DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis

Explainability of Fake News Detection Models for Social Media

Batuhan Erdogdu

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis

Explainability of Fake News Detection Models for Social Media

Erklärbarkeit von Modellen zur Fake-News-Erkennung in Sozialen Medien

Author:	Batuhan Erdogdu
Supervisor:	Prof. Dr. Georg Groh
Advisor:	M.Sc. Carolin Schuster
Submission Date:	Submission date

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, Submission date Batuhan Erdogdu

Acknowledgments

Abstract

Fake news is becoming a more significant issue as the news source becomes social media. Social media is capable of disseminating information at a rapid pace, which leads to widespread fake news in a matter of hours. The consequences of sharing wrong information can be destructive at many levels. This is why fake news, rumor, deception, and other aspects of false information were studied to be automatically detected. Many studies focused on different aspects, creating a variety of perspectives for automated detection. However, this is not an easy and straightforward task. There exist many challenges considering fake news, from human psychology to language modeling. Humans are not good lie detectors and can easily be fooled by misdirections, untrustworthy sources, or social status. Therefore, we need objective systems that can tell the difference.

Nevertheless, languages have many features which need to be utilized efficiently by automated detection systems to distinguish real news from fake news. As we will discuss, even though news content is important, social context also proves to be essential in detecting fake news.

This thesis investigates two types of fake news detection models: a news content-based model and a model that utilizes both news content and social context called a mixed model or mixed approach. These models were developed in other studies. We reproduce their work using their datasets and attempt to explain their models. We investigate the explainability of two kinds of neural networks, an inductive graph neural network and a Transformer model. We explain the Transformer model with the partition explainer from the SHAP framework. The mixed model is explained by GNNExplainer, a recent and only study that attempts to explain the graph neural networks, and provide the library for it. We report our findings as well as improvement suggestions for GNNExplainer.

We suggest the adoption of input sensitivity analysis along with explanation techniques to discover unintended weaknesses of the model. We also draw attention to the importance of interpretable and understandable explanations, and suggest better visualization techniques where necessary.

Contents

1 Introduction

With the rapid development of communication technologies, social media has become one of the most frequently used news sources. It is easier, faster, and offers interaction with people. Walker and Matsa (2021) report that 48% of adults in the U.S. "often" or "sometimes" get their news from social media. Furthermore, global data presented by Watson (2022) shows that the rate of adults who use social media as a news source is under 40% in The United Kingdom, The Netherlands, Germany, and Japan while this rate is over 70% in Kenya, Malaysia, Phillippines, Bulgaria, and Greece.
While ensuring convenience, interactivity, and speed, social media can also be used as a ground to spread any kind of information as there are no regulatory authorities checking the posts. As a result, floods of false and misleading information can be observed on social media (Allcott & Gentzkow, 2017).
Up to now, the research community has introduced numerous approaches to counteract the uncontrolled dissemination of fake news. For instance, some studies focus on building and analyzing datasets (Dou et al., 2021; Nakamura et al., 2020; Santia & Williams, 2018; Shu et al., 2017; Tacchini et al., 2017; Wang, 2017), and some studies leveraged the power of *Machine Learning* (ML) to automatically detect fake news by learning features from data. Due to the number of posts and the limitation of staff to check the posts, ML-based techniques can reduce manual labor when used with human supervision to counter the spreading of fake news. Since the task of detecting fake news is not trivial, the adopted ML methods are intrinsically complex. ML-based techniques with high complexity, such as *Deep Neural Networks* (DNNs), are harder to understand and interpret since they act like black-boxes (Castelvecchi, 2016).
The integration of ML-based methods into society gains more impact every day. While incredibly helpful in some aspects, ML-based techniques do not offer a reason for a particular prediction. Furthermore, we can not simply accept classification accuracy as a metric to evaluate real-world problems (Doshi-Velez & Kim, 2017). Therefore, integrating ML-based methods into society makes interpretability a requirement to increase social acceptance (Molnar, 2022).
Consequently, a new research field called *eXplainable Artificial Intelligence* (XAI) has surfaced to fill this missing link between humans and *Artificial Intelligence* (AI). XAI proposes creating a set of ML techniques that deliver more explainable models while preserving learning performance and helping humans to understand, properly trust,

and effectively handle the emerging generation of artificially intelligent partners (Gunning & Aha, 2019). While incorporating XAI increases social acceptance, it also aims to create more privacy-aware (Edwards & Veale, 2017), fairer, and trustworthy systems (Z. C. Lipton, 2016).
Like all ML techniques, *Fake News Detection* (FND) models need interpretability, particularly when implementing countermeasures for fake news. However, the interpretability of a model is not often considered despite the large amount of research produced in the last decade. Incorporating social context (Shu et al., 2018), representing the propagation networks as graphs (Dou et al., 2021), and using *Graph Neural Networks* (GNNs) to produce *State Of The Art* (SOTA) models (Monti et al., 2019) have increased the complexity, but also the performance of FND models. For instance, using social context data alone has proved more effective than utilizing textual data alone in recent studies (Dou et al., 2021). However, it is not clear which social features impact the decision process of these models.
This thesis focuses on the explainability of FND models using tools from the XAI suite. Specifically, we focus on news content-based models and mixed approaches (news content and social context) to elaborate on their interpretability. Thus, we define three research objectives:

RO1 Determine the tools for explaining FND models.

RO2 Show that explanations of FND models play an essential role in understanding the shortcomings of the FND models.

RO3 Investigate the understandability of explanations and suggest improvements where necessary.

In Chapter 2, we elaborate on fake news, FND methods, and XAI. We give foundations of fake news and define its characteristics. We then categorize FND models and give important examples from the literature. After examining fake news and its detection methods, we focus on the characterization of XAI and give definitions that will be used throughout this thesis.
In Chapter 3, we introduce the background for the news content model we used, examining how it is constructed. With this information, we then discuss the news content dataset which is used in the training of our news content model. After sharing our insights and statistics from the dataset, we move on to reporting the model performance. In this chapter, we also explain the reasons behind our model's performance by conducting several experiments using the explanation tool provided in 3.2.1. We show that our model can indeed be improved with the help of explanation techniques.
In Chapter 4, we introduce a different type of neural network, namely GNNs. We summarize how GNNs are constructed, what kind of GNNs exist for FND, and how

to work with them. For FND, We use a mixed approach that utilizes news content and social context data. We give our insights from the dataset and report our model's performance on the dataset. Lastly, we make use of a recently introduced explanation technique for GNNs to understand our model's behavior.
In Chapter 5, we talk about our findings from Chapter 3 and Chapter 4. We show that research objectives are attained. Additionally, we discuss the limitations that we have encountered and possible future works.

2 Background and Related Work

We explain two research fields that create the bedrock of this thesis, namely, fake news detection and explainable artificial intelligence. Both areas provide the foundation of tools used in this work. The first provides the mechanisms and approaches to detect fake news, and the second offers a suite of techniques to interpret these mechanisms and strategies.
Initially, in 2.1, we discuss societal challenges, the characteristics, and the history of fake news. Then we talk about the detection methods that were developed over the years. After showing the challenges of creating FND models, we conclude the first section with SOTA FND models.
After fake news detection, in 2.2, we first examine when XAI is necessary and its importance. Then, we define the suite of XAI and the goals of it, and finally, we determine the suite that aims to satisfy these goals.

2.1 Fake News Detection

In the past decade, social media has become a place where anyone can share information. Although fast, free, and easy to access, obtaining real news from social media can be difficult, and one should do so at their own risk and always check the facts (Allcott & Gentzkow, 2017; Lazer et al., 2018). Nevertheless, the news stream never ends; thus, the need to verify the credibility of news using automated systems arises. To address this necessity, the number of studies involving *Fake News* or *Fake News Detection* has dramatically increased in the last decade (Fig. 2.1).
In 2.1.1, we briefly present the history of fake news and look at studies that display the impact of fake news on society. In this section, we also define the terms fake news, disinformation, and misinformation.
In 2.1.2, we explore social sciences and human psychology, delivering insights into why humans fall for or tend to believe fake news. Furthermore, we draw some insights from the social, technical, and data-oriented foundations of fake news.
We then list the available datasets used in FND and deliberate their advantages and disadvantages in 2.1.3. Finally, in 2.1.3, we summarize the evolution of detection algorithms, then we classify FND algorithms according their input data type and focus.

