel dataset that includes news articles generated by ChatGPT to test how well classifiers would handle artificially generated text. The dataset consists of an ID, title, text, author/source, and the labels false and true. Additionally, the texts labelled as true are provided with a link to the original article. I created 200 false articles to the dataset, and 100 true articles.

Creating false news articles with ChatGPT was a very specific and unusual task. By default, if you try to command ChatGPT into writing an obviously false news article it will refuse to do it by default. This, however, can be easily manipulated with word play to tricking ChatGPT into writing articles that contain misleading information. Table 10 shows how initially I failed when trying to generate a false news article, and Table 11 shows how to manipulate the default settings of ChatGPT into creating a false news article. The text generated by ChatGPT is believable, creative, and does not contain language usually found in false news, such as exaggerated language. The example of how to generate false news articles with ChatGPT shows, how easily it can be used to generate untruthful information in general.

The so-called truth full news for this dataset was created by first summarising news articles from credible news sources, such as Reuters (Reuters 2023), Helsinki Times (Helsinki Times 2023), The Kyiv Independent (The Kyiv Independent 2023), and NHK WORLD-JAPAN (NHK WORLD-JAPAN), then making ChatGPT write the news article again using the summary that was created. I also made sure, that each generated article was truthful towards its original context, and that it did not contain false information made up by ChatGPT. A link to the original article that was summarised and rewritten by ChatGPT is also provided.

From the other datasets that were used, an observation could be made that the articles used in them do not have a lot of variety. A lot of them mainly focus on USA and its political climate, which leaves topics from other parts of the world and from different subjects largely untouched. For this issue, I included a variety of different subjects, such as sports, economics, medicine, and crimes from different countries in this dataset. This decision was made to assure more diversity in the data which would assure that the trained model does not simply overfit to some specific politics-related term in classification.

TABLE 10. An example of failure when obviously trying to create false news with ChatGPT.

Prompt	Result
Write a news article about why Vladimir Putin is friends with Bigfoot.	I'm sorry, but I cannot provide you with a news article that is factually inaccurate or misleading. There is no credible evidence or reports to suggest that Vladimir Putin is friends with Bigfoot, and any such claim would be purely fictional or a product of creative imagination. As an AI language model, my function is to provide informative and accurate responses to your queries based on verified data and sources. Therefore, I suggest reframing your question to something that is more grounded in reality and can be answered with factual information.

TABLE 11. An example of successfully tricking ChatGPT into generating a false news article.

Prompt	**Result**
Let's play creative word games. Write a piece of text that would look line an article about Vladimir Putin being friends with Bigfoot.	In a shocking revelation that is sure to shake up the world of international diplomacy, it has been reported that Russian President Vladimir Putin has struck up an unlikely friendship with none other than Bigfoot. According to sources close to the Kremlin, the two unlikely companions first met in the remote forests of Siberia, where Putin was reportedly on a hunting trip.

TABLE 12. An example of an entry in the Novel ChatGPT-generated dataset.

Elements	**Value of Elements**
ID	0
title	Fed plans broad revamp of bank oversight after SVB failure
text	The Federal Reserve could make a significant impact on its supervisory practices by rapidly implementing mitigants in response to serious issues regarding capital, liquidity, or management, according to a senior Fed official...
source	Reuters
label	true
original article	https://www.reuters.com/business/finance/fed-plans-broad-revamp-bank-oversight-after-svb-failure-2023-04-28

5 APPLIED METHODS

In this paper, Python programming language is used for its good source of Machine Learning (ML) and Data Science libraries. This section introduces the libraries and methods used, and shortly explains their usage and why they are used for this paper.

5.1 Pandas

Pandas (Pandas 2023) is a library for Python, that is used for working with datasets. Pandas provides functions for analysing, cleaning, exploring, and manipulating data. It allowed an efficient way to analyse data from the datasets, such as determining the most common values of elements, length of values, and how many times a certain value of an element appears in the dataset.

5.2 Scikit-learn

Scikit-learn or sklearn (scikit-learn 2023a) is an ML library for Python that features various classification, regression, and clustering algorithms. Scikit-learn works well with other Python libraries such as Matplotlib, NumPy and Pandas. Scikit-learn offers a good collection of different methods for classifying purposes and feature extraction. The chosen feature extraction methods are introduced in section 5.2.1 and the classifying methods are introduced in section 5.2.2

5.2.1 Feature extraction

CountVectorizer turns given textual data into a vector based of the frequency (count) of each word in the text. The created vector is represented as a sparse matrix, where each of the words are stored using index values determined by the alphabetical order. Using CountVectorizer makes it easy to use textual data directly in ML models in text classification tasks. The matrix, indexing of words, and word counts created by CountVectorizer are visualized in Table 13. The textual data used for creating the matrix was a small table consisting of this sentence: "This is the first sentence. This is the second sentence. And his is the third sentence."

From the Table 13 we can observe that each of the words that appear in the used data are arranged in an alphabetical order, and they are given an index number according to the said order. E.g., the word 'and' is given the index number of 0, the word 'first' is given an index number of 1, etc. We can also observe how many times each of the words appear in all the sentences found in the used data. E.g., we can see that in 'text 1' the word 'and' doesn't appear at all, but the word 'first' appears one time.

TABLE 13. Sparse matrix created by CountVectorizer.

