INTERNATIONAL
HELLENIC
UNIVERSITY

Fake News Detection

Chouliara Vasiliki

SID: 3308200006

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

JANUARY 2023

THESSALONIKI – GREECE

Fake News Detection

Chouliara Vasiliki

SID: 3308200006

Supervisor:	Assoc. Prof. Christos Tjortjis
Supervising Committee Members:	Dr Christos Berberidis
	Dr Paraskevas Koukaras

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

JANUARY 2023

THESSALONIKI – GREECE

Abstract

Easy and quick information diffusion on the web and especially in social media (i.e., Facebook, Twitter, etc.) has been rapidly proliferating during the past decades. As information is posted without any kind of verification of its veracity, fake news has become a problem of great influence in our information driven society. With the current rate of news generated in social media, the differentiation between real and fake news has become challenging. Thus, to mitigate the consequences of fake news and its propagation, considerable research has been conducted both by the academia and the industry, to create automated approaches to detect malicious content. A plethora of approaches have been investigated, most of which identify patterns on fake news after they are already disseminated. The need for early detection methods is crucial.

The goal of this thesis is to review the current approaches for detecting disinformation and propose an effective framework that utilizes only the text features of the news, without using any other related metadata. Several Machine Learning models and Natural Language Processing techniques have been used during experimentation. The findings reveal that a combination of linguistic features and text-based word vector representations through ensemble methods can predict fake news with high accuracy.

Keywords: Fake news, Text classification, Linguistic features, word embeddings, Machine Learning (ML), Ensemble Machine Learning (EML), Deep Learning (DL)

Acknowledgments

I would like to thank my supervisor Prof. Christos Tjortjis whose guidance and encouragement has been invaluable throughout the duration of my dissertation. The meetings and conversations were vital in inspiring me to think outside the box, from multiple perspectives in order to form a comprehensive study. Finally, I would like to express my most profound gratitude to my family for their unwavering support and motivation throughout my years of study and through the process of writing this thesis.

Chouliara Vasiliki
27/01/2023

Contents

1 Introduction

Social media have become the main source of information transmission during the past years, as most people have access to news via online channels avoiding the interaction with traditional sources of information. Social networks provide easy access and circulation of information published, but there is no verification of the news that propagate the network. Thus, social media offer a breeding ground for developing and spreading malicious content. It is apparent that people are exposed to a huge amount of fake news daily while the quality of news in general becomes questionable [1].

Fake news is unreliable content that intends to deliberately manipulate public opinion on different subjects, especially political affairs. It is of great concern though, that malevolent content seems to spread faster, compared to real news, and has greater impact on aspects of life, politics, and economy [2]. A more recent example of the huge influence of disinformation in our everyday lives was during the COVID-19 outbreak, which made the work of health professionals more difficult, while confusing the public and placing millions of individuals at risk [3]–[5].

Due to the immense impact that fake news has, of critical interest is the ability to detect and distinguish fake from real news in real time. This is challenging for many reasons. First of all, through social media, content is generated and spread fast, which leads to huge amounts of information that need to be validated. Moreover, content is diverse, referring to many different subjects, which makes the task even more complex [6]. It is an established fact that people lack the ability to effectively discern fake content. Studies in social psychology and communications have manifested that human's ability to differentiate fake from real topics is slightly better than chance [7].

On the other hand, fact-checking organizations that try to combat the proliferation of fake news, have limited applicability due to time latency and the quick propagation of information in social media. In recent years, extensive research on establishing an automated framework for online fake news detection has been made, to counter the misinformation diffusion [6]. Many machine learning models have been proposed in association with text-relevant features (i.e., TF-IDF), text-based linguistic features, visual and social features to automatically detect malicious content adequately and deal with the volume, variety and velocity of fake news [6].

In this thesis, we approach the problem of fake news by employing Machine learning (ML) and Deep learning (DL) algorithms utilizing only textual data (content-based approach). Additionally, we test the assumption that word vector representations enhanced with linguistic features improve the prediction ability of the models, and finally we succeeded in verifying this theory.

The following is an outline of the remaining research. Chapter 2 provides information from the literature about fake news definition, related areas and detection methods, followed by a discussion of the impact of concept drift on the fake news research area. Chapter 3 includes an overview of the related work that has been done in the field considering both machine learning and deep learning approaches. Chapter 4 presents in detail the research methodology, describing the dataset, data preprocessing, feature extraction techniques as well as the classification models utilized to provide a solution to the problem. Chapter 5 focuses on analyzing the experiments conducted and evaluating the performance of each experiment. Chapter 6 discusses the findings of this thesis. Chapter 7 summarizes the whole process and findings and sets some future work suggestions.

2 Literature Review

This chapter provides a review of the research literature on fake news detection. Relative papers were investigated to enrich this part.

2.1 Fake News Definition

Fake news is now considered to be an important menace to democracy and journalism. Even though the term has a long legacy, it suddenly gained immense popularity during the 2016 U.S. presidential campaign [7]. However, there has been no universal definition of the term "fake news" up to this point. A widely adopted definition of fake news as introduced by Shu et al. [8], mentions that "Fake news is a news article that is intentionally and verifiably false". On top of that definition, Allcott et al. [9] added that this type of news has an intention of misleading people. Another formal definition was stated by Golbeck et al. [10], who claim that fake news is "information presented as a news story that is factually incorrect and designed to deceive the consumer into believing it is true". Later, Sharma et al. [11] captured the broader scope of the term and proposed a new definition as "a news article or message published and propagated through media, carrying false information regardless of the means and motives behind it".

Existing studies often relate fake news with other types of news like deceptive news, satire news, rumors etc. The characteristics that differentiate these concepts are: authenticity and intention [7]. Fake news includes unverified information and is created with an intention to manipulate and deceive the public. Likewise, it is apparent that fake news tries to imitate the format and writing style of real news, so as to amplify the level of its veracity [12]. Ultimately, fake news can have real repercussions in many aspects of life, which makes it an important subject of study.

2.2 Fake News Related Areas

A plethora of scientific studies on the field of Fake News identifies a variety of false information types. However, there is no commonly agreed typology framework or specific categorization criteria on this matter. Rashkin, H., et al., in their research, use two dimensions to classify fake news, the quality of the article and the intention of the writer to deceive [13]. This approach is also adopted by many other researchers. It is important to mention that not all types of fake news have the same level of deceptiveness and intention to harm. In the next section, some of the most common fake news categories are presented.

2.2.1 Propaganda

The term propaganda is used to describe the premeditated attempt to manipulate and influence public perceptions. This endeavor to affect the common belief, is achieved through the activation of strong emotions of the targeted audience, the breeding of fear and the projection of simplified ideas [14]. Propaganda is mainly utilized by political entities to mislead people and impose damage to a particular political party [15]. In recent years, a new term has been widely used, the computational propaganda. The term refers to the usage of political bots to propagate specific opinions through major social networking applications and manipulate conversations. Taking into consideration all of these, propaganda must be detected promptly, as it can result in significant alterations in the course of history. Some examples of propaganda are, the 2016 U.S. presidential election, stories about the Brexit referendum (2020) and the most recent Russian propaganda in the Russian-Ukrainian war (2022).

2.2.2 Conspiracy Theories

Conspiracy theories refer to stories that try to interpret an event or situation that invokes a conspiracy. These stories are based on insufficient or false evidence, though they appear to be relatively widespread among citizens, as they can offer a coherent explanation of the given situation. Studies relate belief in conspiracy theories with low self-esteem, distrust in authority and political cynicism [16]. Conspiracy theories have a wide range of topics, from science, health and economy to politics. Throughout the years, many such theories have been developed that created confusion to the public, with the most recent to be the COVID-19 conspiracy theories. The pandemic proved to be fertile ground for a dozen of theories to be developed, including the origins of COVID-19, its association with the 5G mobile networks and the use of the virus as a population control [17].

2.2.3 Rumors

Many definitions have been used throughout the years to describe the term rumor. The prevailing one was given by Allport and Postman (1947), according to whom, a rumor is a story or a statement whose truth value is unverified [18]. This does not always mean that a rumor is a false piece of information, but rather that its veracity is unconfirmed at the time of posting. Many studies have identified two types of rumors, long-lasting rumors and breaking news [19]. Long-lasting rumors may circulate for long periods of time and thus, training data can be gathered and studied to classify ongoing discussions. The same does not apply for the breaking news rumors, as they last for short periods of time and may include unseen cases which require real-time processing.

The importance and the interest of a topic is highly correlated with the spread of the rumor. Sensitive topics can be a fertile ground for rumors to be generated and spread across social media networks. The proliferation of rumors can result in major chaos and have a negative impact on society [18]. For this reason, a great percentage of the research aims to develop systems to detect rumors using supervised, unsupervised and hybrid methods.

2.2.4 Click bait

Clickbait describes the intentional use of gaudy headlines in articles in a way that attracts the reader’s attention. Sort messages are displayed, often misleading and inaccurate, that lead to the spread of fake news across social media platforms. Clickbait is a kind of web content advertisement that aims at luring the user to click the link of the article and redirect to a specific website. In this way, the owner of the link, increases the popularity and profits, by proliferating the traffic in the link. Although this approach seems to be an effective marketing strategy, it also tends to become a means of manipulating the crowds [20]. Due to this, and the rapid propagation of this kind of stories through social media, clickbait should not be ignored as it can result in huge consequences in the journalistic integrity and the public good [21].

2.2.5 News Satire

Satire is a form of fake news which employs exaggeration and humor to present a story. Satire news is produced to entertain the audience and not to deliberately spread misinformation. However, this type of news is perceived as credible from many people who usually don’t read beyond the headlines. Another reason lies in the fact that when a story is shared in social media, readers tend to associate the number of likes and comments of the post with the veracity of the story. All these factors contribute to classify satire in the fake news category that deceives people.

This statement is also verified in Horne & Adali (2017) research where they examined correlations among satire, fake and real news and they concluded that satire news is more closely related to fake than real ones. They found that the majority of the features distributions they examined are common between satire and fake news [22]. The detection of satire is deemed to be a difficult problem, and it has been addressed in only a few studies. Thus, further research should be conducted in this direction.

2.2.6 Hoaxes

Hoaxes are stories that contain inaccurate information, yet presented as genuine. The research community identifies hoaxes also as half-truth or factoid stories [15]. Tacchini, E., et al. (2017) in their study, retrieved Facebook posts and tried to classify them as hoaxes and non-hoaxes on the basis of the users who "liked" them. They concluded with an accuracy above 99% that the user's interaction with the posts is a good indicator of identifying hoaxes in the news [23].

Kumar, S., et al. (2016) identify in their study, two types of hoaxes, those detected promptly and a small number of hoaxes that is well cited across the Web. They also try to quantify the impact of the hoaxes in the Web based on how long they survive, the traffic they receive and how often they are cited by others. Another interesting finding in this study is that human accuracy in detecting hoaxes (63%) is surprisingly weak compared to the automated detection tool which reached a performance of 98% [24].

2.2.7 Biased/ one-sided

This type of fake news refers to stories that are one-sided or biased against a person, a political party or a situation [15]. It is considered also as a form of political misinformation and is known as hyper partisan news. These stories report true news, where certain facts are selectively highlighted or omitted to serve a purpose. They cannot be described as totally false news, but they are definitely misleading for the public.

Potthast, M., et al. (2018) contributed with their survey to the detection of hyper partisan news. Specifically, they tried to find relations between the writing styles of articles with opposing orientations and those that are neutral and they concluded that the writing styles of the left-wing and right-wing appear to have a common style of extremism. Further research should be conducted to this direction, as hyper partisan news sites penetrate more and more into the media landscape [25].