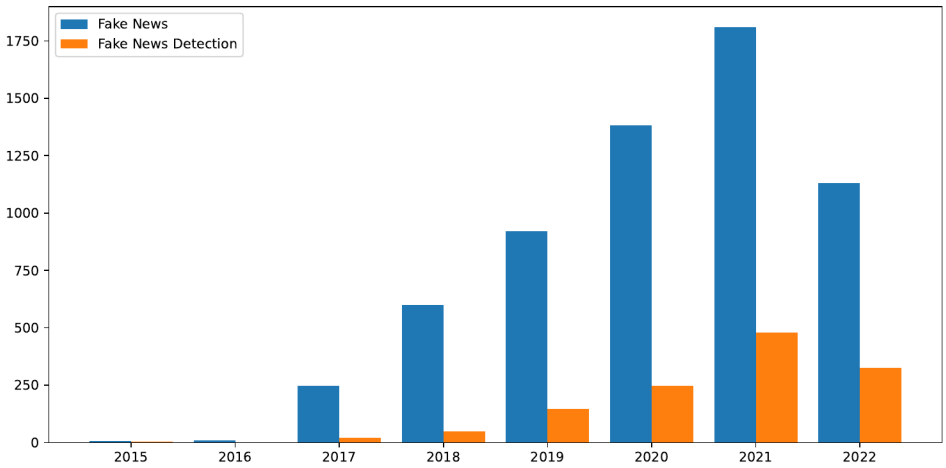


Figure 2.1: Total number of publications that include (1) *Fake News* (blue) and (2) *Fake News Detection* (orange) publications by year. Source: Scopus; Search Arguments: (1) TITLE-ABS-KEY("fake news*") PUBYEAR AFT 2014 (2) TITLE-ABS-KEY("fake news detection")

2.1.1 Fake News

Throughout history, various forms of widespread misinformation have been recorded. In the thirteenth century BC, Rameses the Great decorated his temples with paintings that tell stories of victory in the Battle of Kadesh. However, the treaty between the two sides reveals that the battle's outcome was a stalemate (Weir, 2009). Later in history, just after the printing press was invented in 1439, the circulation of fake news began. One of history's most famous examples of fake news is the "Great Moon Hoax" (Foster, 2016). In 1835, The Sun newspaper of New York published articles about a real-life astronomer and a made-up colleague who had observed life on the moon. It turns out that these fictionalized articles brought them new customers and almost no backlash after the newspaper admitted that the articles mentioned earlier were a hoax[1].
In order to highlight the difference, using the definitions from (Pennycook & Rand, 2021), we formally introduce the terms disinformation and misinformation as follows.

Definition 2.1.1 (*Disinformation*)**.** "*Information that is false or inaccurate and that was created with a deliberate intention to mislead people.*" (Pennycook & Rand, 2021)

Definition 2.1.2 (*Misinformation*)**.** "*Information that is false, inaccurate, or misleading.*

[1]https://www.politico.com/magazine/story/2016/12/fake-news-history-long-violent-214535/

Unlike disinformation, misinformation does not necessarily need to be created deliberately to mislead." (Pennycook & Rand, 2021)

There is no fixed definition for fake news. Thus, we elaborate on the definitions of fake news. A limited definition is news articles that are intentionally or verifiably false (Allcott & Gentzkow, 2017). This definition stresses authenticity and intent. The inclusion of false information that can be confirmed refers to authenticity. On the other hand, intent refers to the deceitful intention to delude news consumers (Shu et al., 2017). This definition is widely used in other studies (Conroy et al., 2015; Mustafaraj & Metaxas, 2017; Shu et al., 2017).
Furthermore, recent social sciences studies (Lazer et al., 2018; Pennycook & Rand, 2021) define fake news as fabricated information that mimics news media content in form but not in organizational process or intent. Similarly, this definition covers authenticity and intent; additionally, it includes the organizational process. More general definitions for fake news consider satire news as fake news due to the inclusion of false information even though satire news aim to entertain and inherently reveals its deception to the consumer (Balmas, 2014; Brewer et al., 2013; Jin et al., 2016; V. Rubin et al., 2016). Further definitions include hoaxes, satires, and blatant fabrications (V. L. Rubin et al., 2015) In this thesis, we are not interested in the organizational process and do not consider conspiracy theories (Sunstein & Vermeule, 2009), superstitions (Lindeman & Aarnio, 2007), rumors (Berinsky, 2017), satire, or hoaxes. Therefore, we use the limited definition from (Allcott & Gentzkow, 2017) and formally introduce it.

Definition 2.1.3 (*Fake News*)**.** *"News articles that are intentionally or verifiably false.*" (Allcott & Gentzkow, 2017)

Fake news can lead to disastrous situations, such as crashes in stock markets, resulting in millions of dollars of loss. For example, Dow Jones industrial average went down like a bullet (see Fig. 2.2) after a tweet about an explosion injuring President Obama went out due to a hack (ElBoghdady, 2013).
The detrimental impacts of fake news further extend to societal issues. During the campaigns for the U.S: presidential elections in 2016, fake news piece became very believable and widespread. This led to a disastrous event in which a man, convinced by what he read on social media about a pizzeria trafficking humans, went on a shooting spree in that pizzeria. Later named Pizzagate (Fisher et al., 2016), this incident illustrates the deadly impact of fake news. In fact, fake news can even affect presidential elections (Allcott & Gentzkow, 2017; Read, 2016).
Recent history exhibits that some fake news spreads like wildfires through social media. Evidence shows that the most popular fake news stories were more widely shared than the most popular mainstream news stories (Silverman, 2016).

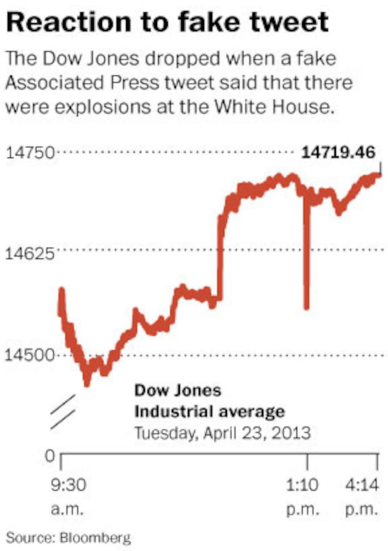


Figure 2.2: The market's reaction to the fake tweet. The sharp decline caused by a single tweet. Image obtained from (ElBoghdady, 2013)

Digital News Report 2022 (N. Newman et al., 2022) shares in its key findings that trust in the news is 42% globally, the highest (69%) in Finland, and the lowest (26%) in the U.S.A. Additionally, the same study shows that in early 2022, in the week of the survey, between 45% and 55% of the surveyed social media consumers worldwide witnessed false or misleading information about COVID-19. The same study also reports the appearance of fake news in politics was between 34% and 51%, and between 9% and 48% for fake news about celebrities, global warming, and immigration (Watson, 2022).

2.1.2 Foundations of Fake News

The environment for fake news has been the traditional news media for a long time. First started with newsprint, then continued with radio and television, and now with social media and the web, the dissemination of fake news reached its peak. Next, we discuss the psychological and social foundations of fake news to stress the importance of human psychology, especially when accepting fake news as genuine and sharing it with others. Then we focus on the technical foundations where we discuss how social media and technology have accelerated the diffusion of fake news.

Psychological Foundations. Understanding the difference between real and fake news is not an easy task for a human. Two psychological theories, namely, *naive realism* and *confirmation bias*, examine why humans fall for fake news. The first refers to a person's

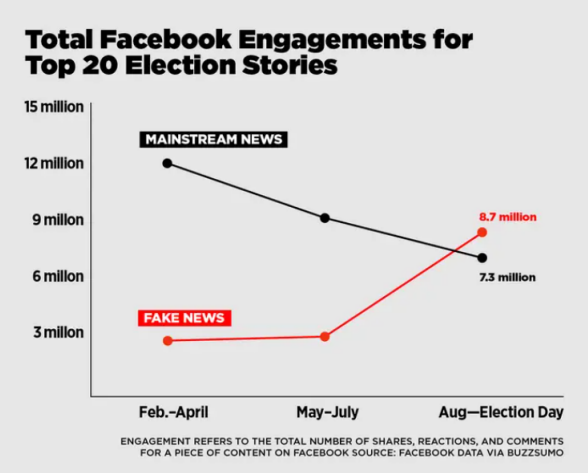


Figure 2.3: The rising engagement for fake news stories observed after May-July, just before Presidential Elections. Image obtained from (Silverman, 2016)

disposition to believe that their point of view is the mere accurate one, while people who believe otherwise are uninformed or biased (Reed et al., 2013). The second, often called selective exposure, is the proclivity to prefer information that confirms existing views (Nickerson, 1998).
Another reason for human fallacy in fake news is that once a misperception is formed, it becomes difficult to correct. In fact, it turns out that correcting people leads them to believe false information more, especially when given factual information that refutes their beliefs (Nyhan & Reifler, 2010).
Social Foundations. The prospect theory explains the human decision-making process as a mechanism based on maximizing relative gains and minimizing losses with respect to the current state (Kahneman & Tversky, 1979; Tversky & Kahneman, 1992). This inherent inclination to get the highest reward also applies to social cases in which a person will seek social networks that provide them with social acceptance. Consequently, people with different views tend to form separate groups, which makes them feel safer, leading to the consumption and dissemination of information that agrees with their opinions. These behaviors are explained by social identity theory (Ashforth & Mael, 1989) and normative social influence (Asch & Guetzkow, 1951).
Two psychological factors play a crucial role here (Paul & Matthews, 2016). The first, social credibility, is explained by a person's tendency to recognize a source as credible when that source is deemed credible by other people. The second, called the frequency heuristic, is the acceptance of a news piece by repetitively being exposed to it. Collectively, these psychological phenomena are closely related to the well-known filter

bubble (Pariser, 2011), also called echo chamber, which is the formation of homogenous bubbles in which the users are people of similar ideologies and share similar ideas. Being isolated from different views, these users usually are inclined to have highly polarized opinions (Sunstein, 2001). As a result, the main reason for misinformation dispersal turned out to be the echo chambers (Vicario et al., 2016).

