IBERIFIER
Iberian Media Research
& Fact-Checking

IBERIFIER — Iberian Digital Media Research and Fact-Checking Hub

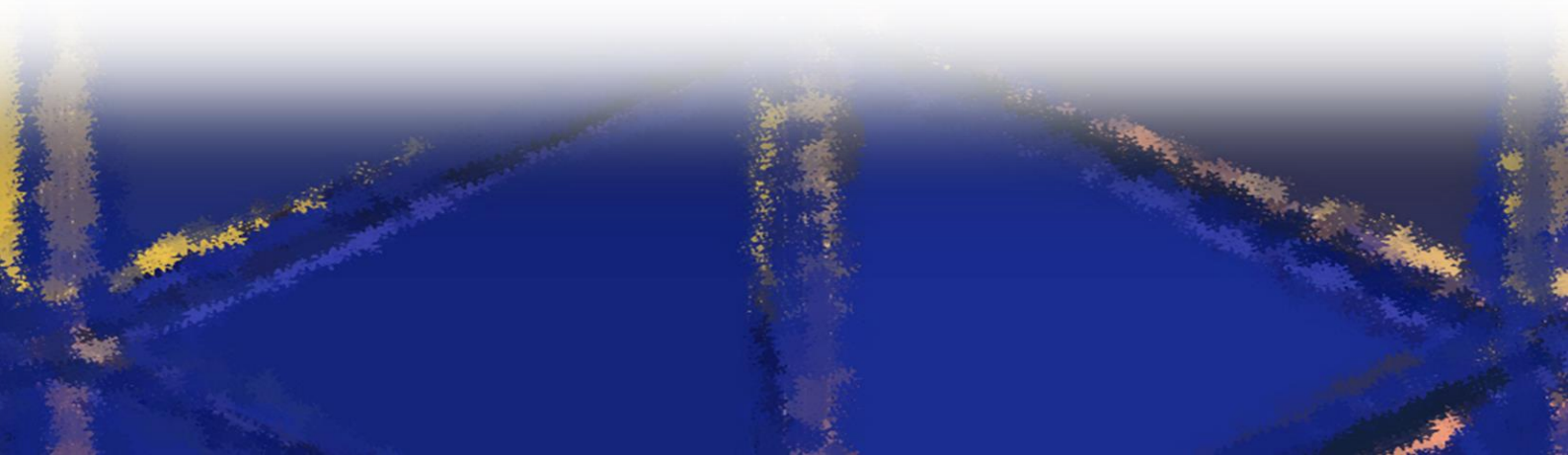
Is the ‘AI toolbox for disinformation’ ready?

Report

March 2023

European Commission

IBERIFIER has received funding from the European Commission
under the agreement CEF-TC-2020-2 (European Digital Media Observatory)
with reference 2020-EU-IA-0252



IBERIFIER
Iberian Media Research
& Fact-Checking

IBERIFIER
Iberian Media Research and Fact-Checking

Activity A3

Computer and data research

Deliverable 12

Is the 'AI toolbox for disinformation' ready?

Funding Instrument:	Connecting Europe Facility
Call:	CEF-TC-2020-2
Call Topic:	European Digital Media Observatory
Project Start:	1 September 2021
Project Duration:	30 months
Beneficiary in Charge:	UGR

Dissemination Level		
PU	Public	✓
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the Consortium (including the Commission Services)	
CO	Confidential, only for members of the Consortium (including the Commission Services)	

Website: iberifier.eu
Twitter: @iberifier

This project has received funding from the European Commission under the call CEF-TC-2020-2 (European Digital Media Observatory) Reference: **2020-EU-IA-0252**

Deliverable Information

Document Administrative Information	
Project Acronym:	IBERIFIER
Project Number:	2020-EU-IA-0252
Deliverable Number:	12
Deliverable Full Title:	Is the 'AI toolbox for disinformation' ready?
Deliverable Short Title:	Review on AI & Disinformation
Document Identifier:	iberifier-12-ReviewOnAIDisinformation-final
Beneficiary in Charge:	UGR
Report Version:	v1.0
Dissemination Level:	PU
Nature:	Report
Lead Author(s):	Andrés Montoro Montarroso (UGR) David Camacho (UPM) Alejandro Martín (UPM) Javier Torregrosa (UPM) Paolo Rosso (UPV) Berta Chulvi (UPV) María J. Rementería (BSC) Blanca Calvo Figueras (BSC) Olivier Philippe (BSC) Miguel Molina Solana (UGR) Javier Cantón Correa (UGR) Juan Gómez Romero (UGR)
Co-author(s):	...
Keywords:	artificial intelligence, machine learning, disinformation, natural language processing, social networks analysis, deepfakes
Status:	__ draft, _x_ final, __ submitted

Table of Contents

Executive Summary

0.1 English

The Internet and, more recently, social networks deliver continuous content to their users, who are more connected than ever but overwhelmed and not necessarily better informed. It is often challenging to distinguish trustworthy from malicious content, especially when it appeals to emotions and beliefs and comes from familiar sources. In recent years, we have witnessed an increased volume of false messages in social networks, which tend to spread faster and broader than truthful information. Not surprisingly, disinformation, meaning incorrect information purposely intended to harm, has been highlighted as a significant contributor to the polarisation and demeaning of democratic institutions.

Debunking false information is complicated and time-consuming, requiring expert participation and manual work. Hence, computational methods based on massive data processing technologies are envisioned as essential to better understanding and mitigating disinformation. Accordingly, Artificial Intelligence (AI) has emerged as a suitable toolbox for disinformation detection and fact-checking. However, the remarkable AI advances in natural language processing, social network analysis or even synthetic content generation have yet to permeate outside the research labs. The motivation behind this report is the realization that there is still a considerable gap between AI research labs and practitioners' daily challenges on fighting disinformation.

This report provides a concise guide to navigating the recent literature on AI for fighting disinformation, emphasizing the region covered by the IBERIFIER project. The report covers the fundamentals of Machine Learning (ML) algorithms, describes their application to several facets of disinformation analysis, and maps datasets and tools generated in the Iberian research community. The main conclusion is that there is not only a great need to transfer technologies from research to the industry but also to redirect research efforts toward human-supported tools rather than fully automated solutions —which are often biased and very domain-specific.

0.2 Español

Internet y las redes sociales proporcionan un flujo de contenido continuo a los usuarios, que están más conectados que nunca pero también saturados y no necesariamente mejor informados. A menudo resulta difícil distinguir los contenidos fiables de los engañosos, sobre todo cuando estos apelan a emociones y creencias y proceden de fuentes conocidas. Así, en los últimos años hemos asistido a un aumento del volumen de mensajes falsos en las redes sociales, que tienden a difundirse con mayor rapidez y alcance que la información veraz. No es de extrañar que la desinformación, es decir, la información incorrecta con la intención deliberada de perjudicar, haya sido señalada como un importante factor en el aumento de la polarización y el desprestigio de las instituciones democráticas.

Verificar la información es complicado y requiere mucho tiempo, así como la participación de expertos y trabajo manual. Por ello, los métodos computacionales basados en tecnologías de procesamiento masivo de datos se consideren esenciales para comprender mejor y mitigar la desinformación. En este sentido, el interés en la Inteligencia Artificial (IA) como herramienta para la detección de la desinformación y la verificación de los hechos ha aumentado notablemente. Sin embargo, los grandes avances de la IA en el procesamiento del lenguaje natural, el análisis de redes sociales o incluso la generación de contenidos sintéticos aún no han tenido mucha penetración fuera de los laboratorios de investigación. El punto de partida de este informe es la constatación de que

sigue existiendo una brecha considerable entre los laboratorios de investigación en IA y los retos cotidianos de los profesionales que luchan contra la desinformación.

Este informe proporciona una guía concisa para navegar por la literatura reciente sobre IA para combatir la desinformación, haciendo hincapié en los desarrollos y la problemática de la zona geográfica del proyecto IBERIFIER. El informe describe los fundamentos de los algoritmos de aprendizaje automático (ML), revisa su aplicación a varias facetas del análisis de la desinformación y lista conjuntos de datos y herramientas generados en la comunidad investigadora ibérica. La principal conclusión es que no sólo existe una gran necesidad de transferir tecnologías de los laboratorios a la industria, sino también de reorientar los esfuerzos de investigación hacia herramientas apoyadas por humanos en lugar de soluciones totalmente automatizadas —que a menudo presentan sesgos y son muy específicas de un dominio.

0.3 Português

A Internet e, mais recentemente, as redes sociais fornecem conteúdo contínuo aos seus utilizadores, que estão mais ligados do que nunca, mas sobrecarregados e não necessariamente melhor informados. É muitas vezes desafiante distinguir conteúdo de confiança de conteúdo malicioso, especialmente quando apela a emoções e crenças e vem de fontes familiares. Nos últimos anos, temos testemunhado um volume crescente de mensagens falsas nas redes sociais, que tendem a espalhar-se mais rapidamente e de forma mais ampla do que a informação verdadeira. Não surpreendentemente, a desinformação, que significa informação incorrecta propositadamente destinada a prejudicar, tem sido destacada como um contributo significativo para a polarização e aviltamento das instituições democráticas.

Desmascarar informação falsa é complicado e moroso, exigindo a participação de peritos e trabalho manual. Assim, os métodos computacionais baseados em tecnologias de processamento de dados maciços são vistos como essenciais para uma melhor compreensão e mitigação da desinformação. Consequentemente, a Inteligência Artificial (IA) surgiu como uma caixa de ferramentas adequada para a detecção de desinformação e verificação de factos. Contudo, os notáveis avanços da IA no processamento de linguagem natural, análise de redes sociais ou mesmo geração de conteúdos sintéticos ainda têm de permear fora dos laboratórios de investigação. A motivação subjacente a este relatório é a constatação de que ainda existe um fosso considerável entre os laboratórios de pesquisa de IA e os desafios diários dos profissionais.

Este relatório fornece um guia conciso para navegar na literatura recente sobre a IA para combater a desinformação, enfatizando a região abrangida pelo projecto IBERIFIER. O relatório cobre os fundamentos dos algoritmos de Machine Learning (ML), descreve a sua aplicação a várias facetas da análise da desinformação, e mapeia conjuntos de dados e ferramentas geradas na comunidade de investigação ibérica. A principal conclusão é que não só existe uma grande necessidade de transferir tecnologias da investigação para a indústria, mas também de redireccionar os esforços de investigação para ferramentas apoiadas pelo homem, em vez de soluções totalmente automatizadas — que são frequentemente tendenciosas e muito específicas do domínio.

1 Introduction

In the current post-truth era, individuals are overwhelmed by a massive and continuous stream of information, within which it is often challenging to distinguish credible content from others that aim to deceive, either intentionally (as in disinformation) or unintentionally (as in misinformation). The first one, disinformation, is a hazardous and far-reaching phenomenon able to bring about profound changes in any community's political, economic, and cultural framework and, in this way, undermine the foundations of democratic societies.

Although sometimes hoaxes could be relatively easy to disprove by specialized fact-checkers and domain experts, more efforts are needed to mitigate the vast flow of false content. This overgrowing need to develop new methods and tools to support information verification, particularly in the increasingly chaotic online ecosystem, has recently made disinformation analysis a popular area of research. Focusing on Artificial Intelligence, the most remarkable contributions to fight disinformation have arisen from the field of Machine Learning, given that the problem of automatic identification of disinformative content can be modelled as a supervised classification problem —-that is, we can tell the likely truth or falsehood of new content by applying a model trained with examples already labelled as true or false. Open social networking platforms (e.g., Twitter) have been the main target of these research initiatives, given their ability to spread messages to broad audiences and permeate different communities quickly.

The most widespread form of disinformation in social and digital media to date has been text, so the main features considered to train machine learning algorithms are related to the grammar and content of the messages —-such as syntactic, lexical, stylistic, and semantic characteristics. Nonetheless, disinformative messages frequently incorporate images and video to increase their credibility, so in recent years, multimedia content has been incorporated into machine learning models to improve the accuracy of hoax detection. Furthermore, the structure of networks spreading disinformation is also a focus of analysis since there are groups of users and content dissemination paths more prone to nurture disinformation than others.

The principal challenge we can anticipate in applying Machine Learning methods to this problem is that disinformation is a very heterogeneous and dynamic phenomenon, strongly connected to society and human behavior and with conflicting actors and approaches. This situation limits the effectiveness of Machine Learning models, which are typically highly dependent on labelled datasets and cover particular events, thus making it difficult to extend the implemented models to other hoaxes.

1.1 Purpose and Scope of the Document

This report provides an overview of the Artificial Intelligence techniques used in the literature to characterise and detect disinformative content in social media automatically. We aim to summarise the current state of the art in this field and to determine whether the results achieved so far are enough to be used by different practitioners, particularly fact-checkers.

The report is framed within the Iberian context of IBERIFIER. Therefore, much of the effort will be to study the tools developed by the partners of the observatory, and to highlight the available datasets in Spanish and Portuguese.

To introduce the more technical concepts and terminology to the reader, the first part of the report will describe artificial intelligence techniques and algorithms that will then be applied to combat disinformation in its different forms.

1.2 Structure

This document is organised as follows:

- **Section 2** describes the IBERIFIER context inside the EDMO hubs and the main purposes of the activities of the project.
- **Section 3** describes background concepts in Machine Learning, Natural Language Processing and Social Network Analysis.
- **Section 4** reviews progress in the fight against disinformation from an Artificial Intelligence and Machine Learning perspective.
- **Section 5** presents relevant datasets for training learning models to identify disinformation and disinformative users and communities, with particular emphasis on datasets in Spanish (no datasets in Portuguese were found in our study).
- **Section 6** describes software tools useful in the information analysis and verification process, ranging from the more generalist tools (such as those that allow the creation of natural language models) to more domain-specific tools designed directly for the detection of false information, among others.
- **Section 7** presents the conclusions of this report and its main takeaways.

1.3 Terminology

Disinformation is a hazardous and far-reaching phenomenon, capable of causing profound changes in any community's political, economic, and cultural framework and thus undermining the foundations of societies around the world, either intentionally or through unconscious mistakes. There is no consensus for a standard definition of this phenomenon involving untruthful information. In the Anglo-Saxon world, the most acknowledged classification is the one proposed in (Wardle & Derakhshan, 2017), which divides the alteration or manipulation of information into three typologies:

- **Misinformation**: information that is false or misleading but not intended to cause harm.
- **Disinformation**: malicious false information, i.e., with a motivation to cause harm.
- **Malinformation**: information that is truthful but disseminated with the aim of causing damage.

This report discusses applications to combat disinformation, leaving aside the concept of "malinformation".

It is worth mentioning that, in the literature, the term "fake news" has been used indiscriminately to refer to this phenomenon. In IBERIFIER, we consider its use to be incorrect because a piece of news, by definition, is understood to be contrasted, so there is no such thing as fake news. Despite the clarification, the word "fake news" is used in this document because many previous works employ it, and therefore, we have considered appropriate to keep it so as not to alter the completeness of the analysis.

2 Context

This report has been developed within the framework of IBERIFIER, the Iberian Digital Media Research and Fact-Checking Hub[1], coordinated by the University of Navarra and made up of twelve universities, five verification organisations and news agencies, and six multidisciplinary research centres. Its primary mission is to analyse the Iberian (Spanish and Portuguese) digital media ecosystem and tackle the problem of disinformation.

IBERIFIER is one of the hubs of the European Digital Media Observatory (EDMO)[2], a publicly-funded platform led by the European University Institute in Florence (Italy) that brings together fact-checkers, media literacy experts and academic research to deal disinformation. The initial network includes eight media and disinformation observatories approved by the European Commission to bring together all the regions that constitute Europe in the fight against disinformation:

- IBERIFIER.
- Ireland hub (EDMO Ireland[3]).
- Belgium-Netherlands Digital Media and Disinformation Observatory (BENEDMO[4]).
- Central European Digital Media Observatory (CEDMO[5]).
- The Nordic Observatory for Digital Media and Information Disorder (NORDIS[6]).
- Belgium-Luxembourg Research Hub on Digital Media and Disinformation (EDMO BELUX[7]).
- Observatoire de L'information et des Medias (DE FACTO[8]).
- Italian Digital Media Observatory (IDMO[9]).

Recently, six more observatories have been invited to join the EDMO network, thus covering all the countries and areas of influence of the European Union:

- LAKMUSZ – EDMO Hungarian hub against disinformation
- GADMO – German-Austrian Digital Media Observatory
- BROD – Bulgarian-Romanian Observatory of Digital Media
- MedDMO – Mediterranean Digital Media Observatory (covering Greece, Malta and Cyprus)
- ADMO – Adria Digital Media Observatory (covering Croatia and Slovenia)
- BECID – Baltic Engagement Centre for Combating Information Disorders (covering Estonia, Latvia and Lithuania)

The EDMO hubs aim to combat disinformation and misinformation by developing media literacy actions, publications, reports, tools and fact-checks. To achieve these objectives, a series of activities have been defined. IBERIFIER tasks are organized into five different dimensions of the disinformation problem:

[1] https://iberifier.eu
[2] https://edmo.eu/
[3] https://edmohub.ie/
[4] https://benedmo.eu
[5] https://www.cedmohub.eu
[6] https://datalab.au.dk/nordis
[7] https://belux.edmo.eu/
[8] https://defacto-observatoire.fr
[9] https://www.idmo.it

- **Activity 1: Digital media research.** This work package aims to map digital media in Spain and Portugal and research the different aspects and trends of news and misinformation consumption.
- **Activity 2: Fact-checking.** The purpose of this work package is to develop partnerships and coordinate activities between the different fact-checking partners and platforms, to create a repository of fact-checks, and to develop tools for fact-checkers.
- **Activity 3: Computer and data research.** This work package aims to map existing technologies to combat disinformation in the Iberian scenario, characterise disinformation and its propagation and support the development of technological tools based on artificial intelligence for fact-checkers.
- **Activity 4: Strategic analysis.** The purpose of this work package is to contribute to strategic analyses of the impacts of disinformation on several interest areas.
- **Activity 5: Media literacy, communication and dissemination.** This work package consists of the dissemination and promotion of all other activities through media literacy, publication of reports, scientific articles, good practice guides, etc.

This report aims to reflect the progress made in Activity 3, specifically in task T3.2. The mission of this task is to map the existing Artificial Intelligence technologies within the Iberian landscape, also including the most relevant datasets to research disinformation. The working team for this activity comprises four academic institutions: Universidad de Granada (UGR), Universitat Politècnica de València (UPV), Universidad Politécnica de Madrid (UPM) and Barcelona Super Computing Center (BSC). The scope of this task is in line with the rest of the European hubs that are also studying state of the art in terms of the tools available to combat disinformation and the technological challenges to be addressed in this fight for objective and truthful information.

3 Machine Learning Technologies

In the age of the information society, in which we are immersed, millions of data points are being generated each minute around the world. This massive amount of data is continuously growing, and there is no sign that the trend will change. This data explosion has enormous potential for economic and social disruption, and unsurprisingly, the phrase “data is the new oil” has been echoed for some time now by scientists, entrepreneurs and news media.

Nevertheless, data is as valuable as it is leveraged into knowledge and put into action. The concept of Big Data, booming until recently, explained the creation of value that occurs when data is converted into knowledge (Mayer-Schönberger & Cukier, 2013). Accordingly, along with the increment of available data, Artificial Intelligence methods have emerged as a suitable toolbox to address this challenge. In the last decade, Machine Learning, a subset of Artificial Intelligence focused on finding relevant patterns in data to build prediction models, has become the prevalent approach to data exploitation problems. Machine Learning has been applied over the years to multiple disciplines, including misinformation and disinformation detection (Bondielli & Marcelloni, 2019; Choraś et al., 2021). Progress in the area has been boosted by neural network techniques, which have shown significant capabilities to analyze, summarize, and make predictions on large volumes of data.

Next, we describe the concept of Machine Learning, the different approaches to the subject, and the most successful techniques in the field, with a particular focus on neural networks.

3.1 Foundations

Machine Learning encompasses several methods, techniques, and tools aimed at making machines *more intelligent* as they are presented with more data about a given problem. While the definition of the term *intelligent* is elusive, a soft version of it simply meaning algorithmic processes that extract and exploit useful data patterns is widely accepted. In this section, we briefly introduce Machine Learning from a historical and practical perspective.

An early predecessor of Machine Learning is Knowledge Discovery from Databases (or KDD), which was coined in the early 90s to frame research works aimed at the processing and extraction of useful knowledge from the increasing volumes of data managed by organizations. KDD is defined as a non-trivial process of discovering useful knowledge from data (Fayyad, Piatetsky-Shapiro, & Smyth, 1996). The KDD process is composed of the following steps:

- selection of data from a larger set;
- preprocessing of the data;
- transformation;
- data mining;
- evaluation and interpretation of those patterns which will result in knowledge.

The core of the KDD process lies in the data mining stage, which consists of applying different algorithms to extract patterns. Machine Learning refers to a specific set of techniques applied in this phase.

One of the first approximations to a definition of Machine Learning appeared in the early years of Artificial Intelligence by Samuel (1959), who stated that Machine Learning consists of programming computers to learn from experience to do away with the effort involved in explicit programming. Mitchell (1997) provided a more formal definition of Machine Learning, which stated that a computer

program learns from experience if its performance on a task, as measured by a quantitative metric, improves over time.