2.2.8 News Fabrication

Fabricated news refers to articles that have no factual basis and are published in a legitimate style of news content. The intention of this type of news, is to create a false impression in the reader’s mind and deceive people. The verification of the fabricated news is deemed to be difficult, as they are published through many unverified sources, websites, blogs or social media platforms such as Twitter and Facebook, rather than legitimate news sites. Due to this, a fabricated story can circulate in these media and gain popularity leading into deceiving many more of its recipients [12].

A representative example of the consequences of a fabricated story is the false news reported back in 2008 referring to the death of Steve Jobs due to a heart attack. Despite the fact that this information was rebutted promptly, it caused great confusion among the public which resulted in a rapid fluctuation of his company’s stock on that day [26].

2.3 Fake News Detection Approaches

According to many researchers the methods used for fake news detection can be classified in three categories: knowledge-based, content-based and social context-based. Most of the recent detection approaches, extract content-based features to address the problem of fake news. On the other side, social context-based approaches are fewer, due to the limited published datasets that include information from social media platforms. Furthermore, there are many cases where authors use a combination of fake news detection approaches and achieve great results. An advanced approach to detect fake news that combines text and image features, is the deep neural network called TI-CNN proposed by Yang, Y. et al [27].

2.3.1 Knowledge - based

When it comes to the knowledge-based approach, one uses a procedure called fact-checking. The scope of this approach is to compare the knowledge extracted from news articles with known facts to determine its veracity. Existing fact-checking approaches can be classified as expert-oriented, crowdsourcing-oriented and automatic [7].

- **Expert-oriented** is a manual fact-checking process which relies on domain expert fact checkers. This small group of people is accountable for identifying the credibility of an article based on research or other previous annotated texts. Although this process achieves high quality results, it is costly and time-consuming. Popular fact-checking websites are Politifact[1], Snopes[2] and FactCheck[3] [7].

- **Crowdsourcing-oriented** fact-checking exploits the "wisdom of the crowd" to manually annotate news articles [8]. CREDBANK, a publicly available fake news dataset was constructed using this approach. Unlike expert-oriented fact checking, the number of crowdsourcing websites is quite small. A characteristic example is the Fiskkit[4] page, which allows users to annotate specific parts of news articles. The weakness of this algorithm though, is that the annotations are less credible due to the political bias of the fact-checkers [7].

- **Automatic fact-checking** techniques have been developed to deal with the rapid increase of the news' volume and provide scalability. This method is based on Natural Language Processing (NLP) and ML techniques as well as network theory. Automatic fact-checking can be segmented into two stages, fact extraction and fact verification. During the first stage, information is extracted either from single reliable sources like Wikipedia, or from open sources where information provided by multiple sources is consolidated. This information is used to construct a Knowledge Graph where entities are represented as nodes and relations between them are represented as edges. Finally, the last stage of the process requires comparison between the knowledge extracted and the facts [7].

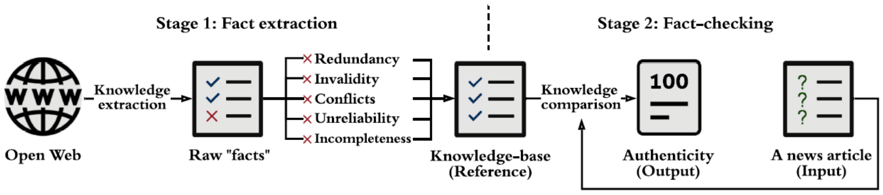

Figure 1: Automatic News Fact-checking process

Throughout the years, many ML models have been built to cope with the classification of fake news. However, misleading news with different styles and features is being spread

[1] https://www.politifact.com/

[2] https://www.snopes.com/

[3] https://www.factcheck.org/

[4] https://fiskkit.com/

among social media every day. All this new type of information which needs to be classified, can be problematic for ML algorithms, that have to unveil previously unseen patterns from the data. Knowledge-based analysis can become a feasible solution to this case. The collection and annotation of huge amounts of data by experts, can reinforce the performance of the algorithms [6].

2.3.2 Content – based

Content-based analysis plays an important role in the fake news identification problem, as it tries to capture the different writing styles between legitimate users and deceivers. Though the deceivers make great efforts to present a story as credible, the style of their language, can often expose them [28]. A benefit of the content-based detection is that it relies on news content, which gives the opportunity to the user to recognize the fake content before this propagates on social media. The goal of this method is to exploit linguistic features from different levels of intricacy of the text, such as words, sentences and documents and thus distinguish fake from real content [8]. Apart from the linguistics, many researchers extract information from visual content such as images to further enrich the feature set.

The **linguistic and syntactic** characteristics which can be extracted from the headline and the text message of an article, can be categorized as follows.

- **Stylistic features**. The stylistic features try to measure the news creator's language proficiency by examining the syntax, text style and grammar of the article. The most common way of investigating the syntactic structure of an article (raw text) is using "Bag-of-words" and "n-grams". These approaches represent a text as a set of words but they don't take into consideration the grammar or the word order. To overcome this problem, new approaches have been proposed, such as word2vec, long short-term memory (LSTM) neural network and so on [6]. Other ways of testing the differences in syntax is by utilizing part of speech tagging (POS) and count the number of tags that appear in the article. Furthermore, counting the frequency of function words, punctuation, and sensory words can further constitute clues for assessing the style of the article [22].

- **Complexity features**. The complexity features are studied in two levels of elaborateness: word and sentence level. On a sentence level, the number of words per sentence, the noun phrase and verb phrase lengths are examined. Moreover, an identifier of complexity can be the number of clauses per sentence, which can be extracted using the Stanford Parser[5] software [29]. The more frequent the usage of longer phrases, the more complicated the structure of the sentence. Regarding the word level complexity, one can count the total number of words or the syllables per word. Another important way to capture the word complexity is by the use of readability indexes, which measure if a reader of a certain level of literacy could understand the text. Gunning Fog and Flesch-Kincaid grade level index are some examples [22]. Previous studies have explored a range of readability indexes and computed their correlation with fake news. Pérez-Rosas et al, extracted 26 readability features and achieved a high score in classification of fake news [30].

- **Psychological features**. These features try to captivate the emotions and behaviors of the creators of the news. They are based on well-researched word counts that are associated with psychological processes and sentiment analysis [22]. Many online tools exist to help researchers quantify the positive or negative emotions of a document. The number of certainty and affective words can also be utilized to amplify the correlation with fake news.

Visual based approaches depend on the visual characteristics extracted by images accompanying the news text. Currently, not many studies exist that detect fake news through image exploration. A novel approach for Microblogs news verification, that uses visual information, was contacted by Jin, Z. et al (2016). In their research, the authors state the impact of images in news diffusion in microblogs and propose a method that uses visual and statistical features to identify fake news. The researchers studied images and extracted five image distribution characteristics such as clarity score, visual coherence score and others, and seven statistical features like image ratio, multi-image ratio and hot-image ratio. Their research achieved an accuracy of 83.6% [31].

[5] https://nlp.stanford.edu/software/lex-parser.shtml

2.3.3 Social Context – based

Every day, millions of online news are published in social media and there is a great need to distill trustable information from these sources. Social context analysis is the study that analyzes the characteristics of information diffusion in social media and tries to identify anomalous information. These characteristics can be categorized in three dimensions as follows.

- **User-based**. Investigating user profiles in social networks is very crucial for the detection of fake news. It is known that a plethora of fake profiles exist in social media platforms, some examples of which are social bots and cyborgs. The intention of these accounts is to disseminate fake news and mislead the public perception. Thus, it is really important to capture those characteristics that differentiate fake from real user profiles. The registration age, the number of followers/ followees and the number of posts that the user has shared can be used to measure the reliability of each user [32]. Sahoo, S. R. et al, in their research combine news and user content features along with deep learning (DL) algorithms and achieve a 99.4% accuracy in detecting fake news in real time [33].

- **Post-based**. People tend to express their viewpoints and emotions through social media posts. Thus, post-based features can be extracted to identify spurious news content. Word embedding approaches and linguistic-based features from the text of the post can be utilized to infer the validity of news articles and obtain user stance. Moreover, many researchers use Latent Dirichlet Allocation (LDA), a common technique used to retrieve the relevant topics from a post. The combination of stance and topic features extracted from posts, along with supervised or unsupervised methods, is used to capture the fake content in social media platforms [8].

- **Propagation-based**. Propagation-based fake news utilizes context information such as, how fake news propagates in social media platforms, which users spread this information, etc., to detect fake news. All this information is represented using different types of graphs. Jin, Z. et al, in their study, built a stance graph network, where the nodes represent user posts related to the news and the edges are the weights of similarity between two articles to track down fake news [34]. Another type of network, is the friendship network which provides the structure to understand the set of relationships among users [8].

2.4 The impact of Concept Drift

A quite important aspect that needs to be addressed in this study, is the impact of concept drift. When data interpretation changes over time, this leads to an impact on performance of previously trained models [35]. This notion is referred to as *concept drift* in ML literature. There are two types of drift in the news. The natural drift can be caused because news differentiates radically over time and there is a different prioritization in what media choose to broadcast at each given time. On the other side, the writing style or the language used in misleading news can change, as an effort to create content that is equivalent to verified news and thus propagate efficiently through the media network. This type of drift in news is called artificial drift. Given that fake news detection algorithms will probably face such kinds of drifts in the content of news, certain features of misinformation are expected to be modified [36].

Horne et al. [36] provided a thoroughly examined survey on the impact of concept drift in state-of-the-art supervised classifiers for fake news detection incorporating content-based features. They concluded that the performance of content-based models declined over time, but in a slower pace than they considered, which led to the deduction that hand-crafted content-based features that are not topic-specific, are robust to changes. In the same vein, Raza et al. [35] tested how concept drift affects the models in their case, and they supported the same conclusion. Likewise, they also established that the content of fake news does not change frequently as is the case for real news.

Overall, the concept drift impacts the classification algorithms, but different ways were introduced to overcome this problem and improve performance. In general, the results from both mentioned researches [35], [36] suggest that simply retraining the model on a regular basis is enough to keep track of the news changes and avoid concept drift.

3 Related work

Social media have become the main source of information these past years. Due to the dissemination of disinformation in the social media, many approaches have been developed by researchers to address this problem. Moreover, different datasets have been constructed, simulating fake information and these were used for evaluating many machine learning and deep learning methods. In this section, we present the related work that has been done in the field of Fake News Detection.

3.1 Fake News Detection with conventional machine learning approaches

Yazdi et al. [37] proposed a novel approach to address the problem of fake news detection using the SVM classifier. The authors first employed the K-means clustering algorithm as a feature selection method to reduce the dimensions of the dataset and then utilized the SVM for the classification. They analyzed the performance of their proposed approach on BuzzFeed-News dataset and achieved an accuracy of 95.34%. They also tested this approach on the LIAR and BS Detector datasets and obtained accuracy of 94.19% and 93.89% respectively. The results outperformed similar surveys that utilized other feature selection techniques.

Notable work has also been proposed by Hamdi et al. [38] that developed a hybrid approach for fake news detection leveraging user features and graph embeddings. The authors studied the CREDBANK dataset, a large-scale corpus of tweets, from which they created a graph that represents the relations among Twitter users. Then, they used the node2vec model to extract the information incorporated into the graph, along with the Light GBM which was utilized to find the optimal dimensions of the vector representation of nodes in the graph. The selected node embedding features were enhanced with user-based features like followers, friends, etc. and were used as input to train some quite popular supervised models. The best results were achieved with the Linear Discriminant Analysis (LDA) algorithm trained upon the combination of the features and reaching a Recall and F1 score equal to 0.98. The authors concluded that node embedding features provide sufficient information for the reliability of a user and can outperform most surveys that are based only on user-related features.

In the past few decades, the research community expressed interest towards the ensemble learning approaches that train several classifiers in an effective and efficient manner and aggregate the results to extract an outcome. Numerous studies have shown that ensemble approaches have rendered more accurate results than traditional model approaches.