Technical Foundations. Social media's easy-to-use and connected nature give rise to more people selecting or even creating their own news source. Naturally, this gives way to more junk information echoing in a group of people on social media. As algorithms evolve to understand user preferences, social media platforms recommend similar people or groups to those in echo chambers. A recent study (Cinus et al., 2022) shows that these recommenders can strengthen these echo chambers. They discuss that some of these recommenders contribute to the polarization on social media. In other words, people can convince themselves that any fake news is real by staying in their echo chambers. One main reason that some fake news spreads so rapidly on social media is the existence of malicious accounts. The account user can be an actual human or a social bot since creating accounts on social media is no cost and almost no effort. While many social bots provide valuable services, some were designed to harm, mislead, exploit, and manipulate social media discourse. Formally, a social bot is a social media account governed by an algorithm to fabricate content and interact with other users (Ferrara et al., 2016). A more recent study from the same author shows that malicious social bots were heavily used in the 2016 U.S. Presidential Elections (Bessi & Ferrara, 2016). On the other hand, malicious accounts that are not bots, such as online trolls who aim to trigger negative emotions and humans that provoke people on social media to get an emotional response, contribute to the proliferation of fake news (Cheng et al., 2017). Building upon three foundations, we draw some results for fake news to be considered when building a fake news detection model:

1. *Invasive*: Fake news can appear on anyone's feed if it spreads for a sufficient amount of time.

2. *Hard to discern*: Fake news is fabricated in such a way that it resembles the authenticity of a real news source. This indistinguishability leads to issues when working with news-content-based FND models.

3. *The source is crucial*: The credibility of a news source is essential. We can use news from credible sources to teach the model to distinguish genuine from fabricated.

4. *Fake news has hot spots*: The echo chambers are invaluable examples when trying to understand the behaviors of fake news. We can leverage this attribute and use social models, such as graphs, to successfully detect fake news.

5. *Early detection is essential*: As discussed in psychological foundations, the volume of exposure to a piece of fake news can significantly affect one's opinions, thus leading to more misinformed individuals.

Data-Oriented Foundations. We define features for news content and social context to represent the news pieces in a structured manner. First, we introduce attributes for news content (Shu et al., 2017):

- *Source*: Publisher of the news piece.
- *Headline*: Short title text that aims to catch the readers' attention and describes the article's main topic.
- *Body Text*: The main text piece that details the news story.
- *Image/Video*: Part of the body content supplies visual input to articulate the story.

Using these attributes, we extract two types of features for news content:

Linguistic-based features: The news content is heavily based on textual content. Thus, the first feature that belongs to this class is lexical features which make use of character and word level frequency information that can be obtained by the utilization of *term frequency-inverse term frequency* (TF-IDF) (Jones, 1972; Luhn, 1957) or bag-of-words (BoW). The second feature is based on syntactic features, which include sentence-level features that can be obtained via *n-grams* and punctuation and *parts-of-speech* (POS) (Daelemans, 2010) tagging. We can extend these features to domain-specific ones, such as external links and the number of graphs (Potthast et al., 2017).

Visual-based features: Particularly for fake news, the visual content is a vital tool for establishing belief (Dan et al., 2021). Hence, the features that reside in images and videos become significant. Fake images and videos that bring the fake story together are commonly used(e.g., Harding, 2012; Sawer, 2020). A recent study (Qi et al., 2019) examines visual and statistical information for fake news detection to counteract the effects of misleading visual input. Visual features consist of clarity score, similarity distribution histogram, diversity score, and clustering score. Statistical features are listed as count, image ratio, multi-image ratio, etc. (Shu et al., 2017).

Now, we define features for social context, which has recently drawn much attention from the research community (Shu et al., 2020; Shu et al., 2019). Overall, we will concern with three aspects of social context data: user-based, post-based, and network-based features.

User-based: As mentioned in the Technical Foundations part of this subsection, fake news has various ways of disseminating, such as via echo chambers, malicious accounts, or bots. Therefore, analyzing user-based information can prove useful. We distinguish user-based features at the group and individual levels (Shu et al., 2017). Individual levels are extracted to deduce the credibility of each user by utilizing, for example, the number of followers and followees, the number of tweets authored by a user, or similar features (Castillo et al., 2011). On the other hand, group-level user-based features are the general characteristics of groups of users related to the news (Yang et al., 2012). Parallel to the social identity theory and normative social influence idea, the assumption is that the real and fake news consumers tend to form different groups, which may lead to unique characteristics. Typical group-level features stem from individual-level features by obtaining the share of verified users and the average number of followers and followees (Ma et al., 2015).

Post-based: Analysis of reactions by users can prove helpful when determining whether a news piece is real or not. For example, if a news piece is getting doubtful comments, this can help determine the news piece's credibility. As such, post-based features are based on inferring the integrity of a news piece from three levels. Namely, post-level, group-level, and temporal-level (Shu et al., 2017). Post-level features can be embedding values for each post or take forms mentioned in linguistic-based features. For post-level features, we can also consider general approaches such as topic extraction (e.g., using latent Dirichlet allocation (LDA) (Blei et al., 2003)), stance extraction, which provides information about users' opinions (e.g., supports, opposes (Jin et al., 2016)), and finally credibility extraction, which deals with estimating the degree of trust for each post (Castillo et al., 2011). Group-level post-based features collect feature values for all relevant posts and apply an operation to extract pooled information. When determining the credibility of news, group-level features proved to be helpful (Jin et al., 2016). Temporal-level features deal with changes in post-level features over time. Typically, unsupervised learning methods, such as Recurrent Neural Networks (RNN) are employed to capture the changes over time (Ma et al., 2016).

Network-based: As discussed in the Technical Foundations part, fake news is likely to give rise to echo chambers, which leads to the idea of a network-based approach. When represented as networks, the propagation behavior of fake news can be analyzed further, and patterns can be discovered (Shu et al., 2017). In literature, various types of networks exist. The most common ones are stance networks, occurrence networks, and friendship networks. Stance networks are constructed upon stance detections which is a part of sentiment analysis and deal with

determining a user's viewpoint using text and social data (Du et al., 2017). Using all users' stances, a network is built in which the nodes are the tweets relevant to the news piece and the edges represent the similarity of stances between nodes (Jin et al., 2016; Tacchini et al., 2017). On the other hand, occurrence networks leverage the frequency of mentions or replies about the same news piece (Kwon et al., 2013). Friendship networks are based on the follower/followee relationship of users who share posts connected to the news piece. Derived from friendship networks, in the form of one of the datasets we use in our experiments (Dou et al., 2021), diffusion networks are designed to track the course of the dissemination of news (Kwon et al., 2013). Briefly, a diffusion network consists of nodes characterizing users and diffusion paths representing user relationships and interactions. In detail, a diffusion path between two users u_i and u_j exists if and only if u_j follows u_i, and u_j shares a post about a news piece that u_i has already shared a post about (Shu et al., 2017). It has been shown that characterizing these networks is possible (Kwon et al., 2013). Approaches for these networks have gained traction recently, especially with some SOTA GNNs, e.g., (Monti et al., 2019).

To conclude this subsection, we have covered psychological, social, technical, and data-oriented foundations in this section. We established that, from different aspects, there are various reasons for the dissemination of fake news. Accordingly, we consider these reasons when building FND systems. In the next section, we discuss FND approaches and how they have evolved. Moreover, we characterize FND models and talk about each type of approach.

2.1.3 Characterization of Fake News Detection

Fake news detection is as old as fake news itself. Before social media became a hub for news consumers, fact-checkers, i.e., fake news detectors, were only journalists and educated people. Following the source shift of the news from printed paper to online, then social media, the detection of fabricated news has become costly, cumbersome, and not as rewarding due to the endless stream of information and decreasing trust in journalism. Automatic detection for news thus became a necessity in our world (Chen et al., 2015).

Similar to what we did in the Data-Oriented Foundations part of the previous subsection, we classify fake news detection models as *news content models* and *social context models* (see Fig. 2.4) and start with news content models by following the classification principles in (Shu et al., 2017).

News Content Models. Based on news content and fact-checking methodologies, these models are the starting point of fake news detection. News content models are

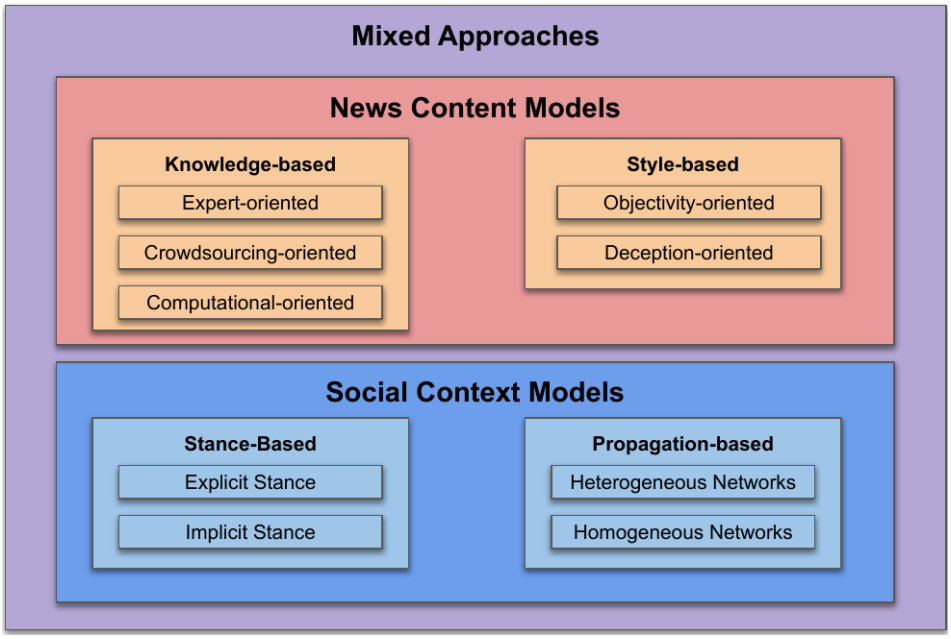


Figure 2.4: Characterization of Fake News Detection Models, Figure inspired by Figure 1 in Shu et al. (2017).

classified as knowledge-based and style-based. We first introduce style-based models as they are the initial approaches for FND.

