UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

MASTER'S THESIS

Natural language processing for scam detection. Classic and alternative analysis techniques

Author:
Ignacio PALACIO MARÍN

Supervisor:
David ARROYO GUARDEÑO

A thesis submitted in fulfillment of the requirements for the degree of

Msc in BigData and DataScience

September 28, 2019

“Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism.”

Dave Barry

UNIVERSIDAD AUTÓNOMA DE MADRID

Abstract

Escuela Politécnica Superior

Msc in BigData and DataScience

Natural language processing for scam detection. Classic and alternative analysis techniques

by Ignacio PALACIO MARÍN

We have seen, over the past decades, an overwhelming increase in the volumes of information being generated, distributed and shared, specially but not only, through social media networks. It has led to an exponential growth on the importance of data in the decision taking processes of most industries and economic sectors, proving the criticality of ensuring the quality of the information we are gathering and using.

Whilst most of this information is, or at least is intended to be, true, a non-negligible portion of it contains false information. Miss-information campaigns have played an important role in recent and critical decision-taking processes such as the Brexit referendum or the 2016 U.S. presidential elections.

The current spread of incorrect information constitutes a meaningful potential risk on information systems' management. This problem becomes even greater when considering decision taking automatic algorithms. As a matter of fact, social media and opened access to data may constitute a way to break the information's asymmetry that has traditionally affected areas such as the financial industry.

This paper will propose different techniques of natural language processing, from the more traditional ones to a brief approach to more recently developed techniques on deep learning approaches. They are all intended to enable an automatic texts' classification in different discussion forums and constructing procedures to pursue users or groups of users' classifications as an open gate to generate attribution procedures for information sources.

Acknowledgements

I would first like to thank my thesis advisor David ARROYO GUARDEÑO. His office door was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work, but steered me in the right direction whenever he thought I needed it.

Finally, I am pleased to express my very profound gratitude to my wife and loving friend, Luisa-Pilar, for her unfailing support and continuous coverage through my years of study and the process of researching and writing this thesis. This accomplishment would not have been possible without her. Thank you.

Ignacio PALACIO MARÍN

Contents

Abstract	v
Acknowledgements	vii
1 Introduction	1
1.1 Motivation	1
1.1.1 False information taxonomy	2
1.1.2 Regulatory concern	3
1.2 Objectives	4
1.3 Thesis organization	5
2 Strategies for campaigns against fake news	7
2.1 State of the art	7
2.1.1 Following the path of flawed information: network analysis and hub characterization	7
2.1.2 Content based verification	9
2.1.3 Detection using computational methods	10
2.2 Our proposal's approach	13
2.2.1 Text structure	13
2.2.2 What? How?	15
3 Proposed techniques: Conventional NLP approach	17
3.1 As a consequence of the <i>"no free lunch" theorem</i>	17
3.2 Methodology	17
3.2.1 Token based analysis	21
3.2.2 Embedding based analysis	22
3.3 Testing	23
4 Proposed techniques: Advanced deep learning techniques	27
4.1 Methodology	27
4.1.1 Long Short-Term Memory (<i>LSTM</i>) neural networks	28
4.1.2 Gated Recurrent Units (<i>GRU</i>) neural networks	29
4.2 Testing	30
5 Conclusions and future work	33
5.1 Conclusions	33
5.2 Future work	34
A Topic modelling visualization	35
B Implemented algorithm architecture diagram	37
C Classification algorithms parametrization	39

D LSTM and GRU networks underlying equations	43
D.1 LSTM hidden layer equations	43
D.2 GRU equations	43
Bibliography	45

List of Figures

1.1	Fake news - Google Trends	1
1.2	Bloomberg - Dow Jones industrial average index	3
2.1	Polifact - Barack Obama's file	9
2.2	Twitter dataset - Stories content distribution	14
2.3	Two labeled Twitter dataset - Ordered <i>countplot</i>	14
2.4	Two labeled Twitter dataset - Stories content distribution	14
2.5	Cloud representation - Stories lemmatization content representation	15
3.1	<i>scikit-learn</i> - Machine learning algorithms cheat sheet	18
3.2	Traditional NLP techniques - <i>Accuracy</i> and <i>roc_auc</i> scores	25
4.1	AI - A back-of-the-envelope explainer. <i>Karen Hao</i>	27
4.2	Greff et al. - <i>LSTM</i> unit	28
4.3	<i>LSTM</i> vs <i>GRU</i> units	30
4.4	Twitter stories length distribution	30
4.5	<i>LSTM</i> : <i>Loss/Accuracy</i> scores - <i>sigmoid</i> activation function	31
4.6	<i>LSTM</i> : <i>Loss/Accuracy</i> scores - <i>linear</i> activation function	31
4.7	<i>GRU</i> : <i>Loss/Accuracy</i> scores	32
5.1	Obtained <i>accuracy</i> and <i>roc_auc</i> scores.	33
5.2	Obtained <i>precision</i> , <i>recall</i> and <i>F1</i> scores.	34
A.1	Coherence score - Topic classification	35
A.2	pyLDAvis - Topic visualization	36
B.1	Text classification proposed algorithms architecture	37

List of Tables

2.1	Zannettou <i>et al.</i> [84] - On the detection of false information	10
2.2	Yang <i>et al.</i> [79] - Test set <i>accuracy</i> results by step	12
3.1	<i>Random Forest</i> - Grid searched parameters	19
3.2	<i>k-NN</i> - Grid searched parameters	19
3.3	<i>SVM</i> - Grid searched parameters	20
3.4	<i>XGBoost</i> - Grid searched parameters	20
3.5	Traditional NLP - ML <i>accuracy</i> and <i>roc_auc</i> scores	24
4.1	<i>LSTM</i> - Grid searched parameters	31
C.1	<i>TF-IDF</i> - Classification algorithms parameterization	39
C.2	<i>Word2Vec</i> - Classification algorithms parameterization	40
C.3	<i>Doc2Vec</i> - Classification algorithms parameterization	41

List of Abbreviations

NLP	Natural Language Processing
NLI	Natural Language Inference
IR	Information Retrieval
AI	Artificial Intelligence
AI HLEG	Artificial Intelligence High-Level Expert Group
CCN-CERT	Centro Criptológico Nacional - Computer Emergency Response Team
HBPT	Homogeneity-Based Transmissive Process
TLV	Tweet Latent Vector
OSN	Online Social Networks
FNC	Fake News Challenge
TF-IDF	Term Frequency – Inverse Document Frequency
LD	Linear Discriminant
RF	Random Forest
SVM	Support Vector Machine
k-NN	k-Nearest Neighbours
XGBoost	Extreme Gradient Boosting
ROC	Receiver Operating Characteristic
AUC	Area Under Curve
TPR	True Positive Rate
FPR	False Positive Rate
MLP	Multi Layer Perceptron
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
SA-LSTM	Sequence Autoencoder Long Short-Term Memory
C-LSTM	Contextual LSTM
HAN	Hierarchical Attention Network
Conv-HAN	Convolutional Hierarchical Attention Network
BoW	Bag of Words
GRU	Gated Recurrent Units
BERT	Bidirectional Encoder Representations (<i>from</i>) Transformers
LDA	Latent Dirichlet Allocation
LSA	Latent Semantic Analysis

Chapter 1

Introduction

1.1 Motivation

Currently data is a critical fuel to the prevailing economic and social systems. It has definitively become a commonplace to refer it as the ‘new oil’ of the global economy [18]. Therefore, techniques and infrastructure to store, process and analyse it are as well in the core of most of the latest technology trends.

Decision making processes, at governmental, enterprise and even personal levels, are increasingly sustained on a permanently growing amount of data, leading, on an accelerating pace, to a data-driven world.

Organizations of all kind now have stockpots of raw data along with sophisticated analysis tools that combined drive their decision-making processes when trying to create new market opportunities. Decisions are no longer made based on gut instinct but on pure evidence, or so they expect to be. When analysing the evolution of companies’ investments and strategies over the past decade big bets on data and analytics are quite common and almost obliged.

The increasing concern over the recent years as well as the importance of fake news is easily understood when addressing *Google trends*, see *Figure 1.1*, to evidence how fake news has gained popularity worldwide.

FIGURE 1.1: Fake news. Source: *Google Trends*

The wide spread of fake news may have a negative impact on society and individuals. An increasing exposure, and therefore desensitization, to fake news definitively affects the way people approach and respond to real news and undermine their capability to separate what is true from what is not. It is important to understand that even though fake and captious news have existed from long ago we are witnesses to an overwhelming increase tied up with the increase of information we are exposed to.

Furthermore, there is overall consensus that good data is important for any analysis but there is no independent definition of what constitutes good data and what does not [25].

1.1.1 False information taxonomy

A clear description of every faced problem is always solid ground over which start discussing and proposing new approaches. Fake news won't be different in that matter.

Zannettou *et al.* proposed, on their study of false information on online social networks [84], a taxonomy of the Web's false information ecosystem, based on [73] and extended. A brief description of this classification is detailed bellow.

Based on the types of false information Zannettou *et al.* distinguished between:

1. **Fake news**, identifying the following types of false information within it:
 - *Fabricated*, or completely fictional stories.
 - *Propaganda*, aiming to hurt the interest of a particular party or promote those from another.
 - *Imposter*, news whose author or source has been impersonated.
 - *Conspiracy theories*, addressing an event or set of circumstances invoking a conspiracy without further evidence.
2. **Biased/Inaccurate news**, as to news that are misleading but do reflect truth in some extent:
 - *Hoaxes*, containing false or inaccurate facts being presented as legitimate.
 - *Hyper-partisan*, referring to stories that are extremely one-sided.
 - *Fallacy*. News making invalid reasoning when constructing an argument.
3. **Misleading/Ambiguous news**, including:
 - *Rumors*, widely disseminated news with no discernible source and without known authority for its truth.
 - *Click-bait*, or the deliberate use of misleading headlines.
 - *Satire news*, where its' main concern is the irony or humor touch they are draw up with, usually obfuscating the facts.

They also elaborate on the taxonomy of false information actors, identifying a hand-full of different players such as *bots*, *criminal/terrorist organizations*, *political persons*, *hidden paid posters* and *state-sponsored trolls*, *journalists*, *trolls* and *useful idiots*. Being the latest those users who share fake news because they are manipulated or because they are naive.

Besides, they proposed as well an assortment of the underlying motives for the propagation of false information, including, *malicious intent*, *political influence*, *profit*, *passion* and *fun*.

1.1.2 Regulatory concern

The outstanding rising of false news is not only a problem of miss-information but it has become an increasing concern for a growing number of actors and may be considered as one of the greatest threats to journalism and, what is even more worrisome, to democracy itself [88]. Examples of such are the 2016 U.S. presidential election campaign, where top twenty fake election stories generated above 8M interactions on Facebook [86][5][28], and the 2016 Brexit referendum, where the *Digital, Culture, Media and Sport Committee*, appointed by the *House of Commons*¹ concluded that "the UK is clearly vulnerable to covert digital influence campaigns" [34].