Based on these two definitions, we can conclude that Machine Learning is a subfield of Artificial Intelligence that aims at building computational systems that improve automatically through experience. In other words, Machine Learning involves the development of algorithms capable of learning patterns from sample data to be applied to new data. Furthermore, these patterns have two primary purposes: forecasting future events (predictive) or gaining knowledge from the data (descriptive). Depending on how experience is collected (learning) and patterns are elicited, Machine Learning techniques have three families: Supervised Learning, Unsupervised Learning and Reinforcement Learning.

3.1.1 Supervised Learning

The main goal of Supervised Learning is to generate a model that, from a set of a priori known labelled example data, can predict the label of unseen or future examples[10].

There are two basic types of Supervised Learning problems according to the type of target variable:

- When the target variable takes a (usually small) number of discrete values, we have a classification problem. Classification algorithms create a model that learns from labelled examples in order to predict the textual label (or class) of new instances not seen during training.
- When the target variable is continuous, we have a regression problem. That is, regression differs from classification by offering a quantitative response rather than a qualitative one.

3.1.2 Unsupervised Learning

Unsupervised Learning refers to Machine Learning techniques in which we do not know the output for the input examples[11].

The most widely-used technique in Unsupervised Learning is clustering. Clustering algorithms group different data into multiple groups in such a way that objects within a cluster are very similar to each other, but very dissimilar to objects in other clusters. These similarities and dissimilarities are evaluated in terms of attribute values, often using distance measures (Han, Kamber, & Pei, 2012).

Another well-known technique that falls into the family of Unsupervised Learning is association rules. The purpose of association rules is to identify and represent the dependencies between a set of items in a database, i.e., to find co-occurrences of items or events (Agrawal, Imieliński, & Swami, 1993). The form of association rules is $A \rightarrow C$, where A and C are a set of items (or itemsets) meaning that C (usually) happens when A happens (Adamo, 2001).

[10] Formally, given a training set X_i, Y_i composed of n training examples $i = \{1, ..., n\}$, where $X_i = \{x_1, ..., x_n\}$ are the input variables (the features) and $Y_i = \{y_1, ..., y_n\}$ the output or target variables (the labels), Supervised Learning algorithms build a function $f : X \implies Y$ such that $f(x_i)$ is a good predictor of the corresponding value of y_i.

[11] Given a set of unlabeled examples $X = \{x_1, ..., x_n\}$, the goal of unsupervised learning is to find a function $f(X)$ that provides a compact description of the set of examples.

3.1.3 Reinforcement Learning

Reinforcement Learning is based on an agent's trial and error interactions with its environment. Therefore, Reinforcement Learning is a process of iterative learning in which the agent performs a series of actions and favours one behaviour against another based on how they align with its goal. The agent is deployed within an environment where it receives rewards if its actions lead to desirable states. Hence, the agent's goal is to discover which actions lead to maximising that reward.

Reinforcement Learning was born in the late 1980s when the two main approaches on which it is based converged. The first one was the study of trial and error learning based on behavioural psychology and classical conditioning. The second one was the problem of optimal control introduced in the 1950s. Unlike Supervised Learning, which uses labelled examples to learn, in Reinforcement Learning the agent must be able to learn from its experience, which is usually unlabelled data. Reinforcement Learning also differs from Unsupervised Learning: while the latter looks for patterns to describe the data, Reinforcement Learning is about maximising a reward.

Some classical methods used in Reinforcement Learning are Monte Carlo methods and Temporal-Difference Learning (Sutton & Barto, 2018). As the complexity of the environments grows, these Reinforcement Learning methods become unfeasible. In recent years, the combination of Deep Learning and Reinforcement Learning, namely Deep Reinforcement Learning, has made it possible to scale Reinforcement Learning to numerous applications in real and complex environments (Mnih et al., 2013). In this setup, neural networks are used to: (i) estimate the expected reward of an agent after performing a sequence of actions, (ii) encode and learn the decision rules that govern the agent, (iii) obtain a numerical representation of the perceived environment that better suits to the agent purposes.

3.1.4 Other Machine Learning approaches

There are Machine Learning paradigms that differ from the previous three principal approaches. One of them is Semi-supervised Learning, which is halfway between supervised and unsupervised learning. Most Semi-supervised Learning strategies involve improving the performance of Supervised Learning or Unsupervised Learning by using another paradigm as a complement for either of them (Engelen & Hoos, 2020). For example, in a semi-supervised classification problem, both labelled and unlabelled data are used to improve the classification process with additional information. Clustering problems can also be tackled with this approach, e.g., by shaping the groups with a number of constraints regarding which instances can belong to a group and which ones cannot (X. Zhu & Goldberg, 2009).

The main trend nowadays in Machine Learning is Deep Learning, initially conceived within Supervised Learning and recently extended to other paradigms. Deep Learning is an evolution of classical neural network systems, which are computational models inspired by the early conceptualization of the human brain and can learn from example data. The main feature of Deep Neural Networks is the use of multiple computation stages arranged in stacked layers. The use of such *hidden layers* is based on the assumption that more complex high-level features can be built by combining simpler lower-level features. Generally, the greater the number of hidden layers, the greater the hierarchy of features learned by the network. Section 3.3 will describe the main characteristics and methods of this type of learning.

3.1.5 Trends, opportunities and challenges

Machine Learning has experienced spectacular progress in the last decade, moving from laboratory environments to playing an essential role in our lives. It has the potential to be applied for countless tasks, such as medical diagnosis, anti-spam filters, voice recognition, recommendation systems, facial recognition, robotics, autonomous driving, and many more. One of the drivers of progress has been the capacity to acquire, store, and process massive data, i.e., big data. This requires Machine Learning algorithms to be computationally efficient in terms of running time and use of data. In this regard, one of the current trends in Machine Learning is improving classical Machine Learning algorithms to make their time requirements manageable when faced with large volumes of data. Likewise, storage and distributed computing resources must be properly managed to cope with this massive data explosion (Jordan & Mitchell, 2015).

As with any other branch of science and technology, Machine Learning also presents some legal and ethical issues. The first relates to data ownership, as many companies currently capture data for specific profit-making uses. Companies often obtain this data without explicit consent to its use or financial compensation to the owner. Other questions raised by these technologies, which often support decision-making, is whether these algorithms are fair or even respectful of human rights. One of the most representative examples is the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) software, which measures prisoners' recidivism risk. An investigation showed a bias as this software gave a higher probability of recidivism to an African-American offender than to a Caucasian offender regardless of the offence committed[12], i.e., there was a higher false positive ratio for African-American prisoners than Caucasian prisoners. Such algorithms are considered unfair since they favour an individual or group based on inherent or sensitive characteristics such as gender, religion, race, etc. Apart from the COMPAS case, there are many other examples of biases in Artificial Intelligence systems (Mehrabi, Morstatter, Saxena, Lerman, & Galstyan, 2021), such as in facial recognition, recommendation systems, automatic evaluation of CVs, etc.

Last but not least, another major problem many Machine Learning algorithms face is explainability, understood as the property of an Artificial Intelligence system to provide the user with the necessary information to understand how it behaves in certain circumstances, guaranteeing reliability, transparency, fairness, and ethics. Many Machine Learning algorithms are black-box algorithms, meaning that their inner workings and the reason for the output they provide are too complex to summarize or unknown. The most popular black-box methods are neural networks, particularly Multi-layer Neural Networks, Convolutional Neural Networks (used for image processing) and Recurrent Neural Networks (used for sequence data, such as time series and text). Consequently, the field of eXplainable Artificial Intelligence (XAI) has gained popularity in recent years (Linardatos, Papastefanopoulos, & Kotsiantis, 2021; Arrieta et al., 2020; Kaur, Uslu, Rittichier, & Durresi, 2023).

The European Union is exploring new legal formulas to address Machine Learning challenges. In particular, the *Report on Artificial Intelligence in a digital age*[13] establishes new standards and measures for the implementation of Artificial Intelligence. The conclusions of this report are expected to be applied in all Member States, establishing guidelines for the development and deployment of Artificial Intelligence systems based on the risk involved. Not surprisingly, one of the major concerns is automatic decision-making by *intelligent* algorithms, and one the high-risk categories is that of systems whose outcomes are not explainable.

[12] https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing
[13] https://www.europarl.europa.eu/doceo/document/A-9-2022-0088_EN.html

3.2 Machine Learning Techniques

In this section, we will focus on the foundations of Supervised and Unsupervised Learning, describing some of the basic approaches to them. We group them into the category of Machine Learning for the sake of simplicity. However, it is arguable whether some of these techniques belong to the Statistics or the Artificial Intelligence area. We explicitly leave out the section Deep Learning with neural networks, which is studied in Section 3.3.

This is the more technical section of this report and can be skipped if the reader is not very familiar with the mathematical language or, conversely, has a background in Artificial Intelligence.

3.2.1 Supervised Statistical Learning methods

Linear Regression is a simple Supervised Learning approach used to predict quantitative values. We start from a single predictor variable X to predict a quantitative response Y, and we assume that an approximate linear relationship exists between them. This approximate relation can be expressed as follow:

$$Y = \beta_0 + \beta_1 X + \epsilon \tag{1}$$

where β_0 and β_1 are two a priori unknown constants representing the intercept and slope, respectively, and ϵ a random error. Together (β_0 and β_1) represent the weights or parameters of the model.

Fitting the line to the input data is necessary to estimate the parameters (β_0 and β_1). To do that, we need to minimize the residual of the linear model, i.e., the difference between the observed response and the prediction. Some approaches to estimate β_0 and β_1 are least squares criterion, gradient descent, maximum likelihood, and lasso method.

Once estimations of β_0 and β_1 (denoted as $\hat{\beta}_0$ and $\hat{\beta}_1$) are obtained from the input data, we can predict $\hat{y}$ from a particular value of X, where $\hat{y}$ indicates a prediction of Y based on $x_i \in X$ (James, Witten, Hastie, & Tibshirani, 2021).

Nonlinear Regression. Linear models have limitations concerning predictive power since the linearity assumption is almost always due to an approximation, and often that approximation does not adequately fit the data. To achieve a substantial improvement in linear models, the linearity assumption must be relaxed, for which there are extensions of linear models as follows (James et al., 2021; Motulsky & Ransnas, 1987):

- Polynomial regression: It extends the linear regression model by adding additional parameters and exponents to the existent predictors according to a polynomial degree. Thus,

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots + \beta_n x_i^d + \epsilon_i \tag{2}$$

  where d is the polynomial degree. Polynomial regression has one main drawback: for very large values of d, the polynomial curve adopts very odd shapes and, therefore, is not a good generalizer. Accordingly, values of d greater than four are not normally used.

- Step functions: Instead of imposing a global structure with the model, the step functions split the variable X into distinct regions (or bins). We need to define the cutpoints

$(c_1, c_2, c_3, ..., c_n)$ to obtain an ordered categorical variable from a continuous variable. The resulting variables (from the original one) are called *dummy* variables. So the relationship with the predictors (*dummy* variables) and the response can be expressed as:

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + ... + \beta_n C_n(x_i) + \epsilon_i \quad (3)$$

where $C(x)$ is the dummy variable created from a range of two cutpoints.

- Regression splines: They are an extension of the two previous approaches. This regression technique involves two main steps. First, instead of using a high degree single polynomial over the whole domain, we can use different polynomials in regions. Using such piecewise polynomial regression generates points where the parameters change, namely knots. The problem with piecewise polynomial regression is that the result function is usually discontinuous in the knots. The second step involves the solution of the piecewise polynomial regression. The solution requires two restrictions: the function must be continuous and smooth at the knots. So, the definition of regression spline is a piecewise polynomial with continuity in the derivatives up to degree $d - 1$ at each knot.

Logistic Regression. Despite the name, logistic regression (Hosmer, Lemeshow, & Sturdivant, 2013; James et al., 2021) is a classification method because the output variable is discrete (usually binary, but multinomial too). To understand logistic regression, let us imagine a dataset whose output variable is binary (0 or 1). If we try to fit a regression line, we could predict values greater than 1 and less than 0. This does not seem appropriate for this kind of problem, so we need a function such that, for all values of X, the output ranges between 0 and 1. One of these functions is the S-shaped logistic function (sigmoid). This function expresses the probability of Y given X, so if the calculated output probability of an instance is greater than 0.5, it instance belongs to class 1 and vice versa. Mathematically, the logistic function is the following:

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

Logistic regression does not have residuals like linear regression, so we can not use the least squares method. Instead, in logistic regression, maximum likelihood is used to fit the function. The idea of the maximum likelihood method is finding an estimation of β_0 and β_1 such that the predicted probability $\hat{p}(x_i)$ corresponds as closely as possible to the observed predictors. The maths behind the calculus of maximum likelihood is out of the scope of this report.

Decision Trees is a supervised learning technique by induction (i.e., generalization from particular cases) that allows the identification of concepts (classes) from the features of a representative dataset (Ethem, 2020; James et al., 2021; Tan, Steinbach, Karpatne, & Kumar, 2019; Han et al., 2012). A decision tree has a hierarchical structure formed by nodes and edges. There are two types of tree nodes: decision nodes and leaf nodes. Decision nodes successively test the existence or not of characteristics in each instance, while the leaf nodes provide the output of the tree –the class of the instances. The sequence of characteristics and the values to test are calculated from training data to make the tree provide outputs as close as possible to the known classes.

Decision Trees can be used for regression and classification problems. The difference between the two, apart from the output, is the selection criteria to obtain the best splitter:

- With classification trees, the quality of a split is measured with an impurity measure, such as the entropy or the gini index. For example, the entropy of a node n is defined by:

$$I(n) = -\sum_{i=1}^{C} p_i(n) \log_2 p_i(n)$$

where C is the number of different class in the node n and $p_i(n)$ is the probability that the instances at node n belong to class C_i. The selection of the best splitter is made by choosing the one that produces the greatest decrease in the impurity of the analyzed node.

The Gini index is similar, but provides the degree of diversity of information at node n. Therefore, the lower the value of the Gini index, the lower the entropy and the better it will be as a splitter. The Gini index of a node n is defined by:

$$G(n) = 1 - \sum_{i=1}^{C} p_i(n)^2$$

Another selection criterion is the classification error or misclassification error:

$$E(n) = 1 - max_i[p_i(n)]$$

- In regression trees, the suitability of a split, like in line regression, is achieved by minimizing the residual sum of squares (RSS).

A common phenomenon in decision trees is overfitting, which means that the tree estimates correct outputs for the training dataset but not for unknown instances. To avoid this problem, decision trees implement a pruning mechanism. Pruning can be carried out during the tree construction process by defining a threshold of instances arriving at the node, or after the tree has been constructed by looking for those subtrees that cause overfitting, and these are pruned. The first approach has less computational cost, but the second approach is more efficient.

Some of the most popular algorithms for developing decision trees are ID3, C4.5 (evolution of ID3 that among other improvements implements a pruning mechanism) and CART (used for both regression and classification).

Rule-based Learning is an extension of first-order logic to handle relational representations. This classification method use a collection of if-then rules to classify instances. A classification rule has the following form:

$$\textit{if } C_1 \wedge C_2 \wedge \textbf{...} \wedge C_n \textit{ then } y$$

where y is the class label (the consequent of the rule) and $C_1, C_2, ..., C_n$ are the conditions, that is, a conjunctions of attributes (the antecedent of the rule).

There are two main families of methods to extract rules: by creating rules from other classification model, which are called indirect methods; and directly from data, namely direct methods.

- **Indirect methods for rules extraction.**

  Rules can be extracted from a decision tree by traversing the paths from the root node to a leaf node. The number of extracted rules from a tree is therefore equal to the

number of leaf nodes. Because the rules are extracted from a decision tree, they are mutually exclusive, as no conflict is possible between two rules being triggered at the same time, and they are also exhaustive as they do not overlap, i.e. they cover the entire instance space. Another indirect method for rule extraction is from neural networks (Fu, 1994). This method makes it possible to transform a black box algorithm such as neural networks into an interpretable model, which is one of the great advantages of rule-based learning.

- **Direct methods for rule extraction.**

  We can extract rules directly from data. The process is simple, given a set of instances of a given class (usually the positive class), the goal is to learn a rule that covers the maximum number of these instances in each iteration until a set of rules is obtained that covers (as far as possible) the entire space of instances of the positive class. This form of constructing rules is called sequential covering algorithm. There are many sequential covering algorithms (AQ, CN2, FOIL) but one of the most famous induction algorithm is Ripper (Cohen, 1995). This was the first rule learning method robust against overfitting. The main different with the others methods (besides a good pruning mechanism) was the post-processing for the rule set optimization. To measure the quality of a rule, the coverage of a rule (a rule covers an instance if the features of the instance satisfy the condition of the rule) and accuracy of a rule (measure by the fraction of instances satisfying both antecedent and consequent), are used.

Support Vector Machines (SVM) is a statistical learning algorithm and is part of a broad family of algorithms known as the Kernel Machine. The SVM is a generalisation of a simpler classifier, called the Maximum Margin Classifier based on the hyperplane concept (James et al., 2021; Ethem, 2020; Tan et al., 2019). In n-dimensional space, a hyperplane is defined as the flat affine subspace of dimension $n-1$. For example, the hyperplane of a two-dimensional space is a line. A mathematical definition of a hyperplane in an n-dimensional space is the following:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n = 0$$

If point X is not exactly on the hyperplane, the point does not satisfy the previous equation, and therefore the equation value is greater or lesser than 0. Therefore, we can use a hyperplane to classify instances that fall on one side or the other of the hyperplane. The maximal margin classifier uses this hyperplane idea to separate the data. To avoid an infinite number of hyperplanes, it uses the maximal margin hyperplane, i.e. the one with the farthest minimum distance from the training set. Observations that are equidistant to the maximal margin hyperplane are known as support vectors.

The support vector classifier relaxes the maximal margin hyperplane assumption by allowing some observations to lie within the defined margin or even on the wrong side of the hyperplane to avoid overfitting. In many cases, the data are not linearly separable and therefore, algorithms whose decision boundaries are non-linear are needed.

The Support Vector Machine is an evolution of the support vector classifier that allows the classification of instances that are non-linearly separable. The underlying idea of support vector machines is to apply transformations to the data so that the data is linearly separable by a hyperplane. To avoid applying these transformations to the original data, the support vector machine employs a function called kernel. The kernel function measures the relationship between every pair of instances as if they were in a higher dimension. The kernel function

does not apply the transformation to the data. The kernels systematically find support vector classifiers in high-dimensional spaces. Some of the most famous kernel functions are:

- Polynomial kernel:

$$K(a, b) = (a * b + \theta)^d$$

where a and b are two instances of the dataset, θ is the coefficient of the polynomial and d, the degree of the polynomial.

- Radial Basis Function (RBF) kernel:

$$K(a, b) = e^{-\gamma(a-b)^2}$$

where γ is a positive constant.

k-Nearest Neighbor (K-NN) belongs to a family of algorithms called instance-based learning. Such algorithms explicitly use the training set to make predictions, i.e. to predict a new instance they use a distance function to determine which training instance is closest to the unknown test instance (Witten, Frank, Hall, & Pal, 2017; Han et al., 2012).

K-NN algorithm labels an unknown test example searching the k training instances closest to it. The k training instances are the k nearest neighbors of the unknown instance.

The nearest instance is defined in terms of distance. There are several distance metrics, being one of the most popular the Euclidean distance. The Euclidean distance between two points in the n-dimensional feature space is:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Other distance measures are the Manhattan distance, which belongs to the family of Minkowski distances (like the Euclidean distance) and is used in the case of numerical variables, and the Hamming distance which is used for text data types.

When the k-NN algorithm is applied to a dataset, it is usually normalised beforehand, because if certain features have very large values relative to the rest, the distance measure will be biased by these large values.

Ensemble methods are the combination of several other learning methods. In general, they improve the accuracy of individual learning models.

The logic behind the ensembles is simple. An ensemble consists of combining a set of basic machine learning algorithms, training them, and using the models on the same dataset to predict the results. In the case of classification problems, the consensus is reached by voting, while in regression algorithms, the arithmetic mean is usually used.

The most well-known ensembles are bagging, boosting and random forest (Tan et al., 2019; Witten et al., 2017; James et al., 2021).

- **Bagging or Boostrapping**

The bagging procedure was proposed Breiman Breiman (1996). It consists of the combination of multiple machine learning algorithms, typically decision trees, to reduce the variance of the model. To obtain this reduction in variance, a statistical tool known as bootstrap is used. The procedure is simple: from a single training set T, T_i bootstrap samples of the same size as the initial set are created. Sampling with replacement is used for this purpose. In theory, averaging a set of independent observations reduces the variance and the bagging procedure simulates this with the bootstrap samples.

To apply bagging, M algorithms (usually decision trees) are constructed using M bootstrap training sets. If the problem is a regression problem the results are averaged while if the problem is a classification problem the majority vote is taken.