One such ensemble approach was presented by Gravanis et al. [1]. The authors extracted many content-based features which were used along with multiple ML algorithms to classify fake news in five different corpora. Moreover, the authors enhanced the proposed feature sets with word2vec embeddings and noticed a uniform increase in accuracy across the five datasets. Next, popular ML models and ensemble learning classifiers like Ada Boost and Bagging were tested and evaluated in terms of performance. The authors concluded that the experiments conducted with the ensemble learning approaches and SVM outperformed all the other algorithms in terms of accuracy.

In the same vein, Reis et al [39] worked on a wide variety of features from news articles and posts so as to predict fake news with great accuracy. Apart from the textual features extracted from the headline and body of the news source, they also combined features that provide information about the publisher of the news, such as bias, reliability and domain location. A last addition, was features related to the user's engagement and some temporal patterns from the user's activity. The total set of features was evaluated using several state-of-the-art classifiers and the best results were achieved with an XGBoost algorithm reaching an accuracy of 86%.

Later, Mahabub [40] in his research, compared eleven well-known ML algorithms like Naïve Bayes, K-NN, SVM, Random Forest and others to classify fake news. The three algorithms that yielded the best results, were used in an Ensemble Voting Classifier. The results exhibited that Ensemble Voting classifier demonstrated better accuracy scores (94.5%) compared to the results obtained by the individual classifiers.

3.2 Fake News Detection with deep learning approaches

Benamira et al. [41] achieved great success in distinguishing fake from true news by developing a semi-supervised method based on graph neural networks. First, they represented each article with the mean vector of the corresponding GloVe embeddings. Following, the authors constructed the similarity graph network using the k-nearest neighbors and Euclidean distance. Finally, they proceeded to the classification part, where they used two graph neural network algorithms, Graph Convolutional Network (GCN) and Attention Graph Neural Network (AGNN). The results returned where reinforced by 3% in terms of performance compared to other researches.

A benchmark study of machine learning models for fake news detection was presented by Khan et al. [42]. The authors conducted extensive research where they studied 19 models along several dimensions, and in particular eight traditional learning models (SVM, LR, AdaBoost, etc.), six deep learning models (CNN, LSTM, Bi-LSTM, etc.) and five advanced pretrained models (BERT, RoBERTa, ELMo, etc.). These models were tested upon three different datasets from which lexical and sentiment features were extracted along with word embeddings produced by the GloVe algorithm. In this research, the authors concluded that the pretrained models performed significantly better than deep learning and traditional models due to the fact that they were trained upon huge text corpus. Finally, they noticed that only a few models performed quite well when trained upon a small dataset, as most of them were prone to overfitting. The models that returned impressive results upon such datasets where Naïve Bayes and pretrained model RoBERTa.

Meel and Vishwakarma [43] proposed a multimodal fake news detection framework, which utilizes both text-based techniques like the hierarchical attention network (HAN) and visual-based features extracted using image captioning and forensic analysis. Afterwards, they tested four types of multimodal analysis, HAN architecture, CHM, NVI, and Error Level Analysis (ELA) to generate headlines that matched the news content. A total of three datasets were used to examine these approaches, both individually and then cumulatively in a max voting Ensemble classifier. The results returned from the ensemble method outperformed all of the other experiments with an accuracy of 95.90% and led to the significant conclusion that image features add great value to the fake news detection.

In the study by Samadi et al. [44], the combination of different state-of-the-art pre-trained models and neural classifiers is introduced to address the problem of diffusion of misinformation. The proposed framework, commences with an embedding layer which consists of pre-trained models like BERT, RoBERTa, GPT2 and Funnel Transformer, and then connects one of the classifiers, Single-Layer Perceptron (SLP), Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN). Three well-known datasets, LIAR, ISOT and COVID-19 [45] were used to evaluate the above framework. The results revealed the superiority of the proposed method compared to other researches. An impressive outcome was also the improvement of the accuracy by 7% on LIAR dataset.

Another novel approach to fake news detection in social media was introduced by Li et al. [46]. The researchers designed a self-learning, semi-supervised DL network which simultaneously trains a DL machine using predefined labels while also returning a confident pseudo label of unlabeled data to further enhance the labeled dataset. The experimentation was applied to the FakeNewsNet dataset where the results showed that even when the authors used only 20% of the labeled data for training the algorithm, they obtained a precision of 88%. Arguably, this approach yielded impressive results utilizing only a small part of the dataset, and as was proven by the authors, exceeded both supervised and DL approaches.

Notable work has also been performed by Raza et al. [35] where they proposed a framework that can detect fake news at an early stage before this news propagates through the social media network. The authors exploit information from both news content and social context and incorporate this concatenated information into a Transformer architecture to classify the news. The proposed model named FND-NS (Fake News Detection through News content and Social context) is based on bidirectional and autoregressive Transformers (BART architecture) and achieved an accuracy equal to 74.89% surpassing many baseline algorithms. Furthermore, the authors addressed in their study the label shortage issue and created an effective weak supervision scheme for labelling new data.

4 Data and Methods

4.1 Methodology

The proposed methodology concerns the amelioration of the state-of-the-art techniques for detecting fake news by utilizing linguistic features and word vector features from the text of news articles. Various models were trained and evaluated to find the best performing one. An overview of the analysis framework presented in this thesis is illustrated in the Figure 2.

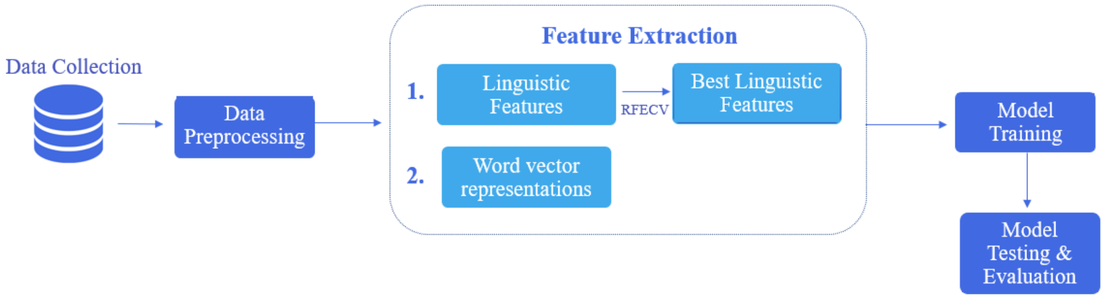

Figure 2: Flow chart of the overall methodology outline

More in detail, the first step of the process is to remove the noise from the data and keep only the necessary information. To accomplish that, many preprocessing steps were applied to the combined raw text of title and main body of the articles. Duplicates and stopwords removal, tokenization and stemming are some of the techniques that were implemented and are presented in section 4.3. Then the cleaned text was used to create word vector representations, which were inserted as an input to the ML models. Later, many linguistic features were constructed to train the ML models on an individual level, as well as to enhance the feature set created by the word representation techniques. The performance of all the tested classifiers was assessed by a widely known measure called Accuracy. Accuracy is the percentage of correct classifications that a trained ML model achieved. It is determined by dividing the number of correct predictions by the total number of predictions made.

The experiments on the dataset were deployed using the Python programming language which is mainly used by the research community due to the vast number of libraries for processing and machine learning that it incorporates.

4.2 Dataset Description

For the purpose of this thesis, the ISOT Fake News dataset provided by Ahmed et al. [47], [48] was used to investigate the issue of fake news detection. The verified news was acquired by crawling news articles from Reuters.com, while the fake news was obtained by unreliable sources, annotated by Politifact (a fact-checking organization in U.S.A.) and Wikipedia. The dataset consists of 21417 real and 23481 fake news, which results in a balanced dataset. Different types of articles on different topics are presented in the dataset, with the majority of them focusing on political and World news topics. Each article comprises information about the title, the body, and the date that the article was published. The title and the body of each article, were combined in a single feature during the analysis to acquire broader information.

The percentage of news articles in each category along with the distribution of the topics, are presented in Figure 3.

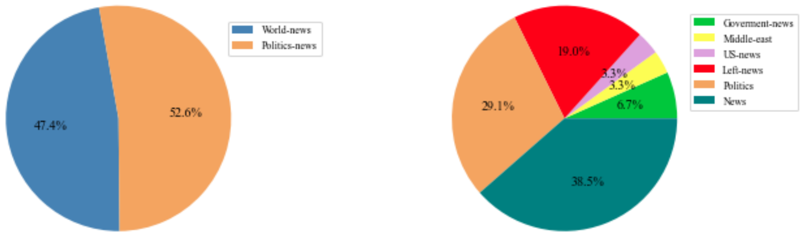

Figure 3: (a) Real News articles topics, (b) Fake News articles topics

4.3 Data Preprocessing

Raw text is an unstructured form of data that contains noisy content, and its quality can directly affect the performance of our models. Data preprocessing is a data mining technique that transforms raw data into concrete information. Thus, data preprocessing is an integral step in Machine Learning, in order to ensure robust results and avoid overfitting. Here, we utilized the Natural Language Toolkit (NLTK) along with Regular Expressions (RE), which are Python libraries, for the implementation of data preprocessing of the ISOT dataset. The preprocessing steps used in this analysis are presented as follows:

Duplicates removal: The first step in data preprocessing is to inspect the dataset and search for duplicate entries. Duplicate values can affect the model performance and increase overfitting. This means that the model will perform best for this dataset, but will not generalize to other types of data. Regarding the experimentation dataset, entries that shared the same title and text were removed from the total sample. The distribution of real and fake news remained balanced after this process.

Lowercase: Python language which was used as an analysis tool in this thesis, is a case sensitive language, which means that it treats uppercase and lowercase characters in a different way. For this reason, all words in news text were transformed into lowercase, which further reduces the vocabulary size.

URL removal: Many articles contain URLs of other links to reference the claims mentioned. The URLs does not provide any insightful information so they were removed from the text.

Punctuation removal: In this part of preprocessing, punctuation marks were eliminated from the test, as they do not convey significant information.

Numbers removal: It is quite difficult for a ML algorithm to grasp the numbers as they take infinite values. Therefore, we deleted all the mentioned numbers from the text.

Stop-words removal: The most used preprocessing step in NLP tasks, is the stop-words removal. Stop-words refer to words that can be found in every document and are not discriminative. Some examples are ‘the’, ‘a’, ‘so’, etc. The removal of these words, reduces the vocabulary size and the training time of the models.

Tokenization: After applying all the previous steps which result in a clean representation of the text of the article, each sentence was converted into a list of words (tokens) to be used as input to the models.

Stemming: Stemming is the process where a word is reduced into its root form. In this way, words in single and plural can be identified as the same word. This applies also when a word is present in a document in different tenses. Stemming helps in dimensionality reduction.

4.4 Feature Extraction

This part of this thesis includes the feature extraction techniques that were used to construct the feature set that will be used in the analysis part to extract meaningful outcomes.

4.4.1 Linguistic features

Burgoon et al. (2003) [49] tested 16 linguistic features to discriminate deceptive communications from truthful ones. In the same spirit, Zhou et al. (2004) [29] proposed a set of 27 linguistic features for differentiating fake from real content. Later, Gravanis et al. (2019) [1] incorporated in their research the combined feature set proposed by Burgoon and Zhou to classify fake news. All these three works concluded that these features count specific cues in text and can capture adequately the characteristics of deceptive and truthful language.

In this thesis, a linguistic- based cue of 35 features taking into consideration the combination of the previously mentioned works [1], [29], [49], were extracted from the available dataset to investigate the fake news problem. Table 1 presents the complete list of the features explored which are conceptually divided into seven categories.