Style-based: Previous research in psychology has mainly focused on style-based approaches to detect *manipulators* in the text. Particularly, deception detection techniques were popular and commonly developed in early works in criminology and psychology. We describe two different ways to approach style-based news content models, namely, *Deception-oriented* and *Objectivity-oriented* (Shu et al., 2017).

- *Deception-oriented*: The initial approaches for automated fake news detection focus on news context and stem from deception detection in language. The first study that focuses on deception detection in the language (Undeutsch, 1954) hypothesized that the truthfulness of the statement is more important than the integrity of the reporting person, and there exist definable and descriptive criteria that form a crucial mechanism for determining the truthfulness of statements. Even though this study is from experimental psychology, it stresses the feasibility of defining a set of rules that determine the truthfulness of a statement.

An early study from criminology, Scientific Content Analysis (SCAN) (Sapir, 1987), analyzes freely written statements. In this process, SCAN claims to detect potential instances of deception in the text but cannot label a statement as a lie or truth. The subsequent study for SCAN (Smith, 2001) is the first known study that correlates linguistic features with deceptive behavior using high-stakes data. Similar to SCAN, the subsequent studies (Adams, 2002; M. L. Newman et al., 2003) that link linguistic features to deception classify the owner of the statement as a truth-teller or liar according to the frequency of deception indicators in the statement.

Although it is more challenging to define a methodology for automated deception detection (DePaulo et al., 1997), early studies have shown that this task is achievable. L. Zhou et al. (2004) create a structured approach using linguistic-based cues and draw attention to further studies for automating deception detection. In this study, the authors extend linguistic-based cues with complexity, expressivity, informality, and content diversity. Instead of using humans as cue identifiers, the authors use *Natural Language Processing* (NLP) techniques, namely an NLP tool called iSkim (L. Zhou et al., 2002), to extract cues automatically. Another study also focuses on linguistic cue analysis. Employing the C4.5 algorithm (Salzberg, 1994) on a small dataset, the authors reach 60.72% accuracy using 15-fold cross-validation.

Similarly, in Bachenko et al. (2008), the authors developed a system for automatically identifying 275 truthful or deceitful statements with the use of verbal cues using the Classification and Regression Tree (CART) (Breiman et al., 1984). Additionally, the studies (Hancock et al., 2007; V. L. Rubin, 2010) make use of a relatively small dataset and analyze linguistic-based cues. Rubin's series of studies (V. Rubin et al., 2015; V. L. Rubin, 2010; V. L. Rubin & Lukoianova, 2015; V. L. Rubin & Vashchilko, 2012) adopt Rhetorical Structure Theory (RST) and Vector Space Modeling (VSM). The first captures the coherence of a story using functional relations among meaningful text units and delivers a hierarchical structure for each news story (Mann & Thompson, 1988). The second is a way to represent rhetorical relations in high-dimensional space. The authors utilized logistic regression as their classifier and reached 63% accuracy.

Furthermore, Afroz et al. (2012) investigate stylistic deception and use lexical, syntactic, and content-specific features. Lexical features include both character- and word-based features. Syntactic features represent sentence-level style and include frequency of function words from LIWC (Pennebaker et al., 2007), punctuation, and POS tagging in which a text is assigned its morphosyntactic category (Daelemans, 2010). Lastly, content-specific features

are keywords for a specific topic. For classification, the authors leveraged Support Vector Machines (SVM) (Hearst et al., 1998). More comprehensive and modern approaches, such as Wang (2017) also leveraged the power of *Convolutional Neural Networks* (CNNs) to determine the veracity of the news.

- *Objectivity-oriented*: Objectivity-oriented news content models aim to detect indicators of objectivity decrease in the news content (Shu et al., 2017). These indicators are observed in the news content from misleading sources, such as hyperpartisan sources, which display highly polarized opinions in favor of or against a particular political party. Consequently, this polarized behavior motivates the fabrication of news that supports the sources' political views or undermines the opposing political party. *Hyperpartisan news* is a subtle form of fake news. It is defined as misleading coverage of events that did actually occur with a strong partisan bias (Pennycook & Rand, 2019). Since the spread of hyperpartisan news can be detrimental, many approaches to detect hyperpartisanship in news articles have been developed. For instance, in Potthast et al. (2017), the authors take a stylometric methodology that adopts 10 readability scores, and dictionary features based on the frequency of words. A competition for detecting hyperpartisan news (Kiesel et al., 2019) hosted several teams with various ideas including the utilization of n-grams, word embeddings, stylometry, sentiment analysis, and many more approaches. The most popular method was the usage of embeddings, particularly the models that leveraged BERT (Devlin et al., 2018).
  Also used for the dissemination of hyperpartisan news (Kiesel et al., 2019), another form of fake news that is evaluated under this focus is *yellow-journalism*. It utilizes clickbaits such as catchy headlines and images that invoke strong emotions, and aims to generate revenue (Agrawal, 2016; Palau-Sampio, 2016). Studies that aim to detect click baits mainly focus on headlines. For example, Rony et al. (2017) construct a DNN in which they use distributed subword embeddings (Bojanowski et al., 2016; Joulin et al., 2016) as features with an extension of the skip-gram model (Mikolov, Sutskever, et al., 2013).

Knowledge-based: Being the most direct way of detecting fake news, these approaches make use of external fact-checkers to verify the claims in news content (Shu et al., 2017). Fact-checkers can be sophisticated algorithms, domain experts, or crowdsourcing methods that assess the truthfulness of a claim in a specific context (Vlachos & Riedel, 2014). With growing consideration for fake news detection, automated fact-checking has drawn much attention, and notable efforts have been made in this area (Barrón-Cedeño et al., 2020; Thorne & Vlachos, 2018). We categorize knowledge-based news content models as *expert-oriented*,

crowdsourcing-oriented, and *computational-oriented*.

- *Expert-oriented* approaches depend on domain experts who investigate the integrity of a news piece collecting relevant information and documents to decide the truthfulness of a claim [1]. Platforms like Politifact [2] and EUfactcheck [3] are examples of expert-oriented fact-checking for all news from various sources. These platforms label news in a range such that the label reflects the veracity of the news. A different approach for labeling is exercised by Snopes [4], which extends the same logic of Politifact by including different aspects of fact-checking, such as "scam", "miscaptioned", and "outdated" [5]. Recently replaced by an irrelevant magazine website, another instance was Gossipcop [6], which dealt with celebrity fact-checking and contributed to the creation of a fake news dataset (Shu et al., 2018). Even though expert-based fact-checking is reliable, with the increasing magnitude of news stream and speed of spread, it is not scalable to fact-check every news piece by hand; thus, manual validation alone becomes insufficient (Guo et al., 2022).
- *Crowdsourcing-oriented* fact-checking is powered by the wisdom of crowds (Galton, 1907), and is a collection of annotations that are afterward aggregated to obtain an overall result indicating the veracity of the news. Unlike professional fact-checkers, who are in short supply, this approach is scalable given that the crowd contains enough educated people (Allen et al., 2021). For instance, Twitter launched a program called Birdwatch [7], in which users can leave notes for tweets that they think contain misinformation. Furthermore, this tool allows users to rate each other's notes, leading to diverse perspectives [8]. Another example is from Facebook [9], which uses a third party of crowdsourced fact-checkers called International Fact-Checking Network [10] (IFCN).
- *Computational-oriented* methods, unlike expert-oriented systems, are scalable and designed to automatically predict whether a claim is truthful. These

[1] https://www.politifact.com/article/2018/feb/12/principles-truth-o-meter-politifacts-methodology-i/
[2] https://www.politifact.com/
[3] https://eufactcheck.eu/
[4] https://www.snopes.com/
[5] https://www.snopes.com/fact-check-ratings/
[6] https://web.archive.org/web/20190807002653/https://www.gossipcop.com/about/
[7] https://twitter.github.io/birdwatch/overview/
[8] https://twitter.github.io/birdwatch/diversity-of-perspectives/
[9] https://www.facebook.com/formedia/blog/third-party-fact-checking-how-it-works
[10] https://www.poynter.org/ifcn/

systems are dependent on external sources. The studies that focus on this type of approach mainly try to solve two issues: (i) *identifying check-worthy claims* and (ii) *estimating the integrity of claims* (Shu et al., 2017). The first issue requires the extraction of factual claims from news content or other related textual content. For example, Hassan et al. (2015) collect presidential debate transcripts, then label them into three classes with the help of crowdsourcing. The authors uncover interesting patterns in these transcripts using annotated data and supervised learning techniques. Another study that covers both issues uses Wikipedia information to generate factual claims, then check whether a given claim is truthful (Thorne et al., 2018). Compared to the first one, the second issue requires the utilization of structured external sources. *Open web* and structured *knowledge graphs* are the two most prominent tools when tackling this issue. Open web tools analyze features like mutual information statistics (Etzioni et al., 2005), frequency, and web-based statistics (Magdy & Wanas, 2010). On the other hand, knowledge graphs are interconnected. One noteworthy example is ontologies such as DBPedia (Auer et al., 2007). Using DBPedia, one can define semantic relations and rules to infer whether a claim is correct (Braşoveanu & Andonie, 2019).