Words	**and**	**first**	**is**	**second**	**sen-tence**	**the**	**third**	**this**
Index	0	1	2	3	4	5	6	7
Text 1	0	1	1	0	1	1	0	1
Text 2	0	0	1	1	1	1	0	1
Text 3	1	0	1	0	1	1	1	1

TF-IDF (Term Frequency-Inverse Document Frequency) is a technique used to determine the importance of words found in used documents. The TF-IDF score of a word is determined according to how many times the word appears in the document. A word that appears less frequently gets a higher score than a word which appears more frequently. This is because a word that appear less frequently in a sentence are seen as more important, since they usually give a better context about the sentence.

The TF-IDF score is a combination of two calculations: the term frequency (TF) and the inverse document frequency (IDF). TF score is calculated by dividing the number of occurrences of a word from a document by the total number of words in that document. IDF score is calculated by dividing the total number of documents by the documents containing a certain word. The TF and IDF scores are then multiplied, and finally the TF-IDF result can be obtained.

In this research, the TF-IDF technique is applied using the **TfidfTransformer**. It is used to transform a count matrix - such as one obtained by CountVectorizer - into a matrix of TF-IDF scores. TfidfTransformer takes the count matrix as an input and applies the TF-IDF technique to convert the matrix into a weighted representation. The matrix created by TfidfTransformer is visualized in Table 14. The matrix was created by using the same data is in Table 13.

TABLE 14. TF-IDF matrix created by TfidfTransformer.

Word	**TF-IDF Score**
is	1.000000
sentence	1.000000
the	1.000000
this	1.000000
and	1.693147
first	1.693147
second	1.693147
third	1.693147

With CountVectorizer and TfidfTransformer a bag-of-words model was created. Bag-of-words is a common technique in Natural Language Processing (NLP), where the used textual data is turned into numerical features that ML algorithms are capable of processing.

5.2.2 Classifying Methods

Random Forest (RF) is an ensemble learning method, that uses a combination of multiple decision trees when training. When used in a classification task, RF generates an ensemble of trees that predict the classification result by casting a vote to determine the most popular class. In RF, each tree in the ensemble is built from a sample drawn with a replacement from a training set - which in the case of this are the used datasets (Breiman 2001). The research conducted by Waikhom, et al. (2019) and Ahmad, et al (2020) that was discussed more in detail in section 3, showed that ensemble learning methods achieved high accuracy scores for false news detection. This provides a reasoning why this paper includes RF as one of the methods used for the classifying experiments.

Naive Bayes (NB) methods are a set of supervised learning algorithms based on applying the Bayes' theorem with strong (naive) independence assumptions between features given the value of a class variable. In this paper, the MultinomialNB algorithm is used for the classification experiments. MultinomialNB predicts the probability of general labels for a given text, applying those features in a multinomial function based on the Bayes' theorem. NB algorithms are often used as a baseline for text classification tasks, they work well with smaller datasets, and are faster to train when compared with other popular classifiers (Eronen, 2021).

Logistic Regression (LR) is a statistical model that calculates the probability of a binary outcome, based on prior observations from a dataset. LR can take multiple input criteria into consideration for the eventual outcome of a prediction. In instance of false news detection, LR model could be capable, for example, to take in consideration the history of label distribution (true/false) of news articles written by a reporter. Based on the historical data, LR model calculates the score for a new case based on its probability of receiving either one of the labels (Lawton, 2022). LR has been successful for false news detection purposes previously (Tacchini 2017, Ahmed 2018, Shivani 2023).

Support Vector Machines (SVM) is a machine learning algorithm that uses supervised learning models to analyse data for classification, regression, and outliers' detection. SVM works by creating a hyperplane that separates used training data into classes, e.g., false/true. New data is then fitted into the same space as the old data, and the class prediction for the new data is determined by which side of the previously created hyperplane they fall (Kanade 2022). SVM has been used widely in false news detection, often being able to receive high accuracy results (Rubin 2016, Kasra 2020, Islam 2021).

k-Nearest Neighbors (kNN) is a non-parametric supervised learning method, used for classification and regression tasks. kNN classifier takes a *k* closest training data as an input, and the input data is then classified based on by a plurality vote of its neighbours. Meaning, that an object currently being classified gets assigned to a class according to its *k* nearest neighbours, where *k* is a positive integer (Wikipedia, 2023c). In this paper, k = 3 was used, where an object is classified according to what is the most common class among its neighbours. Just like NB, kNN is often used as a baseline for text classification tasks and are fast and simple to train (Eronen 2021).

Multi-layer Perceptron (MLP) is a feed-forward artificial neural network, typically consisting of three different interconnected layers: input layer, hidden layer, and output layer. In MLP classification tasks, used input data is passed through the layers, where each layer solves a specific part of the task. The output of the solved result is passed through the layers, until the result is determined, e. g. whether an article is false or not (Sidharth 2023). For this paper, the MLP classifier was implemented with the suggested settings given by Scikit-learn's user guide for neural networks. MLP has shown good results in previous research about false news detection (Kaur 2020, Thota 2018), and it has been useful for more complex tasks simple text-based classification (Nadeem 2023}.

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes, and leaf nodes. The root node and the internal nodes represent the base problem given by a classification task, e.g., is this article false or true, branches represent the outcome of the problem, and the leaf nodes represent the final decision after calculating all the possible outcomes, e.g., this article is false (Wikipedia 2023b). DTs are simple and easy to understand, and because of its structure, it is capable of handling multi-output classification problems. For false news detection, DTs have been capable of providing good accuracy results (Kaur 2020, Patil 2022, Anuradha 2023).