Furthermore, even though miss-information and financial markets has been a long run problem for a while, the raising of the fake news phenomenon has taken it to a new level. In 2013, over \$130 billion, see *Figure 1.2*, were wiped out after a 'bogus' tweet suggesting that Barack Obama had been injured in an explosion [57].

FIGURE 1.2: Source: Bloomberg - Dow Jones industrial average index, April 23rd 2013 plunge.

As a matter of fact, the erosion to democracy, and public trust posed by the unprecedented growth in miss-information has led to a sustained demand of further analysis of this phenomenon [87], as to newer regulation at all levels [34].

Moreover, legislation recently sanctioned, in both Germany and France [34], come up as meaningful examples of local regulation against harmful online content. Both examples constitute successful demonstrations as to how local Governments can enforce rules empowering judges to order the immediate removal of disinformation online articles. Paraphrasing the French formula, both regulations seek "information that is fair, clear and transparent" on personal data usage sanctioning actions like "deliberately disseminate false information likely to affect the sincerity of the ballot".

Earlier this year, the European Commission High-Level Expert Group on Artificial Intelligence (AI HLEG) delivered the Policy and investment recommendations for trustworthy Artificial Intelligence [1], establishing as a point of special criticality the

¹ Lower house of the Parliament of the United Kingdom.

study of the relationship between AI and Cybersecurity to develop and support AI-specific cybersecurity infrastructures to leverage Europe's enablers for trustworthy AI. It does state that "user centric cybersecurity policies must protect users' rights online, by design and by default".

Additionally, the AI HLEG dedicates a full chapter to "*Establishing an appropriate governance and regulatory framework*" encouraging the definition of an appropriate framework, ensuring and respecting fundamental rights, in order to safeguard individuals and society. Finally, among the fostered measures presented in their conclusions they promote the "data-driven economy as a cornerstone for the EU's future success in the global market".

Finally, local Spanish authorities such as *Centro Criptológico Nacional (CCN-CERT)*, recently published as well a reference guide [6] to help addressing the main features and methodology of the current disinformation techniques convening citizens and digital media users to acquire the necessary tools and skills so they can consume and share information avoiding unwillingly spreading miss-information. Recently, attacks to a country's interest have migrated from targeting their IT infrastructure or communications systems, corporate or governmental, into trying to undermine an essential gear of modern democracy, public opinion.

As a matter of fact, according to the *CCN-CERT* guide [6], there are, at least, six factors contributing to the increasing frequency of miss-information campaigns:

- High level of effectiveness, along with limited costs and broad diffusion.
- Difficulty establishing a direct attribution of the miss-information campaigns. Anonymous profiling, automation messages distribution processes, software and technology helping to cover trace and IP addresses, etc. highly limit the tracing of the information and its sources as well as the reaction capability to establish a direct attribution of the attacks.
- Complex regulation as in opposition to classical opened battlefield war.
- Causal relationship establishment limitations.
- Usage of existing social vulnerabilities.
- Infiltration of illegitimate misinformation in legitimate social and political communication channels, challenging the audience capability of separating fact from fake and discern what opinions and information among those being distributed are or not legitimate.

The *CCN-CERT* guide suggests as well a decalogue of recommendations to "*communication war victims*": analyzing the information sources, being aware of distributed screenshots, analyzing who shared a certain piece of information and in what context, detecting bots or multiple-profile actors, read into the details, enhance critical thinking and conscience of the impact of individual actions, etc.

1.2 Objectives

This thesis intent is to present a broad comparison of various machine learning and deep learning techniques trying to identify fake news from a sample of texts. Further

detail will be presented in future chapters, nevertheless we depict in this section an general overview of the developed implementations and analysed framework.

Using the already cleaned and tagged Twitter based dataset used by Jooyeon Kim *et al.* on their HBPT² proposal [42], a road-map from the more traditional natural language processing, hereinafter NLP, techniques to newer deep learning implementations such as LSTM³ neural networks is presented.

It is not within the objectives of this master thesis to provide a deeper extent or profound analysis of any of the proposed classification algorithms, nor of the used analysis techniques.

1.3 Thesis organization

This master thesis memorandum is opened into the following chapters:

- *Strategies for campaigns against fake news.*
 - *State of the art*, a critical review in the context of the presented work and related issues.
 - *Our proposal's approach*, as to the different approaches presented on the *State of the art*, this section introduces this thesis implemented approach.
 - * *Text structure*, an overview on the used data-set.
 - * *What and how*, further detail on the proposed approach.
- *Proposed techniques: Conventional natural language processing approach.*
 - *As a consequence of the "no free lunch theorem"*, referencing the ideas presented by David Wolpert and William Macready [76] [75] where they proposed that "any two optimization algorithms are equivalent when their performance is averaged across all possible problems".
 - *Methodology*, presenting a description and justification of the methods and approaches used.
 - *Testing*, detail of the performed tests and obtained results.
- *Proposed techniques: Advanced deep learning techniques.*
 - *Methodology*, see above.
 - *Testing*, see above.
- *Conclusions*, presenting a wider perspective including further implications of the obtained results and future work on the matter.

² Homogeneity-Based Transmissive Process.

³ Long short-term memory.

Chapter 2

Strategies for campaigns against fake news

2.1 State of the art

As to the more recent and meaningful research made over the classification, propagation and cascade patterns, analysis or separating algorithms for fake news we will present hereafter a carefully selected picture of the existing papers on the matter that have enlightened this thesis.

Furthermore, Zannettou *et al.* provided a detailed overview of the most relevant research papers following their own classification proposal, on their study of false information on online social networks [84], providing great understanding on users' perceptions on false information, its' propagation, detection and containment, and a review the most relevant studies of fake information on the political stage.

The fake news phenomenon does include, in fact, miss-information, dis-information and mal-information, along with all their modalities [74].

2.1.1 Following the path of flawed information: network analysis and hub characterization

There is a quite a long history regarding the research on the very foundations of miss-information. There are many theories as to why the very nature of them is so rooted on our history. From the application of the early theories of Eric Fromm [21] where he explored humanity's relationship with freedom, to the confirmation bias theory¹ [52], and Barabasi's proposal [4], considered, in fact, the very origin of network analysis studies. However, people actually are not keen on separating fact from fake, only about 70% to 75% of humans are actually able to effectively identify fake news [54] on a broader context [59].

The detail presented hereinafter regarding the dishonest information propagation phenomenon will be presented by means of its application to the fake news problem in hand.

Moreover, Del Vicario *et al.* provided evidence on how users tend to gather themselves in communities by interest causing reinforcement on their opinions and shared

¹ Seeking or interpreting of evidence in ways that are partial to existing beliefs, expectations, or a hypothesis in hand.

content, and fostering confirmation bias, segregation and polarization at the expense of the quality of the shared information leading to the proliferation of biased narratives [16].

Recent work on the differences between true and false information in social networks by Vosoughi *et al.* based on their classification by topics, cascade patterns and sizes, diffusion speeds, emotion, and sentiment [72], leading to some meaningful conclusions including that false information is diffused significantly farther, faster, deeper, and more broadly. As to the spreading pattern of true and false news in social media networks [78], early-fact-check [68] and building fact checking systems [69] there are multiple academic papers demonstrating the significant progress made over the recent years.

On the role of user's profiling, i.e. node and hub analysis, and its' relationship with fake news Raymond S. Nickerson [52] presented a deep dive on its understanding, demonstrating that user's features can provide a key stone on helping to detect fake news. When considering the profiling option on the analysis, usually both *explicit* and *implicit* features are taken into account [64].

Extended examples of explicit users' features analysis may be credibility classification or identity linkage. In terms of credibility assessment, news-worthy topics can be separated from others by analyzing users' explicit features such as the extension and amount of previously written messages by user [8]. As to identity linkage, users' unique behavioral patterns and users' information redundancies due to these behavioral patterns can be tied up together [83].

On the other hand, implicit features can be exploit to, for example, attribute age from writing style, even predicting gender or personality as in Schwartz *et al.* [61], where from Facebook messages and through personality tests they identified variations in language by characterizing personality, gender and age. Finally, political bias can also be deduced [11].

Comprehending the diffusion process of fake news is another key aspect to understanding the phenomenon and, obviously, social media networks have a great share on it. Information's diffusion is in the core of the very nature of social media, however, inflammatory and emotional news spread faster than real news. Therefore, understanding their mushroom on social networks may have an impact on addressing their detection [53].

Furthermore, Shao *et al.* studied on [63], by means of 14 million messages spreading 400 thousand articles on Twitter during and following the 2016 U.S. presidential elections, the role that social bots play on the diffusion of fake news, providing evidence of the unequal role they play on the process. Altogether with how vulnerable we are to miss-leading information and manipulation suggests that restraining social bots may be an effective countermeasure.

Propagation based methods, for fake news detection, have gained popularity over the past years. For example, Zhiwei Jin *et al.* proposed a hierarchical propagation model based on user posts and considering credibility propagation through a three-layer credibility network to detect fake news [38].

Further use of propagation based methods include the application of epidemiological modeling techniques. Fang Jin *et al.* proposed on [37] a characterization of tweets diffusion though a comparison of different epidemiological models to characterize

propagation cascades. Finally, statistical modeling has also been used when analyzing the propagation of fake news on online social networks.

2.1.2 Content based verification

There are as well, investigations on the linguistic side of fake news, demonstrating the positive impact of fact-checkers [36]. As a matter of fact, big platform companies such as Google or Facebook have increased their reliance on fact-checking as a response to the raising on miss-information. They have both deployed fact-checking services to help determine the veracity of factual statements in social media [12][26].

Broadly speaking fact-checking techniques may be divided into expert-based, crowd-sourced and automatic fact-checking systems [87]. While expert-based approaches rely on domain-experts, are easy managed and often lead to high-quality results. They are, however, costly and hardly scalable. Actually there are quite a number of websites allowing expert-based fact-checking. See *Polifact* caption on Figure 2.1.

FIGURE 2.1: Polifact fact-checking website - B. Obama's file.

On the other hand, crowd-sourced fact-checkers rely instead on large populations of regular individuals. This alternative is deeply connected with the concept of *collective intelligence* by means of the use of collaboration, collective efforts and competition of many individuals to produce a consensus decision making. They are however, when compared with expert-based systems, difficult to manage, less credible and accurate. Unlike expert-based, crowd-sourced fact-checking websites are still in early stages. This systems do rely, as a matter of fact, on the concept of "*wisdom-of-the-crowds*", however "*stupidity-of-the-herds*" is still a challenge they face, actually limiting their effectiveness [82].

However, traditional approaches based on human verification do not scale to the volume generated in online social networks, thus this requires the development of new computational techniques to complement expert verification.