- **Random Forest**

  Also proposed by Breiman (2001), Random Forests are an improvement on the bagging procedure. As in bagging, M decision trees are implemented in M bootstrap samples with the difference that each time a split in a tree is considered, a subset of the features or predictors is randomly selected as split candidates. Then, an attribute is chosen within that subset that reduces as much as possible a measure of impurity for splitting. Thus, in the random forest not only the bootstrap training instances are manipulated, but also the selection of the input attributes. Typically the number of attributes considered in each split is the square root of the total number of attributes.

  The prediction is the same as in bagging, i.e., the arithmetic mean of the prediction of each tree for regression and, the majority vote for classification.

- **Boosting**

  Boosting is a sequential procedure where each tree grows using information from the previous trees. Like bagging (and of course random forest), the boosting method is usually implemented with multiple decision trees. Like the two previous methods, it can be used for both regression and classification. The main difference is that this iterative procedure adaptively changes the distribution of training examples for learning trees.

  The boosting procedure works as follows: we start from an initial dataset in which each example is assigned a weight (in the first iteration all training examples have the same weight); then a subset of data is selected by sampling with replacement taking into account the assigned weight, i.e. the weight acts as the probability that an example is chosen; then the classifier is implemented and trained with the obtained sample; the trained model is used to classify all instances of the original dataset; once the results are obtained, the weights of the dataset are updated such that incorrectly classified examples will have their weights increased, while correctly classified examples will have their weights decreased; as the boosting rounds are completed, those examples that are more difficult to classify will become more frequent; once the ensemble is trained, prediction is carried out by weighted voting of the classifiers (a classifier that performs well on the data on which it was implemented will receive a higher weighting than one that has performed worse).

3.2.2 Unsupervised Statistical Learning methods

Within unsupervised learning there are two main families of algorithms: association rules and clustering methods. One of the main subdivisions of clustering techniques are partitional clustering, where the clusters obtained are disjoint and normally cover the whole set of items, and hierarchical

clustering where, as the name suggests, a hierarchy of nested clusters is obtained in which each cluster at one level is subdivided into several at the next level.

K-means clustering is a partitional clustering method where each group is represented by a prototype in terms of a centroid (when the data features are continuous) (Tan et al., 2019; Han et al., 2012; James et al., 2021). The centroid is the mean of a group of points in n-dimensional space (in the case of categorical variables the most representative pattern is often the medoid).

The K-means method is simple, first, we need to define the number of initial K clusters and K centroids. Then, each point is assigned to the nearest centroid to form initial clusters. The centroid of each cluster is updated iteratively according to the points assigned to the cluster, the updating process is repeated until the centroids do not change.

Proximity measures are used to assign points to the nearest centroid, depending on the type of data. Some of the most common proximity measures are the Euclidean distance, the Manhattan distance, the cosine similarity measure, and the Jaccard measure.

If we use the Euclidean distance as a measure of proximity, the sum of the squared error (SSE) is used to measure the goodness of clustering, which must be minimised to achieve the best fit.

DBSCAN is a clustering method based on density analysis where high-density regions of space that are separated by low-density regions of space are analysed (Tan et al., 2019; Han et al., 2012).

There are several ways to define the concept of density in a region of space. In particular, DBSCAN is based on the centre-based approach. In this approach, the density is estimated for a particular point by counting the number of points that are within a space determined by a prefixed radius (Eps). The density depends on the specified radius. Another parameter to be specified before applying the DBSCAN clustering algorithm is the minimum number of points (MinPts) required for a region to be considered sufficiently dense to form a cluster.

The centre-based approach allows us to classify every point in the pattern space as:

- Core points: a point is considered core if there is a minimum number of neighbouring points (MinPts) fall within a specified radius (eps).
- Border points: these are those that lie within an eps radius environment that has one or more core points as its centre.
- Noise points: these points are located in very sparse regions and are neither core nor border points, i.e. points that are not part of the cluster.

Hierarchical clustering is a succession of nested partitions represented in an intuitive structure called a dendrogram (similar to a tree) (Tan et al., 2019; Han et al., 2012). There are two approaches to building a hierarchical clustering (Kaufman & Rousseeuw, 1990; Roux, 2018):

- Agglomerative: it starts with individual items in which each item forms a cluster and in each step these are joined together to form larger clusters (bottom-up strategy). Cluster proximity measures are needed to join clusters into larger clusters. Some of theses proximity measures are: MIN defined by the proximity between two closets points that are in different clusters, MAX where the proximity is measured by the farthest two points in different clusters, group average defined by the average distance of all pair of points, distance between cluster centroids (where each cluster is represented by a centroid) and the Ward's method which each cluster is also represented by a centroid, but the

proximity measure is achieve in terms of minimizing the sum of the squared error (SSE) like k-means.

- Divisive: start with all the items in the same cluster and, at each step, divide them into smaller clusters until only individual items are remaining in clusters (top-down strategy). At each step we obtain a bipartition of the former cluster. There are several splitting procedures, for example using k-means with $k = 2$, another approach is selecting the two most dissimilar points of the cluster to be split and using them like seeds to build the new clusters, then aggregate to this seeds the points which are closer. Divisive hierarchical clustering, the same proximity measures as in agglomerative clustering's can be used to evaluate dissimilarities between clusters.

Association rule learning

Another approach to unsupervised learning is association rules (C. Zhang & Zhang, 2002; Tan et al., 2019; Han et al., 2012), which are used for searching and extracting frequent patterns in databases. Formally an association rule is defined as follows, given a set of (unlabelled) examples or transactions, an association rule is an implication expression of the type $X \rightarrow Y$, where X and Y are a set of items (itemset). The previous expression is interpreted as if X then Y, meaning that transactions containing X tend to contain Y. An item is an attribute-value tuple, an itemset is a collection of 0 or more items and the transactions define particular instances of relationships between items, i.e. a transaction is a subset of selected items from the total set of items.

The interest of an association rule can be measured in terms of its support and confidence. The support of a rule is the frequency with which the itemset forming the rule occurs for the total number of transactions in the database and the confidence determines the frequency with which items of the consequent appear in transactions containing the antecedent. The formal definition is:

$$support = \frac{(X \cup Y)(t)}{T} = P(X, Y)$$

$$confidence = \frac{(X \cup Y)(t)}{X(t)} = P(Y|X)$$

where t is a transaction and T is the transaction database. Confidence can also be measured based on the support of the rule divided by the support of the antecedent itemsets.

The generation of association rules can be reduced to the extraction of frequent itemsets and from these, generate strong association rules, i.e. rules that satisfy a minimum support and confidence threshold. The threshold for support and confidence is predefined by the user.

One of the classic algorithms for generating association rules is the Apriori algorithm. This algorithm employ an iterative level-wise search approach where from a set of frequent k-itemsets (at the first level $k = 1$) a set of candidate itemsets $C(k + 1)$ are generated and are evaluated based on their support. And, if the candidates satisfy the minimum support threshold, they will be considered frequent itemsets and will be used to generate a new set of candidates in the next iteration. This process will be repeated until no more frequent k-itemsets can be found.

3.3 Deep Learning

Neural networks emerged as an attempt to emulate the biological brain as it has always been associated with great computational capacity. If we look at the brain as a machine, it would be a very complex machine, non-linear, presenting massive parallelism over a distributed representation.

In 1958, F. Rosenblatt published his work on perceptrons, and established the first architecture of an artificial neural network. The perceptron is a basic type of artificial neural network consisting of a series of input nodes representing the input attributes, and a single output node to represent the output of the model. Each input node is connected to the output node by a weighted link. This link is used to simulate the strength of the synaptic connection between neurons. The output node is a mathematical function that calculates the weighted sum of the inputs, adds a bias factor to the sum and finally examines the sign of the result to produce the output. This function that examines the sign is called the activation function.

Neural networks are powerful classification models capable of learning complex and non-linear decision boundaries from data.

Neural network-based algorithms have affected the modern machine learning scene. In the last 15 years, they have experienced a huge boost with the emergence of deep learning, driven by today's high computational power and the availability of massive datasets.

The increase in the size of models (deep neural networks) and larger datasets has revolutionised the field of machine learning. Deep neural networks have a larger number of intermediate layers and more units (neurons) within each layer than conventional neural networks. This has allowed deep neural networks to be able to represent much more complex functions exploiting hidden information in massive datasets. The rise of deep learning has had a great impact both in the world of industry and, of course, in the field of research, and since the last decade there have been a large number of scientific papers applying deep learning to different disciplines (Islam, Liu, Wang, & Xu, 2020; Abdullah & Ahmet, 2022; Liu, Tantithamthavorn, Li, & Liu, 2022; C. Ma, Zhang, Guo, Wang, & Sheng, 2022).

3.3.1 Multilayer neural networks

Perceptrons have a limitation. There are certain types of functions that they are not able to approximate, for example, a perceptron does not properly classify a set of examples that is not linearly separable. To solve this problem, more layers are added to the network, so a multilayer neural network usually has more than one hidden layer and many units per layer.

In order for the neural network to make predictions from a data set, it needs to be trained. The training process is summarised simply, the network has to adjust the weights so that the prediction is as close as possible to the training instances. The key algorithm for the network learning the weights is the backpropagation algorithm (Rumelhart, Hinton, & Williams, 1986). The objective of this algorithm is to minimise the cost function by adjusting the weights and bias of the network by propagating the error from the network output to the input and using the gradient descent optimisation technique to calculate the value of the weights that minimise the error.

As in the rest of the machine learning algorithms, there are different cost or loss functions. Some of the most common ones are the mean square error (MSE),

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

and the entropy,

$$H = -\sum_{i=1}^{n} y_i \log \hat{y}_i$$

where n is the number of training examples, y_i is the real output and $\hat{y}_i$ is the networks' prediction.

Gradient descent is used to iteratively adjust the weight and bias parameters to minimise the cost function. This is done by taking the partial derivatives of the error concerning each parameter. For example, if we take the weight at a given instant $w(t)$ as the parameter to be updated, the equation is as follows:

$$w(t+1) = w(t) - \alpha \frac{\partial E(w)}{\partial w}$$

where E is the cost function and α is the learning rate that determines how quickly the parameters are updated. Each of the iterations over the dataset to adjust the network parameters are called epoch. Usually, the dataset is divided into batches to reduce the parameter update time.

The output of the network (and of each of the internal units) is calculated as a linear combination of the inputs to which an activation function is then applied,

$$f\left(\sum_i w_i x_i + b\right) \tag{4}$$

where $f(z)$ is an activation function, which allows adding non-linearity to the neural network. Some of the best-known activation functions are the sigmoid, hyperbolic tangent, softmax, and rectified linear unit (ReLU).

In deep neural networks, the recommended default activation function in the hidden layers is the ReLU function, as these are quasi-linear functions that are easier to optimise with gradient-based methods and generalise better (I. Goodfellow, Bengio, & Courville, 2016). Another important change that has improved the performance of deep neural networks is the change from the mean square error cost function to the family of cross-entropy loss functions.

3.3.2 Convolutional neural networks

Convolutional neural networks (CNNs) (LeCun et al., 1989) are a specialised type of feedforward neural network for processing data with a known topology, such as images. The name CNN is given because it involves a mathematical operation called convolution, which is a specialised type of linear operation.

The convolution operation applies a filter or kernel to an input argument such as an image pixel matrix. This filter is applied by multiplying a section of the input pixel matrix by a matrix of weights. The resulting matrix, called feature map, is obtained by translating the filter over the entire image.

Each convolution layer consists of a large number of filters that will vary according to the value of the weights (James et al., 2021; I. Goodfellow et al., 2016; Witten et al., 2017).

In CNNs there are other commonly used layers, such as the pooling layer. This layer is used is used to reduce the dimension of the input matrix (feature map). To do this, it uses a function to summarise the sub-regions of the matrix. These functions are usually the maximum value (max pooling), the minimum, the average (average pooling) or the sum of the values of the sub-regions of the matrix. Finally, the last layers of a CNN are usually fully connected layers that work as classifier where learn the high-level features represented by the outputs of the convolutional layer.

3.3.3 Recurrent neural networks

Recurrent neural networks (RNNs) are a type of neural network specialised in processing sequential data such as text or time series. The hidden layers of an RNN are connected in a directed cycle (itself and with other hidden units) of each hidden units, i.e. they make a decision based on current and previous input. RNNs are typically used for applications such as language modeling, machine translation, time series prediction and even to image processing and video sequence analysis.

RNNs are not good at capturing dependencies when the network is very deep, i.e. they have no long-term memory. This occurs because the gradients of the cost function decrease exponentially as it propagates through the layers and approaches zero making the learning task difficult. This is known as the Vanishing Gradient Problem. To solve this problem long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) neural network was introduced. These networks identify relevant short and long-term information and discard the rest. LSTM incorporate an element called a memory cell that allows data to be added to or removed from memory and gates units that control the memory cell. Each gate controls the information to be removed, the input to update the memory state, and the output (which is based on the input and memory state).

Sequence models still have the problem of loss of relationship between very distant elements of the sequence. The current sequence model is based on complex architectures with an attention mechanism. Attention in sequence models allows calculating the representation of the sequence by relating different positions of the sequence to each other. But these models combining recurrent networks and attention mechanisms have a high computational cost. In 2017, a new network architecture called Transformer (Vaswani et al., 2017) is proposed that removes the recurrent neural networks and keeps only the attention mechanism. Transformers are more parallelizable and more efficient in training time. Currently, many of the major natural language models are based on Transformers such as BERT or GPT-3, among others.

3.4 Natural Language Processing

Natural Language Processing (NLP) is a set of computational linguistics methodologies aimed at automatically extracting precise information from text written or spoken in a given language (Manning & Schutze, 1999). Its applications are varied, but such procedures appear particularly suited to analysing, from various perspectives, the large flow of information circulating online and on social network platforms. Given the unstructured nature of natural language, before building a Machine Learning we need to address three main challenges: text preprocessing, to clean the text and remove uninformative units; feature extraction, to numerically quantify relevant aspects of the text; and creation of semantic representations, to build a meaningful numerical encoding of the text (namely, *embeddings*). Once the text is properly encoded, it is ready for the resolution of Machine Learning downstream tasks, such as classification, prediction, or translation.

3.4.1 Text preprocessing

After extracting a corpus of texts and before any analysis or training of a model, any Natural Language Processing procedure requires a preparatory phase called text preprocessing, which allows only the portion of information useful for the type of analysis to be filtered from the raw data. In this regard, one can generally observe various methods of text preparation, and various combinations thereof. The most basic techniques are:

- tokenization, i.e. splitting the input raw texts into single units, called tokens;
- lemmatisation and stemming are the reduction of the inflected form to the root form, morphological or not respectively;
- stop word removal;
- removal of duplicates;
- calculation of readability criteria, such as the minimum number of unique words, the threshold for the proportion of tokens to other elements, etc.

Other frequent preprocessing operations, especially in the context of social content analysis, consist of the removal of certain precise elements from the raw corpus that may penalise analysis and introduce statistical noise: URLs, user mentions and hashtags, punctuation, non-alphanumeric characters, numbers, and infrequent terms. Also, it is possible to deal with character flooding —the repetition of characters within the word—, and conversion of emojis into tokens to better interpret their meaning. In addition, there are more complex procedures usually applied to larger texts: parsing —i.e., phrase and word detection—, part-of-speech tagging —identifying lexical categories—, named entity recognition —finding relevant locations, people, etc.—, disambiguation —distinguishing between homonyms—, and coreference resolution —detecting the different kinds of mentions to certain entities and linking them.

3.4.2 Feature extraction

The next phase typically consists of extracting features that can represent in a machine-readable format. This conversion can take place through the application of various methodologies, including vectorisation with bag-of-words, sentiment analysis, and embeddings.

Bag-of-words is a straightforward feature extraction method based on measuring the frequency of tokens, characters, or a combination of both. Contiguous elements can be grouped together to generate the so-called n-grams, that is, an n-gram is a sequence of text elements with an associated frequency or probability of occurrence. The most popular counting metric for n-grams is the term frequency-inverse document frequency (TF-IDF). TF-IDF considers how many times an element set appears within a selected text, and also penalises those tokens that appear globally with high frequency, i.e. that are too frequent to be helpful to uniquely characterize a piece of text.

Sentiment analysis aims at automatically grasping the emotions conveyed by a text. Lexicon-based approaches are common to address this task: first, a sentiment value is assigned to each text unit, based on a ‘dictionary’ of terms with associated predefined scores; second, the sentiment values are aggregated to obtain a global value of the larger piece of text. Some well-known linguistic resources that support sentiment analysis are: the NRC Emotion Lexicon (Mohammad & Turney, 2013) —which considers eight basic emotional dimensional (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive); SenticNet (Satapathy,

Cambria, & Hussain, 2017) —which provides an emotional tag for more than 200,000 linguistic concepts according to a set of vector representations about both semantic and polarity dimension; and SentiWordNet (Baccianella, Esuli, & Sebastiani, 2010) —a version of the lexical resource WordNet annotated with sentiment categories.

Other features can be extracted following a similar approach to sentiment analysis. In particular, psychological features of (the author of) a text can be obtained by using specific-purpose annotated lexicons. Among the most popular methodologies for extracting this category of features, we can highlight two of them. The first one is the Linguistic Inquiry and Word Count (LIWC) (Pennebaker, Boyd, Jordan, & Blackburn, 2015), a software which maps the text into a dense representation composed of 73 psychologically-meaningful pre-defined linguistic categories —i.e., linguistic/grammars classes and mental processes of various kinds, like affective, social, cognitive, ect.). The second one and the Five Factor Model (John & Srivastava, 1999), which condenses the personality of an individual in five basic factors. This model is commonly applied following Neuman and Cohen's proposal (Neuman & Cohen, 2014) by calculating the semantic similarity between the input text and sets of benchmark adjectives associated with each basic factor.

3.4.3 Semantic representation

The features presented in the previous section mostly focus on the syntactic and lexical levels of the text. Sentiment analysis goes beyond structural aspects, but it is based on dimensions and scores defined a priori. Word embedding, in turn, aims at building a semantic representation of the text, attempting to condense all the facets and meanings of each word into a numerical vector. In contrast to sentiment lexicons, these embeddings are automatically calculated from the texts and capture their semantics. A typical example of proper embeddings is the following: if we have vectors encoding the concepts $< king >$ and $< man >$, $< king > - < man >$ would yield $< queen >$.

There are many techniques for the calculation of embeddings. The most common is Word2Vec (Mikolov, Chen, Corrado, & Dean, 2013), which learns the probabilistic relationships between words from the training data and generates a language model thanks to which it is then able to infer the context —i.e. the semantic representation— of a single token. When applying this technique, the learning phase can be carried out by means of two mirror-image architectures: a continuous bag-of-words (CBOW), which produces the word embeddings by formulating a prediction task of a given word (called "center word") given the other context words, i.e. the "surrounding words" which in turn have been associated with their own word embeddings; and Skip-Gram, which instead learns to infer the context words given the center word. Another method is Global Vector for Word Representation (GloVe) (Pennington, Socher, & Manning, 2014), which is based on the collection of statistics from the word co-occurrence matrix. GloVe computes the similarity between two tokens as the ratio of their respective co-occurrence probabilities with other benchmark words, each representing a precise context.

3.4.4 Language model building

Once the input words have been vectorised and condensed into a unique dense vector representing the whole document, the features can be given as input to a Machine Learning model. In its simplest form, the model will learn the relationships between these dense semantic representations in order to distinguish between target labels (e.g., *true* or *false*). Given the sequential nature of the text, the models that explicitly consider adjacent segments usually work better than those which do not. For

example, recurrent neural networks (RNNs) will retain a portion of information related to the previous tokens to make predictions about the current one.

However, basic RNNs have limitations when processing larger documents, because they might not be able to retain information when related segments are far away in the text. As mentioned in Section 3.3, RNNs have evolved into models that can handle longer-term dependencies such as long short-term memory (LSTM) and gated recurrent unit (GRU).

Although these neural networks overcome many of the problems of more classical techniques, and are potentially capable of achieving significantly better results, they still have limitations. The main one is that they shrink the representation of the input token sequence to summarise it into a single dense vector, which causes a loss of information. The attention mechanism, a special kind of memory that learns to retain the most important parts of the global sequence, has been proposed to overcome this weakness (Vaswani et al., 2017). A second issue is the adaptation of the classification and regression architectures to address more complex tasks, such as machine translation, text synthesis, or image generation from text. Broadly speaking, these tasks can be modeled as transformations between input and output sequences. Transformers are neural networks aimed at this purpose (Bahdanau, Cho, & Bengio, 2014). They integrate two parts which can be optimized together; namely, an encoder to calculate a compact representation of the input in a latent space (similar to embeddings), and a decoder to reconstruct an output sequence. These two sections are optimized together, and can also incorporate improvements such as multiple simultaneous attention mechanisms (which is called self-attention) and information about the precise position of the word within the sentence (positional embedding).

Currently, there are many public implementations of advanced neural networks combining attention mechanisms and transformers, as well as weight values calculated for predictions tasks in different languages from open corpora. Here we highlight two of them: Bidirectional Encoder Representations from Transformers (BERT) (Devlin, Chang, Lee, & Toutanova, 2019), and Generative Pre-trained Transformer (GPT) (Q. Zhu & Luo, 2022). Interestingly enough, it is possible to adapt these models with weights to other domains than the ones calculated in training, a process called transfer learning.