Table 1: Extracted Linguistic features

Category	Feature	Category	Feature
Quantity	# words	Uncertainty	Modifiers
	# syllables		# modal verbs
	# sentences		# uncertainty words
	# verbs		% other reference
	# noun phrases	Non immediacy	passive voice
Complexity	# big words		Subjectivity
	avg syllables per word		% self-reference
	avg # clauses		% group-reference
	avg word length	Expressivity	Emotiveness index
	avg sentence length		Lexical diversity
	avg noun phrase length		Content word diversity
	Pausality		Redundancy
	# short sentences		Typographical error rate
	# long sentences	Specificity	Rate of adjectives & adverbs
	Flesch reading ease		# affective terms
	sentence complexity		# sensory words
	# conjunctions		Spatio-temporal information
		Affect	Sentiment

Quantity: The words, syllables, verbs, sentences and noun phrases of the text were captured and their frequencies were measured.

Complexity: The complexity features aim to capture the total complexity of the document. The articles were examined in two terms of complication, word and sentence level. Features like the number of big words, average syllables per word, average word length were constructed to evaluate the word complexity. Regarding the sentence level complexity, the average number of clauses, the average sentence and noun phrase length were calculated. Additionally, we captured the number of short and long sentences along with the number of conjunctions in the article. To measure the sentence complexity, we used the

Stanford CoreNLP API[6] to parse the sentences and find the independent and subordinate clauses, the presence of which characterizes a sentence as complex.

Pausality is a measure that calculates the number of punctuation marks per sentence.

Readability is the ease with which a reader can comprehend a text. To measure the readability of each article here, we used the Flesch reading ease sentence complexity score which is defined as follows.

$$Reading\ ease = 206.8 - (1.015 \times AVG\ sentence\ length) - (84.6 \times AVG\ word\ length) \quad (1)$$

The formula returns scores in a range from 0 to 100, where a high score indicates that the article is easier to be read, and a low score indicates difficulty in understanding the text.

Uncertainty: Modifiers are words that are used to clarify another word in the sentence. There are two parts of speech that are described as modifiers: adjectives and adverbs.

To calculate the number of uncertainty words in the articles, we followed a lexicon-based approach by utilizing the word list in the Loughran-McDonald Master Dictionary[7] [50]. Finally, the number of modal verbs and the percentage of the third person pronouns (other reference) found in the text were considered.

Non immediacy: In this category of features, we took into consideration the first person singular (self-reference) and plural (group-reference) pronouns. The subjectivity score of each article was defined using the TextBlob's API[8]. The sentences that appeared in passive voice were used as another feature in the final feature set.

Expressivity: Emotiveness index and redundancy were calculated as follows:

$$Emotiveness = \frac{total\ \#\ adjectives + total\ \#\ adverbs}{total\ \#\ nouns + total\ \#\ verbs} \quad (2)$$

[6] https://github.com/nltk/nltk/wiki/Stanford-CoreNLP-API-in-NLTK

[7] https://sraf.nd.edu/loughranmcdonald-master-dictionary/

[8] https://textblob.readthedocs.io/en/dev/api_reference.html

$$Redundancy = \frac{total\ \#\ function\ words}{total\ \#\ sentences} \quad (3)$$

In linguistics, function words express grammatical relationships among other words in the sentence. Pronouns, prepositions, conjunctions and others are some function parts of speech. The total number of unique words divided by the total number of words in each article returned the lexical diversity. In a same way, the total number of content words divided by the total number of words, was defined as the content word diversity. Last, the typographical error rate was added to the feature set.

Specificity: The feature set was expanded further including the rate of adjectives and adverbs and the number of sensory words; words that indicate sensorial experiences such as sounds, smells, etc. The number of affective terms was also calculated following a lexicon-based approach which uses the lists of words in Liu and Hu Lexicon [51], [52]. The lexicon is composed of two lists of words, the first with a positive classification and the second with a negative one. Eventually, information about location and events was also gathered employing Python’s NLTK toolkit POS-tagger.

Affect: The last feature that was constructed, was the sentiment (overall polarity) of each article which was defined using the TextBlob’s API.

4.4.2 Text representation techniques

1) TF-IDF

TF-IDF (Term Frequency–Inverse Document Frequency) is a widely known feature extraction technique which is often used in information retrieval, text mining and text summarization. The TF-IDF is a statistical measure used to identify the importance or relevance of a word in a given document. The TF-IDF score for a given term t in a document d is defined as:

$$tfidf(t,d) = tf(t,d) \times idf(t) \quad (4)$$

where $tf(t,d)$ represents the frequency of term t, how many times the term appears in the document, and $idf(t)$ represents the inverse document frequency.

The Inverse Document Frequency is a metric that measures the informativeness of a term and is defined as:

$$idf(t) = \log\left(\frac{N}{df(d,t)}\right) + 1 \quad (5)$$

where N is the total number of documents and $df(d,t)$ is the number of documents that incorporate the term t. IDF assigns higher weights to words that appear rarely in a document and can be more informative and lower weights to the ones appearing more frequently. Examples of frequent words can be the stop-words that can be found many times in every document and are of little importance. Thus, frequent terms have high impact, while infrequent terms become negligible.

TF-IDF is quite simple to calculate and computationally cheap. But it can suffer from the curse of dimensionality, as a document may include a huge number of words. Here, we utilized the TfidfVectorizer[9] provided by the sklearn library that takes user defined parameters. In order to avoid the dimensionality problem, we set the parameter max_features=10000. This parameter, considers the top max features ordered by the term-frequency (tf) and removes all the rest.

2) word2vec

Word embeddings, is one of the most popular representations of text found in literature. They represent each word as a vector in a predefined vector space, where words with similar meaning can be found in close proximity to one another within the space. Apart from representing words, they also capture the context of the word, the semantic and syntactic similarity and relations between other words in the document.

A recent breakthrough in the world of NLP is Word2vec introduced by Tomas Mikolov in 2013 [53]. Word2vec uses two different model architectures to produce the word embeddings, Continuous Bag-of-Words (CBOW) model and Skip-Gram (SG) model. These

[9] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

models are shallow, two-layer neural networks trained to reconstruct contexts of words. The CBOW model is similar to a feed forward neural network and tries to predict a word given the context of this word [54]. On the other side, the Skip-Gram model tries to learn and predict the context words around the specified input word. In simple terms, the Skip-Gram is exactly the opposite process of the CBOW model [54]. In both approaches, the size of the context window, is user-defined and determines how many words before and after a given word will create the context. Small values are recommended.

Figure 4 depicts the architecture of these two models.

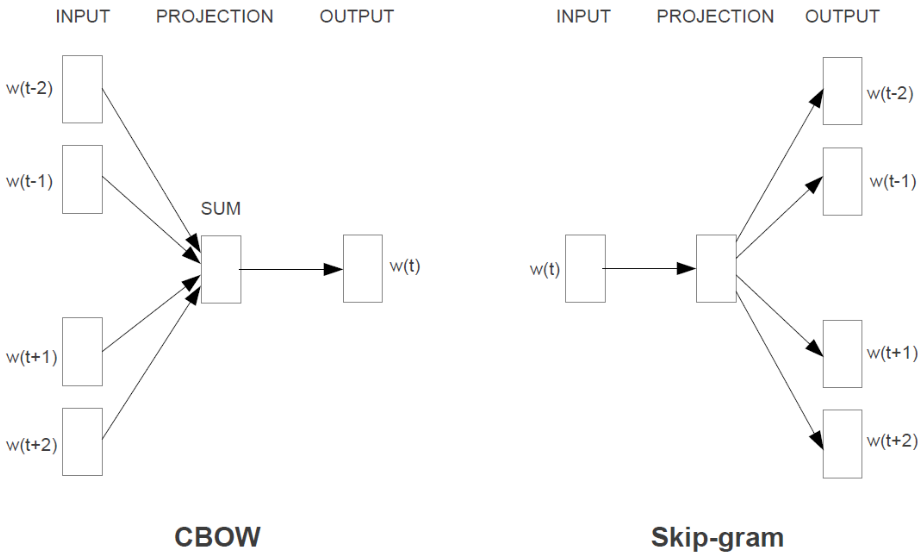

Figure 4: (a) CBOW architecture, (b) Skip-gram architecture [53]

Word embeddings can also be utilized in the concept of transfer learning. Transfer learning is an approach where a model developed for one task can be reused for a completely different task. In NLP, transfer learning techniques, are based on pre-trained language models, which are firstly trained on large amounts of documents and then applied in many other cases. In this way, the learning parameters and the training time are reduced. Additionally, overfitting is avoided as these models generalize well even with small datasets. The most used pre-trained word embeddings across the research community are the pre-trained vectors trained on part of Google News dataset. The model includes a total of 3 million words and phrases in a 300-dimension vector representation. This architecture was used in this project to create word embeddings for each article in the dataset.

3) GloVe

GloVe is an acronym for Global Vectors and is an alternative method to create word embeddings. It is an unsupervised algorithm developed by researchers at Stanford University which created word embeddings by aggregating word co-occurrence matrices from a given corpus [55]. GloVe incorporates both local and global statistics in order to produce the word vectors, compared to word2vec which derives the semantics of a word only by the local surrounding words. The idea behind the GloVe algorithm is that the ratios of word-word co-occurrence probabilities can better distinguish relevant from irrelevant terms.

As a first step, the GloVe algorithm constructs the co-occurrence matrix X_{ij}, where each element represents how many times the word i appears in context of word j. The corpus is scanned with a predefined window size which determines the context of the word. Less weights are assigned to distant words. The next step in training is to learn word vectors such as their dot product equals the logarithm of the words' probability co-occurrence as defined in the equation below.

$$w_i^T w_j + b_i + b_j = \log(X_{ij}) \quad (6)$$

The final step of the algorithm is the definition of the cost function. By minimizing the cost function, the model tries to find the lower-dimensional representations which explain most of the variance of the data in the original dimension.

$$J = \sum_{i,j=1}^{V} f(X_{ij}) \left(w_i^T w_j + b_i + b_j - \log(X_{ij})\right) \quad (7)$$

where $f(X_{ij})$ is a weighting function that helps the algorithm remove the noisy rare co-occurrences that include less information than the more frequent ones [55].

Another important aspect that needs to be highlighted here, is that Glove is a global log-bilinear regression model which is efficient and well-performing on word analogy tasks.

In this project we utilized the Wikipedia 2014 pre-trained GloVe [10] embeddings and tested both the 100-dimension and 300-dimension vectors.

4.5 Machine Learning models

In our research, we selected the most prevailing ML algorithms found in literature related to fake news classification, to successfully evaluate the given dataset. Subsequently, we provide some additional information about each classifier.

4.5.1 Traditional learning models

1) Support Vector Machine (SVM)

It is a supervised learning technique used extensively in binary classification problems and lately extended into multi classification problems as well [56]. The method projects the data into higher dimensions so as to separate them using a hyperplane. To find the optimal hyperplane, the distance between the closest points of the two classes is maximized. This separating hyperplane is called margin. A benefit of the algorithm is that it allows the specification of different kernel functions for defining the separation, such as linear, polynomial, RBF and others. SVM returns high quality results even with little parametrization, but is computationally expensive.

2) K-nearest Neighbors (KNN)

K-nearest Neighbors is another supervised learning model which has many applications in the fields of recommendation systems, semantic searching, anomaly detection as well as fake news detection. The algorithm is computationally efficient and handles noisy data very well. KNN initializes by calculating the distance between the point that needs to be classified and the rest of the training data using the Euclidean distance. It then selects the closest data points to the point examined and uses the Majority vote to classify the unknown point [57].

[10] https://nlp.stanford.edu/projects/glove/

3) Naïve Bayes (NB)

Naïve Bayes is a promising algorithm for text classification problems. Though quite simple, it can yield great results. It is a probabilistic classifier which employs the Bayes theorem to classify data. The prevailing principle upon which the family of Naïve Bayes classifiers has been built, is that the values of the features are statistically independent to each other. Naïve Bayes is widely applied in email filtering and other fields of text analytics [58].