In any case, the role of the human being in news' fact-checking processes is still open and under discussion. Opened discussion schemes are *human-in-the-loop*, *human-on-the-loop* and *human-out-of-the-loop*, terms in fact used on military operation strategy [65], but increasingly applied on *civil* contexts.

2.1.3 Detection using computational methods

Automatic detection techniques have been developed to address the lack of scalability of manual fact-checking systems. Actual developments rely, mostly on Information Retrieval (IR) and Natural Language Processing (NLP) techniques, as well as on graph theory and other models/techniques.

Zannettou *et al.* collected and presented a well detailed and quite extensive summary of the works addressing the false information detection problem [84]. Papers presented in such table considering only Twitter and Facebook are detailed on Table 2.1 bellow.

Social Network	Machine learning	Deep learning	Other algorithms
Twitter	Castillo <i>et al.</i> [8]	Volkova <i>et al.</i> [70]	Resnick <i>et al.</i> [58]
	Gupta <i>et al.</i> [29]		Vosoughi <i>et al.</i> [71]
	Kwon <i>et al.</i> [45]		Jaho <i>et al.</i> [35]
	Yang <i>et al.</i> [80]		Qazvinian <i>et al.</i> [56]
	Liu <i>et al.</i> [47]		Zhao <i>et al.</i> [85]
	Wu <i>et al.</i> [77]		Farajtabar <i>et al.</i> [19]
	Gupta <i>et al.</i> [30]		Kumar <i>et al.</i> [43]
	AlRubaiyan <i>et al.</i> [2]		
	Hamidian <i>et al.</i> [31]		
	Giasemidis <i>et al.</i> [24]		
	Kwon <i>et al.</i> [44]		
Facebook	Tacchini <i>et al.</i> [67]		
	Conti <i>et al.</i> [13]		

TABLE 2.1: Zannettou *et al.* [84] - On the detection of false information

As the brief overview on the loads of existing papers on the topic provided by the table above demonstrates, miss-information detection is not a straightforward task. There are, however, a number of metrics and approaches to the problem. A great number of the existing approaches try to use machine learning techniques when applying computational methods to the problem in hand. The difficulty of implementing effective *human-on-the-loop* schemes lies in the attacker's ability to carry out counterintelligence dynamics, that is, adversary procedures and characterization/detection evasion techniques.

Hereinafter, we will provide an overview on the recent studies by walking through the references proposed by Zannettou *et al.*

On the referenced papers applying machine learning techniques on social network analysis for fake news detection, Castillo *et al.* [8] classified a number of tweets with *precision* and *recall* scores in the range from 70% to 80% by means of conventional machine learning techniques (*SVM*, *Decision Trees*, *Decision Rules*, and *Bayes Networks*).

Gupta *et al.* [29] adopted a supervised machine learning and relevance feedback approach prominent content based and user based features to rank tweets according to their credibility score. Kwon *et al.* [45] classified rumors with high *precision* and *recall* results in the range of 87 to 92% by taking to exam temporal, structural and linguistic aspects of diffusion by traditional machine language techniques (*Decision Trees*, *Random Forest* and *SVMs*). Yang *et al.* [80] combined bursty term identification and multi-dimension sentence modeling, working along with conventional machine learning techniques (*Naïve Bayes*, *Logistic Regression* and *Random Forest*), to automatically detect emerging hot topics for rumor identification.

Furthermore, Liu *et al.* [47] proposed a real time rumor debunking algorithm, using *SVMs*. Wu *et al.* [77] analyzed whether knowledge learned from historical data could potentially help identify newly emerging rumors. Gupta *et al.* [30], presented a semi-supervised tweets credibility scoring model using *TweetCred* real-time. Moreover, AlRubaian *et al.* introduced a multi-stage credibility framework using *Naïve Bayes* classifier obtaining *accuracy* 90.3%, 86.24% *precision* and 98.8% *recall*.

Finally, on the Twitter based references, Hamidian *et al.* [31] adopted a supervised rumor classification employing the *Tweet Latent Vector (TLV)* feature, creating a 100 dimension vector representing each tweet, achieving a 97.2% *precision* with a *SVM tree kernel* model. Giasemidis *et al.* [24] aimed, over 80 trustworthiness measures from 72 identified rumours, to produce trustworthiness scores for classifying tweet with a maximum *accuracy* of 96.6%. Kwon *et al.* [44] studied the classification performance levels over varying time windows—from the first three days to nearly two months, providing insight into the cumulative spreading patterns of rumors over time.

As to those referred papers on Facebook data-sets based on machine learning analysis, Tacchini *et al.* [67] showed that Facebook post can be highly accurately classified based on the users who "liked" them, obtaining classification *accuracy* over 99%. Additionally, Conti *et al.* [13] focused on Online Social Network (OSN) structural properties of the information cascade as they are inherently difficult to be manipulated obtaining, on a highly imbalanced dataset and using a total of 28 features over three distinct classifiers (*Linear Discriminant (LD)*, *Random Forest (RF)*, *Multi-Layer Perceptron (MLP)*) classification *F-score* never exceeded 0.7.

Further work on machine learning approaches to fake news detection, even if not strictly applied to OSN and therefore not referred by Zannettou *et al.* [84], is hereafter introduced. Khan *et al.* [41] conducted a benchmark study assessing the performance of different machine learning algorithms (*SVM*², *Linear regression*, *decision trees*, *AdaBoost*, *Naïve Bayes* and *k-NN*³), extending their research into neural network-based and deep learning models (*CNN*, *LSTM*, *Bi-LSTM*, *C-LSTM*⁴, *HAN*⁵, *Conv-HAN*⁶ and *Char-level C-LSTM*) as well.

On the other hand, as to the papers presenting deep learning approaches to OSN fake news detection, Volkova *et al.* [70] built predictive deep learning models to predict four sub-types of suspicious news: satire, hoaxes, clickbait and propaganda,

² SVM stands for Support Vector Machines.

³ k-nearest neighbours also referred as k-NN.

⁴ C-LSTM or Contextual LSTM networks.

⁵ HAN stands for Hierarchical Attention Network.

⁶ Conv-HAN, Convolutional Hierarchical Attention Network.

the implemented recurrent and convolutional neural networks exhibit significant performance on making such predictions.

Further work on deep learning approaches to fake news detection, even if not strictly applied to OSN analysis but still meaningful, is hereafter detailed. Davis *et al.* [15] introduced in the 2017 FNC-1⁷ four neural network models, two using a feed-forward architecture (Concatenated multi-layer perceptron (*Concat MLP*) and *Bag-of-words multi-layer perceptron (BoW MLP)*) and two using a recurrent architecture (*Dual GRU* and *Bi-directional concatenated and stacked LSTM (Bi-dir LSTM)*), achieving categorical test-set accuracy of 93%. Price *et al.* [55] focused their proposal on the exploitation of *LSTM* for classification of news by comparing the performance of a basic *LSTM* network against a *sequence autoencoder LSTM network (SA-LSTM)* obtaining accuracy scores on the test set of 78.74% and 70.61% respectively.

Besides, Mrowca *et al.* [51] approached the problem by feeding headlines into a *Bi-directional LSTM* and concatenate the output with headlines' global statistical features achieving an overall score of 87.4% and a mean *F1 score* of 69.5%. Samir Bajaj [3] built a content based classifier, exploring several architectures (*logistic regression*, *two-layer feedforward neural network*, *recurrent neural network (RNN)*, *LSTM* and *GRU networks*, *Bi-dir LSTM*, *Convolutional Neural Network (CNN) with Max Pooling* and *Attention-Augmented Convolutional Neural Network*) obtaining precision scores ranging from 0.87 (*CNN with Max Pooling*) to 0.97 (*CNN with Max Pooling and Attention*), recall scores from 0.003 (*CNN with Max Pooling and Attention*) to 0.79 (*GRUs*) and *F1* scores from 0.06 (*CNN with Max Pooling and Attention*) to 0.84 (*GRUs*).

Additionally, there are quite a number of papers discussing the latest advances on text processing applied neural networks. Hereafter some examples of *BERT*⁸ implementation applied to fake news detection papers are presented. Yang *et al.* at [79] treated the tasks of natural language inference (NLI) training a number of NLI models including *BERT* achieving test set accuracy scores of 88.06% by blending predictions from first level ensembled models (using *LightGBM* and densely connected feed-forward network) along with *BERT* networks and the took those as soft pseudo-labels and used them to fine-tune the pre-trained NLI models, using the output to re-feed a second level ensemble (*LightGBM* and a multi-layer perceptron) who's output is connected to a fine tuned *BERT* with a final bleeding and taking into account the transitive relations on the used data.

Model	Accuracy
Ensemble (1)	0.86741
BERT (1)	0.86689
Blended (1)	0.86963
Ensemble (2)	0.87990
BERT (2)	0.87484
Blended (2)	0.88019
Blended (2) + Transitivity	0.88063

TABLE 2.2: Yang *et al.* [79] - Test set accuracy results by step

⁷ (FNC-1), Fake News Challenge, is a public competition that aims to find automatic methods for detecting fake news.

⁸ BERT, Bidirectional Encoder Representations from Transformers neural networks.

The detailed accuracy results presented by Yang *et al.* (see Table 2.2) for every of the aforementioned steps are detailed in order to evidenced the shocking and compromising relationship between accuracy gain and complexity increase. Furthermore, Table 2.2, demonstrates the implementation effort and complexity increase required to gain not even a 2% in test set accuracy results.

2.2 Our proposal's approach

This thesis presents a walk-through and comparison over different fake news detection approaches. It's intention does not lay on introducing novel approaches in fake news detection problems but instead present a narrative and juxtaposition over some of the main existing techniques.

The implementation proposed on this thesis uses the cleaned dataset provided by Kim *et al.* on [42], previously used in rumor detection researches [48][49][50], taken from the sources shared on their project's [GitHub](#)⁹. It contains fact-checked Twitter rumors and the referenced articles within them. This thesis does not consider the data extraction step but instead relies on already cleaned data for its purposes.

Based on that source a three way path is proposed:

1. Implementation of traditional NLP techniques, more precisely *TF-IDF*¹⁰.
2. Implementation of embedding based classification architectures. *Word2Vec* and *Doc2Vec* embedding algorithms are put to test.
3. Use of deep learning algorithms for text classification purposes.

News are previously treated applying standard NLP methods when required such as lemmatization, *stop words* removal, etc. And four machine learning classification algorithms (*Random Forest*, *k-Nearest Neighbours*, *SVM* and *XGBoost*) where applied on both the first two proposed paths (*TF-IDF*, *Word2Vec* and *Doc2Vec*).

Hereinafter we provide first, a full description of the used data (section 2.2.1), and then a detailed description of the implemented architecture (section 2.2.2).