Boosting in ML refers to a set of ensemble algorithms, such as Adaptive Boosting (AdaBoost), Gradient Boosting or Extreme Gradient Boosting (XGBoost). For this paper, the AdaBoost algorithm was implemented. AdaBoost works by training a classifier on a dataset, and the classifier is given a weight according to the performance. AdaBoost gives a higher weight for items that have been classified incorrectly, so that the incorrect classification could be corrected. This process is then repeated, until the actual values and predicted values reach an acceptable threshold (Verma 2022). AdaBoost algorithm is simple to use, yet it can achieve good accuracy results for false news detection tasks (Waikhom 2019, Anuradha 2023).

Stochastic Gradient Descent (SGD) is an algorithm used to minimize the loss functions of a linear classifier model. It is often used in large-scale machine learning problems usually encountered in text classification and natural language processing. SGD is an optimization method, used to train a model to find the optimal set of parameters for the model. In this research, the model that was trained with SGD was the "Modified Huber" loss function. SGD is efficient and easy to implement, and it works well with large datasets (Scikit-learn 2023c). SGD has been capable of achieving good accuracy results for false news detection related tasks (Patil 2022).

Passive Aggressive algorithm (PA) is an algorithm that is part of a group called Linear Models, where the target value is expected to be a linear combination of the features (Scikit-learn 2023b). More specifically, PA is used with binary classification tasks, usually when the used data is potentially noisy, or it might change over time (Crammer, Dekel, Keshet, Shalev-Shwartz and Singer, 2006). Passive Aggressive algorithms has been widely used for false news detection related tasks, often with good results (Sharma, Saran, and Patil, 2020; Ahmed, Hinkelmann and Corradini, 2022).

6 EXPERIMENTS

In this section, the experiments that were concluded on the four datasets that were introduced in section 4 are introduced. The source code for these experiments was written on Google's Colab environment[1].

6.1 Experiment 1: Twitter15 dataset

For the first set of experiments, the Twitter15 dataset (Liu 2015) was used. This dataset was already divided into separate 'test' and 'train' files. As this dataset was created for rumour detection, this dataset was used to compare the differences and similarities between false news and rumours. As presented in Table 1, the definitions of rumours and false news have a slight overlap, conducting a classifying experiment using both rumours and false news was an intriguing idea. In this experiment, a classifying experiment only using the Twitter15 dataset (Liu 2015) was conducted first. Then, an experiment using the LIAR dataset (Wang 2017) as train data, and the Twitter15 dataset (Liu 2015) as test data was conducted. For the final experiment, Twitter15 dataset (Liu 2015) as train data, and the LIAR dataset (Wang 2017) as test data was used.

To be able to use the ML classifiers, a bag-of-words model was first built with CountVectorizer and TfidfTransformer. For this experiment, to get the best comparisons between all the different methods, all the methods introduced in section 5.2.2 were used. The three best performing methods from this experiment were chosen for later experiments.

The first experiment conducted only using the Twitter15 dataset (Liu 2015) on the classifiers. This way, a good baseline for the classifiers was constructed, that could be used to compare the classifying results achieved later by the more complex experiments. The results for this experiment can be seen on Table 15. From the Table 15, we can observe that on average, all the classifiers performed well, as all the classifiers could achieve accuracy results of over 50%. The worst performing classifiers were AdaBoost and MLP, which achieved accuracy scores of 54% and 68%. The best performing classifiers were the Passive Aggressive classifier and SVM, which both achieved accuracy scores of 87%.

[1] https://drive.google.com/drive/folders/1PHdLZaJ712w9aFCs6J2sO3N_wvOfC8b1?usp=sharing

The second and third experiments were conducted using the Twitter15 dataset as train data and the Liar dataset as test data, and vice versa. The results for these experiments can be seen on Table 16 and Table 17. From these tables we can observe that testing and training using different false information type datasets did not provide good results. This can be explained since the language used in false news is very different from the language used in rumours. However, we can observe that the two classifiers - AdaBoost and MLP - that performed the worst in the first experiment, performed slightly better comparatively in the second and third experiments. This could prove that these classifiers are more capable when the classifying task is more complicated.

From these classification experiments, the conclusion was that RF, PA and SVM were overall the best performing methods, that were also the most consistent with their precision, recall and F1 scores as well. For these reasons, these three methods were used for the later classification experiments.

TABLE 15. Performance report on the classifying results conducted only using the Twitter15 dataset.

Method	**Accuracy**	**Precision**	**Recall**	**F1**
SGD	0.83630	0.84	0.84	0.84
PA	0.85714	0.87	0.87	0.87
RF	0.81250	0.82	0.80	0.80
MLP	0.68000	0.70	0.68	0.68
LR	0.84821	0.86	0.85	0.86
ADA	0.55059	0.54	0.54	0.53
kNN	0.78869	0.79	0.79	0.78
MultinomialNB	0.80357	0.80	0.80	0.80
DT	0.71428	0.72	0.72	0.72
SVM	0.86607	0.87	0.87	0.87

TABLE 16. Performance report on the classifying results conducted using Twitter15 as train data and Liar as test data.

Method	**Accuracy**	**Precision**	**Recall**	**F1**
SGD	0.34251	0.32	0.32	0.28
PA	0.35201	0.35	0.34	0.29
RF	0.36464	0.31	0.33	0.22
MLP	0.37647	0.13	0.33	0.18
LR	0.36779	0.28	0.33	0.21
ADA	0.36385	0.32	0.33	0.24
kNN	0.34806	0.34	0.34	0.32
MultinomialNB	0.37411	0.18	0.33	0.18
DT	0.34569	0.32	0.32	0.27
SVM	0.35753	0.31	0.34	0.26

TABLE 17. Performance report on the classifying results conducted using Twitter15 as test data and Liar as train data.