3.5 Social Network Analysis

Social Network Analysis (SNA) is a computational science approach focused on the study of relationships between social entities, as well as their patterns and meanings (Wasserman & Faust, 1994). The different connections or links (relationships) between the social actors of the network determine its structure (topology), allowing to analyze the existing synergies among them to extract insights. This discipline has benefited from the development of Social Platforms, such as Twitter or Facebook, as they facilitate the creation of a user' social network thanks to their functionalities and the recording of his/her social interactions. Currently, SNA is applied in several research areas, including healthcare (Smith & Christakis, 2008), marketing (Harrigan, Evers, Miles, & Daly, 2017), tourism and hospitality (Li, Xu, Tang, Wang, & Li, 2018), cyber security (Lalou, Tahraoui, & Kheddouci, 2018), politics (Panizo-LLedot, Torregrosa, Bello-Orgaz, Thorburn, & Camacho, 2019) or fake news detection (Vosoughi, Roy, & Aral, 2018), among others.

Graph theory is the traditional approach used to represent the content and interactions of a social network (Van der Hulst, 2009). A graph G = (V, E) is a mathematical model that is composed of a set of *nodes* or *vertex* $V = \{v_1, .., v_n\}$ and another set of *links* or *edges* $E = \{e_{ij} | v_i \in V \wedge v_j \in V\}$. The most common way of representing a graph is through its *adjacency matrix* (A), defined as a square matrix *n x n*, where *n* indicates the number of nodes from the graph, and each coefficient a_{ij}

satisfies the equation 5:

$$a_{ij} = \begin{cases} k, if\ e_{ij} \in E \\ 0, otherwise \end{cases} \quad (5)$$

In the classical graph model, k value always shall be equal to 1. However, when k value represents the weight of the relationship, it may have values different from 1; therefore, the graph is known as *weighted graph*. Finally, a graph is considered *directed* if $\exists i, j, a_{ij} \neq a_{ji}$, and *non-directed* if $\forall i, j, a_{ij} = a_{ji}$.

From an analytical perspective, a social network can be analyzed through two different approaches (Aggarwal, 2011): *structural data*, representing the connections, interactions and the topology, and *content data*, which focuses on providing information about the social actors, the content of the interactions, etc.

3.5.1 Structural-based analysis and community detection

This research approach involves the study of the properties of the network's topology using graph theory. Among the different structural metrics that networks can present, some of the most common are:

- Centrality: this measure is used to evaluate the relevance reached by a node or actor inside an specific network. There are several metrics regarding centrality, each of them based on different variables, including the number of interactions with other nodes, their closeness to other relevant nodes, or the closest paths that travel through the node (therefore, the relevance of the node to keep the network's cohesion).
- Transitivity: this measure analyzes the chance of interaction between three different nodes. This means that higher transitivity leads to a more dense graph (with more links among the nodes).
- Density: proportion of actors with connections in the network divided by the total of connections available inside it. A dense network will be one that presents more links between all the users.
- Closeness: distance between two actors inside the same network, based on the number of interactions that separate them.
- Degree: number of interactions established with or from a specific node. When there is a comparison between common interactions among nodes, this concept is known as *reciprocity*.
- Diameter: maximum distance between nodes inside the network.

One of the main applications of SNA is the analysis of how actors (and their interactions) group themselves, statically or dynamically, in *specific communities* inside a bigger network. This process, known as community detection, presents several similarities with the concept of partition of a graph inside graph theory (Bedi & Sharma, 2016). While in other structural-based metrics, such as centrality, the analysis is focused on the node and its qualities inside a group, community detection analyzes groups connected among themselves. Therefore, considering a graph $G = (V, E)$, community detection approaches group the nodes from that graph $V = \{v_1 ... v_n\}$ in clusters or communities $C_i = \{v_0 ... v_k\}$ considering the edges $E = \{e_1 ... e_n\}$ from that graph . The whole set of communities inside a graph $CS = \{C_1 ... C_m\}$ is known as *community structure*. When a graph's node may belong to more than one community at the same time ($\bigcap_{C_i \in CS} C_i \neq \emptyset$), it is said that

communities are *overlapped*. However, if the node cannot belong to more than one community ($\bigcap_{C_i \in CS} C_i = \emptyset$), it is said that the communities are *not-overlapped*.

There are several ways to detect communities (also known as clusters) on a graph. These approaches include techniques such as random walks, spectral clustering, or modularity maximization, among others (Fortunato, 2010). This kind of algorithm uses the topology of the graph to create the partitions that are validated by taking into account the density of the resulting sub-graph (i.e., a sub-graph is highly connected), and connections from these nodes to the rest (Camacho, Panizo-LLedot, Bello-Orgaz, Gonzalez-Pardo, & Cambria, 2020). A good community is one whose nodes are highly connected and it has few connections to the nodes of other communities (Kannan, Vempala, & Vetta, 2004).

To present an example of different techniques to detect communities, the variable "time" can be considered. Using this variable, and grouping different detection techniques depending on it, it is possible to talk about *static* and *dynamic* community finding algorithms [14].

- Static community finding algorithms: it refers to the techniques and methods applied when the variable "time" is not considered in the system. Therefore, the term *static* refers to the single snapshot of the network that is considered for the analysis. While these algorithms are quite easy to apply to any problem, the outcomes may not be very representative, considering that several real life communities are not isolated groups, but the actors tend to enter and leave it dynamically.

  Depending on the scope of the method, there are different static community finding algorithms. They can be divided in:

  - Node-centric: each node of the network must satisfy different properties (mutuality, reachability and degrees).
  - Group-centric: the connections inside a community as a whole are considered to detect the community.
  - Network-centric: instead of considering the connection just inside a community, these algorithms consider all the connections of the network.
  - Hierarchy-centric: these algorithms build a hierarchical structure of communities by taking into account the structure of the network.

- Dynamic community finding algorithms: it refers to the techniques and methods applied when the variable "time" is considered in the system. Therefore, the term *dynamic* refers to the consideration of the changes that the network experiences over time. There are two possible groups of algorithms considering the type of time-lapse considered for the time analysis:

  - Snapshot-based methods: it considers an ordered sequence of graphs, where each graph represents the state of the network at a given point in time.
  - Temporal networks-based methods: it avoids doing any aggregation at all, representing the network as a set of timestamped nodes and edges that precisely define when an element appear and disappear from a network.

[14] This section is just aimed to be a short summary of the state of the art. For further information and content, the reader can check the state of the art regarding SNA published by Camacho et al. (Camacho et al., 2020)

3.5.2 Content analysis

This approach focuses on analyzing the content and meaning of the nodes and the links among them. Therefore, a mixed approach is used to provide meaning to the social networks, using, for example, NLP as a complement (Cambria, Wang, & White, 2014). NLP provides a set of methods and algorithms that enable the processing of multimodal information circulating on social networks, and therefore allow to structure information and to add meaning to the interactions. The most representative applications of content analysis are:

- User profiling: while SNA is focused on analyzing the interactions between nodes, content analysis can be useful to obtain information about the nodes themselves; therefore, it may help to obtain information about the human actors, when studying a human network. The different profiles are established based on the behavioral patterns using different techniques, including clustering, behavioral analysis, qualitative analysis or facial detection, among others.

  Not only the user, but the messages and the content of the network' communication can be profiled. For example, when a Twitter network of users is analyzed, using NLP can help understanding the content of the messages among users, which content is more spreaded in the network, or to study the different characteristics of messages between one community and another one. Some examples of this approach include the detection of probable communication networks between users (Bar-Yossef, Guy, Lempel, Maarek, & Soroka, 2008; Pal & McCallum, 2006), the identification of influencers based on behaviour or the tagging of social roles (Harrigan et al., 2021), such as student, director, teacher, etc (De Choudhury, Mason, Hofman, & Watts, 2010).

- Topic extraction: Topic extraction is a technique used for discovering the abstract "topics" that occur in a collection of documents, which is useful for tasks such as text auto-categorization, sentiment analysis but also SNA. Common approaches include mixture of unigrams, latent semantic indexing, LDA, and knowledge-drive methods (Chaturvedi, Ong, Tsang, Welsch, & Cambria, 2016). Applied to the field of SNA, it is helpful to extract topics debated by a group of nodes, facilitating the analysis of the analysis of the discussions and the opinions of a group.

  This approach has been used to detect the topics approached by different communities (Pathak, Delong, Banerjee, & Erickson, 2008), the detection of interests from active or inactive users based on the social links among them (T. Wang, Liu, He, & Du, 2013) or the creation of a hierarchy based on the interests expressed by the actors, depending on their "thematical partners" (Faralli, Stilo, & Velardi, 2017).

- Sentiment analysis: Sentiment analysis has also been applied to better understand online social networks dynamics by looking at the exchange of information between network nodes (Camacho et al., 2020). Analysing the emotional valence underlying the interactions inside a network can help discovering influence patterns, polarization or political interest, but it can also be used as a complement to user profiling (friends vs enemies) and topic extraction (opinion mining).

  Inside the different examples of sentiment analysis combined with SNA, there are studies focused on political interest (Gryc & Moilanen, 2010), on the creation of classification rules to determine the consumer preferences (Shams, Saffar, Shakery, & Faili, 2012), or on the modelling of the existence of consensus and decision making processes between a network nodes, to lately analyze the relational preference between those nodes (Morente-Molinera, Kou, Samuylov, Ureña, & Herrera-Viedma, 2019).

4 Disinformation analysis as a Machine Learning task

4.1 Identification with supervised classification

Over the years, a large number of strategies have been tried out to automatically identify disinformation. In general, this problem can be modelled as a binary classification one: given an informational item A and a set of attributes $\overrightarrow{c_A}$ that represent it, the task consists in predicting whether A is truthful or not. If $\overrightarrow{c_A}$ can be numerically encoded and we have a set of already labeled individuals, it is straightforward to: (i) apply statistical analysis to determine the most relevant components of A correlated with the presence of disinformation (Oehmichen et al., 2019), (ii) train a classifier to predict unseen instances (Molina-Solana, Amador Diaz Lopez, & Gómez-Romero, 2018; Amador, Molina-Solana, & Gómez-Romero, 2019).

Recently, however, the need has arisen to further study the phenomenon through the definition of a finer-grained labelling that can capture the more subtle nuances of a phenomenon made so heterogeneous by the rapid evolution of the virtual contexts in which information circulates. These efforts, in the most recent literature, have led to experiments with different types of formulations of the problem. For example, Nakamura et al. (Nakamura, Levy, & Wang, 2020) proposed a classification based on 6 categories: *True*, *Satire/Parody*, *Misleading Content* —i.e. information intentionally manipulated to persuade the reader—, *Bot-generated Content*, *False Connection* —textual or visual items taken out of context and commented on in a misleading manner—, *Manipulated Content* — portion of information manually re-edited with a predefined malicious intent. Another example is (W. Y. Wang, 2017), in which 6 categories are also defined after the degree of reliability of the content, from *false* to *true*. More information on the nuances of dataset annotation is provided in Section 5.

The next question that arises is how to obtain a set of training examples already labeled with the target categories. We can find two principal approaches in the literature: expert-driven and data-driven. Expert-driven resembles fact-checking (see Section 4.4), since the verification of contents to build the training dataset is done by knowledgeable annotators. There are proposals in which annotation is performed manually by domain experts and journalists, and others in which annotation is crowdsourced and the annotators are not necessarily experts. The most successful ones are those which combine domain knowledge and computer assistance, e.g., methodologies relying on Open Web (Banko, Cafarella, Soderland, Broadhead, & Etzioni, 2007) —like the state-of-the-art tools WeVerify (Marinova et al., 2019) and ClaimBuster (Hassan, Arslan, Li, & Tremayne, 2017)— or on curated knowledge graphs —as in (Shiralkar, Flammini, Menczer, & Ciampaglia, 2017; Pan et al., 2018).

Focusing on the classification methods, a first approach is modeling fake news detection as a Multi-Criteria Decision Making problem (MCDM) (Pasi, Grandis, & Viviani, 2020). In MCDM, the expert provides a set of criteria and probability weights, which encode the credibility characteristics to be taken into account, and the process selects the optimal alternative from a set of candidate solutions (i.e., news items). To avoid manually defining criteria and weights, the trend in the last years is to apply Machine Learning for data-driven knowledge extraction. The most successful methodologies are categorized according to the type of features chosen to represent the news item: (i) content-based features, i.e., related to the content of the news item itself; and (ii) features related to the social context, i.e. related to the internal dynamics of the virtual platform on which the posts are published. These two perspectives are discussed below.

4.2 Natural language processing for stylistic characterization

The majority of recent work in disinformation analysis has been conducted with Computational Linguistic methods (Ruffo, Semeraro, Giachanou, & Rosso, 2023). Some recent (and straightforward) approaches put together an NLP pipeline (preprocessing, feature extraction, model building) to exploit traditional text analysis techniques (Asaad & Erascu, 2018; Koloski, Pollak, & Škrlj, 2020) or to train neural networks (Reddy, Suman, Saha, & Bhattacharyya, 2020; Umer et al., 2020; Eldesoky & Moussa, 2021; Qazi, Khan, & Ali, 2020; Tida, Hsu, & Hei, 2022; Dun, Tu, Chen, Hou, & Yuan, 2021) for fake news classification.

A more targeted approach to the identification of disinformation is through detecting the presence of certain writing styles, contents, or even intentions. Machine Learning has been thus used to identify the linguistic differences between reliable and unreliable news. Researchers have used language models to find which characteristics stand out in disinformation texts (Castelo et al., 2019; Giachanou, Rosso, & Crestani, 2019; Vogel & Meghana, 2020; Bonet-Jover, Piad-Morffis, Saquete, Martínez-Barco, & Ángel García-Cumbreras, 2021). This includes using syntactic, lexical, semantic, discursive, and morphological features; but also other properties of text, such a readability, similarity, punctuation, quality, informality, subjectivity, diversity, uncertainty, complexity, specificity, sentiment, and emotions. It has been found that part-of-speech counts, lexical diversity, informality and readability are some of the most distinctive features of fake news (Azevedo, D'aquin, Davis, & Zarrouk, 2021; Castelo et al., 2019). Conversely, stylometry can be learned by disinformers in order to replicate the styles of reliable news, as shown in (Schuster, Schuster, Shah, & Barzilay, 2020), which calls for approaches that continuously update over time.

More specifically, Afroz, Brennan, and Greenstadt (2012) obtained an F-measure score of 96.6% by considering the count of syllables and words, vocabulary and grammatical complexity and part-of-speech tags for a binary classification into *false* and *real* news; Rashkin, Choi, Jang, Volkova, and Choi (2017) found that special patterns in the use of personal pronouns and swear words become an indicator of less credibility; and Mendoza, Poblete, and Castillo (2010) showed how misleading posts are characterised by a higher proportion of negations and contradictory expressions, and at the same time by a lower variability in the vocabulary used. On the other hand, the systematic use of polarised language patterns is often considered as a factor of low credibility (Stella, Ferrara, & Domenico, 2018; Ghanem, Ponzetto, Rosso, & Rangel, 2021). Moreover, disinformation aim to engage the audience by trying to provoke certain negative emotions, such as disgust, anger and fear, in order to increase their impact and redistribution. Sentiment analysis has been applied for text classification in combination with classical Machine Learning models (Del Vicario et al., 2016) and Deep Learning architectures (Giachanou et al., 2019). The impact of emotional signals and psycholinguistic patterns has been investigated further in (Giachanou, Rosso, & Crestani, 2021; Giachanou et al., 2022). In (Ghanem et al., 2021) the authors modeled the flow of affective information in fake news to capture exaggerations added in order to affect the readers' emotions.

Related to this phenomena, features relating to personality characteristics and mental processes also play an important role in analysing the phenomenon of disinformation. Since psychological characteristics regulate behaviour and interaction in the real world, it is reasonable to assume that they are also influential within virtual communities. Indeed, the psychological traits distort users' understanding, makes them differently inclined to spreading false information and toxic narratives. For example, mental disorders like paranoia, insensitivity and aggressiveness appear to have a large influence in this sense (Shu, Sliva, Wang, Tang, & Liu, 2017). It is therefore useful to implement computational linguistics techniques capable of mining this psychological data and examining their correlation with the tendency to spread and/or produce false information.

4.3 Contextual aspects of disinformation generation and dissemination

In addition to exploiting information on all elements of individual content, it has been shown in several cases how useful it is to extend the analysis to a broader perspective, taking into account social context features. In fact, as stated by Zhou and Zafarani (2020), there are two relevant topics regarding the context where fake news are spread: propagation-based fake news detection, which implies the analysis of how fake news are disseminated from their origin through a network of users, including the peaks of interactions; and the identification of the source and relevant nodes of the network for the dissemination of the fake news.

In the literature about propagation, we can find works considering (i) user-based features, such as demographics, number of interactions, etc. (J. Ma, Gao, Wei, Lu, & Wong, 2015); (ii) post-based features, e.g., by monitoring discussion and comments on a publication (Ruchansky, Seo, & Liu, 2017); and (iii) network-based features, considering the properties of users' networks and the way content spreads within the various platforms (Guille, Hacid, Favre, & Zighed, 2013; Tacchini, Ballarin, Della Vedova, Moret, & de Alfaro, 2017; Zhou & Zafarani, 2020).

Regarding relevant nodes in a (social) network, the features of fake news spreaders have been studied for instance in the context of the PAN[15] challenges (Vogel & Meghana, 2020; Labadie-Tamayo, Castro-Castro, & Ortega-Bueno, 2020; Giglou, Razmara, Rahgouy, & Sanaei, 2020; Hashemi, Zarei, Moosavi, & Taheri, 2020). This approach has proved very effective to counterfight disinformation propagation, since in some cases it is more useful to focus on identifying accounts that intentionally create and convey false information rather than on the single publications. Here we highlight the work by Buda and Bolonyai (2020), who won the spreaders identification task in PAN 2020. The authors proposed an ensemble model consisting of four classical Machine Learning models trained on n-grams and a fifth one trained on a set of stylistic features.

Multimedia contents can be also incorporated to the Machine Learning models to improve the detection rate, which is often referred as multimodal analysis. Images attached to disinformative messages have been the primarily object of study from different perspectives: forensic (the image has not been manipulated), contextual (the text and the image are falsely associated), distributional (the kind of images associated to disinformation have distinct characteristics). Usually this procedure involves the extraction of the implicit semantic features by means of neural networks (Y. Wang et al., 2018; Qi, Cao, Yang, Guo, & Li, 2019; Khattar, Goud, Gupta, & Varma, 2019; Giachanou, Zhang, & Rosso, 2020; Giahanou, Zhang, & Rosso, 2020), in a similar way as it is done with text but by relying on architectures specialized in image processing. In a recent work (G. Zhang, Giachanou, & Rosso, 2022) the authors combined textual, contextual scene and visual representations. The place, weather and season scenes were extracted from the images showing statistical significance differences regarding their frequency in fake and real news.

4.4 Semi-automated fact-checking: a human in the loop approach

Automated fact-checking has been framed as the ultimate AI-based application to fight against fake news. However, after some years of exploring this research avenue, recent reports show a general distrust from fact-checkers to fully-automated methods (Arnold, 2020). While human fact-checking has scalability issues fueled by the fast spread of false information (Vosoughi et al., 2018; Zaman, Fox, & Bradlow, 2014), this can not compromise the complexity and accuracy of the fact-checking process. For this reason, hybrid methods that focus on assisting fact-checkers have been put on the research focus. So far, assistance applications have been designed for three main tasks: (1)

[15] https://pan.webis.de

identifying check-worthy claims, (2) finding previously-checked claims, and (3) retrieving the relevant evidence for the verification (Nakov, Corney, et al., 2021). Specific datasets have been created to support these approaches (Thorne, Vlachos, Christodoulopoulos, & Mittal, 2018).

Regarding the check-worthiness of claims, previous work has framed this application as a ranking task, in which the goal is to predict a score that prioritises the claims to be fact-checked (Nakov et al., 2018; Hansen, Hansen, Alstrup, Grue Simonsen, & Lioma, 2019; Barrón-Cedeño et al., 2020; Nakov, Da San Martino, et al., 2021). Others have approached it as a classification task, in which a system predicts if a claim is factual, numerical or based on personal beliefs (Arslan, Hassan, Li, & Tremayne, 2020; Courney, 2019; Konstantinovskiy, Price, Babakar, & Zubiaga, 2021), or as a query answering task, expecting that there are verified claims in the dataset (Chen, Fisch, Weston, & Bordes, 2017). Nevertheless, some works also point out that any automated system might introduce biases in the choice of the claims to be fact-checked and that developing tools such as news alerts, speech recognition and translation models might be the optimal way to help fact-checkers filter claims (Nakov, Corney, et al., 2021).

When it comes to detecting previously fact-checked claims, the task consists in finding both claims that were fact-checked in the past and claims that were fact-checked in other countries or languages. In this direction, we find the work from Shaar, Babulkov, Da San Martino, and Nakov (2020) that focuses on matching new claims with previously fact-checked claims with a learning-to-rank approach. This task has been approached in Spanish by Martín, Huertas-Tato, Huertas-García, Villar-Rodríguez, and Camacho (2022), who used semantic textual similarity to match claims.