2.2.1 Text structure

As previously stated, this thesis uses the cleaned dataset provided by Kim *et al.* on [42], previously used in rumor detection researches [48][49][50]. From the scratch Twitter information, Kim *et al.* pre-processing included the exclusion of news whose content could not be retrieved with the Twitter API, cleaning the stories cascades and confirming that there were no "circular" retweets.

After all that processing, the original dataset contains, 1.196 tweets, and for each one, (1) the tweet ID, (2) label, being either true, false, non-rumor or unverified, (3) the tweets' text, (4) news' story URL, (5) news' story title, (6) news' story content and (7) news' story content size.

⁹ GitHub - [dongkwan-kim/HBTP](#)

¹⁰ Term frequency – Inverse document frequency (TF-IDF).

Figure 2.2 presents the dataset's distribution. It plots the used Twitter stories content length as a function of their fact-checked label.

FIGURE 2.2: Cleaned dataset stories' content distribution by label.

The cleaned dataset is divided into 371, 289, 274 and 262 non-rumor, true, unverified and false stories, see Figure 2.3. For the purposes of this thesis it actually made more sense to reduce the existing labels into just two categories. *Unverified* tweets were discarded and *true* and *non-rumor* labels were tied together into a single category.

FIGURE 2.3: Reassigned Twitter stories by fact-checked category.

Find below, on Figure 2.4, the plotted distribution of the used Twitter stories once labels have been reassigned as described. Reducing the number of labels definitively eases the analysis on the plot relationship.

FIGURE 2.4: Reassigned label dataset - Content distribution by label.

Once label cleaned, the resulting dataset is divided into 660 and 262 true and false

stories. See *Figure 2.3*. Even not extreme, a bias on the used dataset is also clear when labels are cleaned. Data is stripped into 28.4% and 71.6% of false and true stores which has to be taken into account when analyzed and classified. Furthermore, a trend on the relation between label and content size can be drawn based on the previous plots. Apparently true stories tend to be larger than false ones.

Based on this data the architecture described on Section 2.2.2 is implemented to classify them and try to separate false news based on an automatic information process.

Further analysis on the data can be performed from visualizing for each story their content based on the relative importance of each appearing lemma, see *Figure 2.5*, to a feature extraction or an LDA¹¹ modeling visualization. Besides, further detail on LDA based topic modeling uncovering the hidden topics on the dataset is provided on *Appendix A*.

FIGURE 2.5: *Cloud representation* - Lemmatized stories content.

As per topic modeling, in the context of NLP, is described as a method of uncovering hidden structure on a corpus. It is useful for text classification, recommendation systems and uncovering themes. *Figure 2.5*, provides a cloud representation of a topic model.

2.2.2 What? How?

This section describes this thesis proposed texts' classification architecture. It is distinctively splitted into three identifiable pieces: NLP analysis pre-processing, NLP traditional techniques combined with machine learning classification algorithms and a deep learning approach.

Our proposal develops the NLP analysis pre-processing of the Twitter stories data based on the *SpaCy* NLP toolkit. *SpaCy* is an open-source software library for advanced NLP published under the MIT licence. It handles many tasks commonly associated with building and end-to-end NLP pipeline, it is written in optimized Cython¹² being the fastest syntactic parser available.

¹¹ Latent Dirichlet Allocation (LDA)

¹² Programming language written in Python and C, designed to give C-like performance. It is a compiled language that is typically used to generate CPython extension modules.

On the text processing stage, it is lemmatized based on the SpaCy implementation and its' available large English [dictionary](#)¹³. Stop words, punctuation and spaces removal processes are also applied. All along with lemmatization, they are both common to all classification processes implemented.

On the other hand of NLP implemented processes, construction of bigrams and tri-grams is only applicable to a subset of the classification techniques. Bigrams are two words, three respectively for trigrams, occurring together in a story piece. *Gensim's* phrases model can build and implement bigrams, trigrams, quadgrams and more. To apply phrase modeling the following implementation was considered, to determine whether two tokens A and B constitute a phrase.

$$\frac{\text{count}(A \setminus B) - \text{count}_{\min}}{\text{count}(A) * \text{count}(B)} * N > \text{threshold}$$

where,

$\text{count}(A)$ and $\text{count}(B)$ are the number of times tokens A or B appear in the corpus.

$\text{count}(A \setminus B)$ is the number of times the tokens $A \setminus B$ appear *in order*.

N is the total size of the corpus vocabulary.

$\text{count}_{\min}$ and threshold are user-defined parameters ensuring, the first, a minimum number of occurrences on accepted phrases, and the second, control on how strong the relationship between two tokens is required to be.

Bigrams (or digrams) help provide the conditional probability of a token given the preceding one, when the relation of conditional probability is applied. Bigrams and tri-grams were in fact implemented and 4-grams, even considered and their implementation verified, did not provided additional meaning to our corpus.

Full pre-processed stories; including lemmatization, removal of stop-words, punctuation and spaces, and construction of bi-grams and tri-grams; is feed into a *IT-IDF* implementation, converting the text for each document into vector models on the basis of occurrence of the words in them. This vectors are them used as input for four machine learning classification algorithms: *Random Forest*, *k-Nearest Neighbours*, *SVM* and *XGBoost*. An analogous process is carried out, but excluding the implementation of bi-grams and tri-grams, to apply *Word2Vect* and *Word2Vect* embedding algorithms as input to the classification processes instead of *IT-IDF*. See [Chapter 3](#) for further detail.

Additionally, a deep learning based implementation was also put up to test including both *LSTM* and *GRU* neural networks. Further detail is provided on [Chapter 4](#).

¹³ *en_core_web_lg* pre-trained English statistical models. English multi-task CNN trained on OntoNotes, with GloVe vectors trained on Common Crawl.

Chapter 3

Proposed techniques: Conventional NLP approach

3.1 As a consequence of the "no free lunch" theorem

In mathematical *folklore* the term "no free lunch" theorem arises from referencing the ideas presented by David Wolpert and William Macready in [76] [75] where they proposed that "any two optimization algorithms are equivalent when their performance is averaged across all possible problems". Besides, Shalev-Shwartz *et al.* work on [62] is also referred in some cases as "no free lunch" theorem.

THEOREM (No-Free lunch, Shalev-Shwartz *et al.* proposal): *Let A be any learning algorithm for the task of binary classification with respect to the 0 – 1 loss over a domain χ . Let m be any number smaller than $|\chi|/2$, representing a training set size. Then, there exists a distribution $\mathcal{D}$ over $\chi \times \{0, 1\}$ such that:*

1. *There exists a function $f : \chi \rightarrow \{0, 1\}$ with $L_{\mathcal{D}}(f) = 0$*
2. *With probability of at least $\frac{1}{7}$ over the choice of $S \sim \mathcal{D}^m$ we have that $\mathcal{L}_{\mathcal{D}}(A(S)) \geq \frac{1}{8}$.*

Which states, as defined on [62] that *for every learner, there exists a task on which it fails, even though that task can be successfully learned by another learner.*

Following that interpretation, this thesis proposes not one but four machine learning classification algorithms (*Random Forest*, *k-Nearest Neighbours*, *Support Vector Machine* and *XGBoost*) to address every proposed implementation. Even though, we do not know how to construct the optimal classifier for every given problem, prior knowledge of it may enable a suitable approximation, and therefore on every proposed case, well known algorithms have been tested for all cases.

3.2 Methodology

All hereinafter proposed implementations are preceded by the NLP corpus pre-processing detailed on [Section 2.2.2](#) and taking into consideration the order, limitations and specifications as described on that section.

As an overview, to ease the comprehension of this proposal's structure, three algorithms are described in this section to address the fake news classification problem in hand, *TF-IDF*, *Word2Vect* and *Doc2Vec*. They will be taking as input the results of

outperforms *boosting* algorithms on average, generally avoids over-fitting problems and it is, in fact, one of the more robust classifiers .

On every execution the following parameters' grid will be inspected though a *Grid-SearchCV*. Detail is provided on [Table 3.1](#).

Tuned parameters	Tested values
<i>max_depth</i>	10, None
<i>max_features</i>	1, 4, 10, auto
<i>min_samples_split</i>	2, 4, 10
<i>bootstrap</i>	True, False
<i>criterion</i>	gini, entropy

TABLE 3.1: *Random Forest* - Grid searched parameters

Regarding *k-Nearest Neighbours*, hereafter *k-NN*, it is a type of supervised *instance-based learning* or *non-generalizing learning*¹ machine learning algorithm. It does assume that similar things exist in close proximity.

scikit-learn proposes two different nearest neighbors classifiers: used *KNeighborsClassifier* implements learning based on the *k* nearest neighbors of each query point, where *k* is one of the grid searched values in the proposed implementation on this thesis. The best fitting value for *k* is actually highly data-dependent, while larger values diminish noise effects, also blur the classification boundaries. Along with the number of neighbours different values on the *weight* and *metric* parameters were put up to test as well. The first defining whether the execution considers uniform or weighted values depending on the distant to the neighbour points, and the second defining the used distance.

On every execution the following parameters' grid will be inspected though a *Grid-SearchCV*. Detail is provided on [Table 3.2](#).

Tuned parameters	Tested values
<i>n_neighbors</i>	4, 5, 10
<i>weights</i>	uniform, distance
<i>metric</i>	euclidean, manhattan, minkowski

TABLE 3.2: *k-NN* - Grid searched parameters

Regarding *Support Vector Machines*, hereinafter *SVM*², they are considered to be one of the most successful machine learning algorithms, relying on three main ideas: linearity, sparseness and kernel trick. The last, as a matter of fact may be the key to their success, since it is one of the features allowing SVM overcoming high dimensional problems, and might be worthy of further detail.

COVER'S THEOREM [14]: *A complex pattern-classification problem, cast in a high-dimensional space nonlinearly, is more likely to be linearly separable than in a low-dimensional space, provided that the space is not densely populated.*

It is one of the primary theoretical motivation for non-linear kernel methods. It is in

¹ It does not attempt to construct a general internal model, but stores instances of the training data.

² Also called Sparse Kernel Machines.

fact in the base of how SVM overcome the curse of highly dimensional problems by making the projection only implicitly (thanks to the kernel trick).

In general terms, SVM implementations have been a really successful classification algorithm based on the following properties/advantages: SVM do not have a local minimum, the optimal solution can be found in polynomial time, their implementation is based on a reduce number of free parameters that can be adjusted by cross-validation, they produce stable results and sparse solutions since they only takes into account the support vectors. Besides, maximizing the margin allows to control the complexity independently of the number of dimensions and they have a great generalization capability.

On every execution the following parameters' grid will be inspected though a *Grid-SearchCV*. Detail is provided on Table 3.3.