4.5.2 Ensemble learning models

Ensemble Learning method, trains multiple models and combines them to find an optimal solution to a problem. It is also known as “committee of experts”. This approach develops better predictive models compared to traditional approaches and returns robust results.

1) Random Forest (RF)

Random Forest is an ensemble classifier that trains multiple decision trees. Each tree generates a vote for a class and then the individual votes are aggregated and the class with the most votes is selected as the model’s decision result. Unlike decision trees, Random Forest can efficiently handle and avoid overfitting issues. It is a unique amalgamation of prediction results and model interpretability. Another significant advantage of RF is that it performs feature selection, as it is capable of measuring the importance of each feature while classification rules are built [59].

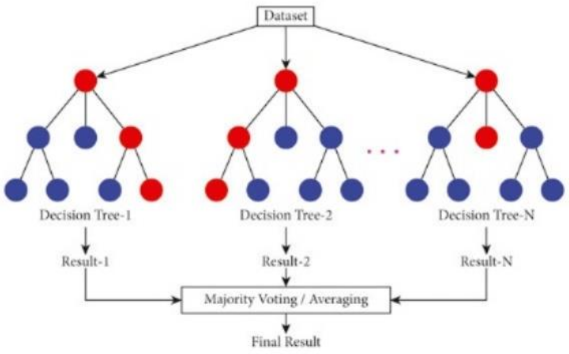

Figure 5: Random Forest architecture [60]

2) Gradient Boosting (GB)

Gradient boosting is a ML algorithm used to solve both regression and classification tasks. It is a mixture of weak classifiers, typically decision trees which are trained in a stage-wise process and combine the results to extract an outcome. To deal with the problem of overfitting the training data, regularization techniques can be applied, like carefully selecting the number of decision trees forming the model (huge number leads to overfitting) and managing the depth of each tree.

3) XG Boost algorithm (XGB)

Extreme Gradient Boosting (XGB) is a decision tree ensemble classifier based on gradient boosting. At each iteration of the algorithm, the residual of a base classifier which is a decision tree, is used in the next classifier in order to optimize the objective function. The minimization of the loss function is used to control the complexity of the trees leading to a pre-pruning strategy. The reduction of tree complexity contributes to the decrease of the training time of the model and the avoidance of overfitting [61].

4) Ensemble Voting classifier

Ensemble Voting classifier is a meta classifier that combines similar or diverse models using majority voting to perform predictions. There are two types of voting schemes used in this classifier: hard and soft voting. In hard voting, the final class is assigned based on the class that got the most votes from the results returned by the classifiers. In soft voting, each classifier returns a probability of each given class label. Then the probabilities from all models are aggregated and the class label with the highest probability is used as a final prediction label. The voting classifier outshines most base models as the prediction is derived from the training of multiple models [62].

4.5.3 Deep learning models

Deep learning approaches (DL) have been widely applied and achieved great results on several complex cognitive tasks. In most cases DL prevails over traditional ML algorithms. DL performs efficiently in different application domains like NLP, cybersecurity, robotics and others. In this thesis, the Convolutional Neural Network and LSTM were used to detect fake news which are presented in greater detail below.

1) Convolutional Neural Network (CNN)

Convolutional Neural Network is considered to be one of the most used architectures for pattern recognition tasks, like computer vision, speech processing, face recognition, etc. Their architecture is inspired by the visual cortex. A quite common type of CNN consists of many convolution layers following sub-sampling (pooling) layers and a fully connected layer (FC) [63]. An example of CNN architecture is illustrated in Figure 6.

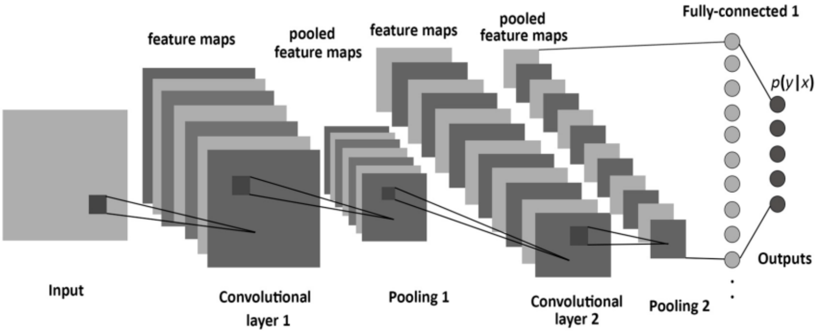

Figure 6: The structure of a CNN, consisting of convolutional, pooling and fully-connected layers [64]

The input layer of the CNN contains the vector representation of text data. Following, the convolutional layer is the first and most significant layer in a CNN architecture which is used to extract features from the input data using a filter or kernel of a particular size $M \times M$. The filter slides over the input data and the dot product of the input data and the kernel is calculated. The values produced by the calculation represent the feature map and an activation function is used to produce the output of the layer. The activation functions that are most commonly used in CNN are sigmoid, tanh and ReLu. Next, a pooling layer is constructed. This layer shrinks large-size feature maps into smaller feature maps, while preserving most of the information of the input. Several types of pooling methods exist, with the most commonly used ones being the max and average pooling. Fully connected layer is the last layer of the CNN architecture. It is called fully connected as every neuron is connected with all neurons in the previous layer. The output of the FC layer is the final output of the CNN algorithm. The benefit of using CNN is the “weight sharing”, which reduces the number of trainable parameters and the training time while controlling the overfitting issue [65].

2) Long-Short Term Memory neural network (LSTM)

Recurrent Neural Networks (RNN) are artificial neural networks that incorporate circular connections between higher and lower-level neurons and allow the output of previous nodes to affect the input of the next nodes [66].

LSTM belongs to the family of RNN and was introduced by Sepp Hochreiter and Jürgen Schmidhuber [67] to address the problem of the vanishing gradient that RNN were facing. LSTM is a dynamic system which propagates data from previous events to current processing steps. Therefore, the behavior of the network is impacted by the input at a given time and its previous state. The LSTM consists of three gates, the input gate i_t, the output gate o_t and the forgetting gate f_t, all of which use the sigmoid as an activation function in the output layer with a range [0,1]. The input gate evaluates if the input should affect the internal state, and the output gate evaluates if the internal state should affect the output. The forgetting gate is the one that determines how much information from the previous states should be preserved at every step. Due to this architecture, LSTM is applicable to tasks such as speech recognition, handwriting recognition, machine translation, image processing and others [66].

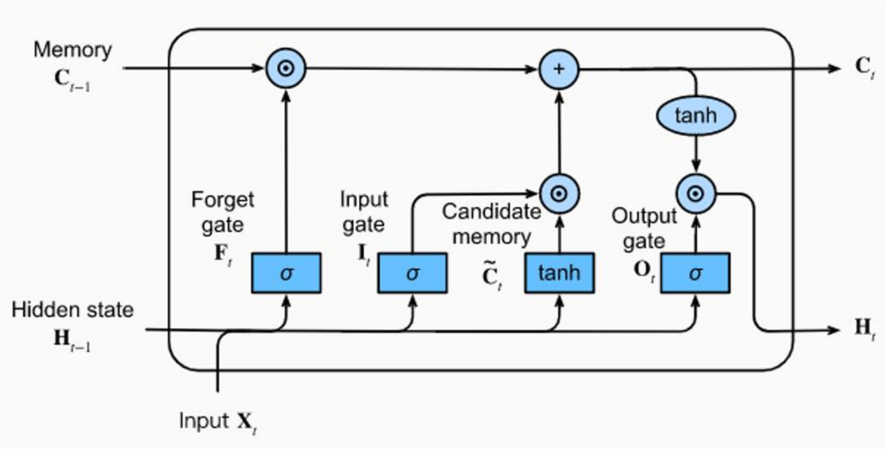

Figure 7: Long-Short Term Memory (LSTM) cell with input, output and forgetting gates[11].

[11] https://d2l.ai/chapter_recurrent-modern/lstm.html

5 Experimental Results

In this section we discuss the experiments we conducted in order to contribute to the fake news classification research area. Many different feature extraction techniques were investigated like TF-IDF, word2vec and GloVe along with linguistic features extraction as described in section 4.4. The word representation techniques were evaluated, and the one that returned the best results was used to train the models examined. Additionally, the best feature set was determined utilizing many feature selection techniques so as to reduce the dimensions of the dataset, avoid overfitting and allow more generalization to the model. Afterwards, the word representation features were enhanced with the best feature set, to verify the claim that linguistic features improve the model performance. The same preprocessing technique was applied across all experiments.

Furthermore, we performed an extensive classification algorithm study for introducing a robust model that detects effectively fake news articles, utilizing the best performing feature set. Traditional learning models (SVM, KNN, etc.) and ensemble models (Random Forest, etc.) were trained using 10-fold cross validation. During the k-fold cross validation procedure, the dataset is equally divided into k subsets. Subsequently, k iterations are performed upon which k-1 subsets are used for training and the remaining fold (different at each iteration) is used for validation. Using cross validation, we achieved more robust results and generalization. Moreover, the grid search approach was implemented to determine the optimal parameters of each classifier and increase its performance. Grid Search performs an exhaustive search over specified parameters and returns the optimal combination based on the specified metric.

Deep learning models were also investigated. Different architectures of these models were examined to conclude to the ones that performed the best. As DL models are complex architectures that can lead to high dimensionality when the documents are large, we limited the vocabulary size to the 10000 most frequent words.

Following, the results obtained by using the proposed algorithms are discussed in more detail.

5.1 Experiments on machine learning models

5.1.1 Linguistic feature set evaluation results

As previously mentioned, we extracted 35 linguistic features considering the literature as cited in section 4.4. As a first step in our analysis, we implemented all these features to both traditional and ensemble ML models and evaluated them in terms of accuracy. Support vector Machine (SVM), K-nearest Neighbors (KNN), Naïve Bayes, Random Forest, Gradient Boosting and XGBoost are the models trained upon the fake news dataset. For the construction of the models, we conducted an extensive grid search to find the user-defined parameters that maximize the performance. Alongside with the grid search, we used a 5-fold cross validation to avoid overfitting to the training set. After obtaining the best parameters for each model, we trained each classifier using a 10-fold cross validation to acquire more robust results. Table 2 presents the results of the fine-tuned tested classifiers.

It is obvious from the results table, that ensemble methods outperform all traditional methods. SVM with little parametrization (linear kernel) reached an accuracy of 94.1%, while KNN and Naïve Bayes did not perform quite well compared to the rest models. On the other hand, XGBoost obtained an accuracy of 97.5% outshining all other classifiers.

Table 2: Model results using Linguistic Features

Classifier	**Accuracy (%)**
SVM	94.1
KNN	89.1
Naïve Bayes	69.8
Random Forest	96.2
Gradient Boosting	97.3
XGBoost	**97.5**

5.1.2 Best Linguistic feature set evaluation results

As a next step, we attempted to limit the dimensions of the examined dataset without sacrificing the achieved accuracy up to this point. The dimensionality reduction results in shorter training times and more efficient algorithms that can perform equally well with other unknown datasets. Additionally, many ML algorithms exhibit a decrease in accuracy when the number of features is significantly higher than the optimal [68]. To find the best combination of features that will achieve the same or higher accuracy numbers, we investigated three approaches. Each of these approaches returned a number of features which were tested upon the classifiers under investigation to evaluate their performance. In the case where the produced feature set degraded the previously achieved accuracy of the models, the method was rejected and we continued with the testing of another method. The feature selection methods that were examined, are discussed in more detail below.