Finally, regarding the retrieval of evidences, some work has focused on extracting the most useful set of evidences given a fixed database of trusted sources (Thorne et al., 2018; Barrón-Cedeño et al., 2020). This has been done both at document and sentence level, as well as from table-structured data. Other important tools for evidence retrieval have been found to be speech recognition, image reverse search, and complex query search.

4.5 Computer-generated contents and disinformation

False information is not only displayed in text format, other types of multimedia content such as images, video or audio are also used as a source of disinformation and misinformation. Multimodal disinformation and misinformation are presumably more devastating than text-only disinformation. In recent years, a new term associated with false information has appeared: deepfakes. Deepfakes are artificially generated multimedia content designed to deceive humans. The word deepfake is a combination of deep learning (see 3.3) and fake (Mirsky & Lee, 2022), as deepfakes are usually generated by artificial neural networks. The most common form of deepfake is the manipulation of people's imagery. The modification ranges from the face manipulation, to the speech falsification, to the generation of people who do not really exist.

Although deepfakes can be used for positive applications such as the generation of avatars for movies and video games, memes for entertainment, audio generation to help hearing-impaired, etc. their emergence has since their inception been linked to the generation of malicious and harmful content. Deepfakes can be used to damage an individual's reputation, for example by creating a fake pornographic video or manipulating an individual's audio and facial expressions in order to alter speech (Greengard, 2019). But the use of deepfakes can go further and be used to manipulate elections or spread hatred in society by employing different actors such as bots, professional trolls, the media and even politicians and foreign governments themselves (Masood et al., 2022).

Dealing with deepfakes can be divided into two main approaches: techniques for deepfakes generation and techniques for deepfakes detection (Mirsky & Lee, 2022; Masood et al., 2022; Malik,

Kuribayashi, Abdullahi, & Khan, 2022; Rana, Nobi, Murali, & Sung, 2022; Saif & Tehseen, 2022; Dagar & Vishwakarma, 2022). Since the emergence in 2014 of Generative Adversarial Networks (GANs) (I. J. Goodfellow et al., 2014), there has been an explosion of work on deepfakes generation and, as a counterpart, deepfakes detection (Masood et al., 2022; Mirsky & Lee, 2022; Dagar & Vishwakarma, 2022; Saif & Tehseen, 2022; Tolosana, Vera-Rodriguez, Fierrez, Morales, & Ortega-Garcia, 2020).

We can differentiate deepfakes depending on whether the manipulation is total or partial. A full face manipulation involves the creation of an image of a non-existing face, some of the architectures for entire face generation are ProGAN (Karras, Aila, Laine, & Lehtinen, 2018) and StyleGAN (Karras, Laine, & Aila, 2021). There are multiple works to detect whether an image is artificial or real, and in recent years the trend has been towards the use of convolutional neural networks (CNNs) with attention mechanisms (Tolosana et al., 2020; Rana et al., 2022; Saif & Tehseen, 2022; Dagar & Vishwakarma, 2022). Partial face manipulation is the most common type of deepfake used to generate false information. Partial manipulation involves swapping one person's face to another (face-swap); attributes manipulation such as hair or skin colour; expression swap, also known as face reenactment; or lip-syncing, which involves modifying a video to make the mouth consistent with the audio. The predominant technique for generating this type of content are GANs and for detecting it, CNNs (Mirsky & Lee, 2022; Tolosana et al., 2020). When training the models, some works generate their own databases (Nataraj et al., 2019; Jung, Kim, & Kim, 2020; L. Zhang, Qiao, Xu, Zheng, & Xie, 2022) and others use public databases (Dang, Liu, Stehouwer, Liu, & Jain, 2020; Peng, Fan, Wang, Dong, & Lyu, 2022; Groshev, Maltseva, Chesakov, Kuznetsov, & Dimitrov, 2022).

To obtain quality deepfake generation, many training examples are needed and as a consequence it takes a long time to produce convincing deepfakes, generating deepfakes for a specific victim is complex and re-training a model for each identity has a high computational cost. Therefore, one of the trends in deepfakes generation is towards the implementation of more generalist models. Another limitation of deepfakes is that they are usually generated from a frontal pose and this generates very static recreations. The previous problem can be exacerbated if the target image has a shadow generated by hair, a hand or any other element, which will result in inconsistent facial aspects. In short, the trend in deepfakes generation is towards higher quality models and real-time models (Mirsky & Lee, 2022; Masood et al., 2022).

Language models are not exactly deepfake generators, but can be used to create synthetic text that eventually can leverage disinformative messages. For example, the current version of GPT, namely GPT-3 (Brown et al., 2020), and its associated conversational agent ChatGPT, are able to generate high quality text using the NLP techniques presented in section 3.4. More generally, generative language models are a class of machine learning models that are trained to generate text that resembles what a human being could have written. These models are trained using large amounts of text, such as news articles or books, so that they can learn to generate text that resembles those examples. One of the possible uses of these models is to generate uninformative content. This could be done by training the model with fake news or uninformative text, so that the model learns to generate text that resembles that type of content. Once trained, the model could be used to generate fake or uninformative news automatically, which could be used to spread misleading or confusing information on a large scale. Another more elaborated use of such models is to automatically create falsified accounts and web pages imitating legitimate sources to give credibility to a false narrative[16].

[16] https://www.poynter.org/fact-checking/2023/chatgpt-build-fake-news-organization-website/

5 Datasets

The task of fake news detection, as seen above, presents a series of complexities linked to the intrinsic nature of the data analysed: the news item is an object composed of different elements (linguistic, visual, structural —i.e. deriving from the architecture of the platform or the diffusion medium—, etc.), which becomes even more complex when considered within virtual dynamics of re-sharing or manipulation. This causes known difficulties when building automatic classification systems —such as, for instance, selecting optimal features, obtaining effective computational strategies for training the model, etc.— but first and foremost when it is necessary to extract correct and useful data.

The construction of a dataset for the fake news detection task is an extremely complicated task because the object of study is not immediately apprehensible in quantitative form, but requires two main phases: *(i)* the mere extraction and *(ii)* the annotation. If the source is a website or a social network platform, the extraction consists of a search for content on the basis of certain criteria defined a priori: the topic dealt with —for which a query can be formulated or a combination of keywords can be given as input—; the type —articles, comments, images, videos, metadata of various kinds—; the users who made the publication. Usually, extraction is done through the use of application programming interfaces (APIs) directly provided by the platform owners themselves, or through web scraping methods, i.e. algorithms that access online content by simulating human web surfing.

Annotation is in turn the process of labelling data, in "true"/"false" and possibly other intermediate classes. This step can be performed in various ways (Simko et al., 2021). A popular strategy is to have the labelling performed manually by groups of expert annotators, who formalise the process by defining a set of unique criteria. Expert annotators are usually journalists, independent fact-checkers or fact-checking platforms, or domain experts. In other cases, all or part of the annotators are not experts or do not coincide with the figures just mentioned. Alternatively, some semi-automatic annotation approaches have been proposed, which require less human intervention —and are therefore less demanding in terms of time and resources— but at the same time have a higher risk of false attribution.

There are numerous works proposing datasets with news items annotated according to the credibility of the content, especially in recent research. These projects differ in the choice of features presented, the method of extraction and annotation and the level of “fineness” of the labelling. Some relevant datasets will be presented hereafter, in terms of their source and main characteristics. They are summarized in Table 1.

Due to its popularity among users, the huge amount of new texts produced daily, and the ease of access to metadata through the API, Twitter is definitely the most common source in current research. A well-known example is the CREDBANK dataset (Mitra & Gilbert, 2015), which collects up to 60 million credibility-rated tweets related to one or more events from an initial set of 1,000 news events manually annotated by 30 experts, over a 96-day period. Another well-known example is PHEME (Zubiaga, Liakata, Procter, Hoi, & Tolmie, 2016), which extracted and binary annotated in rumour and non-rumours the tweets published within conversational threads that related to two specific types of news items: *(i)* breaking news from credible sources, and *(ii)* precise rumours previously identified in Twitter. Then, FakeNewsNet (Shu, Mahudeswaran, Wang, Lee, & Liu, 2020) appears absolutely relevant, because it includes three different categories of features. First, the authors collected news items labelled as “fake news” and “true news” from *PolitiFact* and *GossipCop*. After obtaining implicitly annotated textual data, they downloaded the tweets referring to these news items, introducing two new types of information: related to the social context (i.e. metrics about users’ reactions and engagement), and spatiotemporal (namely, the location and the timestamps of user engagements), thanks to which it is possible to examine the propagation of fake news in the

platform and the evolution of online discussion.

Facebook is most likely to be the second most used platform to access useful data for the research on the fake news detection task. In this respect, the authors of the BuzzFeedNews dataset (Silverman, Strapagiel, Shaban, Hall, & Singer-Vine, 2016) extracted and fact-checked Facebook posts in the period around the 2016 US presidential election from 9 sources (3 with a right-wing political bias, 3 with left-wing bias and 3 credible mainstream political news pages). The annotation of each of these publications was carried out directly by BuzzFeed journalists, in three classes: mostly true, mixture of true and false, and mostly false, thus avoiding assigning an absolute level of truthfulness/falsity to each item. In addition to textual data, the dataset incorporates metadata on the engagement caused by each post —number of shares, comments, and reactions— and on the presence of other elements —links, images/videos. An interesting extension of the BuzzFeedNews dataset is the BuzzFace dataset (Santia & Williams, 2018), which also incorporates the text of the comments triggered by the publication of each post already archived in the original dataset, up to a final count of 1.6 million new texts added from Facebook. Each text is accompanied by metadata about engagement: number of shares, comments, and reactions. In addition, the extension also regarded the sources of extraction: the final dataset gathers the comments under the original news articles by the 9 agencies, as well as posts from Twitter and Reddit that discussed about these same original news articles. Similarly, FacebookHoax (Tacchini et al., 2017) makes available information related to approximately 16.000 posts, which are obtained from 32 Facebook pages reporting scientific news or spreading conspiracy theories, annotated into the two classes hoax and non-hoax.

Other works have recognised the importance of analysing different social network platforms, less popular but still rich in textual content and useful information. Indeed, the number of projects based on the extraction of posts from Reddit has grown recently. FACTOID (Sakketou et al., 2022), for instance, is a dataset derived from the monitoring of the discussion about political topics within certain *subreddits* between January 2020 and June 2021. The data collection represents roughly 4000 users, amounting a total of 3.4 million posts, each of which was annotated in three ways: binary (in fake news or real news), by means of a fine-grained credibility scale (from very low to very high), and according to the degree of political bias (from extreme right to extreme left). There is also a growing need to study the phenomenon of disinformation within more polarised virtual environments, as these act as breeding grounds for false information that is then re-shared and spread on more popular platforms: well-known cases are Gab, 4Chan and 8Kun (Zeng & Schäfer, 2021).

In addition to the complexities already mentioned, the phenomenon under study presents a further aspect that must not be underestimated for proper analysis: the language. Numerous efforts have therefore been concentrated on generating datasets in languages other than English. In Spanish various projects aimed at constructing valid datasets can be found: Posadas-Durán, Gómez-Adorno, Sidorov, and Escobar (2019) mined textual data from different resources on the Web: websites of credible newspapers and media companies, special websites dedicated to the validation of fake news, websites designated by different journalists, as well as webpages that regularly publish fake news. Gómez-Adorno, Posadas-Durán, Bel Enguix, and Porto Capetillo (2021) presented an update of the dataset described above. This new Spanish dataset was proposed in the task for fake news detection FakeDeS 2021 organised by the IberLeF conference. The corpus is divided into training and test data from different sources such as news websites and fact-checkers sites. The data is manually labelled as fake and real news. The collected news are related to nine different topics for the training set (science, sports, economy, education, entertainment, politics, health, safety and society) and the test set with seven topics (science, sport, politics, society, COVID-19, environment, and international) of which three are different from the training set. Another relevant particularity of

the test set is that it also includes posts from social networks. The main feature of this dataset is that it covers different varieties of Spanish as different fact-checking sites have been used to extract the news. These sites include countries such as Argentina, Bolivia, Chile, Colombia, Costa Rica, Ecuador, Spain, United States, France, Peru, Uruguay, England and Venezuela.

Furthermore, Sierra, Soto, and Díaz (2018) made available GECO, an online corpora management software that allows users to upload collections of Spanish-language documents and transform them into digital corpora. The NLI19-SP dataset, created by Martín et al. (2022) to evaluate test the FaCTeR-check software, is a way to test the concept of *Natural Language Inference*, which focuses on the similarity between texts. It includes a pool of 61 hoaxes identified by fact-checker organisations, with another pool of semantically-similar tweets for each hoax labelled as *entailment*, meaning that the tweet endorses the false claim, *contradiction* or *neutral*. These tweets were extracted between the 1st of January 2020 to the 14th of March 2021. Finally mention another Spanish dataset proposed at the shared task of Profiling Fake News Spreaders on Twitter at PAN 2020 (Rangel, Giachanou, Ghanem, & Rosso, 2020), which collects a set of 100 tweets from the timelines of 500 users, for the development of methods that can automatically distinguish accounts that are attributed a tendency to spread fake news from other profiles that have not shown this tendency in their past publications.

Despite the efforts of the scientific community to generate datasets in languages other than English, there is still a lack of them. In this section we have described four datasets in Spanish and this sample represents the vast majority of the datasets available in Spanish to date. This demonstrates the need for developments focused on the production of quality datasets in both Spanish and Portuguese within the IBERIFIER project.

Table 1: Summary of datasets.

Datasets	Application	Data Source	Size	Information used	Labels	Annotations	Feature coverage	Date of generation	Language	Public avail-able	URL
CREDBANK	Credibility assessment	Twitter	>60 millions	Social media posts about 1049 events	Tuple <degree (certainly, probably, uncertain), polarity (accurate, inaccurate, uncertain)>	Mechanical Turk	Content and context features	2015	English	Yes	https://compsocial.github.io/CREDBANK-data/
PHEME	Rumor detection	Twitter	5.802	Social media posts about 1049 events	Rumor (1.972), Non-rumor (3.830)	Expert annotation	Content features	2015	English	Yes	https://figshare.com/articles/dataset/PHEME_dataset_of_rumours_and_non-rumours/4010619
FakeNewsNet	Study fake news on social media	BuzzFeed.com and Politi-Fact.com	422	News content	Fake (211), Real (211)	Expert annotation	Content and context features	2017	English	Yes	https://github.com/KaiDMML/FakeNewsNet
BuzzFeedNews dataset	Fake news detection	Facebook	2.282	Social media posts from 9 sources (3 right-wing bias, 3 left-wing bias and 3 credible)	Most true (1669), No factual content (264), Mixture of true and false (245), Mostly false (104)	Expert annotation	Content and context features	2016	English	Yes	https://webis.de/data/buzzfeed-webis-fake-news-16.html
BuzzFace dataset	Fake news detection and bots detection	Facebook	>1.6 millions	Social media posts verified by BuzzFeed and comments and reactions about this posts	Only source data (BuzzFeedNews dataset) are labelled	Expert annotation	Content and context features	2018	English	Yes	https://github.com/gsantia/BuzzFace
FacebookHoax	Hoax detection	Facebook	15.500	Social media posts 32 pages (14 conspiracy and 18 scientific)	Hoax (8.923), Non-Hoax (6.577)	Pages assumptions	Content and context features	2017	English	Yes	https://github.com/gabll/some-like-it-hoax
FACTOID	Fake news spreaders detection	Reddit	4.150	3.354.450 social media posts authored by 4,150 users	Real news spreader (3.071), Fake news spreader (1.079)	Expert-based automatic annotation	Content and context features	2021	English	Yes	https://github.com/caisa-lab/FACTOID-dataset
Spanish Fake News Corpus	Fake news detection	News Media websites	971	News from 9 different topics	Fake (480), Real (491)	Expert annotation	Content features	2019	Spanish	Yes	https://github.com/jpposadas/FakeNewsCorpusSpanish
Spanish Fake News Corpus 2.0	Fake news detection	News Media websites and Social Networks	1543	News and social media post from 12 different topics	Fake (766), Real (777)	Expert annotation	Content features	2021	Spanish	Yes	https://github.com/jpposadas/FakeNewsCorpusSpanish
NLI19-SP	Misinformation detection	Twitter	46.919	Social media posts related with a pool of 61 hoaxes identified by fact-checker organisations	Contradiction (406), Entailment (2.521), Neutral (43.992)	Automatic annotation	Content and context features	2021	Spanish and English	Under request	https://aida.etsisi.upm.es/download/nli19-sp-dataset-facter-check/
PAN-AP-2020 corpus	Fake news spreaders detection	Twitter	500	Social media users from news posted on Twitter.	Real news spreader (250), fake news spreader (250)	Expert annotation	Content and context features	2020	Spanish and English	Under request	https://zenodo.org/record/4039435#.Y2z2fi8ryRs

6 Software Tools

The aim of this section is to provide a sample of some of the most representative software tools focusing on developments IBERIFIER's partners. Our findings are summarized in Table 2. In the following, we can distinguish two main types of tools; first, those that can be used generically for tasks related to disinformation analysis, and second, those that are specifically purported to address disinformation. As it can be seen, many of the tools described are at a low level of maturity as they are the result of recent research and have not entered a production phase, while those at a high technology readiness level do not cover the full spectrum of fact-checkers' needs and have very limited functionality. The analysis suggests that there is a real need to develop tools with a sufficient degree of maturity to further assist fact-checkers in processing the vast amount of verifiable information that is produced on a daily basis.

6.1 General Purpose

6.1.1 Initiatives outside IBERIFIER

Gephi[17] is an open source software for the visualization and manipulation of graphs. Among the possibilities that Gephi has, it allows both importing and creating graphs. One of the advantages of Gephi is the rendering of 3D graphs in real time, which makes it easier to work with large networks.

Gephi's tools include Layout Properties, which allows an aesthetic visualisation of the graphs by implementing different algorithms such as the "Force-based" algorithm in which linked nodes are attracted to each other and unlinked nodes are pushed apart. Gephi also allows the modification of the colour and size of nodes and edges as well as the possibility of displaying their labels. But the most interesting functionality of Gephi is the different metrics it implements, such as local and global quality measures and the possibility of detecting communities within a network with the Louvain method (Blondel, Guillaume, Lambiotte, & Lefebvre, 2008).

For fack-checkers, this tool is very useful in helping to debunk hoaxes as it allows them, for example, to locate the focus of a hoax by analysing social networks and identifying influential users and the communities that are created around them.

Graphext[18] is a general purpose data analytics tool aimed at professionals. Graphext allows you to transform, explore, visualise and analyse data from different sources as it offers the possibility of integrating with Google Sheets, BigQuery or Amazon S3 among others. One of the advantages of Graphext is that no programming skills are required to get the most out of the tool, which makes it very interesting for a variety of professionals outside the IT field.

Graphext has a very complete data analytic toolkit. Graphext has different clustering, dimensionality reduction and predictive analytics algorithms such as HDBSCAN, UMAP, logistic regression or CatBoost. It also has natural language processing tools for information extraction in text (including sentiment analysis), topic analysis and keyword extraction linked to its social network analysis functionality. Like Gephi, it also allows the community detection in graphs and their manipulation. In short, Graphext allows you to analyse different types of data in a simple and visual way.

[17] https://gephi.org
[18] https://www.graphext.com

GATE[19] is an open source software for all kinds of tasks involving natural language. GATE allows the analysis of all types of text regardless of its size. Since its appearance, GATE has been growing and incorporating new functionalities such as an IDE for natural language processing tasks, a web application (GATE Teamware) for collaborative semantic annotation projects or GATE Cloud for end-to-end text processing on cloud computing infrastructures.

GATE includes different natural language processing tasks such as tokeniser, part of speech tagger, semantic tagger, and many others with multi-language support. One of the most important and useful features of GATE is that it integrates many plugins to work with ontologies, domain-specific resources such as biomedical texts, tools for social network analysis such as Twitter, machine learning toolkit, etc.

Thanks to the large number of utilities and plugins GATE has become a very useful tool in the fight against disinformation. Fack-checkers use GATE for example to extract text from images using OCR, collect relevant information from URLs, implement a rumour classifier[20], classify the reply to a tweet according to its stance[21], etc. Many of these plugins are integrated into the WeVerify/InVID pluging[22].

6.1.2 IBERIFIER partners initiatives

Twitter Analysis Toolkit[23]. The analysis of tweets is often followed by doing the same pipeline for different projects. This toolkit is a compilation and wrapper of many tools to ease the pipeline of analysis in twitter. First of all it provides search utilities, either by searching by a twitter query or by identifier with the last API version of Twitter. Then it integrates some models to infer users age, gender, and if it is a person or an organisation. There is also a location of users inference for Spanish locations based on their *location* text or *description* in their Twitter profile. For the text analysis the toolkit provides a pipeline for topic analysis using the LDA algorithm. The toolkit also has a text pre-processing methods that are helpful for analyzing and before doing a topic analysis. It can clean noisy text in text like emails, links and other expressions often found in tweets. And also a lemmatizing process that is often used in a topic analysis. There is also a sentiment module analysis specially trained for tweets that use a particular language than regular texts. Finally we provide a network creation of the tweets and users function for a network analysis. The network of users and tweets are the resulting two one-mode projection of the bipartite network of users interactions with tweets.