Tuned parameters	Tested values
<i>kernel</i>	rbf
<i>gamma</i>	$10^{-3}, 10^{-4}$
<i>C</i>	1, 10, 100
<i>kernel</i>	linear
<i>C</i>	1, 10, 100

TABLE 3.3: SVM - Grid searched parameters

Regarding *XGBoost*, it is a *Gradient Boosting* based algorithm optimized for speed, allowing to work with highly dispersed data and storing it as to prevent continuously re-arranging attributes. It does include a complexity based lose function penalty:

$$\mathbb{E}[L(y, F(x))] + \sum_{m=1}^M \Omega(f_m)$$

On every execution the following parameters' grid will be inspected though a *Grid-SearchCV*. Detail is provided on Table 3.4.

Tuned parameters	Tested values
<i>learning_rate</i>	0.05, 0.1, 0.2
<i>n_estimators</i>	10, 75, 150
<i>max_depth</i>	3, 5, 7
<i>gamma</i>	0, 0.1, 0.2

TABLE 3.4: XGBoost - Grid searched parameters

XGBoost includes, aside from the lose function penalizing complexity shown before, a great number of parameters to be adjusted. Even if providing great results and an important execution optimized speed it is not easy to parameterize. Initial implementation of *Gradient Boosting* algorithms was developed by Friedman [20]. Later implementations introduced the idea of an iterative functional gradient descent.

On all the previously mentioned classification algorithms both, *roc_auc_score* and *accuracy* metrics are used to calibrate the above mentioned grid searched parameters. Regarding *accuracy*, in binary classification as it is the problem presented, is currently used as an statistical measure the *goodness* of a classification, testing how correctly it does identify or exclude a condition. More precisely, *accuracy* defines the

proportion of true results over the total. It is actually defined as:

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

where,

TP = True positive; FP = False positive; TN = True negative; FN = False negative.

Furthermore, the *roc_auc_score* returns the area under the ROC³ curve, which, in terms, plots parametrically $TPR(T)$ versus $FPR(T)$ with T as the varying parameter. TPR (True positive rate) and FPR (false positive rate) are, respectively, defined as:

True positive rate,

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

and, false positive rate,

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

Assuming, that given a threshold parameter T , a result is classified as "positive" if $X > T$, and "negative" otherwise. $\mathcal{X}$ follows a probability density $f_1(x)$ if the result actually belongs to class "positive", and $f_0(x)$ if otherwise. Therefore, the area under the curve, provided by the *roc_auc_score* metric can be seen as:

$$\begin{aligned} Area &= \int_{x=0}^1 TPR(FPR^{-1}(x))dx \\ &= \int_{-\infty}^\infty TPR(T)FPR'(T)dT \\ &= \int_{-\infty}^\infty \int_{-\infty}^\infty \mathcal{I}(T' > T)f_1(T')f_0(T)dT'dT \\ &= P(X_1 > X_0) \end{aligned}$$

Given $TPR(T) : T \rightarrow y(x)$ and $FPR(T) : T \rightarrow x$. Considering that X_1 is the score for a positive result and X_0 the score for a negative one.

3.2.1 Token based analysis

Fake news classification problem though NLP token based proposed approach considers the implementation of a *Term Frequency – Inverse Document Frequency*, hereinafter *TF-IDF*, algorithm whose result will be used as input to the previously described ML classification algorithms. *TF-IDF* is, in fact, and automatic statistical token based approach into classifying word relevance within a corpus.

It breaks down into the two terms that make up its own name,

- *Term Frequency*, can be defined as the number occurrences of a word in a document, normalized by the document's length.

³ Receiver Operating Characteristic Curve.

- *Inverse Document Frequency*, is inverse to the proportion of documents in the corpus containing each word scaled by the logarithm (aligned with Zipf's law⁴).

TF-IDF value for a word w , on a specific document d of a certain corpus D , is given by:

$$\begin{aligned} TF - IDF(w, d, D) &= TF(w, d) \cdot IDF(d, D) \\ &= \frac{|w \in d|}{|d|} \cdot \log \frac{|D| + 1}{|d \in D : w \in d| + 1} \end{aligned}$$

The actual *TF-IDF* implementation used for the purposes of this thesis is the standard *scikit-learn*, *TfidfVectorizer* proposed function, where minor changes from the standard textbook notation have to be taken into account.

Detail on the hyper-parameter tuning results based on *GridSearchCV*, based on the previously detailed grids for each of the implemented ML classification algorithms. See Table C.1 on Appendix C.

3.2.2 Embedding based analysis

Two sorts of word embeddings have been applied through this thesis, *Word2Vec* and *Doc2Vec*. Word embeddings are NLP techniques creating real numbers mapping vectors for words in a corpus. In mathematics, an embedding is an instance of some algebraic structure contained within another instance, and as a parallelism from that definition, it "defines" how it takes the embedding from a multi-dimensional space per word to a continuous lower dimension vector space. The underlying concept is that a word is characterized by those "close" to it, and therefore, words with "close" meaning are represented close to each other in the resulting vector-space.

We have loaded and used the *Word2Vec* model created and published in 2013 by Google's researcher Tomas Mikolov [46], retrieving the trained model's vector representation for every word in the fed corpus and the averaging all word associated vectors within each story to create a unique vector associated to each Twitter story in the corpus that will be used as an input to the ML classification algorithms described on Section 3.2.

Regarding *Doc2Vec*, this thesis uses the *Gensim* library implementation, providing a unique vector associated to each document in the corpus. Even though, the implementation itself provides a meaningful classification retrieving associated probability to each existing label with remarkable results range accuracy scores over the test set of 85.4%, we have followed the same schema as in previous cases to maintain consistency across experiments.

Detail on the hyper-parameter tuning results based on *GridSearchCV*, based on the parameters' grids detailed on Section 3.2 for each of the implemented ML classification algorithms. See Table C.2 and Table C.3, for *Word2Vec* and *Doc2Vec* associated parameterization respectively on Appendix C.

⁴ Empirically introduced by George K. Zipf in the 1940s, states that for a given language the frequency of occurrence of different words follows a distribution that can be approximated by $P_n \sim 1/n^a$, where P_n represents the frequency of the n -th most frequent word, and a is a real number slightly over 1.

3.3 Testing

For each of the above described algorithms; *TF-IDF*, *Word2Vec* and *Doc2Vec*; the same execution schema has been followed by taken their output results and feeding them into the ML classification algorithms described on Section 3.2. Parameterization of the classification algorithms was performed though a grid search as described in such section, and the results are detailed on Appendix C.

The Twitter based corpus, as described on Section 2.2.1, was split into 80% and 20%, training and testing sets, in order to keep a subset of the data unseen by the classification algorithms that were trained based on the 80% of the data used as training set and their performance evaluated on the resting 20%, based on the following metrics:

- Confusion matrix.
- Classification report, including precision, recall and f1 scores, as well as the support values, for all labels. Additionally, it provides average values including macro average (*averaging the unweighted mean per label*) and the weighted average (*averaging the support-weighted mean per label*).
- *accuracy* score.
- *roc_auc* score.

Detailed description of both *accuracy* and *roc_auc* scores has been already provided on Section 3.2. Regarding the classification report it is the standard report provided on the implementation of *scikit-learn*, including precision, recall and f1 scores.

Precision and *recall*, are well related scores, defined as follows:

$$precision = \frac{|\{RelevantResults\} \cap |\{RetrievedResults\}|}{|\{RetrievedResults\}|}$$

$$recall = \frac{|\{RelevantResults\} \cap |\{RetrievedResults\}|}{|\{RelevantResults\}|}$$

Therefore, precision is understood as the percentage of the retrieved results that are in fact relevant and recall the percentage of the relevant results that were in fact retrieved.

F-1 score relies on the definition of the precision and recall scores. It is, in fact, the harmonic mean of both metrics,

$$F_1 = \frac{2}{recall^{-1} + precision^{-1}} = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

The general formula for any positive real number β is:

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

$$= \frac{(1 + \beta^2) \cdot TruePositive}{(1 + \beta^2) \cdot TruePositive + \beta^2 \cdot FalseNegative + FalsePositive}$$

The obtained results on each of the implemented algorithm architectures for each of the described metrics will be presented on [Section 5.1](#). However, a table with the comparison of the results in terms of *accuracy* and *roc_auc* is presented on [Figure 3.5](#) below.

Architecture	Classification algorithm	Metric	Score
TF-IDF	Random Forest	<i>accuracy</i>	86.49%
		<i>roc_auc</i>	76.85%
	XGBoost	<i>accuracy</i>	87.03%
		<i>roc_auc</i>	81.59%
	k-Nearest Neighbors	<i>accuracy</i>	88.65%
		<i>roc_auc</i>	86.00%
	SVM	<i>accuracy</i>	92.97%
		<i>roc_auc</i>	87.96%
Word2Vec	Random Forest	<i>accuracy</i>	80.54%
		<i>roc_auc</i>	68.84%
	XGBoost	<i>accuracy</i>	81.08%
		<i>roc_auc</i>	50.00%
	k-Nearest Neighbors	<i>accuracy</i>	81.62%
		<i>roc_auc</i>	71.24%
	SVM	<i>accuracy</i>	70.81%
		<i>roc_auc</i>	50.00%
Doc2Vec	Random Forest	<i>accuracy</i>	80.00%
		<i>roc_auc</i>	65.74%
	XGBoost	<i>accuracy</i>	85.95%
		<i>roc_auc</i>	77.01%
	k-Nearest Neighbors	<i>accuracy</i>	82.16%
		<i>roc_auc</i>	79.24%
	SVM	<i>accuracy</i>	81.62%
		<i>roc_auc</i>	76.14%

TABLE 3.5: Traditional NLP - ML *accuracy* and *roc_auc* scores

The highest obtained results, on both *accuracy* and *roc_auc*, were obtained by applying an SVM to the vector space generated after applying *TF-IDF* to the Twitter stories dataset. Achieving results with 92.97% and 87.96%, *accuracy* and *roc_auc* scores respectively, over the test set.

A comparison of the obtained results, by *accuracy* and *roc_auc* scores, is presented on Figure 3.2.

FIGURE 3.2: *Accuracy* and *roc_auc* scores - Traditional NLP techniques.

Chapter 4

Proposed techniques: Advanced deep learning techniques

4.1 Methodology

First things first, before going into further detail regarding deep learning techniques applied to the fake news detection problem in hand, it might worth the effort trying to clear the confusion and detail the differences telling apart *Artificial Intelligence* (AI), *Machine Learning* (ML) and *Deep Learning*. See on Figure 4.1 a back-of-the-envelope explanation.

FIGURE 4.1: Source: technologyreview.com - A back-of-the-envelope AI explainer

Even closely-related, they have a set / sub-set relationship between them being AI the broader and all-encompassing concept set and deep learning the smaller one. A quick definition of each of them might help to clarify.

- Artificial intelligence, as defined by Kaplan *et al.* on [39] is "a system's ability to correctly interpret external data, to learn from such data, and to use those learnings to achieve specific goals and tasks through flexible adaptation".
- Machine learning, as defined by Tom Mitchell, "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E ".