Social Context Models. The interconnected design of social media can be leveraged by extracting user-based, post-based, and network-based features and utilizing these features to supplement news content models. Social context models exploit related user engagements for a news piece by capturing this external information from multiple angles. Two types of social context models are prominent: *stance-based* and *propagation-based* (Shu et al., 2017).

- *Stance-based* approaches estimate the user's stance toward a specific news topic. More formally, stance detection in social media deals with users' viewpoints toward particular topics by means of various aspects related to users' posts and characteristic traits (ALDayel & Magdy, 2021). The user's stance information can be extracted either implicitly or explicitly. Implicit stances can be automatically obtained from social media posts with the help of NLP tools such as sentiment analysis (Mohammad et al., 2016). Explicit stances are easier to obtain since they are direct expressions of opinions or emotions. For example, "like" on Twitter or Facebook, "upvote" and "downvote" ratings on Reddit, and "like" and "dislike" on Youtube are explicit stances of users. Some Like it Hoax, a study by Tacchini et al. (2017) utilize explicit stances with logistic regression and harmonic boolean label crowdsourcing for classification on a dataset they curated from Facebook. In the stance classification process, the authors consider the likes and the issuer of likes for each post. They state that logistic regression comes short in this task since

it cannot learn anything about posts without likes (Tacchini et al., 2017). An early example of implicit stance detection (Walker et al., 2012) leverages the dialogic relations between authors by constructing graphs that represent the interaction between authors. They show that this information can improve the performance of stance-detection models. A more detailed study (Addawood et al., 2017) investigates stance classification considering lexical, syntactic, twitter-specific, and argumentation feature types. Although some twitter-specific features can be considered explicit stances, such as whether the tweet is a retweet, or it contains the title of an article or a hashtag. Those features are later aggregated before it is fed to the classifier. The authors reach the highest F1 score using lexical and argumentation features. In literature, there are also implicit stance-based approaches that aim to detect the veracity of a news piece by exploring the relationship between a headline and the article (Ghanem et al., 2018; Hanselowski et al., 2018).
Another variation of stance-based detection is rumor detection. One example of a rumor detection model is a Bayes classifier that utilizes content-based, network-based, and twitter-specific meme features through *Information Retrieval* (IR) techniques (Qazvinian et al., 2011). In this study, the authors propose a general framework that leverages statistical models and maximizes a linear function of log-likelihood ratios to retrieve rumorous tweets. They show that the features they used contribute to their model's overall performance.

- *Propagation-based* procedures are inspired by the assumption that the veracity of a news event is highly correlated with the credibilities of related social media posts. These models employ the interrelations of related social media posts to classify a news piece as truthful or not (Shu et al., 2017). Propagation-based models can be based on either *homogeneous networks* or *heterogeneous networks*. Homogeneous networks are built upon a single type of entity, such as a post or event. A study by Jin et al. (2016) creates homogeneous credibility networks for each topic which is extracted using an unbalanced version of the Joint Topic Viewpoint Model (Trabelsi & Zaiane, 2014). These credibility networks consist of nodes as tweets and edges as links, defined by either supporting or opposing. On the other hand, heterogeneous networks can contain multiple types of entities, such as events, sub-events, posts, and comments. For example, Jin et al. (2014) builds a hierarchical propagation graph that contains events, sub-events, and messages. Using an iterative method, the authors provide a globally optimal solution for the graph optimization problem in the study.

Mixed Approaches. We examined two types of FND models, namely, news content

models and social context approaches. It is crucial to note that approaches are not necessarily purely news content or social context-based; they can be based on both. The models that use news content and social context are referred to as mixed approaches (Han et al., 2020). *User Preference-Aware Fake News Dataset* (UPFD) (Dou et al., 2021) is a dataset that houses two of these types. UPFD framework uses news content and social engagement information to construct a hierarchical tree. The node features are textual representations of various sources which will be discussed in 4.1.3. The best-performing model uses GraphSAGE (Hamilton et al., 2017) as graph encoder and BERT (Devlin et al., 2018) as the text encoder.
To summarize this section, we have introduced the history and definitions of fake news in subsection 2.1.1. Then, we investigated the foundations of fake news and gave motivations for developing automated FND systems in subsection 2.1.2. Following that, we examined the evolution and characterized FND models in section 2.1.3. We discussed examples for each type of model and briefly summarized their approaches. We also introduced one of the datasets (Dou et al., 2021) and models (Hamilton et al., 2017) used in this thesis; however, in-depth information will be provided in Chaper 4. In 2.2, we elaborate on the techniques available in explainable artificial intelligence. We discuss the qualities of a reasonable explanation and highlight the importance of the interpretability of a model. We give essential definitions that will be used throughout this thesis.

2.2 Explainable Artificial Intelligence

Understanding and interpreting a model's prediction is very important nowadays since this understanding allows us to validate the model's reasoning and extract rich information for a human expert, thus leading to to increased trust in the model (Ribeiro et al., 2016). In addition, explaining a model can help improve the model (Lundberg & Lee, 2017) and alleviate concerns raised by Ethical AI (Angwin et al., 2016; Goodman & Flaxman, 2017). In this section, we introduce the background for XAI techniques used in this thesis. First, in 2.2.1, we characterize XAI by following works from Z. C. Lipton (2016) and Barredo Arrieta et al. (2020) and give definitions to clarify the taxonomy. Then, in 2.2.2, we discuss the properties of good explanations, the goals of XAI, and the evaluation techniques for explanation ethods. Finally, in 2.2.3 we briefly lay out the most frequently mentioned explanation methods in the literature, along with the ones we use in this thesis. We summarize each and cover explanation techniques offered to any neural network.

2.2.1 Foundations of Explainable Artificial Intelligence

Initial AI methods were not sophisticated enough to require additional explanation schemes. In the last years, DNNs have been adopted in various sectors. Although empirically successful thanks to enormous parameter spaces and numerous layers, DNNs are complex *black-box* models in terms of interpretability (Castelvecchi, 2016). In the XAI context, *black-box* or *opaque* models are considered to be the opposite of *transparent* because they require further investigation to understand their inner workings (Z. C. Lipton, 2016). Humans hesitate to use systems that not directly interpretable and reliable, making *interpretability* essential (J. Zhu et al., 2018). Moreover, from a legal perspective, the notion *right to explanation* brings more attention to interpretability (Z. C. Lipton, 2016). Particularly in situations such as when:

- The prediction of AI directly affects human life, e.g., fully autonomous cars in traffic and medical AI assistants.
- The reasons behind an AI system's decision can not be clearly determined.

With the additional demand from the Ethical AI field (Goodman & Flaxman, 2017), the research community has put in a tremendous amount of effort to gap the bridge between a black-box model and its interpretability. However, the lack of consensus on taxonomy has led to synonymous usages of interpretability. The early definitions for interpretability were too broad, describing it as an additional design driver when building a model (Kim, Rudin, et al., 2015) or a requirement for *trust* (Kim, 2015). Nevertheless, can trust be defined objectively? Is the accuracy or F1 score of a model enough to trust the model? To answer the first question, Z. C. Lipton (2016) argues that trust is subjective and can not be technically defined. To answer the second question, taking only the performance metrics as a baseline for trust in the model is shown to be an incorrect approach, particularly in studies that analyze models with *adversarial examples* (Liang et al., 2021; Yuan et al., 2017). Moreover, Doshi-Velez and Kim (2017) argues that the need for interpretability comes from the *incompleteness* of the problem formalization.
Instead of finding a technical definition for interpretability, we can categorize existing systems in terms of transparency. Z. C. Lipton (2016) states two properties for interpretable models: *transparency* and *post-hoc interpretability*. The definition for the first and its related terms are given as in the following,

Definition 2.2.1 (*Understandability*)**.** "*Denotes the characteristic of a model to make a human understand its function - how the model works - without any need for explaining its internal structure or the algorithmic means by which the model processes data internally.*" (Barredo Arrieta et al., 2020; Montavon et al., 2018)

Definition 2.2.2 (*Transparency*)**.** *"A model is considered to be transparent if by itself it is understandable."* (Barredo Arrieta et al., 2020)

To elaborate further, we discuss the degrees of transparent models, as not all models provide the same extent of understandability (Barredo Arrieta et al., 2020). Both in Z. C. Lipton (2016) and Barredo Arrieta et al. (2020) the categorization is made as: *simulability*, *decomposability*, and *algorithmic transparency*. We discuss each of them briefly.

- *Simulability* denotes the model's characteristic to be simulated or thought only by a human. Thus, the complexity of a model plays an important role here. Models that can be presented to a human in terms of text and visualizations are considered interpretable (Ribeiro et al., 2016). Simulatable models are usually elementary models (Barredo Arrieta et al., 2020; Tibshirani, 1996).

- *Decomposability* represents the model's characteristic to explain each part of the model. Basically, when all components of a model are simulable, that model is decomposable (Z. C. Lipton, 2016), given that the inputs are already interpretable (Barredo Arrieta et al., 2020).

- *Algorithmic transparency* deals with the user's comprehension of the input's journey from entering the model to becoming a prediction (Barredo Arrieta et al., 2020; Z. C. Lipton, 2016). For example, linear models can be considered algorithmically transparent since the user can understand how the model can act in a given situation (James et al., 2013).