1) Boruta algorithm

Boruta algorithm is a wrapper feature selection algorithm that uses a classifier to return a feature ranking. For simplicity, the classifier should be simple and computationally efficient. Most of the time, Random Forest (RF) is utilized in this method, as it does not require heavy tuning of the parameters and can estimate the feature importance [68]. The algorithm starts adding randomness to the given dataset by creating copies of shuffled values of the attributes which are called “shadow features”. Thereinafter it continues with the training of the extended dataset using the RF classifier and measures the importance of each attribute. At each iteration, Boruta uses as reference the highest importance of the shadow features and removes the original features that have a lower accuracy and are deemed unimportant. The process is repeated until the importance is computed for all features or when the maximum number of iterations defined is reached [68].

In this experiment, the Boruta algorithm deemed as important **34 out of 35 features**, which is not a significant reduction in dimensionality. Furthermore, some of the classifiers were tested upon the reduced feature set and a degradation in accuracy was noted. Therefore, we did not continue with this method.

2) Select-From-Model algorithm

Select-From-Model is an embedding method for feature selection which at each iteration of the training process, returns the features that contribute most to the training. The algorithm initializes by defining a threshold value as a boundary between the features that will be kept and those that will be removed. The threshold is user-defined. After that, all features are sorted by the Gini importance score and those that are below the determined threshold will be eliminated from the dataset. In the same vein as with the previous algorithm, Select-From-Model can be used with classifiers that measure features' importance [69].

To run the Select-From-Model algorithm, we used the respective package[12] from sklearn library. The classifier tested was Random Forest and the threshold after some experimentation was set equal to 0.001. The mentioned algorithm deemed as important **30 out of 35 features**. Nevertheless, the new feature set decreased again the accuracy in some of the classifiers examined, so we did not proceed with this method either.

3) Recursive Feature Elimination with Cross Validation algorithm (RFECV)

RFECV is a feature selection method that uses the coefficients or features' importance to determine the optimal number of features that maximize performance. At each iteration of the method, Random Forest is fitted to the data and the importance of each feature is calculated. Next, the features are ranked based on the importance score and the one with the lowest importance value is removed from the dataset. The method is repeated until it concludes to the final feature set [70]. The number of features that can be eliminated at each iteration is defined by the user in order to optimize the algorithm.

We utilized again the RFECV[13] package from sklearn with a base RF classifier, Gini criterion and at each step of the method we eliminated one feature at a time. RFECV returned **28 out of 35** features as important. After testing this feature set on the models, we noticed an increase in accuracy in some models while in others we achieved the same accuracy that was returned using the whole feature set. Due to this, we proceeded with these 28-feature subset which will be mentioned as "Best Lexical Features" for the rest of the analysis. Table 3 presents the optimal subset of features.

[12] https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFromModel.html
[13] https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html

Table 3: Best Linguistic feature set obtained by RFECV

Best linguistic features	
# words	Flesch reading ease
# syllables	sentence complexity
# sentences	Modifiers
# verbs	% other reference
# noun phrases	Subjectivity
# big words	% self-reference
avg syllables per word	Emotiveness index
avg # clauses	Lexical diversity
avg word length	Content word diversity
avg sentence length	Redundancy
avg noun phrase length	Typographical error rate
Pausality	Rate of adjectives & adverbs
# short sentences	# affective terms
# long sentences	Spatio-temporal information

All the previously mentioned traditional and ensemble models with the same parameters were trained upon the dataset with the best linguistic features and Table 4 presents the accuracy scores that were obtained. Naïve Bayes increased its accuracy to 72.2% compared to using all the features. On the other hand, all other models retained the same accuracy values while using a smallest part of the feature set. Thus, we managed to reduce the dimensions with all the benefits that follow, while preserving the accuracy levels for all models.

Table 4: Model results using Best Linguistic Features

Classifier	Accuracy (%)
SVM	94.1
KNN	89.3
Naïve Bayes	72.2
Random Forest	96.3

Gradient Boosting	97.3
XGBoost	**97.5**

5.1.3 Text representations evaluation results

A very popular approach to fake news detection is the use of text representation techniques and word embeddings. Driven by that, we conducted experiments using this approach and compared it with the linguistic approach. To begin with, we have chosen to test three text representation techniques, TF-IDF, word2vec and GloVe using a linear SVM model and select the one with the highest performance to train the rest of the models. The results obtained during this process are presented in Table 5.

Table 5: SVM results using text representations

SVM classifier	**Accuracy (%)**
TF-IDF	**98.8**
Word2vec	94.3
GloVe 100 dim	92.8
GloVe 300 dim	94.2

The results indicate that even though TF-IDF is a simple feature extraction model, it achieved the best accuracy scores compared to word embeddings. Based on this outcome, we trained all the models using the TF-IDF representation and provide the findings in Table 6. Once again, ensemble classifiers seem to outperform traditional ones, with XGBoost reaching an accuracy equal to 99%. As compared to the linguistic approach (all features and best features), all classifiers improved their accuracy values with Naïve Bayes showing a significant increase of 0.21.

Table 6: Model results using TF-IDF

Classifier	**Accuracy (%)**
SVM	98.7
KNN	90.4
Naïve Bayes	92.8
Random Forest	98.3

Gradient Boosting	98.7
XGBoost	**99.0**

5.1.4 TF-IDF features enhanced with Best Linguistic features evaluation results

As a final step, we combined the features extracted using TF-IDF with the best linguistic features. A model trained upon the best linguistic features set and the same model trained upon TF-IDF features were combined using an ensemble voting classifier. The same process was followed for all the models examined in this section. The voting classifiers assigned votes to the predictions and yielded the final prediction set which was measured in terms of accuracy. The results are illustrated in Table 7.

An improvement in all scores is prevalent in the results obtained from this method. Once again, XGBoost showed the highest performance with 99.6% accuracy. SVM even though is a simple classifier, it achieved a comparable accuracy to the ensemble methods equal to 99.3%.

Table 7: Model results using TF-IDF & Best Feature set

Classifier	**Accuracy (%)**
SVM	99.3
KNN	94.3
Naïve Bayes	92.8
Random Forest	98.4
Gradient Boosting	99.3
XGBoost	**99.6**

5.2 Experiments on deep learning models

We also performed experiments with DL models and we evaluated their performance compared to the rest models. The main difficulty in training DL models was the hyper parameter tuning and finding the appropriate depth of the network. These neural networks have a wide range of parameters that need to be tested before concluding to the optimal. Additionally, many different layers can be used, with different number of neurons at each layer. All these indicate that usually a significant amount of time is needed for experimentations until concluding to the neural network architecture that maximizes performance.

In this thesis, we ran experiments using the Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM) which are widely used in text classification problems across the research community. Each DL model was trained upon multiple architectures, while some common layers were used in both models to avoid overfitting the training set. As overfitting represents a major issue in neural networks, many intuitive concepts exist to overcome this issue. In our examples we selected to apply Batch Normalization and Dropout.

Batch Normalization is a layer that normalizes the inputs by transforming them to have a mean of zero and a standard deviation of one. This technique speeds up the training process and can be applied at any stage in the model. Another commonly used overfitting improvement technique is Dropout. Dropout is a regularization method for neural networks that at each training step a user defined number of units (neurons) is not taken into consideration, resulting in a smaller network [65].

1) Convolutional Neural Network (CNN) training

Many experiments were conducted using CNN to reach an optimal network. The ones we discuss here, are those that achieved a satisfying performance.

CNN architecture # 1

In this CNN, an embedding layer of 300 dimensions was used, along with two dense layers. As mentioned before, Batch Normalization and Dropout prevented the model from overfitting the data. Global Average Pooling (GAP) was used to reduce the feature map. Global Max Pooling (GMP) was also tested, but it proved to lower the accuracy of the model, so we proceeded with the first option. After experimentation with the number of neurons, a Convolutional layer of 128 neurons was added to the architecture of the model. A notable insight here was that when the number of neurons in the convolutional layer increased, the performance declined.

The CNN model described in Table 8, returned **96.81%** accuracy and is the one with the highest performance compared to the rest of the trained CNNs.

Figure 8 illustrates the accuracy and the loss in the training and validation set, which is a good indicator that the model generalizes well. The training and validation accuracy increases with the epochs and the respective loss values decline, both indicating that the model learns to classify the articles better.

Table 8: CNN architecture #1

Layers	Output Dimension
Input layer	120
Embedding layer	300
Convolutional (1D) layer	128
Global Average Pooling	128
Dense layer	100
Batch Normalization	100
Dropout	100
Dense layer	1

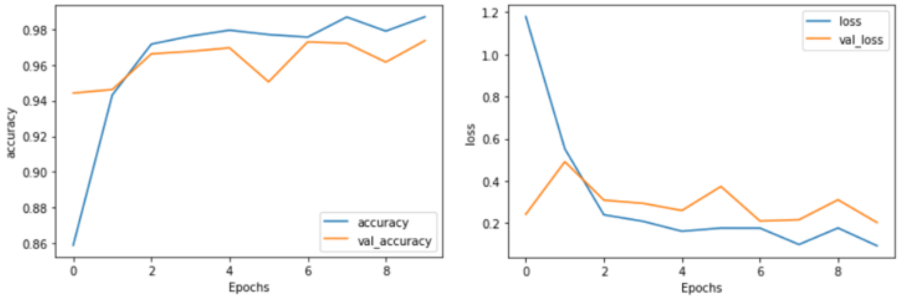

Figure 8: Learning curves of the accuracy and the loss for training and validation sets (CNN#1)

CNN architecture # 2

As a later experiment, we wanted to test if additional convolutional layers would lead to an increase in accuracy. Thus, we added two convolutional layers, one with 128 neurons and another with 64, after conducting several experiments to conclude to the number of neurons. The results showed that the accuracy decreased with the addition of another layer and reached 95.81%.

Table 9: CNN architecture #2

Layers	**Output Dimension**
Input layer	120
Embedding layer	300
Convolutional (1D) layer	128
Convolutional (1D) layer	64
Global Average Pooling	64
Dense layer	100
Batch Normalization	100
Dropout	100
Dense layer	1

CNN architecture # 3

Finally, we examined the influence of feeding the model with pretrained word embeddings instead of using the ones produced directly by the neural network. First, we utilized the pretrained word2vec embeddings (300 dimensions) in the embedding layer of the CNN, while keeping the same structure as the CNN #1. The CNN with the word2vec resulted in an accuracy of 93.60%. Later, we replaced the embedding layer with the pretrained GloVe embeddings (300 dimensions) which returned 74.26% accuracy score. Using both embedding approaches, we observed degradation in accuracy, especially while using the GloVe algorithm.

2) Long-Short Term Memory (LSTM) training

Another extensively used algorithm for fake news detection that was also tested here, is a variation of LSTM called Bi-directional LSTM. Various layers with different number of neurons were employed to conclude to the architecture with the best results. Some of the best performing structures are discussed below.

Bi-LSTM architecture # 1

An embedding layer of 300 dimensions preceded the Bi-directional LSTM layer with 32 neurons. A similar sequence of a Dense layer, Batch normalization and Dropout followed. This neural network acquired an accuracy score equal to **95.78%**, which is the highest accuracy obtained by using the Bi-LSTM architectures. The learning curves demonstrated in Figure 9 are indicative of the good performance of the model.