- Deep learning, Deng *et al.* proposed on [17], among others, the following definition: "A class of machine learning techniques that exploit many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification."

Altogether, problem solving based on pre-defined rules (*algorithm*), as far as it can be considered "intelligent" behaviour, is what we understand by artificial intelligence. Meanwhile, machine learning, being as subset of it, it is an method of training algorithms so that they can learn the best approach to decision making. Finally, deep learning is just an specific machine learning technique based on the exploit of many non-linear information processing layers.

On the discussed problem of fake news classification, this chapter introduces this thesis' proposed deep learning approach. Taking, again, the data set described in Section 2.2.1 and applying the lemmatization and stop words removal processes described in Section 3.2, two alternative algorithms' implementations are proposed and hereafter detailed, on the one hand a LSTM¹ neural network and on the other a GRU² neural network.

4.1.1 Long Short-Term Memory (LSTM) neural networks

Long Shot-Term Memory neural networks, hereinafter LSTM, are, in fact, a sort of artificial recurrent neural network (RNN) architecture. RNN networks are based on Rumerhart *et al.* work on [60] in 1986. LSTM on the other hand, were proposed by Hochreiter and Schmidhuber in 1997 on [33].

An LSTM network looks quite like a standard RNN, but for the summation units in the hidden layer being replaced by memory blocks. The following architecture diagram, as provided by Greff *et al.* on [27], a provides clarification on the network's architecture basis. See Figure 4.2.

FIGURE 4.2: Source: Greff *et al.* [27]. - The LSTM unit.

¹Long Short-Term Memory.

²Gated Recurrent Units.

In its original form, *LSTM* contained only input and output gates. The forget gates, by Gers *et al.* on [22] and the peephole weights on [23] connecting the gates to the memory cell were further extensions to the original model. The purpose of the forget gates was to provide a way for the memory cells to reset themselves and forget previous input values.

Additionally, having feedback connections they are keen on processing text since it is a sort of sequenced based data. Obviously, its advantages are most favoured when facing problems requiring the use of long range contextual information, such as text classification problems like the one in hand.

Detailed equations regarding both, the forward (activation) and backward (gradient calculation) passes of an *LSTM* hidden layer are provided on *Appendix D*.

Among the described advantages of *LSTM*, preservation of gradient information is one of the more remarkable [40]. Through input gates the network can decide when an input is important enough to be memorized, reset gates allow the network to decide when a memory is no longer useful and output gates when to release a particular memory to compute the current network output. Therefore, a memorized value can be retained indefinitely and thus they do not vanish gradients. They deal quite successfully with the exploding and vanishing gradient problems faced by *RNNs*.

The implementation presented on this thesis relies on the Keras' opened libraries. The proposed model is based on a Keras' sequential model, i.e. a linear stack of layers. A 200 neuron *LSTM* layer with dropout is presented after a Keras embedding layer. Various implementations were tested modifying the final dense layer activation function.

4.1.2 Gated Recurrent Units (*GRU*) neural networks

Gated Recurrent Units, hereinafter *GRU*, neural networks were firstly introduced by Kyunghyun Cho *et al.* in 2014 on [9]. Further detail on the formulation proposed by Kyunghyun Cho *et al.* is provided on *Appendix D*. *GRUs* have been shown to exhibit even better performance on certain smaller datasets, which as a matter of fact comes as a very useful feature to the problem in hand in this thesis. Performance comparison between *LSTM* and *GRU* is, however, a problem still under review. *GRU* networks, combine write and rest gates into a single update gate with not read gate. They have therefore a lower computational cost.

As a matter of fact, Junyoung Chung *et al.* [10] provide a valuable evaluation of *GRU* networks against *LSTM* neural networks. In fact they concluded that "by using fixed number of parameters for all models on some datasets *GRU*, can outperform *LSTM* units both in terms of convergence in CPU time and in terms of parameter updates and generalization. Furthermore, Wenpeng Yin *et al.* on [81] performed systematic comparison of CNN and the two prevailing *RNN*, *LSTM* and *GRU*, on a wide range of NLP tasks.

GRU neural network presented on this thesis relies as well on the Keras' opened libraries. The proposed model is based on a Keras' sequential model, i.e. a linear stack of layers. A 64 neuron *GRU* layer without dropout is presented after a Keras embedding layer. Various implementations were tested modifying the final dense layer activation function.

A helpful diagram comparing *LSTM* and *GRU* units' architecture, as pictured by Kyunghyun Cho *et al.*[9] is detailed as well on [10], and provided here for further comprehension. See Figure 4.3.

FIGURE 4.3: Source: Kyunghyun Cho *et al.* [9] LSTM vs GRU units.

4.2 Testing

First thing to be addressed before testing the proposed *LSTM* and *GRU* neural network implementations was leveling the Twitter stories lengths within the dataset. An standardization of the stories length distribution was performed. Based on Figure 4.4 stories were pad using Keras' *pad_sequence* into same length texts. Shorter texts are padded with value at the end, and longer sequences truncated so that they fit the desired length.

FIGURE 4.4: Twitter stories length distribution

A broad spectrum of padding lengths were tried³. From lower lengths, closer however to the average length, to higher extreme values. Along with the different padding lengths, a cluster of activation functions, *sigmoid*, *softmax* and *linear* was tested as well.

Achieved results were verified though *StratifiedKfold* performing 10-fold stratified cross-validation. This is a re-sampling technique that can provide a robust estimate of the performance of a machine learning model on unseen data. *scikit-learn* function *cross_val_score* was used to evaluate our model using the cross-validation scheme and print the results. Furthermore, Deep Learning Model Parameters were grid-searched to evaluate different configurations for the proposed neural network model without significant improvements on the achieved scores. See Table 4.1 for further

³ 150, 200, 250, 300, 350, 400, 500 and 600 words padding lengths.

detail on the grid-searched parameters and the possible values sequentially tested arrays.

Tuned parameters	Tested values
<i>optimizer</i>	SGD, adam
<i>init</i>	glorot_uniform, normal, uniform
<i>batches</i>	10, 32, 64
<i>activation</i>	softmax, relu, linear, sigmoid
<i>number of LSTM layers</i>	10, 100, 200
<i>LSTM dropout</i>	0.02, 0.2

TABLE 4.1: *LSTM* - Grid searched parameters

Data augmentation processes were tested as well trying to re-balance the dataset in order to feed both neural networks a more equitably labeled datasets without further performance improvement.

LSTM neural networks, training/test loss vs accuracy results using a sigmoid activation function are demonstrated on Figure 4.5.

FIGURE 4.5: *LSTM*: Loss/Accuracy scores - *sigmoid* activation function

LSTM neural networks, using a linear activation function, training/test loss vs accuracy results. See Figure 4.6.

FIGURE 4.6: *LSTM*: Loss/Accuracy scores - *linear* activation function

Best *accuracy* scores reported were of 0.7027 with a *roc_auc* score of 0.5016. Further tuning, considering all data augmentation, hyper-parameters grid search or K-Folds validation only increased accuracy up to 0.7081.

Furthermore, GRU neural networks training/test loss vs accuracy results are plotted on Figure 4.7.

FIGURE 4.7: GRU: Loss/Accuracy scores

Chapter 5

Conclusions and future work

5.1 Conclusions

Based on the implementations described on *Chapters 3* and *4*, and metrics presented on *Chapter 3*, the obtained *accuracy* and *roc_auc* scores are detailed bellow for the different algorithm architectures.

FIGURE 5.1: Obtained *accuracy* and *roc_auc* scores.

Figure 5.1 represented scores demonstrate that testing results previously addressed on *Chapter 3*, even representing old-fashioned techniques, still worth trying, specially when available data is scarce. Best obtained results, based on *accuracy* and *roc_auc* scores, for the used dataset were provided by a combination of *TF-IDF* NLP algorithm and *SVM* classification implementation, even when more recent techniques, such as text-based embeddings (*Word2Vec*) or even *LSTM* and *GRU* neural networks, were put up to test as well.

Furthermore, detail on the obtained *precision*, *recall* and *F1* scores is introduced as well in Figure 5.2 bellow, as part of the performed analysis. The presented information is detailed by each of the tested implementations.

FIGURE 5.2: Obtained *precision*, *recall* and *F1* scores.

As evidenced by the previous figure, best architecture for a certain problem may highly depend on the task in hand. While *accuracy* or *roc_auc* scores define the best overall solution. If an specific result is sought, *precision* or *recall* metrics should be contemplated. Those, as defined on Chapter 3, are key when retrieving the maximum data of highest precision on a certain label is the final objective, and based on them, as shown before, the best suited implementation may change.

The obtained results demonstrate that taking into account the complexity level, execution's times and resources' requirements, basic algorithms are, at least when minor increases on accuracy are not crucial to the sought outcomes, worth trying and may be in some cases, such as the one in hand, even more accurate. It is however, not a definitively conclusion due to the size and limitations of the used dataset. Fake news classification task it is not the one with the highest available tagged data and therefore unbalanced data is of common usage, this is main reason impeding the deployment of human-out-of-the-loop solutions. Additionally, hyper-parameter tuning for new neural network architectures is definitively not an easy tasks, but highly time consuming and of high computational cost [66].

5.2 Future work

As this thesis was focused on presenting an overview on the main existing text analysis techniques, not deep dive was addressed for none of them. Further hyper-parameter testing could be performed on the presented neural networks as well as their combination with the implemented embeddings.

Additionally, newly introduced *transformer* network based implementations, such as *BERT*, are surely worthy exploring when addressing the fake news classification problem. Even though, further data might be necessary to fully exploit the advantages and characteristics of these implementations, the actual fake news and data availability increasing pace will sure lead to an abundance of available data to feed into the different tested strategies.

Appendix A

Topic modelling visualization

Main topic modeling algorithms are Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA) or Latent Semantic Indexing (LSI), Non-Negative Matrix Factorization (NMF) and Biterm topic modeling. This thesis approach was limited to the first as an overall and initial approximation to the modeled corpus.

LDA is a generative statistical model allowing sets of observations to be explained by un-observed groups explaining why some parts of the data are similar. It was developed by Blei *et al.* in 2003 and it tells us what topics are present in a given document by observing the words in it and producing a topic distribution. By using *Bag-of-Words* as a method to extract features from our corpus, by means of the *Gensim's LDAMulticore* streamed implementation a topic analysis on the dataset can be addressed. Additionally, topic visualization and result interpretation can be performed by means of *pyLDavis* interactive LDA visualization python package. It extracts information from a fitted LDA topic model to inform an interactive web-based visualization.

FIGURE A.1: *Coherence score* - Best fitting number of topics.

Furthermore, in order to build LDA models and modify the number of topics considered in each case, we have checked the *coherence score*, based on the *gensim* implementation¹.