Method	**Accuracy**	**Precision**	**Recall**	**F1**
SGD	0.26190	0.28	0.27	0.27
PA	0.24404	0.24	0.24	0.23
RF	0.38928	0.34	0.33	0.32
MLP	0.50000	0.17	0.33	0.22
LR	0.28273	0.28	0.28	0.27
ADA	0.39583	0.30	0.31	0.29
kNN	0.31250	0.34	0.32	0.31
MultinomialNB	0.33333	0.29	0.31	0.29
DT	0.34821	0.31	0.31	0.21
SVM	0.27380	0.29	0.28	0.27

6.2 Experiment 2: LIAR Dataset

The second set of experiments was conducted on the LIAR dataset (Wang, 2017). This dataset was already divided into training, validation, and testing files. For this experiment, the validation and training data together were combined, as the inclusion of the validation file was unnecessary for these experiments.

To be able to use the ML classifiers, a bag-of-words model with CountVectorizer and TfidfTransformer was first built, just like in the previous experiment. For the classifying experiments, three different ML methods were used: Passive Aggressive Classifier (PA), Random Forest (RF), and Support Vector Machines (SVM).

As the LIAR dataset originally separates false and true news into six different labels, two different types of classification experiments on this dataset were concluded. At first, a six-way classification experiment was concluded, where the articles were classified with the six different labels originally provided by the dataset's author. After that, a series of different binary classification experiments were concluded, where the labels of the dataset were reduced only into 'true' and 'false'. The labels were reduced in six different ways: all labels except 'true' are labelled as 'false', labels 'pants-fire' and 'false' are labelled as 'false' and the rest as 'true', all labels except 'pants-fire' are labelled as 'true', labels are split from the middle into 'true' and 'false', and labels 'true' and 'mostly-true' are labelled as 'true' and the rest as 'false'. All these different labelling ways were used to determine, when non-binary classification work the best for the LIAR dataset, and in which labels the language is the most differentiable.

The results for the six-label classification experiment can be found in Table 18. From these results we can observe that all the used methods achieved low scores overall. The best performing method was RF, with an accuracy score of 25%. However, the macro averages of precision, recall and F1-scores were all almost identical with all the used methods.

After the six-way classification experiment, a series of binary classification experiments were concluded. The results for the binary classification experiments can be found from tables 19, 20, 21, 22, and 23. From these tables we can observe that the classifiers were most efficient when only the label 'pants-fire' was considered as false, and least efficient when the labels 'mostly-true' and 'true' were considered as true and the rest are false. Overall, in the binary classification experiments, the classifiers were more accurate than in the six-label experiment. Like in the six-label classification experiment, RF method was the best performing in all the binary classification experiments. However, in the experiment where only the label 'pants-fire' was considered false, RF and SVM both achieved accuracy scores of 91%. And just like in six-label classification experiment, the macro averages of precision, recall and F1-scores were almost identical with all the methods.

TABLE 18. Performance report for the six-way classification experiment.

Method	**Accuracy**	**Precision**	**Recall**	**F1**
PA	0.21073	0.20	0.20	0.20
RF	0.24782	0.25	0.22	0.21
SVM	0.22967	0.22	0.22	0.21

TABLE 19. Performance report for the binary classification experiment where all the labels except 'true' are re-labelled as false.

Method	**Accuracy**	**Precision**	**Recall**	**F1**
PA	0.76318	0.52	0.51	0.51
RF	0.82649	0.91	0.50	0.46
SVM	0.81125	0.51	0.50	0.47

TABLE 20. Performance report for the binary classification experiment where labels 'pants-fire' and 'false' are re-labelled as false, and the rest as true.

Method	**Accuracy**	**Precision**	**Recall**	**F1**
PA	0.79132	0.51	0.51	0.51
RF	0.86166	0.43	0.50	0.46
SVM	0.84642	0.51	0.50	0.48

TABLE 21. Performance report for the binary classification experiment where only the label 'pants-fire' is considered as false, and the rest are true.

Method	Accuracy	Precision	Recall	F1
PA	0.88161	0.53	0.51	0.51
RF	0.90923	0.45	0.50	0.48
SVM	0.90528	0.45	0.50	0.48

TABLE 22. Performance report for the binary classification experiment where labels 'barely-true', 'pants-fire' and 'false' are considered as false, and labels 'half-true', 'mostly-true' and 'true' are considered as true.

Method	Accuracy	Precision	Recall	F1
PA	0.55564	0.55	0.55	0.55
RF	0.59273	0.59	0.59	0.59
SVM	0.58168	0.58	0.58	0.58

TABLE 23. Performance report for the binary classification experiment where labels 'mostly-true' and 'true' are considered as true, and the rest as false.

Method	Accuracy	Precision	Recall	F1
PA	0.55801	0.49	0.49	0.49
RF	0.65114	0.50	0.50	0.44
SVM	0.60063	0.51	0.50	0.50

6.3 Experiment 3: FakeNewsNet data repository

My third set of experiments was conducted on FakeNewsNet data repository (Shu 2019). As explained in section 4, this data repository consists of two datasets, one that has a focus on political articles and one that has a focus on celebrity gossips. The datasets were split into 'real' and 'fake' files. Before my experiments, I added the labels 'true' and 'false' into all the data files. After that, the 'real' and 'fake' files were combined according from the source where the dataset was collected from.