The second property of interpretable models, *post-hoc explainability*, aims to improve the interpretability of not readily interpretable models. It does so through *text explanations*, *visual explanations*, *local explanations*, *explanations by example*, *explanations by simplification*, and *feature relevance explanations* techniques (Barredo Arrieta et al., 2020; Z. C. Lipton, 2016). In a more general sense, post-hoc explainability methods can be grouped into three categories in terms of the knowledge of the target model (*model-specific or model-agnostic*), the granularity of focus (*local or global*), and form (*text* or *visual*). The first category refers to the explainability method's assumption of the model's structure. *Model-specific* techniques can be utilized with a limited set of models since these techniques make an assumption about the model to be explained.
On the other hand, *model-agnostic* techniques are designed not to require knowledge about the model's inner workings (Barredo Arrieta et al., 2020; Guidotti et al., 2018). The second category denotes the explanation's domain. *Local* explanations reason about a particular prediction of a model at the feature level (Doshi-Velez & Kim, 2017) (e.g., compute a saliency map by taking the gradient of the output with respect to a given input vector (Z. C. Lipton, 2016)), whereas global explanations aim to outline

the model's general behavior on the dataset (Barredo Arrieta et al., 2020; Doshi-Velez & Kim, 2017; Guidotti et al., 2018). Global explanations are usually presented in the structure of a series of rules (Lakkaraju et al., 2016). The third and last category, the form of the explanation, can be *visual* or *text*. We will discuss these forms of explanation in detail after we give a definition of *explainability*. From now on we will talk about the explainability of a model rather than its interpretability.

Definition 2.2.3 (Explainability)**.** "*Explainability is associated with the notion of explanation as an interface between humans and a decision maker that is, at the same time, both an accurate proxy of the decision maker and comprehensible to humans.*" (Barredo Arrieta et al., 2020; Guidotti et al., 2018)

- *Text explanations* are techniques that learn to produce textual expressions that assist user in understanding the model's outcomes (Bennetot et al., 2019).
- *Visual explanations* are techniques that supplement a model's explainability by visualizing the model's behavior. Due to the mismatch between high-dimensional nature of complex ML systems and the capacity of human reasoning (Burrell, 2016), visual explanations often employ dimensionality reduction practices (Barredo Arrieta et al., 2020).

From the perspective of explainability, one would intuitively prefer transparency since transparent models can be easily explained. However, some works argue that as the transparency of a model increases, its performance usually tends to decrease (Dosilovic et al., 2018). Although, other works argue that this is not necessarily true, particularly in cases where the data is well structured and the quality and value of available features are outstanding (Rudin, 2018). Considering FND models, extensive and complex models can not be avoided since news pieces tend to be long texts, and fake news pieces are fabricated to mimic real news. Alternatively, social networks are represented as graphs, thus forcing SOTA FND models to utilize complex approaches that decrease transparency, such as word embeddings, data fusion, and graph data structure (Dou et al., 2021).
On the other hand, global explanations can be helpful to domain experts by providing information about what model has learned on a global level; however, global explanations might be difficult to obtain (Z. C. Lipton, 2016). Instead, local explanation methods are easier to obtain and more practical for real-world applications. For example, if a user requests an explanation for a prediction, local explanations can provide it, which also complies with the "right to explanation" (Goodman & Flaxman, 2017).
Depending on the model adopted for FND, we might be required to use model-specific

or model-agnostic approaches. When dealing with DNNs in a post-hoc setting, usually, it is better to opt for a model-specific explanation as it will give a better insight into the model. However, in cases where a model-specific approach can not be employed, model-agnostic methods are the choice. Below are listed the types of post-hoc approaches (Barredo Arrieta et al., 2020).

- *Explanations by example*, a method suggested by Caruana et al. (1999), focus on obtaining representative information from a model by providing explanations for an example that sufficiently illustrates the inner workings of the model (Baehrens et al., 2010; Barredo Arrieta et al., 2020).
- *Explanations by simplification* techniques construct a new and simplified system to provide explanations for a model. These simplified systems keep the performance of the original model while displaying less complexity (Barredo Arrieta et al., 2020).
- *Feature relevance explanations* compute feature relevance scores for a model's variables in order to determine the effect a feature has upon a model (Barredo Arrieta et al., 2020).

We discussed the types of explanation methods we can adopt and how these methods can shape the design of a model. Then we laid out forms of explanations that aims to convey information about the model's behavior. It is possible to see a combination of the previously mentioned explanation forms. In order to present the user with an comprehensible explanation, we characterize a good explanation and define its important properties in the following subsection.

2.2.2 What Makes A Good Explanation

There is no clear definition of the characteristics of a good explanation (Barredo Arrieta et al., 2020). XAI draws wisdom from social and cognitive sciences to handle the subjectivity in explanations. A comprehensive study on social sciences and XAI by Miller (2017), analyzes explanations in terms of the content, the *explainer* and the *explainee*. The author argues that the AI research lacks knowledge about the properties and structure of an explanation. The major findings from a collection of works about a good explanation's characteristics are outlined below.

- Explanations are *contrastive* (P. Lipton, 1990; Miller et al., 2017). When presented with counterfactual explanations, users understand the model's decision easier (Byrne, 2019; Escalante et al., 2018; Lopez-Paz et al., 2016). For example, rather than asking why event A occurred, we ask why event A occurred instead of event B (Barredo Arrieta et al., 2020; Miller, 2017).

- Explanations are *selective*. Presenting all the causes for an event to a human is pointless since humans are inclined to select a couple of leading causes out of numerous, sometimes countless, causes (Herlocker et al., 2000). Accordingly, Miller (2017) argues that specific cognitive biases shape this selection process.

- Explanations are *social* and conveyed from explainer to explainee via social interaction. Hence, explanations are transferred through the frame of the explainer's beliefs about the explainee's beliefs (Miller, 2017).

- Probabilities may be insignificant. Even though probabilities matter when creating the explanations, the usage of these statistical relations in explanations is not as effective as that of causes. If the underlying causal explanation is not included, then the utilization of statistical generalizations is not sufficient (Miller, 2017).

It should be noted that the characteristics of a good explanation are not limited to the ones mentioned above. These are the most prominent characteristics discussed by Miller (2017). An important aspect is that the explanations are provided to an *explainee*, also called an *audience*. The audience is the person or people receving the explanation (Barredo Arrieta et al., 2020). It is further noted in (Barredo Arrieta et al., 2020) that explanations depend on the audience, i.e., an explanation meant for an end-user will not be enough for a domain expert, or an explanation for a domain expert might be too complicated for an end-user. Also, we will refer to the expainee as the *audience* from now on.
The main target audience is the primary driver when considering the needed outcomes for an explanation. Barredo Arrieta et al. (2020) summarize the pursued goals when trying to attain explainability. These goals are listed below.

- *Trustworthiness* deals with the assurance of a model's intended behavior when the model is presented with real-world scenarios (Antunes et al., 2008; Z. C. Lipton, 2016). Some studies highlight the importance of *trustworthiness* as a requirement for explainability (Kim, Glassman, et al., 2015; Ribeiro et al., 2016). The target audience for this goal are domain experts and users affected by the model decisions (Barredo Arrieta et al., 2020).

- *Confidence* refers to the robustness and stability of a model (Barredo Arrieta et al., 2020). Yu (2013) argues that stability is a prerequisite when obtaining explanations from a model. Moreover, Barredo Arrieta et al. (2020) argues that an explanation method should not provide trustworthy explanations for unstable models. The audience relevant to this goal are domain experts, developers, managers, and regulatory entities.

- *Fairness* refers to a model's potential to assure a fair prediction for a user affected by the model's prediction based on factors such as age, race, and gender (Barredo Arrieta et al., 2020; Oneto & Chiappa, 2020). The audience for this goal consists of users affected by model decisions and regulatory entities.
- *Transferability* refers to the model's capability to perform on unseen data. It is the desired goal not just for explainability but also for obtaining good performance from the model (Kuhn & Johnson, 2013). The audience for this goal is domain experts and data scientists.
- *Causality* denotes the causal relationships among model variables (Pearl, 2009). Its primary audience is domain experts, managers, and regulatory entities (Barredo Arrieta et al., 2020).
- *Informativeness* is a goal meant for all users, and it deals with the information provided by the model. In order to fill the gap between the user's decision and the prediction of a model, a massive amount of information about the problem at hand needs to be conveyed to the end user (Barredo Arrieta et al., 2020).
- *Accessibility* refers to the possibility of end users getting more involved in a model's development or improvement (Miller et al., 2017). The audience for this goal includes product owners, managers, and users affected by model decisions.
- *Interactivity* denotes a model's capability to interact with the user (Kim, 2015). The audience consists of domain experts and users affected by model decisions.
- *Privacy awareness* is a goal not frequently seen in the literature. It deals with the learnings of a model's internal representation such that these learnings might pose a privacy breach. From the opposite perspective, it is a differential privacy breach when an unauthorized third party can explain the inner workings of a trained model (Barredo Arrieta et al., 2020).