Analysis based on topic modeling uncovering the hidden topics on the dataset Twitter stories can also be performed. Popular topic modeling algorithms include latent semantic analysis (LSA), hierarchical Dirichlet process (HDP), and latent Dirichlet allocation (LDA), showing the latest excellent results. *pyLDavis* is a interactive LDA visualization python package allowing a visualization of the composition of every identified topic, see Figure A.2. Reference from Section 2.2.1.

¹ Based on Michael Roeder, Andreas Both and Alexander Hinneburg: "Exploring the space of topic coherence measures"

FIGURE A.2: *pyLDavis* - Topic visualization.

Appendix B

Implemented algorithm architecture diagram

Detail on the data flow as to the different algorithm architecture is provided bellow. It is distinctively split into three identifiable pieces: NLP analysis pre-processing, NLP traditional techniques combined with machine learning classification algorithms and a deep learning approach.

FIGURE B.1: Proposed classification algorithm architecture.

Appendix C

Classification algorithms parametrization

Detailed hyper-parameter tuning results based on *GridSearchCV* for each of the implemented ML classification algorithms.

Parameter results for *TF-IDF* architecture:

Algorithm	Score metric	Parameterization
Random Forest	accuracy	<i>bootstrap: False</i> <i>criterion: gini</i> <i>max_depth: None</i> <i>max_features: auto</i> <i>min_samples_split: 2</i>
	roc_auc	<i>bootstrap: False</i> <i>criterion: entropy</i> <i>max_depth: None</i> <i>max_features: auto</i> <i>min_samples_split: 4</i>
XGBoost	accuracy	<i>gamma: 0.1</i> <i>learning_rate: 0.05</i> <i>max_depth: 7</i> <i>n_estimators: 150</i>
	roc_auc	<i>gamma: 0.1</i> <i>learning_rate: 0.2</i> <i>max_depth: 5</i> <i>n_estimators: 75</i>
k-Nearest Neighbors	accuracy	<i>metric: euclidean</i> <i>n_neighbors: 4</i> <i>weights: distance</i>
	roc_auc	<i>metric: euclidean</i> <i>n_neighbors: 5</i> <i>weights: distance</i>
SVM	accuracy	<i>C: 10</i> <i>kernel: linear</i>
	roc_auc	<i>C: 1</i> <i>kernel: linear</i>

TABLE C.1: *TF-IDF* - Classification algorithms parameterization

Parameter results for the *Word2Vec* based architecture:

Algorithm	Score metric	Parameterization
Random Forest	accuracy	<i>bootstrap: True</i> <i>criterion: entropy</i> <i>max_depth: None</i> <i>max_features: 10</i> <i>min_samples_split: 2</i>
	roc_auc	<i>bootstrap: False</i> <i>criterion: entropy</i> <i>max_depth: None</i> <i>max_features: auto</i> <i>min_samples_split: 4</i>
XGBoost	accuracy	<i>gamma: 0.0</i> <i>learning_rate: 0.2</i> <i>max_depth: 3</i> <i>n_estimators: 150</i>
	roc_auc	<i>gamma: 0.1</i> <i>learning_rate: 0.2</i> <i>max_depth: 7</i> <i>n_estimators: 150</i>
k-Nearest Neighbors	accuracy	<i>metric: manhattan</i> <i>n_neighbors: 5</i> <i>weights: distance</i>
	roc_auc	<i>metric: manhattan</i> <i>n_neighbors: 10</i> <i>weights: distance</i>
SVM	accuracy	<i>C: 1</i> <i>gamma: 0.001</i> <i>kernel: rbf</i>
	roc_auc	<i>C: 100</i> <i>gamma: 0.001</i> <i>kernel: rbf</i>

TABLE C.2: *Word2Vec* - Classification algorithms parameterization

Parameter results for the *Doc2Vec* based architecture:

Algorithm	Score metric	Parameterization
Random Forest	accuracy	<i>bootstrap: True</i> <i>criterion: entropy</i> <i>max_depth: None</i> <i>max_features: 10</i> <i>min_samples_split: 2</i>
	roc_auc	<i>bootstrap: False</i> <i>criterion: gini</i> <i>max_depth: None</i> <i>max_features: 1</i> <i>min_samples_split: 2</i>
XGBoost	accuracy	<i>gamma: 0.0</i> <i>learning_rate: 0.1</i> <i>max_depth: 7</i> <i>n_estimators: 150</i>
	roc_auc	<i>gamma: 0.1</i> <i>learning_rate: 0.1</i> <i>max_depth: 5</i> <i>n_estimators: 150</i>
k-Nearest Neighbors	accuracy	<i>metric: manhattan</i> <i>n_neighbors: 4</i> <i>weights: distance</i>
	roc_auc	<i>metric: euclidean</i> <i>n_neighbors: 5</i> <i>weights: distance</i>
SVM	accuracy	<i>C: 10</i> <i>gamma: 0.001</i> <i>kernel: rbf</i>
	roc_auc	<i>C: 100</i> <i>gamma: 0.001</i> <i>kernel: rbf</i>

TABLE C.3: *Doc2Vec* - Classification algorithms parameterization

Appendix D

LSTM and GRU networks underlying equations

D.1 LSTM hidden layer equations

Detailed equations as to how and *LSTM* neural network models the word sequence x , as introduced on [81], is detailed bellow.

$$\begin{aligned} i_t &= \sigma(x_t U^i + h_{t-1} W^i + b_i) \\ f_t &= \sigma(x_t U^f + h_{t-1} W^f + b_f) \\ o_t &= \sigma(x_t U^o + h_{t-1} W^o + b_o) \\ q_t &= \tanh(x_t U^q + h_{t-1} W^q + b_q) \\ p_t &= f_t \star p_{t-1} + i_t \star q_t \\ h_t &= o_t \star \tanh(p_t) \end{aligned}$$

where i_t , f_t and o_t stand for the three LSTM gates; input, forget and output gates, all generated by a *sigmoid* function over the input ensemble x_t and the preceding hidden state h_{t-1} . Additionally, q_t stands for the temporary result generated by a *tanh* non linear operation over the combined x_t and h_{t-1} ensemble, then combined with the history p_{t-1} by input gate i_t and forget gate f_t respectively to get an updated history p_t . Finally, output gate o_t over p_t produces the final hidden state h_t .

D.2 GRU equations

GRU equations addressing x text sequence modelization, as refereed on [81], is detailed bellow.

$$\begin{aligned} z &= \sigma(x_t U^z + h_{t-1} W^z) \\ r &= \sigma(x_t U^r + h_{t-1} W^r) \\ s_t &= \tanh(x_t U^s + (h_{t-1} \circ r) W^s) \\ h_t &= (1 - z) \circ s_t + z \circ h_{t-1} \end{aligned}$$

where x_t is and d -dimensional element representing token x at position t , i.e. $x_t \in \mathbb{R}^d$, $h_t \in \mathbb{R}^h$ represents the hidden state at position. z and r represent the two gates.

Bibliography

- [1] European Comision High-Level Expert Group on AI (AI HLEG). *Policy and investment recommendations for trustworthy Artificial Intelligence*. Tech. rep. June 2019.
- [2] Majed AlRubaian, Muhammad Al-Qurishi, Mabrook Al-Rakhami, Sk Md Mizanur Rahman, and Atif Alamri. "A multistage credibility analysis model for microblogs". In: *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE. 2015, pp. 1434–1440.
- [3] Samir Bajaj. *The Pope Has a New Baby! Fake News Detection Using Deep Learning*. Tech. rep. Technical Report, Stanford Univ, 2017.
- [4] Albert-László Barabási. *Linked: The new science of networks*. 2003.
- [5] Alexandre Bovet and Hernán A Makse. "Influence of fake news in Twitter during the 2016 US presidential election". In: *Nature communications* 10.1 (2019), p. 7.
- [6] CCN-CERT BP/13. *Desinformación en el ciberespacio*. Tech. rep. Feb. 2019.
- [7] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.
- [8] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. "Information credibility on twitter". In: *Proceedings of the 20th international conference on World wide web*. ACM. 2011, pp. 675–684.
- [9] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).
- [10] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555* (2014).
- [11] Michael D Conover, Bruno Gonçalves, Jacob Ratkiewicz, Alessandro Flammini, and Filippo Menczer. "Predicting the political alignment of twitter users". In: *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*. IEEE. 2011, pp. 192–199.
- [12] Josh Constine. *Facebook tries fighting fake news with publisher info button on links*. Ed. by techcrunch.
- [13] Mauro Conti, Daniele Lain, Riccardo Lazzeretti, Giulio Lovisotto, and Walter Quattrociocchi. "It's always April fools' day!: On the difficulty of social network misinformation classification via propagation features". In: *2017 IEEE Workshop on Information Forensics and Security (WIFS)*. IEEE. 2017, pp. 1–6.

- [14] Thomas M Cover. "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition". In: *IEEE transactions on electronic computers* 3 (1965), pp. 326–334.
- [15] Richard Davis and Chris Proctor. *Fake news, real consequences: Recruiting neural networks for the fight against fake news*. 2017.
- [16] Michela Del Vicario, Alessandro Bessi, Fabiana Zollo, Fabio Petroni, Antonio Scala, Guido Caldarelli, H Eugene Stanley, and Walter Quattrociocchi. "The spreading of misinformation online". In: *Proceedings of the National Academy of Sciences* 113.3 (2016), pp. 554–559.
- [17] Li Deng, Dong Yu, et al. "Deep learning: methods and applications". In: *Foundations and Trends® in Signal Processing* 7.3–4 (2014), pp. 197–387.
- [18] T Economist. "The world's most valuable resource is no longer oil, but data". In: *The Economist: New York, NY, USA* (2017).
- [19] Mehrdad Farajtabar, Jiachen Yang, Xiaojing Ye, Huan Xu, Rakshit Trivedi, Elias Khalil, Shuang Li, Le Song, and Hongyuan Zha. "Fake news mitigation via point process based intervention". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1097–1106.
- [20] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine". In: *Annals of statistics* (2001), pp. 1189–1232.
- [21] Erich Fromm. *Escape from freedom*. Rinehart, Incorporated, 1973.
- [22] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. "Learning to forget: Continual prediction with LSTM". In: (1999).
- [23] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. "Learning precise timing with LSTM recurrent networks". In: *Journal of machine learning research* 3.Aug (2002), pp. 115–143.
- [24] Georgios Giasemidis, Colin Singleton, Ioannis Agrafiotis, Jason RC Nurse, Alan Pilgrim, Chris Willis, and Danica Vukadinovic Greetham. "Determining the veracity of rumours on Twitter". In: *International Conference on Social Informatics*. Springer. 2016, pp. 185–205.
- [25] Lisa Gitelman. *Raw data is an oxymoron*. MIT press, 2013.
- [26] Google. *Google fact check feature*. 2018.
- [27] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. "LSTM: A search space odyssey". In: *IEEE transactions on neural networks and learning systems* 28.10 (2016), pp. 2222–2232.
- [28] Nir Grinberg, Kenneth Joseph, Lisa Friedland, Briony Swire-Thompson, and David Lazer. "Fake news on Twitter during the 2016 US presidential election". In: *Science* 363.6425 (2019), pp. 374–378.
- [29] Aditi Gupta and Ponnurangam Kumaraguru. "Credibility ranking of tweets during high impact events". In: *Proceedings of the 1st workshop on privacy and security in online social media*. Acm. 2012, p. 2.
- [30] Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. "Tweetcred: Real-time credibility assessment of content on twitter". In: *International Conference on Social Informatics*. Springer. 2014, pp. 228–243.
- [31] Sardar Hamidian and Mona Diab. "Rumor identification and belief investigation on twitter". In: *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. 2016, pp. 3–8.