Table 10: Bi-LSTM architecture #1

Layers	Output Dimension
Input layer	120
Embedding layer	300
Bi-LSTM	32
Dense layer	100
Batch Normalization	100
Dropout	100
Dense layer	1

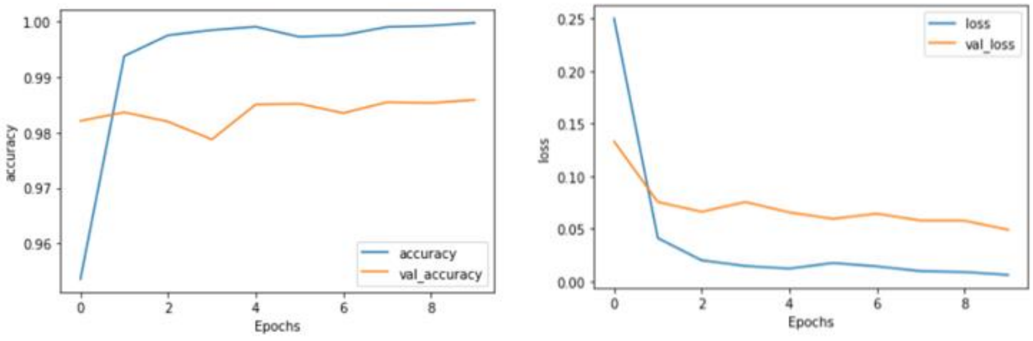

Figure 9: Learning curves of the accuracy and the loss for training and validation sets (LSTM #1)

Bi-LSTM architecture # 2

Another test was carried out with a different architecture of Bi-LSTM network. We applied two sequential layers of Bi-LSTM, while preserving the rest of the structure as previously mentioned. This model peaked at an accuracy of 95.74% which indicates that an additional layer of Bi-LSTM could not improve performance.

Table 11: Bi-LSTM architecture #2

Layers	Output Dimension
Input layer	120
Embedding layer	300
Bi-LSTM	32
Bi-LSTM	64
Dense layer	100
Batch Normalization	100
Dropout	100
Dense layer	1

3) Combination of CNN and Bi-LSTM

Finally, we combined CNN and Bi-LSTM layers into a neural network and evaluated the results. The outcome was quite promising reaching an accuracy of 96.06%, but it did not manage to outperform CNN #1.

Table 12: CNN & Bi-LSTM architecture

Layers	Output Dimension
Input layer	120
Embedding layer	300
Convolutional (1D) layer	128
Global Average Pooling	128
Bi-LSTM	32
Dense layer	100

Dropout	100
Dense layer	1

6 Discussion

The outcomes of this study, support the findings in the literature about linguistic features improving the performance of the classifiers when used as an enhancement to text representations and word embeddings. A variety of traditional and ensemble models with different parametrization were tested to endorse this assumption. Moreover, DL models were also employed to tackle the problem of fake news detection investigated in this study, by utilizing only word embeddings and achieved satisfactory performance. An overview of the results achieved from the models examined, is summarized in Table 13 and Table 14.

Table 13: Accuracy for different feature sets and models

Method	**Model accuracy (%)**					
	SVM	**KNN**	**NB**	**RF**	**GB**	**XGB**
Linguistic Features	94.1	89.1	69.8	96.2	97.3	97.5
Best Linguistic Features	94.1	89.3	72.2	96.3	97.3	97.5
TF-IDF	98.7	90.4	92.8	98.3	98.7	99.0
Best Linguistics & TF-IDF	**99.3**	94.3	92.8	98.4	**99.3**	**99.6**

Table 14: Accuracy for DL models

Model accuracy (%)						
CNN #1	**CNN #2**	**CNN (word2vec)**	**CNN (GloVe)**	**LSTM #1**	**LSTM #2**	**CNN & LSTM**
96.8	95.8	93.6	74.3	95.8	95.7	96.1

During the experimental process, we extracted 35 linguistic features which resulted in 97.5% accuracy produced by training an XGBoost classifier. Subsequently, we searched for a subset of the feature set aiming at reducing the dimensions of the dataset without lowering the accuracy, which was achieved by the RFECV algorithm that deemed 28 features as important. Again, XGBoost performed best in this dataset preserving the same accuracy as the one obtained by using the whole feature set. Additionally, text representations and word embeddings were utilized as feature extraction techniques. It is worth noting that when the article's text was represented using pre-trained word embeddings like word2vec and GloVe, the accuracy scores of all models were reduced compared to the scores returned by the TF-IDF text representation method. There can be several reasons to explain these findings. One might say that it is possible that word embeddings cannot represent efficiently the news articles due to their large length. The TF-IDF method obtained 99% accuracy with an XG Boost method. Furthermore, the combination of the best linguistic feature set with the enhancement of TF-IDF features, reinforced the performance of all the classifiers acquiring an accuracy of 99.6%. Taking into consideration all of the above, we can conclude that detecting deception by using text representation methods achieves better results compared to using only linguistic features. Moreover, the consolidation of these two methods was able to better capture the characteristics of fake and real news and thus result in the highest performance.

The experimental results on traditional and ensemble models, indicate that ensemble methods overshadowed traditional models in terms of performance. Both Random Forest and Gradient Boosting achieved high accuracy scores, while XG Boost surpassed all other models in all cases examined. Another remarkable observation is that linear SVM yielded quite impressive results and can compete with ensemble methods. In the approach that combines linguistic features with TF-IDF features, SVM gained 99.3% accuracy, equivalent to the one returned by the ensemble Gradient Boosting.

The results of the DL models were quite promising, despite lacking any explicit linguistic features. CNN and LSTM were trained deploying different architectures, with CNN outperforming LSTM with an accuracy of 96.8%. The obtained results signify that the normalization techniques applied to the models, led to the prevention of overfitting and delivered robust outcomes. The limitation in our case, is that neural networks require huge amount of data to return impressive results, so the dataset used in this experimentation made it difficult for DL models to overcome the performance of the state-of-the-art ML models. However, both CNN and LSTM have a lot of potential since there are still many combinations and architectures that could be examined further.

Eventually, it is worth pointing out that, Ahmed et al. (2017) [48] extracted TF-IDF features for detecting fake news using the same ISOT dataset examined here. Their model achieved an accuracy of 92%, while our approach peaked at an accuracy of 99.6%. The difference in performance may be an indicator that the combination of linguistic cues with word vector representations can considerably amplify the predictive ability.

7 Conclusions and Future work

The objective of this thesis was to discuss the state-of-the-art models for fake news detection and propose an effective way to remedy the issue of dissemination of misinformation using linguistic cues. Several supervised classifiers were trained using linguistic features, word vector representations and their combination, with the latter scheme outshining in all models acquiring an accuracy of 99.6%. Ensemble learners have shown the greatest performance compared to individual learners. DL algorithms were also explored as a potential solution, but they did not manage to overcome the performance of ensemble models due to the size of the dataset and they obtained a maximum accuracy equal to 96.8%. The promising results returned by the deep neural networks indicate that there is potential for further experimentation to improve performance.

In this study, content-based approaches were utilized to explore patterns in text that differentiate fake from real news, as we aimed to detect fake news in real time before these propagate to social media platforms. For that reason, a possible object of future research could be to employ meta-data about the source and the author of the news, as well as information about the diffusion of these news in social media platforms. Additional information could also be exploited by images and videos included in the articles, employing Transfer learning and pre-trained models. Another possible direction could be the use of unsupervised learning. The limited accessibility to labeled data is a major challenge in the field of fake news detection. To overcome this problem, unsupervised learning algorithms like cluster analysis, outlier analysis, etc. could be explored.

Bibliography

[1] G. Gravanis, A. Vakali, K. Diamantaras, and P. Karadais, “Behind the cues: A benchmarking study for fake news detection,” *Expert Syst Appl*, vol. 128, pp. 201–213, Aug. 2019, doi: 10.1016/j.eswa.2019.03.036.

[2] P. K. Verma, P. Agrawal, I. Amorim, and R. Prodan, “WELFake: Word Embedding over Linguistic Features for Fake News Detection,” *IEEE Trans Comput Soc Syst*, vol. 8, no. 4, pp. 881–893, Aug. 2021, doi: 10.1109/TCSS.2021.3068519.

[3] D. P. Kasseropoulos and C. Tjortjis, “An Approach Utilizing Linguistic Features for Fake News Detection,” in *Artificial Intelligence Applications and Innovations*, 2021, pp. 646–658.

[4] D. P. Kasseropoulos, P. Koukaras, and C. Tjortjis, “Exploiting Textual Information for Fake News Detection,” *Int J Neural Syst*, vol. 32, no. 12, p. 2250058, 2022, doi: 10.1142/S0129065722500587.

[5] V. Chouliara and E. Kapoteli, “Social Media Sentiment Analysis Related to COVID-19 Vaccinations,” in *Artificial Intelligence and Machine Learning for Healthcare: Vol. 2: Emerging Methodologies and Trends*, A. and C. Y.-W. and J. V. and J. L. C. Lim Chee Peng and Vaidya, Ed. Cham: Springer International Publishing, 2023, pp. 47–69. doi: 10.1007/978-3-031-11170-9_3.

[6] X. Zhang and A. A. Ghorbani, “An overview of online fake news: Characterization, detection, and discussion,” *Inf Process Manag*, vol. 57, no. 2, Mar. 2020, doi: 10.1016/j.ipm.2019.03.004.

[7] X. Zhou and R. Zafarani, “A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities,” *ACM Comput Surv*, vol. 53, no. 5, Sep. 2020, doi: 10.1145/3395046.

[8] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake News Detection on Social Media: A Data Mining Perspective,” *SIGKDD Explor. Newsl.*, vol. 19, no. 1, pp. 22–36, Sep. 2017, doi: 10.1145/3137597.3137600.

[9] H. Allcott and M. Gentzkow, “Social media and fake news in the 2016 election,” *Journal of Economic Perspectives*, vol. 31, no. 2. American Economic Association, pp. 211–236, Mar. 01, 2017. doi: 10.1257/jep.31.2.211.

[10] J. Golbeck *et al.*, “Fake news vs satire: A dataset and analysis,” in *WebSci 2018 - Proceedings of the 10th ACM Conference on Web Science*, May 2018, pp. 17–21. doi: 10.1145/3201064.3201100.

[11] K. Sharma, F. Qian, H. Jiang, N. Ruchansky, M. Zhang, and Y. Liu, “Combating fake news: A survey on identification and mitigation techniques,” *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 3. Association for Computing Machinery, Apr. 01, 2019. doi: 10.1145/3305260.

[12] E. C. Tandoc, Z. W. Lim, and R. Ling, “Defining ‘Fake News’: A typology of scholarly definitions,” *Digital Journalism*, vol. 6, no. 2. Routledge, pp. 137–153, Feb. 07, 2018. doi: 10.1080/21670811.2017.1360143.

[13] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi, “Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Sep. 2017, pp. 2931–2937. doi: 10.18653/v1/D17-1317.

[14] R. Hobbs, “Propaganda in an Age of Algorithmic Personalization: Expanding Literacy Research and Practice,” *Read Res Q*, vol. 55, no. 3, pp. 521–533, Jul. 2020, doi: 10.1002/rrq.301.

[15] S. Zannettou, M. Sirivianos, J. Blackburn, and N. Kourtellis, “The web of false information: Rumors, fake news, hoaxes, clickbait, and various other shenanigans,” *Journal of Data and Information Quality*, vol. 11, no. 3, 2019, doi: 10.1145/3309699.

[16] V. Swami, “Social psychological origins of conspiracy theories: The case of the Jewish conspiracy theory in Malaysia,” *Front Psychol*, vol. 3, no. AUG, 2012, doi: 10.3389/fpsyg.2012.00280.

[17] T. K. Hartman *et al.*, “Different Conspiracy Theories Have Different Psychological and Social Determinants: Comparison of Three Theories About the Origins of the COVID-19 Virus in a Representative Sample of the UK Population,” *Front Polit Sci*, vol. 3, Jun. 2021, doi: 10.3389/fpos.2021.642510.

[18] S. A. Alkhodair, S. H. H. Ding, B. C. M. Fung, and J. Liu, “Detecting breaking news rumors of emerging topics in social media,” *Inf Process Manag*, vol. 57, no. 2, Mar. 2020, doi: 10.1016/j.ipm.2019.02.016.