Considering a model, the evaluation metrics from the test set reflect the model's overall performance on unseen data and allow comparing different models that use the same dataset (Olson et al., 2017). For example, metrics like accuracy, F1 score, recall, and precision are often used in the evaluation of models. Given that there are numerous metrics, it should be noted that different domains and models may require different evaluation metrics (Handelman et al., 2019; Hossin & Sulaiman, 2015; McNee et al., 2006). Similar to models, explanations require evaluation methods that can quantify their performance. So far, we have seen that explanations might have different audiences, they can take several forms, and they have desired properties. Therefore, like models, there should be a set of explanation evaluation methods focusing on different

categories of explanation approaches. In fact, a rigorous study by Doshi-Velez and Kim (2017) lays out the categorization of explanation evaluation approaches. The authors split the evaluation methodologies into three:

1. *Application-grounded evaluations* are conducted on domain experts interacting with explanations in a real-world application. This kind of evaluation directly tests the system's objective. Thus attaining high performance with respect to application-grounded evaluations suggests good evidence of the explanation's success. The fact that we need humans to interact with a real-world application in an environment that can be observed for experimentation makes this type of evaluation more specific and, thus, the most costly of all three types of evaluation (Doshi-Velez & Kim, 2017).

2. *Human-grounded evaluations* are constructed by more straightforward experiments on humans who are not necessarily domain experts. Although this type of evaluation is less specific than the application-grounded evaluations, it offers more flexibilty and is less costly. It is a good choice when the task is to test the quality of an explanation in a general sense (Doshi-Velez & Kim, 2017). For example, a recent study (Mohseni et al., 2018) used human attention maps that overlay images as explanations and asked users to rate the decision made by the model. The authors further argue that the evaluation on these attention maps can be utilized to understand the *trustworthiness* of a model.

3. *Functional-grounded evaluations* do not include real humans. Instead, these evaluations use a formal definition of interpretability as a proxy to assess the explanation's quality. The lack of human dependency makes them favorable due to low cost. Typically, these evaluations are preferred, especially when conducting experiments with humans might be unethical. The challenge with these evaluations is to select the suitable proxy models. In addition, when possible, it is considered good practice to obtain verified proxies first, for instance, by human-grounded evaluations (Doshi-Velez & Kim, 2017).

From high cost to low cost, and more specific to more broad, one can opt for an evaluation technique to obtain a performance indicator of an explanation. As discussed above, each approach requires a entirely different setting, which brings its shortcomings with it. Depending of the availability of resources such as time, finances, the expertise of the user, or the sufficiency of human subjects, one might have to opt for a different evaluation technique. Having highlighted essential characteristics of a good explanation by collecting insights from the literature, we now move forward to the frequently mentioned techniques used in XAI. We primarily focus on post-hoc local explanation techniques and outline their contribution to this thesis.

2.2.3 Overview of Techniques in Explainable Artificial Intelligence

As discussed in the last section, when constructing an explanation method, one has to consider the audience, opt between model-specific or model-agnostic, local or global explanations, and utilize various forms of explanations. In literature, there exist various combinations of previously mentioned options. For transparent models, no explanation method is needed; one can obtain relevant information in the form of weights or attention scores, given that the features are simple enough (Barredo Arrieta et al., 2020). In particular, we talk about explanation methods that were frequently mentioned in studies and relevant to this thesis.

First, we discuss the initial methodologies aimed at gaining insight from a black-box model. The one of the initial approaches was to ask: "What happens if we remove this part of the input?". *Sensitivity Analysis* (SA) deals with the analytical assessment of the effect of an omitted input variable on the uncertainty of a model (Novak et al., 2018). SA can be done on two levels, local and global. *Local Sensitivity Analysis* (LSA) assesses the impact of the changes in the input on the output. In contrast, *Global Sensitivity Analysis* (GSA) examines the effect of each variable (feature) with respect to the variations of all parameters (Rao et al., 2019). In the literature, there are a variety of approaches for both GSA and LSA. For instance, Novak et al. (2018) construct a GSA procudure that employs the partial derivative of each parameter in the back-propagation algorithm to explore the change rule, which admits the *Input-Perturbation-Sensitivity* (IPS) that allows for obtaining global sensitivity. An interesting example of a GSA and LSA fusion approach, Kowalski and Kusy (2018) utilizes LSA to reduce the number of input features and GSA to reduce the number of patterns learned by a model.

Another approach is to calculate relevance scores for each feature using saliency maps. The usage of saliency maps were initially adopted for image CNNs (Simonyan et al., 2014), then extended to NLP in Denil et al. (2014) and Bansal et al. (2016). Typically, salience maps compute a gradient to get a relevance score for an input feature. In other words, they convey information about the model's sensitivity based on the input.

A popular method used in XAI is *Layer-wise Relevance Propagation* (LRP) which was first introduced for *Fully Connected Networks* (FCNs) and CNNs by Lapuschkin et al. (2015), then extended to *Recurrent Neural Networks* (RNNs) by Arras et al. (2017). LRP assumes that a model can be *decomposed* into several layers that might contain feature-relevant information. From the last layer to the input layer, LRP computes a relevance score for each dimension of the vector at a layer. As LRP moves backward in the layers, the sum of relevance scores does not change, always staying equal to the prediction probability (Lapuschkin et al., 2015).

Similar to LRP, a study explaining DNNs offers another solution named *Deep Learning Important FeaTures* (DeepLIFT) (Shrikumar et al., 2017). This approach addresses two

shortcomings of LRP, namely, the failure to model saturation caused by activation functions and the possibility of getting misleading importance scores due to discontinuous gradients. Combining techniques from LRP and integrated gradients (Sundararajan et al., 2016), DeepLIFT computes importance scores based on the *difference-from-reference* approach that allows the propagation of information even if the gradient is zero. Difference-from-reference is a procedure that involves determining a reference and then getting the difference between the reference and the output. This method is also later utilized to create DeepSHAP (Lundberg & Lee, 2017).
In contrast to model-specific approaches like LRP and DeepLIFT, *Locally Interpretable Model-agnostic Explanations* (LIME), as the name suggests, is a model-agnostic method. LIME interprets the predictions of any black-box model by approximating the model around a prediction. This approximation allows for obtaining a locally faithful and interpretable version of the model (Ribeiro et al., 2016).
So far, no study unifies all the works to create one explainability framework. To address this lack of unification, Lundberg and Lee (2017) offer *SHapley Additive exPlanation* (SHAP) framework, in which the authors utilize recent studies from game theory based on the pioneering work of Shapley (1952). These studies are *Shapley regression values* (Lipovetsky & Conklin, 2001), *Shapley sampling values* (Štrumbelj & Kononenko, 2014)*quantitative input influence* (Datta et al., 2016), and recent approaches like LIME and DeepLIFT are utilized to create a model-agnostic and model-specific explainers. SHAP values measure the feature importance and are obtained via Shapley values of the conditional expectation function of a model (Lundberg & Lee, 2017). Model-agnostic SHAP values are computed using the Shapley sampling values method, which uses an approximation of a permutation adaption of classic Shapley value estimation. Alternatively, a model-agnostic method, *KernelSHAP*, employs LIME with linear explanations and Shapley values to find a weighting kernel that enables regression-based estimation of SHAP values. The authors also propose model-specific procedures, such as *LinearSHAP*, which can approximate Shapley values using weights for linear models, and *DeepSHAP*, which connects DeepLIFT with Shapley values. Less striking examples are *low-order SHAP* and *max SHAP* for model-specific explainers. We will discuss SHAP values further in Section 3.2.1.
In the literature, there is a lack of explanation tools for GNNs. GNNs require graphs as input and produce a prediction at a level based on the focus of the task (Zhang et al., 2018). Graphs can represent the node feature information as well as rich relational information between nodes (Zhang et al., 2018; J. Zhou et al., 2018). GNNs are potent tools that are designed to learn relational information between nodes and node features. This makes GNNs a perfect candidate for learning patterns from social media networks (Zang et al., 2016). In our case, we want to understand how a GNN behaves when classifying fake and real news pieces by explaining their propagation networks.

A study by Ying et al. (2019) proposes a post-hoc model-agnostic approach called *GNNExplainer* to explain predictions made by GNNs. *GNNExplainer* takes a trained GNN, input graph(s) and its prediction(s), and it returns local explanations in the form of subgraph(s) of input graph(s) along with the most influential node features for the prediction. These subgraphs are constructed by maximizing the mutual information between the subgraph and the input graph with respect to the prediction (Ying et al., 2019).

Bearing in mind the FND models and explanation methods discussed, one could use LIME, DeepLIFT, or SHAP framework for news content models, which are essentially DNNs with textual data as inputs. Since the SHAP framework has implemented every method optimally and addressed the shortcomings of each, and provides text plots and easily interpretable importance scores, which are crucial for understanding the words or the word groups that are most important. Therefore, for explaining our news content model, we employ SHAP framework, in particular, partition explainer. We wanted to work with DeepLIFT, but our model choice was not compatible with DeepLIFT. Details on the partition explainer will be provided in 3.2.1. On the other hand, for GNNs, the choice is straightforward as there is only one option. GNNExplainer can be helpful in identifying the most critical spreaders of a news piece which will be discussed in Chapter 4.2.

In the next chapter, we elaborate on the news content FND model we used in this thesis. We initially introduce foundational terms for our task, we inspect Transformer models, and briefly discuss their architecture. We analyze the news content dataset and share our initial findings. We then report news content model's performance on the dataset and argue the reasons behind its performance. We construct our arguments based on the explanations we obtain from the partition explainer and share results of experiments. We stress the importance of explaining a model and understanding its inner mechanisms.