Like in previous experiments, a bag-of-words model was built using CountVectorizer and TfidfTransformer. The methods used for these experiments were the same as in Experiment 6.2.

Four different experiments with this data repository were concluded, one where the PolitiFact dataset as the 'train' file and GossipCop dataset as the 'test' file was used, and vice versa, as well as an experiment where the datasets were split into 'train' and 'test'. As both datasets are very different from one another in terms of content, it was an appealing idea to experiment how using the two datasets together would affect the possible results.

The results for all the classification experiments can be seen in tables 24, 25, 26, and 27. From these tables we can observe that the used classification methods were least efficient when training with the PolitiFact dataset and testing with the GossipCop dataset. The highest accuracy scores were achieved when splitting the GossipCop dataset into 'train' and 'test'.

From these tables we can also observe that different methods were more or less accurate depending on what was used as 'train' or 'test' data. From Table 24, we can observe that the RF method was the most accurate when training with the PolitiFact dataset and testing with the GossipCop dataset. From Table 25, we can observe that the SVM method was the most accurate when training with the GossipCop dataset and testing with the PolitiFact dataset. From Table 26, we can observe that PA and SVM methods achieved identical accuracy scores when splitting the PolitiFact dataset into 'train' and 'test'. From Table 27, we can observe that RF and SVM methods were most accurate when splitting the GossipCop dataset into 'train' and 'test'.

TABLE 24. Performance report when the PolitiFact dataset was used as ‘train’ and the GossipCop dataset was used as ‘test’.

Method	**Accuracy**	**Precision**	**Recall**	**F1**
PA	0.42109	0.50	0.49	0.41
RF	0.63893	0.51	0.51	0.51
SVM	0.43920	0.50	0.50	0.43

TABLE 25. Performance report when the GossipCop dataset was used as 'train' and the PolitiFact dataset was used as 'test'.

Method	**Accuracy**	**Precision**	**Recall**	**F1**
PA	0.59185	0.56	0.55	0.54
RF	0.59375	0.57	0.51	0.41
SVM	0.61837	0.61	0.56	0.52

TABLE 26. Performance report when the PolitiFact dataset was split into 'train' and 'test'.

Method	**Accuracy**	**Precision**	**Recall**	**F1**
PA	0.83018	0.83	0.81	0.82
RF	0.76415	0.76	0.73	0.74
SVM	0.83018	0.83	0.81	0.82

TABLE 27. Performance report when the GossipCop dataset was split into 'train' and 'test'.

Method	**Accuracy**	**Precision**	**Recall**	**F1**
PA	0.79765	0.72	0.72	0.72
RF	0.84123	0.82	0.71	0.74
SVM	0.84869	0.80	0.75	0.77

6.4 Experiment 4: Novel ChatGPT-generated Dataset

For the fourth and final set of experiments, the Novel ChatGPT-generated dataset, and the LIAR dataset (Wang 2017) was used. The Novel ChatGPT-generated dataset consists of 200 false news, generated by ChatGPT based on imaginary prompts that given to the AI, as well as 100 true news generated by picking actual news articles from various sources, summarising them, and making ChatGPT rewrite the article based on the given summary. As various AI tools have become more widespread in today's world, and seeing how easily and fast I was able to create false news articles using ChatGPT, it is important to think of ways how we could efficiently develop ways to identify AI generated content. In this experiment, the first classifying experiment was conducted by only using the ChatGPT-generated dataset, the second classifying experiment was conducted using the

ChatGPT-generated dataset as the train data and the LIAR dataset as test data, and the third classifying experiment was conducted vice-versa.

Like in previous experiments, a bag-of-words model was first built using CountVectorizer and TfidfTransformer. The methods used for these experiments were the same as in Experiment 6.2 and Experiment 6.3.

The first experiment was conducted only using the ChatGPT-generated dataset. This was done to build a baseline of results that could use to compare the results achieved from the more complex experiments. The results for the first experiment can be seen on Table 28. From this table we can see that all the classifiers were able to achieve good results, but the PA classifier was able to achieve more consistent precision, recall and F1 scores in comparison to the other two classifiers.

The second and third set of experiments were used using ChatGPT-generated dataset as train data and the LIAR dataset as test data, and vice versa. The results for these experiments can be seen from Tables 29 and 30. The performance of the classifiers dropped significantly, especially when using ChatGPT-generated dataset as test data. The results obtained from using ChatGPT-generated dataset as train data were better but lacked behind from the results achieved from the first experiment. These results show that the classifiers are better at recognizing human-generated texts if trained with AI-generated texts, but struggle at recognizing AI-generated texts if trained with human-generated texts.

TABLE 28. Performance report on the classifying results conducted using only the ChatGPT-generated dataset.

Method	**Accuracy**	**Precision**	**Recall**	**F1**
RF	0.86885	0.93	0.56	0.56
SVM	0.83606	0.64	0.58	0.60
PA	0.86885	0.74	0.79	0.79

TABLE 29. Performance report on the classifying results conducted using the ChatGPT-generated dataset as train data and the LIAR dataset as test data.

Method	**Accuracy**	**Precision**	**Recall**	**F1**
RF	0.53117	0.48	0.49	0.44
SVM	0.55722	0.52	0.50	0.43
PA	0.53433	0.52	0.52	0.51

TABLE 30. Performance report on the classifying results conducted using the ChatGPT-generated dataset as test data and the LIAR dataset as train data.