[19] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter, “Detection and resolution of rumours in social media: A survey,” *ACM Computing Surveys*, vol. 51, no. 2. Association for Computing Machinery, Feb. 01, 2018. doi: 10.1145/3161603.

[20] M. Potthast *et al.*, “Crowdsourcing a Large Corpus of Clickbait on Twitter,” 2018.

[21] Y. Chen, N. J. Conroy, and V. L. Rubin, “Misleading online content: Recognizing clickbait as ‘false news,’” in *WMDD 2015 - Proceedings of the ACM Workshop on Multimodal Deception Detection, co-located with ICMI 2015*, Nov. 2015, pp. 15–19. doi: 10.1145/2823465.2823467.

[22] B. Horne and S. Adali, “This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News,” *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, Jan. 2017, doi: 10.1609/icwsm.v11i1.14976.

[23] E. Tacchini, G. Ballarin, M. L. della Vedova, S. Moret, and L. de Alfaro, “Some Like it Hoax: Automated Fake News Detection in Social Networks,” Apr. 2017, [Online]. Available: http://arxiv.org/abs/1704.07506

[24] S. Kumar, R. West, and J. Leskovec, “Disinformation on the web: Impact, characteristics, and detection of wikipedia hoaxes,” in *25th International World Wide Web Conference, WWW 2016*, 2016, pp. 591–602. doi: 10.1145/2872427.2883085.

[25] M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein, “A Stylometric Inquiry into Hyperpartisan and Fake News,” pp. 231–240, 2018, doi: 10.5281/zenodo.1239675.

[26] V. Rubin, N. Conroy, and Y. Chen, “Towards News Verification: Deception Detection Methods for News Discourse,” Jan. 2015. doi: 10.13140/2.1.4822.8166.

[27] Y. Yang, L. Zheng, J. Zhang, Q. Cui, Z. Li, and P. S. Yu, “TI-CNN: Convolutional Neural Networks for Fake News Detection,” *CoRR*, vol. abs/1806.00749, 2018, [Online]. Available: http://arxiv.org/abs/1806.00749

[28] D. de Beer and M. Matthee, “Approaches to Identify Fake News: A Systematic Literature Review,” in *Lecture Notes in Networks and Systems*, vol. 136, Springer, 2021, pp. 13–22. doi: 10.1007/978-3-030-49264-9_2.

[29] L. Zhou, J. K. Burgoon, J. F. Nunamaker, J. R. And, and D. Twitchell, “Automating Linguistics-Based Cues for Detecting Deception in Text-based Asynchronous Computer-Mediated Communication,” 2004.

[30] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, “Automatic Detection of Fake News,” Aug. 2017, [Online]. Available: http://arxiv.org/abs/1708.07104

[31] Z. Jin, J. Cao, Y. Zhang, J. Zhou, and Q. Tian, “Novel Visual and Statistical Image Features for Microblogs News Verification,” *IEEE Trans Multimedia*, vol. 19, no. 3, pp. 598–608, Mar. 2017, doi: 10.1109/TMM.2016.2617078.

[32] C. Castillo, M. Mendoza, and B. Poblete, “Information Credibility on Twitter,” in *Proceedings of the 20th International Conference on World Wide Web*, 2011, pp. 675–684. doi: 10.1145/1963405.1963500.

[33] S. R. Sahoo and B. B. Gupta, “Multiple features based approach for automatic fake news detection on social networks using deep learning,” *Appl Soft Comput*, vol. 100, Mar. 2021, doi: 10.1016/j.asoc.2020.106983.

[34] Z. Jin, J. Cao, Y. Zhang, and J. Luo, “News Verification by Exploiting Conflicting Social Viewpoints in Microblogs,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Mar. 2016, doi: 10.1609/aaai.v30i1.10382.

[35] S. Raza and C. Ding, “Fake news detection based on news content and social contexts: a transformer-based approach,” *Int J Data Sci Anal*, vol. 13, no. 4, pp. 335–362, May 2022, doi: 10.1007/s41060-021-00302-z.

[36] B. D. Horne, J. NØrregaard, and S. Adali, “Robust fake news detection over time and attack,” *ACM Trans Intell Syst Technol*, vol. 11, no. 1, Dec. 2019, doi: 10.1145/3363818.

[37] K. Majbouri Yazdi, A. Majbouri Yazdi, S. Khodayi, J. Hou, W. Zhou, and S. Saedy, “Improving Fake News Detection Using K-means and Support Vector Machine Approaches.”

[38] T. Hamdi, H. Slimi, I. Bounhas, and Y. Slimani, “A Hybrid Approach for Fake News Detection in Twitter Based on User Features and Graph Embedding,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020, vol. 11969 LNCS, pp. 266–280. doi: 10.1007/978-3-030-36987-3_17.

[39] J. C. S. Reis, A. Correia, F. Murai, A. Veloso, F. Benevenuto, and E. Cambria, "Supervised Learning for Fake News Detection," *IEEE Intell Syst*, vol. 34, no. 2, pp. 76–81, Mar. 2019, doi: 10.1109/MIS.2019.2899143.

[40] A. Mahabub, "A robust technique of fake news detection using Ensemble Voting Classifier and comparison with other classifiers," *SN Appl Sci*, vol. 2, no. 4, Apr. 2020, doi: 10.1007/s42452-020-2326-y.

[41] A. Benamira, B. Devillers, E. Lesot, A. K. Ray, M. Saadi, and F. D. Malliaros, "Semi-supervised learning and graph neural networks for fake news detection," in *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2019*, Aug. 2019, pp. 568–569. doi: 10.1145/3341161.3342958.

[42] J. Y. Khan, Md. T. I. Khondaker, S. Afroz, G. Uddin, and A. Iqbal, "A benchmark study of machine learning models for online fake news detection," *Machine Learning with Applications*, vol. 4, p. 100032, Jun. 2021, doi: 10.1016/j.mlwa.2021.100032.

[43] P. Meel and D. K. Vishwakarma, "HAN, image captioning, and forensics ensemble multimodal fake news detection," *Inf Sci (N Y)*, vol. 567, pp. 23–41, Aug. 2021, doi: 10.1016/j.ins.2021.03.037.

[44] M. Samadi, M. Mousavian, and S. Momtazi, "Deep contextualized text representation and learning for fake news detection," *Inf Process Manag*, vol. 58, no. 6, Nov. 2021, doi: 10.1016/j.ipm.2021.102723.

[45] P. Patwa *et al.*, "Fighting an Infodemic: COVID-19 Fake News Dataset," in *Communications in Computer and Information Science*, 2021, vol. 1402 CCIS, pp. 21–29. doi: 10.1007/978-3-030-73696-5_3.

[46] X. Li, P. Lu, L. Hu, X. G. Wang, and L. Lu, "A novel self-learning semi-supervised deep learning network to detect fake news on social media," *Multimed Tools Appl*, vol. 81, no. 14, pp. 19341–19349, Jun. 2022, doi: 10.1007/s11042-021-11065-x.

[47] H. Ahmed, I. Traore, and S. Saad, "Detecting opinion spams and fake news using text classification," *Security and Privacy*, vol. 1, no. 1, p. e9, Jan. 2018, doi: 10.1002/spy2.9.

[48] H. Ahmed, I. Traore, and S. Saad, “Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques,” in *Lecture Notes in Computer Science*, 2017, vol. 10618 LNCS, pp. 127–138. doi: 10.1007/978-3-319-69155-8_9.

[49] J. K. Burgoon, J. P. Blair, T. Qin, and J. F. Nunamaker, “Detecting Deception through Linguistic Analysis,” 2003.

[50] T. Loughran and B. Mcdonald, “When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks,” *Journal of Finance*, vol. 66, no. 1, pp. 35–65, Feb. 2011, doi: 10.1111/j.1540-6261.2010.01625.x.

[51] M. Hu and B. Liu, “Mining and Summarizing Customer Reviews,” 2004.

[52] M. Al-Shabi, “Evaluating the performance of the most important Lexicons used to Sentiment analysis and opinions Mining,” Jan. 2020.

[53] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” Jan. 2013, [Online]. Available: http://arxiv.org/abs/1301.3781

[54] X. Rong, “word2vec Parameter Learning Explained,” Nov. 2014, [Online]. Available: http://arxiv.org/abs/1411.2738

[55] J. Pennington, R. Socher, and C. Manning, “GloVe: Global Vectors for Word Representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Oct. 2014, pp. 1532–1543. doi: 10.3115/v1/D14-1162.

[56] D. Rohera *et al.*, “A Taxonomy of Fake News Classification Techniques: Survey and Implementation Aspects,” *IEEE Access*, vol. 10, pp. 30367–30394, 2022, doi: 10.1109/ACCESS.2022.3159651.

[57] A. Kesarwani, S. S. Chauhan, and A. R. Nair, “Fake News Detection on Social Media using K-Nearest Neighbor Classifier,” in *2020 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, 2020, pp. 1–4. doi: 10.1109/ICACCE49060.2020.9154997.

[58] M. Granik and V. Mesyura, “Fake news detection using naive Bayes classifier,” in *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, 2017, pp. 900–903. doi: 10.1109/UKRCON.2017.8100379.

[59] Y. Qi, “Random Forest for Bioinformatics,” in *Ensemble Machine Learning: Methods and Applications*, Y. Zhang Cha and Ma, Ed. Boston, MA: Springer US, 2012, pp. 307–323. doi: 10.1007/978-1-4419-9326-7_11.

[60] M. Y. Khan, A. Qayoom, M. S. Nizami, M. S. Siddiqui, S. Wasi, and S. M. K. U. R. Raazi, “Automated Prediction of Good Dictionary EXamples (GDEX): A Comprehensive Experiment with Distant Supervision, Machine Learning, and Word Embedding-Based Deep Learning Techniques,” *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/2553199.

[61] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, “A comparative analysis of gradient boosting algorithms,” *Artif Intell Rev*, vol. 54, no. 3, pp. 1937–1967, Mar. 2021, doi: 10.1007/s10462-020-09896-5.

[62] S. Kumari, D. Kumar, and M. Mittal, “An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier,” *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 40–46, Jun. 2021, doi: 10.1016/j.ijcce.2021.01.001.

[63] K. Patel *et al.*, “Facial Sentiment Analysis Using AI Techniques: State-of-the-Art, Taxonomies, and Challenges,” *IEEE Access*, vol. 8, pp. 90495–90519, 2020, doi: 10.1109/ACCESS.2020.2993803.

[64] S. Albelwi and A. Mahmood, “A framework for designing the architectures of deep Convolutional Neural Networks,” *Entropy*, vol. 19, no. 6, Jun. 2017, doi: 10.3390/e19060242.

[65] L. Alzubaidi *et al.*, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00444-8.

[66] R. C. Staudemeyer and E. R. Morris, “Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks,” Sep. 2019, [Online]. Available: http://arxiv.org/abs/1909.09586

[67] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[68] M. B. Kursa and W. R. Rudnicki, “Feature Selection with the Boruta Package,” *J Stat Softw*, vol. 36, no. 11, pp. 1–13, 2010, doi: 10.18637/jss.v036.i11.

[69] M. Huljanah, Z. Rustam, S. Utama, and T. Siswantining, “Feature Selection using Random Forest Classifier for Predicting Prostate Cancer,” in *IOP Conference Series: Materials Science and Engineering*, Jul. 2019, vol. 546, no. 5. doi: 10.1088/1757-899X/546/5/052031.

[70] B. F. Darst, K. C. Malecki, and C. D. Engelman, “Using recursive feature elimination in random forest to account for correlated variables in high dimensional data,” *BMC Genet*, vol. 19, Sep. 2018, doi: 10.1186/s12863-018-0633-8.