Spanish and Catalan massive Language Models and finetunings for part-of-speech (POS), named entity recognition (NER), and query answering (QA) tasks. Large scale language models trained with massive high-quality corpora have revolutionized the field of NLP since (Devlin et al., 2019). The members of the BSC have trained four large models of varying sizes and configurations for Spanish (Gutiérrez-Fandiño et al., 2022), and two large models for Catalan (Armengol-Estapé et al., 2021). These models have been released publicly[24] and can be used to develop tools to deal with disinformation involving NLP.

Additionally, the BSC group has fine-tuned these models for the tasks of Part-of-Speech tagging, Named Entity Recognition, and Question Answering. The knowledge outputted by these models

[19] https://gate.ac.uk
[20] https://cloud.gate.ac.uk/shopfront/displayItem/rumour-veracity
[21] https://cloud.gate.ac.uk/shopfront/displayItem/stance-classification-multilingual
[22] https://weverify.eu/verification-plugin/
[23] https://gitlab.bsc.es/rrssalud/twitter_toolkit
[24] For Spanish: https://huggingface.co/PlanTL-GOB-ES. For Catalan: https://huggingface.co/projecte-aina

has been found to be useful for Information Extraction or Evidence Retrieval, which are common tasks in the fight against disinformation (Nakov, Corney, et al., 2021).

Pyleetspeak: Word camouflaging generator and Data Augmentation package. Content moderation is the process of screening and monitoring user-generated content online to suppress communications that deem undesirable (Gerrard & Thornham, 2020). The growing amount of content uploaded to social media platforms makes it unfeasible to rely exclusively on the human content moderation approach. Nevertheless, content filtering automated systems depend on their capacity to analyse the material uploaded, potentially vulnerable to recent content evasion techniques, such as word camouflaging.

Word camouflage is currently used to evade content moderation in social media. Therefore, Pyleetspeak aims to counter new misinformation that emerges in social media platforms by providing a mechanism for simulating and generating leetspeak and word camouflaging data. The tool is publicly available in the Python PyPI package repository[25]. It includes three different, but compatible, text modifications word camouflaging methods: LeetSpeaker, PuntctuationCamouflage and InversionCamouflage.

- LeetSpeaker: This module apply the canonical 'leetspeak' method of producing visually similar character strings by replacing alphabet characters with special symbols or numbers. There's many different ways you can use leet speak. Ranging from basic vowel substitutions to really advanced combinations of various punctuation marks and glyphs. Different leetspeak levels are included.
- PunctuationCamouflage: This module apply punctuation symbol injections in the text. It is another version of producing visually similar character strings. The location of the punctuation injections and the symbols used can be selected by the user.
- InversionCamouflage: This module create new camouflaged version of words by inverting the order of the syllables. It works by separating a input text in syllabels, select two syllabels and invert them.

These modules can be combined into a string to generate a leetspeak version of an input text. Precisely, this can be achieved by using the Leet_NER_generator method that selects the most semantically relevant words from an input text, applies word camouflage and creates compatible annotations for NER detection.

LeetSpeak-NER Transformer models. Taking into consideration the previous section, two Transformer based models are developed for the detection of camouflaged data using the Pyleetspeak package to generate data for training in Spanish and English for the detection of camouflaged words and content evasion. The base models fine-tuned for the task were roberta-base (Liu et al., 2019) and roberta-base-bne (Gutiérrez-Fandiño et al., 2022) for English and Spanish, respectively. The models were fine-tuned using the Spacy interface (Montani, Honnibal, Van Landeghem, & Boyd, 2020) as the camouflage NER data is Spacy format. The models are available at HuggingFace[26].

LeetSpeak-NER App: Word Camouflaging NER detector. Considering what has been described in the two previous sections, a web application and an API are developed where the previously de-

[25] https://pypi.org/project/pyleetspeak/

[26] https://huggingface.co/Huertas97/en_roberta_base_leetspeak_ner,https://huggingface.co/Huertas97/es_roberta_base_bne_leetspeak_ner

veloped models in Spanish and English for camouflage detection and content evasion are put into production. The app is available in HuggingFace[27].

BERTuit is a Transformer model designed for the analysis of Spanish language in Online Social Networks (Huertas-Tato, Martin, & Camacho, 2022). This model enables users to study social networks like Twitter with an specialized transformer with an inexpensive fine-tuning of the weights. It achieves State-of-the-Art results in tasks like Named Entity Recognition, Sentiment Analysis, among other classification problems, frequently outperforming general-purpose models while using less resources.

The model has been pretrained on a large corpus (200 million) of miscellaneous tweets written in Spanish including Latino-american communities and dialects, allowing for robust mono-lingual recognition of linguistic features. This specialized training allows the model to easily adapt to any task in this domain, even in the presence of obstacles such as lack of labelled data or noisy text.

Non-speciallized models like M-BERT or XLM-RoBERTa are unable to detect some subtleties that BERTuit is capable of understanding as they have been trained on datasets with well-structured and curated language such as journalistic news or wikipedia articles. On the other hand, BERTuit allows the semantic analysis of elements of speech such as emoji, shorthand and even grammatical errors, as it has been trained to recognize and take advantage of such features.

It has been originally developed to counter misinformation on social media, as to analyze the content of malicious tweets and to profile authors who disinform.

FacteR-CheckKey API. Automatic keyword extraction tool for query building developed in the FacTeR-Check (Martín et al., 2022) architecture. It combines KeyBERT (Grootendorst, 2020), using the MSTSb-paraphrase-multilingual-mpnet-base-v2 model fine-tuned in multilingual Semantic Textual Similarity Benchmark (mSTSb) as the semantically aware model, and multilingual Name Entity Recognition (NER) approaches with Spacy (Montani et al., 2020) and Flair (Akbik et al., 2019) frameworks as keyword filtering steps.

KeyBERT package provides the infrastructure for keyword extraction selecting as keywords those words that are the most semantically similar to an input text. For this purpose, it leverages the semantic power of Transformer-based models to compute text embedding and word embeddings. Then it uses the cosine similarity distance metric to find the words most semantically similar to the text. To optimise multilingual keyword extraction for query building purposes, stop words are removed using Spacy v3. Furthermore, multilingual part-of-speech tagging, POS tagging, is accomplished with Flair, removing verbs, auxiliary verbs (AUX), coordinating conjunctions (CCONJ) and subordinating conjunctions (SCONJ), adverbs (ADV), and adpositions (ADP).

In contrast to the straightforward KeyBERT approach, FacteR-CheckKey automatically detects the language introduced to apply the appropriate stopword list. To this end, the FastText lid.176.bin model (Joulin, Grave, Bojanowski, & Mikolov, 2017; Joulin et al., 2016) is used as the language identification system.

[27] https://huggingface.co/spaces/Huertas97/LeetSpeak-NER

6.2 Specific Purpose

6.2.1 European initiatives

Truly Media[28] is a web-based journalism platform focused on the collaborative verification of content from social and digital media, co-developed by Athens Technology Center (ATC) and Deutsche Welle (DW). It is used by DW and other organisations, like fact-checking organisations, broadcasters and news agencies.

By using Truly Media, journalists can first collect and archive content around a topic they are investigating from different digital sources in a "Collection". Collections are like thematic folders where relevant content is added and organised. Content inside a collection can be further annotated and filtered.

Single content items, e.g. photos, videos, posts or social accounts, in a collection can be verified in detail, through each item's verification page. For this purpose, the tool provides important third-party plug-in tools, which offer both high-end functions such as "reverse image search" and detection of deepfakes, as well as technologically basic but important services like "image magnification". Following their investigation, journalists can then mark a content item as "pending", "unclear", "verified" or "fake". Content collections, the verification process as well as single results can be easily shared and discussed with other users through a set of collaboration tools, like shared notes, chat, and direct messages. Although individual journalists can use the tool, the focus of the approach is on remote collaboration across teams.

In summary, the existing functions are: (i) monitoring social networks, (ii) detecting twitter communities, (iii) organising work/findings in collections, (iv) easily importing content, (v) collaborating in real-time (vi) extending verification networks, (vii) managing Collection Items (viii) extracting and visualising useful information (ix) using effective verification tools and functions and (x) searching media archives.

Truly Media is constantly enhanced with the latest technologies applied in the field of media and content analysis, by introducing AI supported functionalities, like Natural Language Processing for named entity recognition and sentiment extraction and advanced Machine Learning methods for unsupervised topic clustering and online community detection.

InVID[29] is a platform that hosts different tools to detect, authenticate and check the reliability and authenticity of videos. The types of users targeted by this tool are media professionals and fack-checkers.

InVID has a series of tools and plugins to analyse the video contents and facilitate the integration of these videos, ensuring the reliability of the content. InVID was born with the aim of being a platform where several video and image verification tools are integrated to help journalists, fact-checkers and human rights activists in the verification process without having to rely on multiple external tools. Some of InVID's functionalities include a verification plugin for Firefox and Chrome that allows obtaining contextual information about videos from different platforms such as Twitter, Facebook or Youtube, as well as the ability to reverse search images to check their veracity or split videos into frames for better analysis, among others. It also has several tools, such as InVID Verification Application, which integrates a set of tools for the verification of user-generated videos, including a video player, the possibility of reverse search, checking the origin and copyright of the video, checking the forensic information of the video through filters or an inspection at frame level. InVID

[28] https://www.truly.media/
[29] https://www.invid-project.eu

also provides a dashboard, called InVID Multimodal Analytics Dashboard to visualise and explore the information of the content verified from the previous tools or even generate PDF reports. As a last tool to highlight, InVID has a mobile application available on iOS and Android to capture and enrich videos with the location and metadata of the device, annotations, etc. and send them directly to the media to incorporate them into the breaking news story.

Ms.W (The Misinformation Widget). The Misinformation Widget[30], or Ms.W, is a science communication tool being developed by TRESCA, a Horizon 2020 project on scientific disinformation. The two main functionalities of Ms.W are as follows:

- Assessing the credibility of a source, which can be broken down into a variety of operations allowed by the REST APIs included in the toolbox: *(i)* check whether the accounts authoring the analyzed posts are human or bots; *(ii)* verify the credentials of the authors who claimed to be experts when posting about a specific topic; *(iii)* detect partisan bias.
- Verifying the veracity of a claim, i.e. making sure that *(i)* the author has sufficient knowledge about the topic he/she is talking about, *(ii)* the claim is supported by sufficient evidence or research, *(iii)* the claim has not been manipulated or taken out of its original context to change its meaning, *(iv)* the title actually match the content of the article, *(v)* the text is not excessively biased and/or attempts to promote a distorted view. For these purposes, the widget offers the possibility to check that a credible fact-checking organisation has validated the claim, or that the source has not republished or manipulated an old news item as if it were relevant now, or to assess the polarity of the text and the emotions it attempts to provoke in the reader.

6.2.2 IBERIFIER partners initiatives

FacTeR-Check (Martín et al., 2022) represents a semantic-aware multilingual Transformer based architecture for semantic similarity evaluation, semi-automated fact-checking and tracking of information pieces in Online Social Networks. This architecture can, on the one hand, help general public in checking the veracity of a claim (i.e. a tweet) through context-aware automated comparison against a databases of hoaxes. On the other hand, it aims at providing useful tools for fact-checking organisations to track and monitor hoaxes circulating in OSNs.

The architecture provides two pipelines, one for semi-automated verification of claims; another for tracking known hoaxes on social media. The pipelines share three modules: a semantic similarity module, a Natural Language Inference (NLI) module and a information retrieval module. By using context-aware semantic similarity, the tool is able to find related fact-checks, while NLI allows to contrast the claim against reputable sources. This double process enables to perform semi-automated fact-checking.

In contrast to other approaches, this tool relies on a semi-automated fact-checking process, using fact-checkers databases as source of verified claims. This ensures the quality of the predictions of the model, instead of relying on training sets of false data that severely limit the capacity of the model to detect the most recent falsehoods. Another major difference lies in the context-aware and multilingual capacities, introduced due to the use of the Transformer architecture, a very important advance to deal with human language understanding and to allow comparisons between different languages without translation. The multilingual capacity helps to do fact check no matter the language of the candidate claim and the verified facts is. Finally, a tracking module is integrated to analyse the whole propagation cascade of the hoax, a very valuable tool to explore its whole story in a social network.

[30] https://trescaproject.eu/2021/07/19/ms-w-the-misinformation-widget/

CheckerOrSpreader. Users play a key role in the creation and spread of fake news. Several fact-checking websites have been developed to refute false fabricated information. As a result of these platforms, some users are interested in sharing their posts to debunk fake news. These users are known as fact-checkers users. The CheckerOrSpreader model can classify a user as a potential fact-checker or a potential spreader of fake news (Giachanou, Ríssola, Ghanem, Crestani, & Rosso, 2020). The model is based on a convolutional neural network (CNN) and combines word embeddings with features that represent users' personality traits and the linguistic patterns used in their tweets. Experimental results show that leveraging linguistic patterns and personality traits can improve performance in differentiating between verifiers and propagators of fake news. The CheckerOrSpreader software module (although it has been named FakeOrFact) is available in the repository: https://github.com/bilalghanem/FakeOrFact

FakeFlow tool is based on neural networks (Emo-analysis) and compare the language of fake news with the language of real news from an emotional perspective, considering a set of information types (propaganda, hoax, clickbait and satire). Experiments have shown that false information has different emotional patterns in each of its types, and emotions play a key role in misleading the reader (Ghanem, Rosso, & Rangel, 2020) (Ghanem et al., 2021). Continuing in this vein, fake news articles often arouse readers' attention through emotional appeals that arouse their feelings. Unlike short news texts, authors of longer articles can exploit these affective factors to manipulate readers by adding exaggerations or fabricating events, in order to affect readers' emotions. To capture this, we propose to model the affective information flow in fake news articles using a neural architecture. The proposed model learns this flow by combining the topic and affective information extracted from the text. The performance of the model is evaluated with several experiments on four real-world datasets. The results show that FakeFlow achieves superior results when compared to state-of-the-art methods, confirming the importance of capturing the flow of affective information in news articles. The FakeFlow module, as well as all its specifications, is available in the repository: https://github.com/bilalghanem/fake_flow

UPV-28-UNITO software module (Ghanem, Cignarella, Bosco, Rosso, & Pardo, 2019) was carried out to participate in the RumorEval 2019 shared task competition (Gorrell et al., 2019), whose main mission was to automatically determine the veracity of rumors. The approach exploits both classical machine learning algorithms and word embeddings and is based on several groups of features: stylistic, lexical, stylistic, emotional, sentimental, and meta-structural Twitter-based features. In addition, a new set of features is introduced to exploit the syntactic information of texts. The UPV-28-UNITO software module is available in the repository: https://github.com/bilalghanem/UPV-28-UNITO

Multimodal Multi-image Fake News Detection. A multimodal system to address the problem of fake news detection has been developed (Giahanou et al., 2020). The proposed system combines textual, visual and semantic information. For the textual representation, it uses BERT-Base to better capture the underlying semantic and contextual meaning of the text. For the visual representation, it extracts image tags from multiple images containing the items using, for example, the VGG-16 model. The semantic information is represented by the image-text similarity which is computed using the cosine similarity of the title and image label embeddings. Then, the different components are concatenated to make the final prediction.

Deep Fake Detection. This is a multiservice application that includes the detection of DeepFakes in images and video. It is presented as a web application[31], a REST API and a plugin for FOCA[32] (open source tool for finding hidden metadata in documents developed by *ElevenPaths*). The services are as follows:

- FaceForensic++ (Rossler et al., 2019): service for the analysis of video manipulation. Some of the main functionalities we found are the creation of a test set for the analysis of DeepFakes and a mechanism for the detection of content manipulation based on the XceptionNet.
- Reverse Engineering: for the detection of synthetic faces based on (Asnani, Yin, Hassner, & Liu, 2021).
- Keras CNN-RNN: Keras implementation for fake video classification based on a two network architecture, one convolutional and another recurrent to take advantage of the spatial and temporal information provided by the frames.
- KerasImg: uploaded to TensorFlow Hub[33]. It also allows a qualitative analysis with the LIME tool (based on (Ribeiro, Singh, & Guestrin, 2016)) to obtain information on why the network has classified an image as real or fake.

Claim Verification model with Semantic Knowledge. Claim Verification is the task of verifying a claim by finding the right evidences and inferring its truth label in an automated way. The benchmark dataset for this task is FEVER (Thorne et al., 2018), an English-language dataset that has both evidence and truth-labels annotated. In (Calvo Figueras, Oller, & Agerri, 2022), we have trained a model that takes advantage of the information embedded in Semantic Role Labels to guide the inference part of this task.

[31] https://reactui-utoehvsqvq-ew.a.run.app
[32] https://github.com/ElevenPaths/FOCA
[33] https://www.tensorflow.org/hub?hl=es-419

Table 2: Summary of software tools to assist in the information verification process

Technology Name	European Initiative	IBERIFIER Tech	Purpose	Core functionality	Technology Readiness Level	Data Type	Language	URL
Gephi	No	No	General	Visualization and manipulation of graphs	TRL9	Tabular data	English	https://gephi.org/
Graphext	No	No	General	Transform, explore, visualise and analyse data from different sources	TRL9	Tabular data and text	English	https://www.graphext.com/
GATE	No	No	General	Analyze of all types of text regardless of its size	TRL9	Text	Mainly in English with some multilingual functionali-ties	https://gate.ac.uk/
Truly Media	Yes	No	Specific	Web-based journalism platform focused on the collaborative verification of content from social and digital media	TRL9	Text, image, video and tabular data	English	https://www.truly.media/
InVID	Yes	No	Specific	Platform that hosts different tools to detect, authenticate and check the reliability and authenticity of images and videos	TRL9	Image and video	English	https://www.invid-project.eu/
FacTeR-Check	No	Yes (UPM)	Specific	Semi-automated fact-checking and tracking of information pieces in Online Social Networks.	TRL4	Text	Multilingual	https://www.sciencedirect.com/science/article/pii/S0950705122006323
CheckerorSpreader	No	Yes (UPV)	Specific	Users classification as checkers or fake news spreaders	TRL4	Text	English	https://github.com/bilalghanem/FakeOrFact
FakeFlow	No	Yes (UPV)	Specific	Fake News detection by modeling the flow of affective information	TRL4	Text	English	https://github.com/bilalghanem/fake_flow
UPV-28-UNITO	No	Yes (UPV)	Specific	Rumor stance classification	TRL4	Text	English	https://github.com/bilalghanem/UPV-28-UNITO
Multimodal Multi-image Fake News Detection	No	Yes (UPV)	Specific	Fake news detection	TRL4	Text and image	English	https://github.com/zgb0537/Multimodal-Fake-News-Detection-with-Textual-Visual-and-Semantic-Information
Deep Fake Detection	No	Yes (UGR)	Specific	Deepfakes detection in videos and images	TRL3	Image and video	English (GUI in Spanish)	https://github.com/PedroMFC/TFM-DeepFakes
Twitter Analysis Toolkit	No	Yes (BSC)	General	Toolkit that wraps different functions to help for tweets analysis	TRL3	Text	English and Spanish	https://gitlab.bsc.es/rrssalud/twitter_toolkit
Spanish and Catalan massive Language Models and finetunings for POS, NER, QA tasks	No	Yes (BSC)	General	NLP tools to parse and understand anything written in Natural Language.	TRL8	Text	Spanish and Catalan	https://huggingface.co/PlanTL-GOB-ES
Claim Verification model with Semantic Knowledge	No	Yes (BSC)	Specific	Claim verification	TRL3	Text	English	https://aclanthology.org/2022.fever-1.5.pdf
Pyleetspeak: Word camouflaging generator and Data Augmentation package	No	Yes (UPM)	General	Python package to generate a camouflaged version of an introduced text	TRL3	Text	Latin-derived alphabet languages	https://pypi.org/project/pyleetspeak/
LeetSpeak-NER Transformer models	No	Yes (UPM)	General	Transformers and Spacy v3 models for word camouflage NER detection	TRL3	Text	Spanish and English	https://huggingface.co/Huertas97/en_roberta_base_leetspeak_ner and https://huggingface.co/Huertas97/es_roberta_base_bne_leetspeak_ner
LeetSpeak-NER App: Word Camouflaging NER detector	No	Yes (UPM)	General	Streamlite app to detect word camouflaging and content evasion	TRL3	Text	Spanish and English	https://huggingface.co/spaces/Huertas97/LeetSpeak-NER
FacteR-ChecKey API	No	Yes (UPM)	General	Automatic semantic-aware keyword extraction system and query generator for searching through Twitter API	TRL4	Text	Multilingual	http://g2.etsisi.upm.es:8854/docs#/
BERTuit	No	Yes (UPM)	General	Transfomer model for Spanish language understanding in Twitter	TRL4	Text	Spanish	https://arxiv.org/abs/2204.03465
Ms.W (Misinformation Widget)	Yes	No	Specific	Misinformation detection	TRL3	Test	English	https://trescaproject.eu/2021/07/19/ms-w-the-misinformation-widget/

7 Conclusions and Future Work

This report began introducing relevant concepts and methods of Machine Learning (ML), the most active area of Artificial Intelligence (AI) addressing disinformation nowadays. In the second part, the report discussed how ML techniques have been applied in disinformation analysis, e.g., from the automatic identification of false news and hoaxes to identifying relevant actors propagating disinformation. Unfortunately, these proposals have not been extensively validated in different contexts and actual use cases. Furthermore, since ML (and mainly supervised ML) strongly relies on the data quality used to train the algorithms, the report revised frequently used datasets. We acknowledged the need for more data in languages other than English, particularly in Spanish and (more critically) Portuguese. Finally, the report enumerated disinformation analysis tools, primarily focused on data collection from social networks and natural language processing. While the number of open-source tools indicates that the ecosystem is very active, the assorted technological readiness level and technical requirements make it difficult to bring them into daily practice without further development.