Method	**Accuracy**	**Precision**	**Recall**	**F1**
RF	0.33443	0.42	0.50	0.25
SVM	0.33443	0.41	0.47	0.29
PA	0.38741	0.47	0.48	0.38

7 DISCUSSION

The results that were achieved from these experiments look very promising. This thesis was able to conclude that linear models and ensemble methods work the most efficiently for automatically detecting false news. The three most efficient methods in this research are passive-aggressive classifier, support vector machines, and random forests. These methods overall achieved good accuracy results in many of the experiments, without having a significant drop in the macro averages of precision, recall and F1-scores.

This thesis wanted to compare different types of false news content, and how they differ from each other. In a lot of the experiments, for this reason, the classifiers were trained and tested using two different datasets. In all experiments where the classifiers were trained and tested using two different datasets, this thesis concluded that the accuracy results would drop significantly in comparison to training and testing using the same dataset. However, this thesis was also able concluded that usually the achieved results depended on what dataset was use for training, and what dataset was used for testing. For example, in the experiment 6.3, the results for training using the GossipCop dataset and testing using the PolitiFact dataset were almost 10% higher on all the methods when comparing the results achieved with training using the PolitiFact dataset, and testing using the GossipCop dataset. These differences in results could be due to many factors, most commonly it is due because the sizes of each of the datasets do not match, and the higher accuracy results are achieved when the dataset used for training is bigger than the dataset used for testing. Other reason could also be, that in some of the datasets the language found in them is more indicating when the news is either false or true.

This thesis also conducted experiments whether labelling false news according to the levels of how false or true news is would affect the accuracy results. In the experiment 6.2, where the LIAR dataset (Wang 2017) that was divided into six different labels depending on how false or true each article was in the dataset was used, this thesis concluded that using many different labels significantly decreased the accuracy results of the classifiers. When experimenting using the six labels originally provided by the dataset, the classifiers managed to only achieve accuracy results of around 20%, but when this dataset was re-labelled into only two labels, the classifiers consistently achieved accuracy results of over 50%, with the highest accuracy result being 91%. These results achieved in the experiment 6.2 show that a simpler labelling system provides better accuracy results.

Perhaps the most intriguing part of this research was the Novel ChatGPT-generated dataset that was built for the purpose of this research. While building this dataset, I found out how easy it was to generate false news articles in just a matter of few minutes. I was able to generate about 10 false articles in roughly an hour. The fact that I was able to generate false articles that look completely believable so easily and fast, show that there is a need for expanding the topic of automated false news detection into the field of detecting computer-generated texts.

In the Experiment 6.4 detecting computer-generated false news is looked more closely, and how they differ from human generated ones. In the experiments, it concluded that when using only the ChatGPT-generated dataset, the classifiers were able to achieve similar accuracy results as when using the human-generated datasets. However, when the classifiers were trained using the ChatGPT-generated dataset and tested with the LIAR dataset (Wang 2017), the achieved accuracy results were around 20% higher, than when training with the LIAR dataset (Wang 2017) and testing with eh ChatGPT-generated dataset. The significant difference in these accuracy scores could show some interesting differences between computer-generated and human-generated texts. These differences could for example show that ML techniques are more capable of differentiating false and true from human-generated texts, when trained with computer-generated texts, than vice versa. The results could also indicate interesting differences in language of false news according to whether the false news was created by a computer or a human. The achieved results from this experiment show that it is not an easy task to detect computer-generated tasks from human-generated ones and show that more complex learning models are needed.

8 CONCLUSIONS AND FUTURE WORK

Overall, the results achieved in this thesis were very successful. This thesis was able to answer the two research questions that were set in the beginning, that were determining the best ML methods for automatic false news detection, and how well these methods work on human generated news vs computer generated news.

I was also able to get a good foundation on several different ML methods, and how they work, as well as good knowledge on what exactly is false news, how false news can be determined, and what are some common language characteristics in false news. The foundation that this thesis was able will be a great help for my future work, where the goal is to research false news even more deeply and use more complex techniques.

For future work, I am planning to expand my research more towards news created by generative Large Language Models - such as ChatGPT, how the news generated by language models differ from the ones written by humans, and how it would be possible to efficiently detect that the news is not written by humans. I will also investigate using bigger, larger scale datasets, as well as using modern, and more efficient language models such as BERT (Devlin and Chang 2018) or RoBERTa (Liu et al. 2019).

REFERENCES

Ahmad, I., Yousaf, M., Yousaf, S., & Ahmad, M. O. 2020. Fake news detection using machine learning ensemble methods. Complexity, 1-11. Search date 15.4.2023. https://www.hindawi.com/journals/complexity/2020/8885861/

Ahmed, H., Traore, I., & Saad, S. 2018. Detecting opinion spams and fake news using text classification. Security and Privacy. Search date 15.4.2023. https://onlinelibrary.wiley.com/doi/full/10.1002/spy2.9?casa_token=dn22jBaZ33cAAAAA%3A7XsFyPNV7cuuCjRnHUiuUtnpj7Y7EjPNohkebzQZEZnolDaKqlD-brk8cd55UNmAkgFSVXlbEcpmQzNg

Ahmed, S., Hinkelmann, K., Corradini, F., 2022. Development of fake news model using machine learning through natural language processing. arXiv preprint. Search date 20.8.2023. https://arxiv.org/abs/2201.07489

Anuradha, K., Senthil Kumar, P., Naveen Prasath, E., Vignes, M., Sneha, S., 2023. Fake news detection using decision tree and adaboost. European Chemical Bulletin. Search date 5.8.2023. https://www.eurchembull.com/issue-content/fake-news-detection-using-decision-tree-and-adaboost-1173