- [32] Tin Kam Ho. "Random decision forests". In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE. 1995, pp. 278–282.
- [33] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.
- [34] House of Commons. Digital, Culture, Media and Sport Committee. *Disinformation and 'fake news': Final Report*. Tech. rep. Feb. 2019.
- [35] Eva Jaho, Efstratios Tzoannos, Aris Papadopoulos, and Nikos Sarris. "Alethiometer: a framework for assessing trustworthiness and content validity in social media". In: *Proceedings of the 23rd International Conference on World Wide Web*. ACM. 2014, pp. 749–752.
- [36] Shan Jiang and Christo Wilson. "Linguistic signals under misinformation and fact-checking: Evidence from user comments on social media". In: *Proceedings of the ACM on Human-Computer Interaction* 2.CSCW (2018), p. 82.
- [37] Fang Jin, Edward Dougherty, Parang Saraf, Yang Cao, and Naren Ramakrishnan. "Epidemiological modeling of news and rumors on twitter". In: *Proceedings of the 7th Workshop on Social Network Mining and Analysis*. ACM. 2013, p. 8.
- [38] Zhiwei Jin, Juan Cao, Yu-Gang Jiang, and Yongdong Zhang. "News credibility evaluation on microblog with a hierarchical propagation model". In: *2014 IEEE International Conference on Data Mining*. IEEE. 2014, pp. 230–239.
- [39] Andreas Kaplan and Michael Haenlein. "Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence". In: *Business Horizons* 62.1 (2019), pp. 15–25.
- [40] Kazuya Kawakami. "Supervised sequence labelling with recurrent neural networks". In: *Ph. D. thesis* (2008).
- [41] Junaed Younus Khan, Md Khondaker, Tawkat Islam, Anindya Iqbal, and Sadia Afroz. "A Benchmark Study on Machine Learning Methods for Fake News Detection". In: *arXiv preprint arXiv:1905.04749* (2019).
- [42] Jooyeon Kim, Dongkwan Kim, and Alice Oh. "Homogeneity-Based Transmissive Process to Model True and False News in Social Networks". In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM. 2019, pp. 348–356.
- [43] KP Krishna Kumar and G Geethakumari. "Detecting misinformation in online social networks using cognitive psychology". In: *Human-centric Computing and Information Sciences* 4.1 (2014), p. 14.
- [44] Sejeong Kwon, Meeyoung Cha, and Kyomin Jung. "Rumor detection over varying time windows". In: *PloS one* 12.1 (2017), e0168344.
- [45] Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. "Prominent features of rumor propagation in online social media". In: *2013 IEEE 13th International Conference on Data Mining*. IEEE. 2013, pp. 1103–1108.
- [46] Quoc Le and Tomas Mikolov. "Distributed representations of sentences and documents". In: *International conference on machine learning*. 2014, pp. 1188–1196.
- [47] Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. "Real-time rumor debunking on twitter". In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM. 2015, pp. 1867–1870.

- [48] Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. "Real-time Rumor Debunking on Twitter". In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 2015, pp. 1867–1870.
- [49] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen, Kam-Fai Wong, and Cha Meeyoung. "Detecting Rumors from Microblogs with Recurrent Neural Networks". In: *The 25th International Joint Conference on Artificial Intelligence*. AAAI. 2016, pp. 3818–3824.
- [50] Jing Ma, Wei Gao, and Kam-Fai Wong. "Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning". In: *The 55th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. 2017, pp. 708–717.
- [51] Damian Mrowca, Elias Wang, and Atli Kosson. *Stance detection for fake news identification*. 2017.
- [52] Raymond S Nickerson. "Confirmation bias: A ubiquitous phenomenon in many guises". In: *Review of general psychology* 2.2 (1998), pp. 175–220.
- [53] Özlem Özgöbek and Jon Atle Gulla. "Towards an Understanding of Fake News". In: *CEUR Workshop Proceedings*. 2017.
- [54] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. "Automatic detection of fake news". In: *arXiv preprint arXiv:1708.07104* (2017).
- [55] Luladay Price and Stephone Christian. *Learning a New(s) Model: An exploration of LSTM classification and Language Modeling of News Articles*. 2018.
- [56] Vahed Qazvinian, Emily Rosengren, Dragomir R Radev, and Qiaozhu Mei. "Rumor has it: Identifying misinformation in microblogs". In: *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics. 2011, pp. 1589–1599.
- [57] Kenneth Rapoza. *Can 'Fake News' Impact The Stock Market?* Ed. by Forbes.
- [58] Paul Resnick, Samuel Carton, Souneil Park, Yuncheng Shen, and Nicole Zeffer. "Rumorlens: A system for analyzing the impact of rumors and corrections in social media". In: *Proc. Computational Journalism Conference*. Vol. 5. 7. 2014.
- [59] Natali Ruchansky, Sungyong Seo, and Yan Liu. "Csi: A hybrid deep model for fake news detection". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM. 2017, pp. 797–806.
- [60] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. "Learning representations by back-propagating errors". In: *Cognitive modeling* 5.3 (1988), p. 1.
- [61] H Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Stephanie M Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin EP Seligman, et al. "Personality, gender, and age in the language of social media: The open-vocabulary approach". In: *PloS one* 8.9 (2013), e73791.
- [62] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

- [63] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. "The spread of low-credibility content by social bots". In: *Nature communications* 9.1 (2018), p. 4787.
- [64] Kai Shu, Xinyi Zhou, Suhang Wang, Reza Zafarani, and Huan Liu. "The Role of User Profile for Fake News Detection". In: *arXiv preprint arXiv:1904.13355* (2019).
- [65] Peter Warren Singer. *Wired for war: The robotics revolution and conflict in the 21st century*. Penguin, 2009.
- [66] Emma Strubell, Ananya Ganesh, and Andrew McCallum. "**Energy and Policy Considerations for Deep Learning in NLP**". In: *arXiv preprint arXiv:1906.02243* (June 2019).
- [67] Eugenio Tacchini, Gabriele Ballarin, Marco L Della Vedova, Stefano Moret, and Luca de Alfaro. "Some like it hoax: Automated fake news detection in social networks". In: *arXiv preprint arXiv:1704.07506* (2017).
- [68] Sebastian Tschiatschek, Adish Singla, Manuel Gomez Rodriguez, Arpit Merchant, and Andreas Krause. "Fake news detection in social networks via crowd signals". In: *Companion Proceedings of the The Web Conference 2018*. International World Wide Web Conferences Steering Committee. 2018, pp. 517–524.
- [69] Nguyen Vo and Kyumin Lee. "The rise of guardians: Fact-checking url recommendation to combat fake news". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM. 2018, pp. 275–284.
- [70] Svitlana Volkova, Kyle Shaffer, Jin Yea Jang, and Nathan Hodas. "Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2017, pp. 647–653.
- [71] Soroush Vosoughi and Deb Roy. "A human-machine collaborative system for identifying rumors on twitter". In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE. 2015, pp. 47–50.
- [72] Soroush Vosoughi, Deb Roy, and Sinan Aral. "The spread of true and false news online". In: *Science* 359.6380 (2018), pp. 1146–1151.
- [73] Claire Wardle. "Fake news. It's complicated". In: *First Draft News* 16 (2017).
- [74] Claire Wardle and Hossein Derakhshan. "Information Disorder: Toward an interdisciplinary framework for research and policy making". In: *Council of Europe Report* 27 (2017).
- [75] David H Wolpert and William G Macready. "Coevolutionary free lunches". In: *IEEE Transactions on Evolutionary Computation* 9.6 (2005), pp. 721–735.
- [76] David H Wolpert, William G Macready, et al. "No free lunch theorems for optimization". In: *IEEE transactions on evolutionary computation* 1.1 (1997), pp. 67–82.
- [77] Liang Wu, Jundong Li, Xia Hu, and Huan Liu. "Gleaning wisdom from the past: Early detection of emerging rumors in social media". In: *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM. 2017, pp. 99–107.
- [78] Liang Wu and Huan Liu. "Tracing fake-news footprints: Characterizing social media messages by how they propagate". In: *Proceedings of the Eleventh ACM*

- International Conference on Web Search and Data Mining*. ACM. 2018, pp. 637–645.
- [79] Kai-Chou Yang, Timothy Niven, and Hung-Yu Kao. “Fake News Detection as Natural Language Inference”. In: (2019).
- [80] Zhifan Yang, Chao Wang, Fan Zhang, Ying Zhang, and Haiwei Zhang. “Emerging rumor identification for social media with hot topic detection”. In: *2015 12th Web Information System and Application Conference (WISA)*. IEEE. 2015, pp. 53–58.
- [81] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. “Comparative study of CNN and RNN for natural language processing”. In: *arXiv preprint arXiv:1702.01923* (2017).
- [82] Tongkui Yu, Shu-Heng Chen, and Connie Houning Wang. “Information aggregation in big data: Wisdom of crowds or stupidity of herds”. In: *International Symposium on Distributed Computing and Artificial Intelligence*. Springer. 2017, pp. 16–27.
- [83] Reza Zafarani and Huan Liu. “Connecting users across social media sites: a behavioral-modeling approach”. In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2013, pp. 41–49.
- [84] Savvas Zannettou, Michael Sirivianos, Jeremy Blackburn, and Nicolas Kourtellis. “The web of false information: Rumors, fake news, hoaxes, clickbait, and various other shenanigans”. In: *Journal of Data and Information Quality (JDIQ)* 11.3 (2019), p. 10.
- [85] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. “Enquiring minds: Early detection of rumors in social media from enquiry posts”. In: *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee. 2015, pp. 1395–1405.
- [86] Xinyi Zhou, Atishay Jain, Vir V Phoha, and Reza Zafarani. “Fake News Early Detection: A Theory-driven Model”. In: *arXiv preprint arXiv:1904.11679* (2019).
- [87] Xinyi Zhou and Reza Zafarani. “Fake news: A survey of research, detection methods, and opportunities”. In: *arXiv preprint arXiv:1812.00315* (2018).
- [88] Xinyi Zhou, Reza Zafarani, Kai Shu, and Huan Liu. “Fake news: Fundamental theories, detection strategies and challenges”. In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM. 2019, pp. 836–837.