Despite the growth of the scientific literature on AI and disinformation in recent years, most studies apply supervised ML techniques to detect false information in social networks. However, disinformation is a very complex phenomenon challenging to address through such a narrow perspective. Furthermore, certain limits must be considered when developing automatic methods, as these methods for detecting disinformation cannot become censors of opinions and thus alter a fundamental right such as freedom of expression. Instead, we propose to incorporate AI to mitigate the impact of disinformation at different stages of their lifecycle:

- in the phase of creation and production of false information, ecosystems prone to be the nucleus of that content can be detected by studying the beliefs and opinions of users about a particular phenomenon;
- in the phase of massive propagation of fraudulent content, it is possible to investigate the types of users who spread hoaxes and how they are spread;
- and finally, once disinformation content has become sufficiently significant to be relevant, mitigation strategies can be implemented through media literacy, for example, within communities prone to sharing conspiracy theories.

Besides this fact, many of the AI methods currently in use have the particularity that they are not explainable, i.e. they do not provide the user with the necessary information to understand how the AI method has made the decision. Explainability is expected to be a requirement of the future European regulation setting out the rules for the implementation of AI-based systems[34], in particular for higher risk systems, e.g., those applied in justice, law enforcement, administration, etc. Therefore, one of the efforts of the scientific community should be to create and implement explainable AI methods to determine how the decision has been made, e.g., to classify information as trustworthy or not.

Last but not least, an essential factor in achieving fair and reliable AI-based systems is curating the data on which the models are trained. The data must be unbiased, objective, and of high quality, which in some scenarios can be impossible. In the case of disinformation analysis, the annotators must have a deep knowledge of the domain where the AI system will be applied in order to be able to label accurately and reliably. This requires strong collaboration of technology developers with transparent fact-checking organizations, as well as a deep knowledge of the contextual nuances of the phenomenon. Being a regional hub including multidisciplinary experts, IBERIFIER is well-equipped to make relevant contributions in this regard.

[34] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206

References

Abdullah, T., & Ahmet, A. (2022). Deep learning in sentiment analysis: A survey of recent architectures. *ACM Computing Surveys*. doi: 10.1145/3548772

Adamo, J.-M. (2001). *Data mining for association rules and sequential patterns: Sequential and parallel algorithms*. Springer. doi: 10.1007/978-1-4613-0085-4

Afroz, S., Brennan, M., & Greenstadt, R. (2012). Detecting hoaxes, frauds, and deception in writing style online. In *Proceedings - 2012 ieee symposium on security and privacy, s and p 2012* (pp. 461–475). Institute of Electrical and Electronics Engineers Inc. (33rd IEEE Symposium on Security and Privacy, S and P 2012 ; Conference date: 21-05-2012 Through 23-05-2012) doi: 10.1109/SP.2012.34

Aggarwal, C. C. (2011). An introduction to social network data analytics. In *Social network data analytics* (pp. 1–15). Springer.

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, *22*, 207–216. doi: 10.1145/170036.170072

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., & Vollgraf, R. (2019). FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 annual conference of the north american chapter of the association for computational linguistics (demonstrations)* (pp. 54–59).

Amador, J., Molina-Solana, M., & Gómez-Romero, J. (2019). Towards easy-to-implement misinformation automatic detection for online social media. In *Conference for truth and trust online (TTO).* London, UK. Retrieved from https://truthandtrustonline.com/

Armengol-Estapé, J., Carrino, C. P., Rodriguez-Penagos, C., de Gibert Bonet, O., Armentano-Oller, C., Gonzalez-Agirre, A., ... Villegas, M. (2021). Are multilingual models the best choice for moderately under-resourced languages? A comprehensive assessment for Catalan. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (pp. 4933–4946). Online: Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.437

Arnold, P. (2020). *The challenges of online fact checking* (Tech. Rep.). Full Fact. Retrieved 2022-08-12, from https://fullfact.org/media/uploads/coof-2020.pdf

Arrieta, A. B., Díaz-Rodríguez, N., Ser, J. D., Bennetot, A., Tabik, S., Barbado, A., ... Herrera, F. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, *58*, 82–115. doi: 10.1016/j.inffus.2019.12.012

Arslan, F., Hassan, N., Li, C., & Tremayne, M. (2020). A Benchmark Dataset of Check-worthy Factual Claims. *Proceedings of the International AAAI Conference on Web and Social Media*, *14*(1), 821–829. doi: 10.1609/icwsm.v14i1.7346

Asaad, B. A., & Erascu, M. (2018). A tool for fake news detection. *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 379–386.

Asnani, V., Yin, X., Hassner, T., & Liu, X. (2021). *Reverse engineering of generative models: Inferring model hyperparameters from generated images.* arXiv. doi: 10.48550/ARXIV.2106.07873

Azevedo, L., D'aquin, M., Davis, B., & Zarrouk, M. (2021). LUX (Linguistic aspects Under eXamination): Discourse Analysis for Automatic Fake News Classification. In *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th Interna-*

tional Joint Conference on Natural Language Processing (ACL-IJCNLP 2021) (pp. 41–56). Online, France: Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.4

Baccianella, S., Esuli, A., & Sebastiani, F. (2010). SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the seventh international conference on language resources and evaluation (LREC'10).* Valletta, Malta: European Language Resources Association (ELRA). Retrieved from http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf

Bahdanau, D., Cho, K., & Bengio, Y. (2014). *Neural machine translation by jointly learning to align and translate.* arXiv. doi: 10.48550/ARXIV.1409.0473

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007). Open information extraction from the web. In *Proceedings of the 20th international joint conference on artifical intelligence* (pp. 2670–2676). Morgan Kaufmann Publishers Inc.

Barrón-Cedeño, A., Elsayed, T., Nakov, P., Da San Martino, G., Hasanain, M., Suwaileh, R., ... Ali, Z. S. (2020). Overview of checkthat! 2020: Automatic identification and verification of claims in social media. In *Experimental ir meets multilinguality, multimodality, and interaction* (pp. 215–236). Springer International Publishing.

Bar-Yossef, O., Guy, I., Lempel, R., Maarek, Y., & Soroka, V. (2008). Cluster ranking with an application to mining mailbox networks. *Knowledge and Information Systems*, *14*(1), 101–139.

Bedi, P., & Sharma, C. (2016). Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *6*(3), 115–135. doi: 10.1002/widm.1178

Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*. doi: 10.1088/1742-5468/2008/10/P10008

Bondielli, A., & Marcelloni, F. (2019). A survey on fake news and rumour detection techniques. *Information Sciences*, *497*, 38–55. doi: 10.1016/j.ins.2019.05.035

Bonet-Jover, A., Piad-Morffis, A., Saquete, E., Martínez-Barco, P., & Ángel García-Cumbreras, M. (2021). Exploiting discourse structure of traditional digital media to enhance automatic fake news detection. *Expert Systems with Applications*, *169*, 114340. doi: 10.1016/j.eswa.2020.114340

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–140. doi: 10.1023/A:1018054314350

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32. doi: 10.1023/A:1010933404324

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... Amodei, D. (2020). Language models are few-shot learners. In *Advances in neural information processing systems 33.*

Buda, J., & Bolonyai, F. (2020). An ensemble model using n-grams and statistical features to identify fake news spreaders on twitter. *In: CLEF 2020 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2020)*, *abs/2009.13859*.

Calvo Figueras, B., Oller, M., & Agerri, R. (2022). A semantics-aware approach to automated claim verification. In *Proceedings of the fifth fact extraction and verification workshop (fever)* (pp. 37–48). Dublin, Ireland: Association for Computational Linguistics. doi: 10.18653/v1/2022.fever-1.5

Camacho, D., Panizo-LLedot, A., Bello-Orgaz, G., Gonzalez-Pardo, A., & Cambria, E. (2020). The four dimensions of social network analysis: An overview of research methods, applications, and software tools. *Information Fusion*, *63*, 88–120. doi: 10.1016/j.inffus.2020.05.009

Cambria, E., Wang, H., & White, B. (2014). Guest editorial: Big social data analysis. *Knowledge-based systems*, *69*(1), 1–2. doi: 10.1016/j.knosys.2014.07.002

Castelo, S., Almeida, T., Elghafari, A., Santos, A., Pham, K., Nakamura, E., & Freire, J. (2019). A Topic-Agnostic Approach for Identifying Fake News Pages. In *Companion Proceedings of The 2019 World Wide Web Conference* (pp. 975–980). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3308560.3316739

Chaturvedi, I., Ong, Y.-S., Tsang, I. W., Welsch, R. E., & Cambria, E. (2016). Learning word dependencies in text by means of a deep recurrent belief network. *Knowledge-Based Systems*, *108*, 144–154. doi: 10.1016/j.knosys.2016.07.019

Chen, D., Fisch, A., Weston, J., & Bordes, A. (2017). Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1870–1879). Vancouver, Canada: Association for Computational Linguistics. doi: 10.18653/v1/P17-1171

Choraś, M., Demestichas, K., Giełczyk, A., Álvaro Herrero, Ksieniewicz, P., Remoundou, K., ... Woźniak, M. (2021). Advanced machine learning techniques for fake news (online disinformation) detection: A systematic mapping study. *Applied Soft Computing*, *101*, 107050–107065. doi: 10.1016/j.asoc.2020.107050

Cohen, W. W. (1995). Fast effective rule induction. In *Machine learning proceedings 1995* (pp. 115–123). Tahoe City, California, USA: Elsevier. doi: 10.1016/B978-1-55860-377-6.50023-2

Courney, D. (2019). *How we use AI to help fact check party manifestos.* Retrieved 2022-08-12, from https://fullfact.org/blog/2019/dec/how-we-use-ai-help-fact-check-party-manifestos/

Dagar, D., & Vishwakarma, D. K. (2022). A literature review and perspectives in deepfakes: generation, detection, and applications. *International Journal of Multimedia Information Retrieval*, *11*, 219–289. doi: 10.1007/s13735-022-00241-w

Dang, H., Liu, F., Stehouwer, J., Liu, X., & Jain, A. K. (2020). On the detection of digital face manipulation. In *Proceedings of the ieee computer society conference on computer vision and pattern recognition* (pp. 5781–5790). doi: 10.1109/CVPR42600.2020.00582

De Choudhury, M., Mason, W. A., Hofman, J. M., & Watts, D. J. (2010). Inferring relevant social networks from interpersonal communication. In *Proceedings of the 19th international conference on world wide web* (pp. 301–310).

Del Vicario, M., Vivaldo, G., Bessi, A., Zollo, F., Scala, A., Caldarelli, G., & Quattrociocchi, W. (2016). Echo chambers: Emotional contagion and group polarization on facebook. *Scientific Reports*, *6*(37825). doi: 10.1038/srep37825

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 4171–4186). Association for Computational Linguistics. doi: 10.18653/v1/N19-1423

Dun, Y., Tu, K., Chen, C., Hou, C., & Yuan, X. (2021). Kan: Knowledge-aware attention network for fake news detection. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 35, pp. 81–89). doi: 10.1609/aaai.v35i1.16080

Eldesoky, I., & Moussa, F. (2021). Fake news detection based on word and document embedding using machine learning classifiers. *Journal of Theoretical and Applied Information Technology*, *99*(8), 1891–1901.

Engelen, J. E. V., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, *109*, 373–440. doi: 10.1007/s10994-019-05855-6

Ethem, A. (2020). *Introduction to machine learning* (Fourth ed.). The MIT Press.

Faralli, S., Stilo, G., & Velardi, P. (2017). Automatic acquisition of a taxonomy of microblogs users' interests. *Web Semantics: Science, Services and Agents on the World Wide Web*, *45*, 23–40.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, *39*, 27–34. doi: 10.1145/240455.240464

Fortunato, S. (2010). Community detection in graphs. *Physics reports*, *486*(3-5), 75–174.

Fu, L. (1994). Rule generation from neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, *24*, 1114–1124. doi: 10.1109/21.299696

Gerrard, Y., & Thornham, H. (2020). Content moderation: Social media's sexist assemblages. *New Media & Society*, *22*(7, SI), 1266–1286. doi: 10.1177/1461444820912540

Ghanem, B., Cignarella, A. T., Bosco, C., Rosso, P., & Pardo, F. M. R. (2019). UPV-28-UNITO at SemEval-2019 Task 7: Exploiting post's nesting and syntax information for rumor stance classification. In *Proceedings of the 13th international workshop on semantic evaluation* (pp. 1125–1131). doi: 10.18653/v1/S19-2197

Ghanem, B., Ponzetto, S. P., Rosso, P., & Rangel, F. (2021). FakeFlow: Fake news detection by modeling the flow of affective information. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: Main volume* (pp. 679–689). Online: Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.56

Ghanem, B., Rosso, P., & Rangel, F. (2020). An emotional analysis of false information in social media and news articles. *ACM Trans. Internet Technol.*, *20*(2). doi: 10.1145/3381750

Giachanou, A., Ghanem, B., Ríssola, E. A., Rosso, P., Crestani, F., & Oberski, D. (2022). The impact of psycholinguistic patterns in discriminating between fake news spreaders and fact checkers. *Data & Knowledge Engineering*, *138*, 101960. doi: 10.1016/j.datak.2021.101960

Giachanou, A., Ríssola, E. A., Ghanem, B., Crestani, F., & Rosso, P. (2020). The role of personality and linguistic patterns in discriminating between fake news spreaders and fact checkers. In E. Métais, F. Meziane, H. Horacek, & P. Cimiano (Eds.), *Natural language processing and information systems - 25th international conference on applications of natural language to information systems, NLDB 2020, saarbrücken, germany, june 24-26, 2020, proceedings* (Vol. 12089, pp. 181–192). Springer. doi: 10.1007/978-3-030-51310-8_17

Giachanou, A., Rosso, P., & Crestani, F. (2019). Leveraging emotional signals for credibility detection. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval* (p. 877—880). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3331184.3331285

Giachanou, A., Rosso, P., & Crestani, F. (2021). The impact of emotional signals on credibility assessment. *Journal of the Association for Information Science and Technology*, *72*(9), 1117–1132. doi: 10.1002/asi.24480

Giachanou, A., Zhang, G., & Rosso, P. (2020). Multimodal fake news detection with textual, visual and semantic information. In (pp. 30–38). Springer International Publishing. doi: 10.1007/978-3-030-58323-1_3

Giahanou, A., Zhang, G., & Rosso, P. (2020). Multimodal multi-image fake news detection. *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, 647–654.

Giglou, H. B., Razmara, J., Rahgouy, M., & Sanaei, M. (2020). Lsaconet: A combination of lexical and conceptual features for analysis of fake news spreaders on twitter. *In: CLEF 2020 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2020)*, *abs/2009.13859*.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. (http://www.deeplearningbook.org)

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, *3*. doi: 10.3156/jsoft.29.5_177_2

Gorrell, G., Kochkina, E., Liakata, M., Aker, A., Zubiaga, A., Bontcheva, K., & Derczynski, L. (2019). SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours. In *Proceedings of the 13th international workshop on semantic evaluation* (pp. 845–854). Minneapolis, Minnesota, USA: Association for Computational Linguistics. doi: 10.18653/v1/S19-2147

Greengard, S. (2019). Will deepfakes do deep damage? *Communications of the ACM*, *63*, 17–19. doi: 10.1145/3371409

Grootendorst, M. (2020). *Keybert: Minimal keyword extraction with bert.* Zenodo. doi: 10.5281/zenodo.4461265

Groshev, A., Maltseva, A., Chesakov, D., Kuznetsov, A., & Dimitrov, D. (2022). Ghost—a new face swap approach for image and video domains. *IEEE Access*, *10*, 83452–83462. doi: 10.1109/ACCESS.2022.3196668

Gryc, W., & Moilanen, K. (2010). Leveraging textual sentiment analysis with social network modeling: Sentiment analysis of political blogs in the 2008 u.s. presidential election. In *From text to political positions: Text analysis across disciplines* (pp. 47–70).

Guille, A., Hacid, H., Favre, C., & Zighed, D. (2013). Information diffusion in online social networks: A survey. *ACM SIGMOD Record*, *42*, 17–28. doi: 10.1145/2503792.2503797

Gutiérrez-Fandiño, A., Armengol-Estapé, J., Pàmies, M., Llop-Palao, J., Silveira-Ocampo, J., Carrino, C. P., ... Villegas, M. (2022). MarIA: Spanish language models. *Procesamiento del Lenguaje Natural*, *68*, 39–60. doi: 10.26342/2022-68-3

Gómez-Adorno, H., Posadas-Durán, J. P., Bel Enguix, G., & Porto Capetillo, C. (2021). Overview of fakedes at iberlef 2021: Fake news detection in spanish shared task. *Procesamiento del Lenguaje Natural*, *67*(0), 223–231. Retrieved from http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6391

Han, J., Kamber, M., & Pei, J. (2012). *Data mining: Concepts and techniques* (Third ed.). Morgan Kaufmann. doi: 10.1016/C2009-0-61819-5

Hansen, C., Hansen, C., Alstrup, S., Grue Simonsen, J., & Lioma, C. (2019). Neural Check-Worthiness Ranking with Weak Supervision: Finding Sentences for Fact-Checking. In *Companion Proceedings of The 2019 World Wide Web Conference* (pp. 994–1000). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3308560.3316736

Harrigan, P., Daly, T. M., Coussement, K., Lee, J. A., Soutar, G. N., & Evers, U. (2021). Identifying influencers on social media. *International Journal of Information Management*, *56*, 102246.

Harrigan, P., Evers, U., Miles, M., & Daly, T. (2017). Customer engagement with tourism social media brands. *Tourism management*, *59*, 597–609.

Hashemi, A., Zarei, M. R., Moosavi, M. R., & Taheri, M. (2020). Fake news spreader identification in twitter using ensemble modeling. In *CLEF 2020 labs and Workshops, Notebook Papers* (Vol. abs/2009.13859).

Hassan, N., Arslan, F., Li, C., & Tremayne, M. (2017). Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining* (p. 1803–1812). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3097983.3098131

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression* (Third ed.). John Wiley & Sons. doi: 10.1002/9781118548387

Huertas-Tato, J., Martin, A., & Camacho, D. (2022). *Bertuit: Understanding spanish language in twitter through a native transformer.* arXiv. doi: 10.48550/arXiv.2204.03465

Islam, M. R., Liu, S., Wang, X., & Xu, G. (2020). Deep learning for misinformation detection on online social networks: a survey and new perspectives. *Social Network Analysis and Mining*, *10*, 82–101. doi: 10.1007/s13278-020-00696-x

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning with applications in r* (Second ed.). Springer.

John, O. P., & Srivastava, S. (1999). The big five trait taxonomy: History, measurement, and theoretical perspectives. In *Handbook of personality: Theory and research* (pp. 102–138).

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*, 255–260. doi: 10.1126/science.aaa8415

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2016). *Fasttext.zip: Compressing text classification models.* arXiv. doi: 10.48550/ARXIV.1612.03651

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th conference of the European chapter of the association for computational linguistics* (pp. 427–431). Association for Computational Linguistics. Retrieved from https://aclanthology.org/E17-2068

Jung, T., Kim, S., & Kim, K. (2020). Deepvision: Deepfakes detection using human eye blinking pattern. *IEEE Access*, *8*, 83144–83154. doi: 10.1109/ACCESS.2020.2988660

Kannan, R., Vempala, S., & Vetta, A. (2004). On clusterings: Good, bad and spectral. *Journal of the ACM (JACM)*, *51*(3), 497–515.

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. In *6th International Conference on Learning Representations, ICLR 2018.*

Karras, T., Laine, S., & Aila, T. (2021). A style-based generator architecture for generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(12), 4217–4228. doi: 10.1109/TPAMI.2020.2970919

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data*. John Wiley & Sons, Inc. doi: 10.1002/9780470316801

Kaur, D., Uslu, S., Rittichier, K. J., & Durresi, A. (2023). Trustworthy artificial intelligence: a review. *ACM Computing Surveys*, *55*(39), 1–38. doi: 10.1145/3491209

Khattar, D., Goud, J. S., Gupta, M., & Varma, V. (2019). MVAE: Multimodal variational autoencoder for fake news detection. In *The world wide web conference* (pp. 2915–2921). doi: 10.1145/3308558.3313552

Koloski, B., Pollak, S., & Škrlj, B. (2020). Multilingual detection of fake news spreaders via sparse matrix factorization. In L. Cappellato, C. Eickhoff, N. Ferro, & A. Névéol (Eds.), *CLEF 2020 labs and workshops, notebook papers* (Vol. abs/2009.13859).