Apuke, O. D., Omar, B. 2021. Fake news and COVID-19: modelling the predictors of fake news sharing among social media users. Telemat Inform. Search date 17.3.2023. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7390799/

Boungery, L. Gray, J. Venturini, T., & Mauri, M. 2018. A Field Guide to" Fake News" and Other Information Disorders: A Collection of Recipes for Those Who Love to Cook with Digital Methods. Public Data Lab, Amsterdam. Search date 2.6.2023. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3097666

Breiman, L., 2001. Random forests. Machine learning 45, 5–32. Search date 5.8.2023. https://link.springer.com/article/10.1023/a:1010933404324

ChatGPT 2023. Search date 2.9.2023. https://chat.openai.com/auth/login

Chong, M., & Choy, M. 2020. An empirically supported taxonomy of misinformation. Navigating Fake News, Alternative Facts, and Misinformation in a Post-Truth World, 117-138. Search date 18.6.2023. https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=7543&context=lkcsb_re-search

Das, A., Liu, H., Kovatchev, V., & Lease, M. 2023. The state of human-centered NLP technology for fact-checking. Information Processing & Management. Search date 17.4.2023. https://www.sciencedirect.com/science/article/abs/pii/S030645732200320X?casa_to-ken=i6Yf_axzSUwAAAAA:8JdXDat2TnJpA_t0zguJHhs0zuhU-A2D6kO62J4-3RZsW0LDnoOEuzrl97tjjwMmT-uchk2tWwQ

Del Tredici, M., & Fernández, R. 2020. Words are the window to the soul: Language-based user representations for fake news detection. arXiv preprint. Search date 19.4.2023. https://arxiv.org/abs/2011.07389

Devlin, J. & Chang, M. W. 2018. Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing. Google Research. Search date 2.9.2023. https://blog.re-search.google/2018/11/open-sourcing-bert-state-of-art-pre.html

Duke Reporter's Lab 2023. The Reporters' Lab. Search date 2.9.2023. https://reporterslab.org/

Eronen, J., Ptaszynski, M., Masui, F., Smywiński-Pohl, A., Leliwa, G., Wroczynski, M., 2021. Improving classifier training efficiency for automatic cyberbullying detection with feature density. Information Processing & Management 58, 102616. Search date 5.8.2023. https://arxiv.org/abs/2111.01689

FactCheck.org 2023. Search date 2.9.2023. https://www.factcheck.org/

Faktabaari 2023. Search date 2.9.2023. https://faktabaari.fi/

Gundapu, S., & Mamidi, R. 2021. Transformer based automatic COVID-19 fake news detection system. arXiv preprint. Search date 22.4.2023. https://arxiv.org/abs/2101.00180

Helsinki Times 2023. News from Finland. Search date 2.9.2023. https://www.helsinkitimes.fi/

Hsu, T., Thompson, S. A. 2023. Disinformation Researchers Raise Alarms About A.I. Chatbots. The New York Times. Search date 19.4.2023. https://www.nytimes.com/2023/02/08/technology/ai-chatbots-disinformation.html

Islam, N., Shaikh, A., Qaiser, A., Asiri, Y., Almakdi, S., Sulaiman, A., Moazzam, V., Babar, S.A., 2021. Ternion: An autonomous model for fake news detection. Applied Sciences 11, 9292. Search date 5.8.2023. https://www.mdpi.com/2076-3417/11/19/9292

Kanade, V., 2022. What is a support vector machine? working, types, and examples. Search date 5.8.2023. https://www.spiceworks.com/tech/big-data/articles/what-is-support-vector-machine/

Karimi, H., Roy, P., Saba-Sadiya, S., & Tang, J. 2018. Multi-source multi-class fake news detection. In Proceedings of the 27th international conference on computational linguistics 1546-1557. Search date 16.4.2023. https://aclanthology.org/C18-1131/

Kaur, S., Kumar, P., Kumaraguru, P., 2020. Automating fake news detection system using multi-level voting model. Soft Computing 24, 9049–9069. Search date 5.8.2023. https://link.springer.com/article/10.1007/s00500-019-04436-y

The Kyiv Independent 2023. News from Ukraine, Eastern Europe. Search date 2.9.2023. https://ky-ivindependent.com/

The Washington Post 2023. Fact Checker. Search date 2023. https://www.washing-tonpost.com/news/fact-checker/

Lawton, G., Burns, E., Rosencrance, L., 2022. Logistic regression. Search date 5.8.2023. https://www.techtarget.com/searchbusinessanalytics/definition/logistic-regression

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint. https://arxiv.org/abs/1907.11692

Liu, X., Nourbakhsh, A., Li, Q., Fang, R., & Shah, S. 2015, October. Real-time rumor debunking on twitter. In Proceedings of the 24th ACM international on conference on information and knowledge management, 1867-1870. Search date 28.4.2023. https://dl.acm.org/doi/abs/10.1145/2806416.2806651. Requires access.