Konstantinovskiy, L., Price, O., Babakar, M., & Zubiaga, A. (2021). Toward Automated Factchecking: Developing an Annotation Schema and Benchmark for Consistent Automated Claim Detection. *Digital Threats: Research and Practice*, *2*(2), 1–16. doi: 10.1145/3412869

Labadie-Tamayo, R., Castro-Castro, D., & Ortega-Bueno, R. (2020). Fusing stylistic features with deep-learning methods for profiling fake news spreader. In L. Cappellato, C. Eickhoff, N. Ferro, & A. Névéol (Eds.), *CLEF 2020 labs and workshops, notebook papers.* (Vol. abs/2009.13859).

Lalou, M., Tahraoui, M. A., & Kheddouci, H. (2018). The critical node detection problem in networks: A survey. *Computer Science Review*, *28*, 92–117.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*, 541–551. doi: 10.1162/neco.1989.1.4.541

Li, J., Xu, L., Tang, L., Wang, S., & Li, L. (2018). Big data in tourism research: A literature review. *Tourism Management*, *68*, 301–323.

Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. (2021). Explainable AI: A review of machine learning interpretability methods. *Entropy*, *23*(1). doi: 10.3390/e23010018

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). *RoBERTa: A robustly optimized BERT Pretraining Approach.* arXiv. doi: 10.48550/ARXIV.1907.11692

Liu, Y., Tantithamthavorn, C., Li, L., & Liu, Y. (2022). Deep learning for android malware defenses: a systematic literature review. *ACM Computing Surveys*. doi: 10.1145/3544968

Ma, C., Zhang, W. E., Guo, M., Wang, H., & Sheng, Q. Z. (2022). Multi-document summarization via deep learning techniques: A survey. *ACM Computing Surveys*. doi: 10.1145/3529754

Ma, J., Gao, W., Wei, Z., Lu, Y., & Wong, K.-F. (2015). Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th acm international on conference on information and knowledge management (cikm '15).* doi: 10.1145/2806416.2806607

Malik, A., Kuribayashi, M., Abdullahi, S. M., & Khan, A. N. (2022). Deepfake detection for human face images and videos: A survey. *IEEE Access*, *10*, 18757–18775. doi: 10.1109/ACCESS.2022.3151186

Manning, C., & Schutze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

Marinova, Z., Teyssou, D., Sarris, N., Spangenberg, J., Papadopoulos, S., Alaphilippe, A., & Bontcheva, K. (2019). WeVerify: Wider and enhanced verification for you - project overview and tool demonstration. In *Proceedings of the Conference for Truth and Trust Online 2019.* doi: 10.36370/tto.2019.23

Martín, A., Huertas-Tato, J., Huertas-García, Á., Villar-Rodríguez, G., & Camacho, D. (2022). FacTeR-Check: Semi-automated fact-checking through Semantic Similarity and Natural Language Inference. *Knowledge-Based Systems*, *251*, 109265. doi: 10.1016/j.knosys.2022.109265

Masood, M., Nawaz, M., Malik, K. M., Javed, A., Irtaza, A., & Malik, H. (2022). Deepfakes generation and detection: state-of-the-art, open challenges, countermeasures, and way forward. *Applied Intelligence*. doi: 10.1007/s10489-022-03766-z

Mayer-Schönberger, V., & Cukier, K. (2013). *Big data: A revolution that will transform how we live, work, and think* (1st ed.). Eamon Dolan/Houghton Mifflin Harcourt.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys*, *54*, 1–35. doi: 10.1145/3457607

Mendoza, M., Poblete, B., & Castillo, C. (2010). Twitter under crisis: Can we trust what we rt? In *Proceedings of the first workshop on social media analytics* (pp. 71–79). Association for Computing Machinery. doi: 10.1145/1964858.1964869

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *1st international conference on learning representations, iclr.*

Mirsky, Y., & Lee, W. (2022). The creation and detection of deepfakes: A survey. *ACM Computing Surveys*, *54*, 1–41. doi: 10.1145/3425780

Mitchell, T. M. (1997). *Machine learning*. McGraw Hill.

Mitra, T., & Gilbert, E. (2015). CREDBANK: A Large-Scale Social Media Corpus with Associated Credibility Annotations. In *Proceedings of the international aaai conference on web and social media* (Vol. 9, p. 258-267). doi: 10.1609/icwsm.v9i1.14625

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). *Playing Atari with Deep Reinforcement Learning.* arXiv. doi: 10.48550/ARXIV.1312.5602

Mohammad, S., & Turney, P. (2013). *NRC emotion lexicon* (Tech. Rep.). National Research Council of Canada. doi: 10.4224/21270984

Molina-Solana, M., Amador Diaz Lopez, J., & Gómez-Romero, J. (2018). Deep learning for fake news classification. In *Proc. I workshop in deep learning (DEPL 2018)* (pp. 1197–1201). Granada, Spain. Retrieved from http://caepia18.aepia.org

Montani, I., Honnibal, M., Van Landeghem, S., & Boyd, A. (2020). *spaCy: Industrial-strength Natural Language Processing in Python.* Zenodo. doi: 10.5281/zenodo.1212303

Morente-Molinera, J., Kou, G., Samuylov, K., Ureña, R., & Herrera-Viedma, E. (2019). Carrying out consensual group decision making processes under social networks using sentiment analysis

over comparative expressions. *Knowledge-Based Systems*, *165*, 335–345. doi: 10.1016/j.knosys.2018.12.006

Motulsky, H. J., & Ransnas, L. A. (1987). Fitting curves to data using nonlinear regression: a practical and nonmathematical review. *The FASEB Journal*, *1*, 365–374. doi: 10.1096/fasebj.1.5.3315805

Nakamura, K., Levy, S., & Wang, W. Y. (2020). Fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection. In *Proceedings of the 12th language resources and evaluation conference* (pp. 6149–6157). Marseille, France: European Language Resources Association. Retrieved from https://aclanthology.org/2020.lrec-1.755

Nakov, P., Barrón-Cedeño, A., Elsayed, T., Suwaileh, R., Màrquez, L., Zaghouani, W., ... Da San Martino, G. (2018). Overview of the CLEF-2018 CheckThat! Lab on Automatic Identification and Verification of Political Claims. In P. Bellot et al. (Eds.), *Experimental IR Meets Multilinguality, Multimodality, and Interaction* (pp. 372–387). Cham: Springer International Publishing. doi: 10.1007/978-3-319-98932-7_32

Nakov, P., Corney, D., Hasanain, M., Alam, F., Elsayed, T., Barrón-Cedeño, A., ... Martino, G. D. S. (2021). Automated Fact-Checking for Assisting Human Fact-Checkers. In *Ijcai international joint conference on artificial intelligence* (pp. 4551–4558). doi: 10.24963/ijcai.2021/619

Nakov, P., Da San Martino, G., Elsayed, T., Barrón-Cedeño, A., Míguez, R., Shaar, S., ... Mandl, T. (2021). The CLEF-2021 CheckThat! Lab on Detecting Check-Worthy Claims, Previously Fact-Checked Claims, and Fake News. In D. Hiemstra, M.-F. Moens, J. Mothe, R. Perego, M. Potthast, & F. Sebastiani (Eds.), *Advances in Information Retrieval* (pp. 639–649). Cham: Springer International Publishing. doi: 10.1007/978-3-030-72240-1_75

Nataraj, L., Mohammed, T. M., Manjunath, B. S., Chandrasekaran, S., Flenner, A., Bappy, J. H., & Roy-Chowdhury, A. K. (2019). Detecting GAN generated fake images using co-occurrence matrices. In *Is and t international symposium on electronic imaging science and technology* (Vol. 2019). doi: 10.2352/ISSN.2470-1173.2019.5.MWSF-532

Neuman, Y., & Cohen, Y. (2014). A vectorial semantics approach to personality assessment. *Scientific reports*, *4*, 4761. doi: 10.1038/srep04761

Oehmichen, A., Hua, K., Amador, J., Molina-Solana, M., Gómez-Romero, J., & Guo, Y. (2019). Not all lies are equal. a study into the engineering of political misinformation in the 2016 us presidential election. *IEEE Access*, *7*, 126305–126314. doi: 10.1109/ACCESS.2019.2938389

Pal, C., & McCallum, A. (2006). CC prediction with graphical models. In *Conference on email and anti-spam.*

Pan, J., Pavlova, S., Li, C., Li, N., Li, Y., & Liu, J. (2018). Content based fake news detection using knowledge graphs. In *17th international semantic web conference* (pp. 669–683). Springer International Publishing. doi: 10.1007/978-3-030-00671-6_39

Panizo-LLedot, A., Torregrosa, J., Bello-Orgaz, G., Thorburn, J., & Camacho, D. (2019). Describing alt-right communities and their discourse on twitter during the 2018 us mid-term elections. In *International conference on complex networks and their applications* (pp. 427–439).

Pasi, G., Grandis, M. D., & Viviani, M. (2020). Decision making over multiple criteria to assess news credibility in microblogging sites. In *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (pp. 1–8).

Pathak, N., Delong, C., Banerjee, A., & Erickson, K. (2008). Social topic models for community extraction. In *SNA-KDD Workshop.*

Peng, B., Fan, H., Wang, W., Dong, J., & Lyu, S. (2022). A unified framework for high fidelity face swap and expression reenactment. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*, 3673–3684. doi: 10.1109/TCSVT.2021.3106047

Pennebaker, J. W., Boyd, R. L., Jordan, K., & Blackburn, K. G. (2015). *The development and psychometric properties of liwc2015* (Tech. Rep.). Austin, TX: University of Texas at Austin. doi: 10.15781/T29G6Z

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543). Doha, Qatar: Association for Computational Linguistics. doi: 10.3115/v1/D14-1162

Posadas-Durán, J.-P., Gómez-Adorno, H., Sidorov, G., & Escobar, J. J. M. (2019). Detection of fake news in a new corpus for the spanish language. *Journal of Intelligent & Fuzzy Systems*, *36*(5), 4869–4876. doi: 10.3233/JIFS-179034

Qazi, M., Khan, M. U., & Ali, M. (2020). Detection of fake news using transformer model. In *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (pp. 1–6). doi: 10.1109/iCoMET48670.2020.9074071

Qi, P., Cao, J., Yang, T., Guo, J., & Li, J. (2019). Exploiting multi-domain visual information for fake news detection. In *2019 IEEE International Conference on Data Mining (ICDM)* (pp. 518–527). doi: 10.1109/ICDM.2019.00062

Rana, M. S., Nobi, M. N., Murali, B., & Sung, A. H. (2022). Deepfake detection: A systematic literature review. *IEEE Access*, *10*, 25494–25513. doi: 10.1109/ACCESS.2022.3154404

Rangel, F., Giachanou, A., Ghanem, B., & Rosso, P. (2020). Overview of the 8th author profiling task at pan 2020: Profiling fake news spreaders on twitter. In L. Cappellato, C. Eickhoff, N. Ferro, & A. Névéol (Eds.), *CLEF 2020 Labs and Workshops, Notebook Papers.*

Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., & Choi, Y. (2017). Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Procs 2017 conference on empirical methods in natural language processing* (pp. 2931–2937). Copenhagen, Denmark: Association for Computational Linguistics. doi: 10.18653/v1/D17-1317

Reddy, S. M., Suman, C., Saha, S., & Bhattacharyya, P. (2020). A GRU-based Fake News Prediction System: Working Notes for UrduFake-FIRE 2020. In *CEUR Workshop Proceedings* (Vol. 2826).

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Procs. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144). doi: 10.1145/2939672.2939778

Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Niessner, M. (2019). FaceForensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1–11). doi: 10.1109/ICCV.2019.00009

Roux, M. (2018). A comparative study of divisive and agglomerative hierarchical clustering algorithms. *Journal of Classification*, *35*, 345–366. doi: 10.1007/s00357-018-9259-9

Ruchansky, N., Seo, S., & Liu, Y. (2017). CSI: A Hybrid Deep Model for Fake News Detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (pp. 797–806). Association for Computing Machinery. doi: 10.1145/3132847.3132877

Ruffo, G., Semeraro, A., Giachanou, A., & Rosso, P. (2023, 2). Studying fake news spreading, polarisation dynamics, and manipulation by bots: A tale of networks and language. *Computer Science Review*, *47*, 100531. doi: 10.1016/j.cosrev.2022.100531

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536. doi: 10.1038/323533a0

Saif, S., & Tehseen, S. (2022). Deepfake videos: Synthesis and detection techniques - a survey. *Journal of Intelligent and Fuzzy Systems*, *42*, 2989–3009. doi: 10.3233/JIFS-210625

Sakketou, F., Plepi, J., Cervero, R., Geiss, H.-J., Rosso, P., & Flek, L. (2022). New dataset for identifying misinformation spreaders and political bias. In *Proc. 13th int. conf. on language resources and evaluation, lrec-2022* (pp. 6149–6157). Marseille, France: European Language Resources Association. Retrieved from https://aclanthology.org/2020.lrec-1.755

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, *3*, 210–229.

Santia, G. C., & Williams, J. R. (2018). BuzzFace: A News Veracity Dataset with Facebook User Commentary and Egos. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 12, pp. 531–540). doi: 10.1609/icwsm.v12i1.14985

Satapathy, R., Cambria, E., & Hussain, A. (2017). SenticNet. In *Sentiment Analysis in the Bio-Medical Domain* (pp. 39–103). doi: 10.1007/978-3-319-68468-0_3

Schuster, T., Schuster, R., Shah, D. J., & Barzilay, R. (2020). The Limitations of Stylometry for Detecting Machine-Generated Fake News. *Computational Linguistics*, *46*(2), 499–510. doi: 10.1162/coli_a_00380

Shaar, S., Babulkov, N., Da San Martino, G., & Nakov, P. (2020). That is a Known Lie: Detecting Previously Fact-Checked Claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 3607–3618). Online: Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.332

Shams, M., Saffar, M., Shakery, A., & Faili, H. (2012). Applying Sentiment and Social Network Analysis in User Modeling. In *Computational Linguistics and Intelligent Text Processing* (pp. 526–539). Springer Berlin Heidelberg.

Shiralkar, P., Flammini, A., Menczer, F., & Ciampaglia, G. L. (2017). Finding Streams in Knowledge Graphs to Support Fact Checking. In *2017 IEEE International Conference on Data Mining (ICDM)* (pp. 859–864). doi: 10.1109/ICDM.2017.105

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2020). FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media. *Big Data*, *8*, 171–188. doi: 10.1089/big.2020.0062

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake News Detection on Social Media: A Data Mining Perspective. *ACM SIGKDD Explorations Newsletter*, *19*(1), 22–36. doi: 10.1145/3137597.3137600

Sierra, G., Soto, J., & Díaz, A. (2018). GECO, un Gestor de Corpus colaborativo basado en web. *Linguamática*, *9*, 57–72. doi: 10.21814/lm.9.2.256

Silverman, C., Strapagiel, L., Shaban, H., Hall, E., & Singer-Vine, J. (2016). *Hyperpartisan Facebook Pages Are Publishing False And Misleading Information At An Alarming Rate.* https://www.buzzfeednews.com/article/craigsilverman/partisan-fb-pages-analysis. BuzzFeedNews.

Simko, J., Racsko, P., Tomlein, M., Hanakova, M., Moro, R., & Bielikova, M. (2021). A study of fake news reading and annotating in social media context. *New Review of Hypermedia and Multimedia*, *27*(1-2), 97–127. doi: 10.1080/13614568.2021.1889691

Smith, K. P., & Christakis, N. A. (2008). Social networks and health. *Annual review of sociology*, *34*(1), 405–429.

Stella, M., Ferrara, E., & Domenico, M. D. (2018). Bots increase exposure to negative and inflammatory content in online social systems. *Proceedings of the National Academy of Sciences*, *115*(49), 12435–12440. doi: 10.1073/pnas.1803470115

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (Second ed., Vol. 3). MIT Press.

Tacchini, E., Ballarin, G., Della Vedova, M. L., Moret, S., & de Alfaro, L. (2017). *Some like it hoax: Automated fake news detection in social networks.* arXiv. doi: 10.48550/ARXIV.1704.07506

Tan, P. N., Steinbach, M., Karpatne, A., & Kumar, V. (2019). *Introduction to data mining* (Second ed.). Pearson.

Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 809–819). New Orleans, Louisiana: Association for Computational Linguistics. doi: 10.18653/v1/N18-1074

Tida, V. S., Hsu, D. S., & Hei, D. X. (2022). *Unified Fake News Detection using Transfer Learning of Bidirectional Encoder Representation from Transformers model.* arXiv. doi: 10.48550/ARXIV.2202.01907

Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., & Ortega-Garcia, J. (2020). Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion*, *64*. doi: 10.1016/j.inffus.2020.06.014

Umer, M., Imtiaz, Z., Ullah, S., Mehmood, A., Choi, G. S., & On, B.-W. (2020). Fake news stance detection using deep learning architecture (cnn-lstm). *IEEE Access*, *8*, 156695–156706. doi: 10.1109/ACCESS.2020.3019735

Van der Hulst, R. C. (2009). Introduction to Social Network Analysis (SNA) as an investigative tool. *Trends in Organized Crime*, *12*(2), 101–121.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)* (Vol. 30).

Vogel, I., & Meghana, M. (2020). Fake news spreader detection on twitter using character n-grams. *In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (eds.) CLEF 2020 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2020)*, *abs/2009.13859*.

Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. *Science*, *359*(6380), 1146–1151. doi: 10.1126/science.aap9559

Wang, T., Liu, H., He, J., & Du, X. (2013). Mining user interests from information sharing behaviors in social media. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 85–98).

Wang, W. Y. (2017). "Liar, Liar Pants on Fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th annual meeting of the association for computational linguistics* (Vol. 2, pp. 422–426). Vancouver, Canada: Association for Computational Linguistics. doi: 10.18653/v1/P17-2067

Wang, Y., Ma, F., Jin, Z., Yuan, Y., Xun, G., Jha, K., ... Gao, J. (2018). EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery and data mining* (pp. 849–857). Association for Computing Machinery. doi: 10.1145/3219819.3219903

Wardle, C., & Derakhshan, H. (2017). *Information disorder: Toward an interdisciplinary framework for research and policy making* (Tech. Rep. No. DGI(2017)09). Council of Europe. Retrieved from https://edoc.coe.int/en/media/7495-information-disorder-toward-an-interdisciplinary-framework-for-research-and-policy-making.html

Wasserman, S., & Faust, K. (1994). *Social Network Analysis: Methods and Applications*. Cambridge university press. doi: 10.1017/CBO9780511815478

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2017). *Data Mining: Practical Machine Learning Tools and Techniques* (Fourth ed.). Elsevier. doi: 10.1016/c2009-0-19715-5

Zaman, T., Fox, E. B., & Bradlow, E. T. (2014). A Bayesian approach for predicting the popularity of tweets. *The Annals of Applied Statistics*, *8*(3). doi: 10.1214/14-AOAS741

Zeng, J., & Schäfer, M. S. (2021). Conceptualizing "Dark Platforms". Covid-19-Related Conspiracy Theories on 8kun and Gab. *Digital Journalism*, *9*(9), 1321–1343. doi: 10.1080/21670811.2021.1938165

Zhang, C., & Zhang, S. (2002). *Association Rule Mining: Models and Algorithms*. Springer.

Zhang, G., Giachanou, A., & Rosso, P. (2022). SceneFND: Multimodal fake news detection by modelling scene context information. *Journal of Information Science*, *0*(0). doi: 10.1177/01655515221087683

Zhang, L., Qiao, T., Xu, M., Zheng, N., & Xie, S. (2022). Unsupervised learning-based framework for deepfake video detection. *IEEE Transactions on Multimedia*, 1–15. doi: 10.1109/TMM.2022.3182509

Zhou, X., & Zafarani, R. (2020). A survey of fake news: fundamental theories, detection methods, and opportunities. *ACM Computing Surveys*, *53*(5), 1–40. doi: 10.1145/3395046

Zhu, Q., & Luo, J. (2022). Generative Pre-Trained Transformer for Design Concept Generation: An Exploration. *Proceedings of the Design Society*, *2*, 1825–1834. doi: 10.1017/pds.2022.185

Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *3*, 1–130. doi: 10.2200/S00196ED1V01Y200906AIM006

Zubiaga, A., Liakata, M., Procter, R., Hoi, G. W. S., & Tolmie, P. (2016). Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PLOS ONE*, *11*(3), e0150989. doi: 10.1371/journal.pone.0150989

IBERIFIER
Iberian Media Research & Fact-Checking

Consortium

Disclaimer

All information provided reflects the status of the IBERIFIER project at the time of writing and may be subject to change.

Neither the IBERIFIER Consortium as a whole, nor any single party within the IBERIFIER Consortium warrant that the information contained in this document is capable of use, nor that the use of such information is free from risk. Neither the IBERIFIER Consortium as a whole, nor any single party within the IBERIFIER Consortium accepts any liability for loss or damage suffered by any person using the information.

This document does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of its content.

Copyright Notice

Website: iberifier.eu
Twitter: @iberifier

This project has received funding from the European Commission under the call CEF-TC-2020-2 (European Digital Media Observatory) Reference: **2020-EU-IA-0252**