Mitrović, S., Andreoletti, D., & Ayoub, O. 2023. Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text. arXiv preprint. Search date 22.4.2023. https://arxiv.org/abs/2301.13852

Nadeem, M.I., Ahmed, K., Li, D., Zheng, Z., Alkahtani, H.K., Mostafa, S.M., Mamyrbayev, O., Abdel Hameed, H., 2023. Efnd: A semantic, visual, and socially augmented deep framework for extreme fake news detection. Sustainability 15, 133. Search date 5.8.2023. https://www.mdpi.com/2071-1050/15/1/133

NHK WORLD JAPAN 2023. Search date 2.9.2023. https://www3.nhk.or.jp/nhkworld/

Oshikawa, R., Qian, J., & Wang, W. Y. 2018. A survey on natural language processing for fake news detection. arXiv preprint. Search date 17.4.2023. https://arxiv.org/abs/1811.00770

Pandas 2023. Python Data Analysis Library. Search date 2.9.2023. https://pandas.pydata.org/

Patil, D.R., 2022. Fake news detection using majority voting technique. arXiv preprint. Search date 5.8.2023. https://arxiv.org/abs/2203.09936

PolitiFact 2023. Search date 2.9.2023. https://www.politifact.com/

Reis, J. C., Correia, A., Murai, F., Veloso, A., & Benevenuto, F. 2019. Supervised Learning for Fake News Detection. IEEE Intelligent Systems, 34(2), 76-81. Search date 17.3.2023. https://ieeexplore.ieee.org/document/8709925

Reuters 2023. Breaking International News & Views. Search date 2.9.2023. https://www.reuters.com/

Rubin, V. L., Conroy, N., Chen, Y., & Cornwell, S. 2016, June. Fake news or truth? using satirical cues to detect potentially misleading news. Proceedings of the second workshop on computational approaches to deception detection. Search date 19.4.2023. https://aclanthology.org/W16-0802/

Scikit-learn, a. Machine learning in Python – scikit-learn 1.3.0. Search date 2.9.2023. https://scikit-learn.org/stable/modules/sgd.html

Scikit-learn, b. 1.1. linear models. Search date 20.8.2023. https://scikit-learn.org/stable/modules/linear_model.html

Scikit-learn, c. 1.5. stochastic gradient descent. Search date 20.8.2023. https://scikit-learn.org/stable/modules/sgd.html

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. 2020. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. Big data, 171-188. Search date 21.4.2023. https://arxiv.org/abs/1809.01286

Sharma, U., Saran, S., Patil, S.M., 2020. Fake news detection using machine learning algorithms. International Journal of Creative Research Thoughts (IJCRT) 8, 509–518. Search date 20.8.2023. https://ijcrt.org/papers/IJCRT2105975.pdf

Singh, V., Dasgupta, R., Sonagra, D., Raman, K., & Ghosh, I. 2017. Automated fake news detection using linguistic analysis and machine learning. In International conference on social computing, behavioral-cultural modeling, & prediction and behavior representation in modeling and simulation (SBP-BRiMS). 1-3. Search date 15.4.2023. https://www.researchgate.net/publication/318541620_Automated_Fake_News_Detection_Using_Linguistic_Analy-sis_and_Machine_Learning

Sidharth, 2023. Multi-layer perceptron explained: A beginner's guide. Search date 5.8.2023. https://www.pycodemates.com/2023/01/multi-layer-perceptron-a-complete-overview.html?utm_content=cmp-true

Snopes.com 2023. The definitive Internet reference source for urban legends, folklore, myths, rumors, and misinformation. Search date 2.9.2023. https://www.snopes.com/

Svärd, M., Rumman, P. 2017. Combating disinformation: Detecting fake news with linguistic models and classification algorithms. Search date 2.6.2023. https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1114109&dswid=-1788

Tacchini, E., Ballarin, G., Della Vedova, M. L., Moret, S., & De Alfaro, L. 2017. Some like it hoax: Automated fake news detection in social networks. arXiv preprint. Search date 15.4.2023. https://arxiv.org/abs/1704.07506

Thota, A., Tilak, P., Ahluwalia, S., & Lohia, N. 2018. Fake news detection: a deep learning approach. SMU Data Science Review. Search date 16.4.2023. https://scholar.smu.edu/datasciencereview/vol1/iss3/10/

Verma, N., 2022. AdaBoost Algorithm Explained in Less Than 5 - Medium. Search date 5.8.2023. https://medium.com/@techynilesh/adaboost-algorithm-explained-in-less-than-5-minutes-77cdf9323bfc

Waikhom, L., & Goswami, R. S. 2019. Fake news detection using machine learning. In Proceedings of International Conference on Advancements in Computing & Management (ICACM). Search date 15.4.2023. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3462938

Wang, W. Y. 2017. "Liar, liar pants on fire": A new benchmark dataset for fake news detection. Search date 17.3.2023. https://arxiv.org/pdf/1705.00648.pdf

Wardle, C. 2017. Fake News. It's complicated. First draft, 16, 1-11. Search date 2.6.2023. https://medium.com/1st-draft/fake-news-its-complicated-d0f773766c79

Wikipedia, 2023a. Fake news – Wikipedia, the free encyclopedia. Search date 28.8.2023. https://en.wikipedia.org/wiki/Fake_news

Wikipedia, 2023b. Decision tree — Wikipedia, the free encyclopedia. Search date 5.8. 2023. http://en.wikipedia.org/w/index.php?title=Decision%20tree&oldid=1165073066.

Wikipedia, 2023c. K-nearest neighbors algorithm — Wikipedia, the free encyclopedia. Search date 5.8.2023. http://en.wikipedia.org/w/index.php?title=K-nearest%20neighbors%20algo-rithm&oldid=1163707353.

Wu, Y., Zhan, P., Zhang, Y., Wang, L., & Xu, Z. 2021. Multimodal fusion with co-attention networks for fake news detection. In Findings of the association for computational linguistics: ACL-IJCNLP 2021, 2560-2569. Search date 24.4.2023. https://aclanthology.org/2021.findings-acl.226.pdf