3 News Content Models for Fake News Detection

The automated detection of fake news on social media comes with its characteristic challenges. The fact that fake news pieces are constructed to misguide its consumers makes them hard to distinguish by only using news content. Fortunately, language models have become robust enough to capture patterns on different levels.
This chapter examines a language model, then inspects the model using explanation methods. In Section 3.1, we lay out definitions and inspect the Transformers architecture. We analyze the dataset and report our findings. Then, we examine our news content model and share its performance on the dataset. Finally, in Section 3.2, we outline the SHAP framework, which we later adopt to inspect our model's performance and behavior.

3.1 News Content Models

A great share of FND methods utilizes news content. Models that base their predictions solely on news content focus on the patterns in the text, especially words or word groups that frequently appear in other instances of the same class. As discussed in Section 2.1, there exist a variety of approaches available for news content models. However, due to unavailable or outdated datasets, we could not work with most news content models or datasets.

3.1.1 Notation and Definitions

Here we introduce the notation used in this section. Note that these notations will appear in its context, which will provide concrete examples for each symbol defined in Table 3.1.
We define some relevant concepts utilizing the notation. First, we talk about terms and definitions for *tokenization*, outline the tokenization process. First, to build upon a concrete foundation, let us consider a news article x^{raw} fed to the tokenizer T.

Definition 3.1.1 (*Tokenizer*)**.** A tokenizer $T : X^{raw} \mapsto X^{tok}$ is a function that maps raw textual data to smaller units called tokens.

$x^{raw} \in X^{raw}$	A news article.
$y^{raw} \in Y^{raw}$	A label of news article.
T	Tokenizer function
ψ	Label mapping function
$x^{tok} \in X^{tok}$	Tokenized news article
$y \in Y$	Vectorized class value.
$\|x^{tok}\|$	The number of tokens in x^{tok}.
$\hat{y}$	Prediction of model.
$x \in X$	Numeric vector of x^{tok}
$\|x\|$	The length of input vector
l	Index of a layer
$l_{embedding}$	Embedding layer
$a_i^{(l)}$	The value of unit i in layer l
$w_{ij}^{(l)}$	Weight between units i in layer l and j in layer $l+1$
σ	Activation function

Table 3.1: Notation used in this section.

A token can be a word, character or a subword. Therefore, we define three types of tokenization techniques:

- *Word tokenization* splits the given text into individual words based on a delimeter such as whitespace, commma, etc. This approach creates a vocabulary from the inputs it was trained on. All words do not appear in the vocabulary are replaced with unknown token ([UNK]), and this concept is called being *Out Of Vocabulary* (OOV). Depending on the task, the size of the vocabulary can grow quite large. The solution for exploding vocabulary sizes was introduced in subword tokenization. Commonly used examples for word tokenizers are Word2Vec (Mikolov, Chen, et al., 2013) and GloVe (Pennington et al., 2014).
- *Character tokenization* splits the text into single characters. Since the size of available characters is limited and known, the OOV problem is solved by encoding the unknown word by means of its characters. Although looks like a good solution, the length of tokens can be massive for long texts.
- *Subword tokenization* splits the given text into subwords, also called *n-gram characters*. For instance, comparative words like harder is segmented into "hard-" and "-er", or superlative words like hardest is segmented into "hard-" and "-est". Most common method for subword tokenization is *Byte Pair Encoding* (BPE). BPE was introduced by Gage, 1994 but adapted to word segmentation by Sennrich et al., 2015. BPE iteratively merges the most frequently appearing character or character sequences. This approach allows for an efficient space usage thus smaller vocabularies (Sennrich et al., 2015).

We say that an input is *tokenized* after it is fed to the tokenizer. A tokenized news article x^{tok} is a vector of tokens in which the order of the words and characters in $x^{raw} \in X^{raw}$ are kept.

$$T(x^{raw}) = x^{tok} = [x_1^{tok}, \ldots, x_n^{tok}], \text{ where } n = |x^{tok}|.$$

Usually, models accept a fixed size of input, hence, to get a fixed length output, the tokenized sequence x^{tok} is padded with padding tokens ([PAD]) where the news article is not long enough. In case it is longer than the fixed length, then it is truncated. Some tokenizer implementations in Huggingface use masks to mask out the pads so that they are not included in the computation. These are called masked transformers (Wolf et al., 2020). In fact, our news content classifier has the same behavior which can be observed in 3.1.
We denote the space of raw label $Y^{raw} = \{"fake", "real"\}$, with $y^{raw} \in Y^{raw}$. We use a label mapping function $\psi : Y^{raw} \mapsto Y$ that maps raw labels to classes, where $Y \in \mathbb{R}^2$ with,

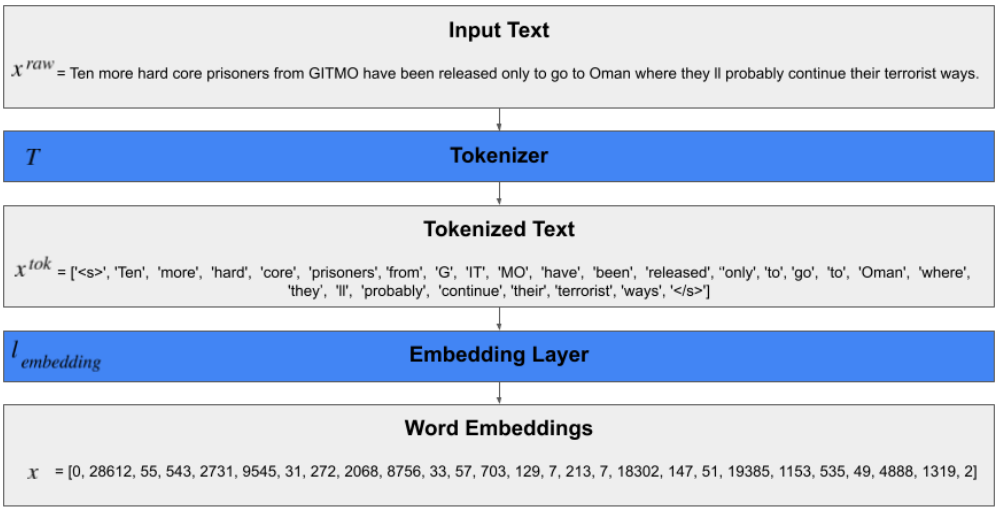


Figure 3.1: The preprocessing pipeline for a textual input.

$$\psi(y^{raw}) = y = \begin{cases} 0, & if \;\; y^{raw} = "fake" \\ 1, & otherwise \end{cases}$$

In order to feed the input to the model, we need numeric representation which can be obtained by various techniques. One widely used approach is BoW representation which produces features based on the number of occurrences of a word or token. An alternative BoW representation uses the presence or absence of words in the vocabulary instead of frequencies. A more sophisticated approach is *Word2Vec*, which encodes words into numeric values by learning word associations. From the perspective of the representation of a word, *Word2Vec* can capture different degrees of similarity between words allowing for the preservation of semantic and syntactic relationships (Mikolov, Chen, et al., 2013). It is clear that the transformation of words into numeric vectors is a very crucial stage for FND since we need to maintain as much contextual information as possible. Nevertheless, the SOTA is an even more sophisticated approach called *Transformer* which is the building block of many powerful language models such as BERT. For ease of notation, we refer to this stage as the *embedding layer* and denote it with $l_{embedding}$. The input transformation pipeline is illustrated in 3.1.

Definition 3.1.2 (*Classifier*)**.** A classifier $f : X \mapsto Y$ is a function that outputs a predicted scores $f(x)_y$ for each class y for a given input x.

Our classifier is a language model that was trained on a large corpus and a news dataset that contain fake and real news. It will predict whether a news piece is real or fake via assigning each label a probability.

Definition 3.1.3 (*Prediction*)**.** A prediction $\hat{y}$ is the maximum of predicted scores $f(x)_y$ of a classifier.

$$\hat{y} = argmax_{y \in Y} f(x)_y$$

We use a neural network classifier that consists of several layers and a complex architecture. Our model is able to work with big vocabularies and can classify news pieces based on various features.

Definition 3.1.4 (*Neural Network Classifier*)**.** A neural network classifier is a *classifier* f that comprises of layers l with $1 \leq l \leq L$, where L denotes the number of layers. Each layer has a set of units $a_i^{(l)}$ with i denoting the position of the unit in a layer l. We say that between two units $a_i^{(l)}$ belonging to layer l and $a_j^{(l+1)}$ belonging to layer $l+1$ have a weight value $w_{ij}^{(l)}$ that connects them. Along with a non-linear activation function σ, we can define the value of the j-th unit $a_j^{(l+1)}$ in terms of weights and unit values from previous layer for an FCN, with N as the number of units in layer l.

$$a_j^{(l+1)} = \sigma(\sum_{i=1}^{N} a_i^{(l)} w_{ij}^{(l)})$$

Neural networks are very powerful models. They can train millions of parameters using matrix multiplications, gradient-based optimization methods, and a fruitful set of other techniques, some of which will be discussed later. Neural networks iteratively optimize the weights between layers so that the produced output is as close to the expected output. This is achieved by using optimization methods, such as *Gradient Descent* (GD) (Cauchy, 1847), *Stochastic Gradient Descent* (SGD) (Robbins & Monro, 1951), *Adaptive Moment Estimation* (Adam) (Kingma & Ba, 2014) that minimize the loss function defined for the problem. While many optimization methods are available for neural networks, we do not examine any of them.
For classification problems, we adopt a layer called *Softmax* (Bridle, 1989) that outputs the predicted scores $f(x)_y$ for each class y by normalizing outputs $Z_y(x)$ from the previous layer.

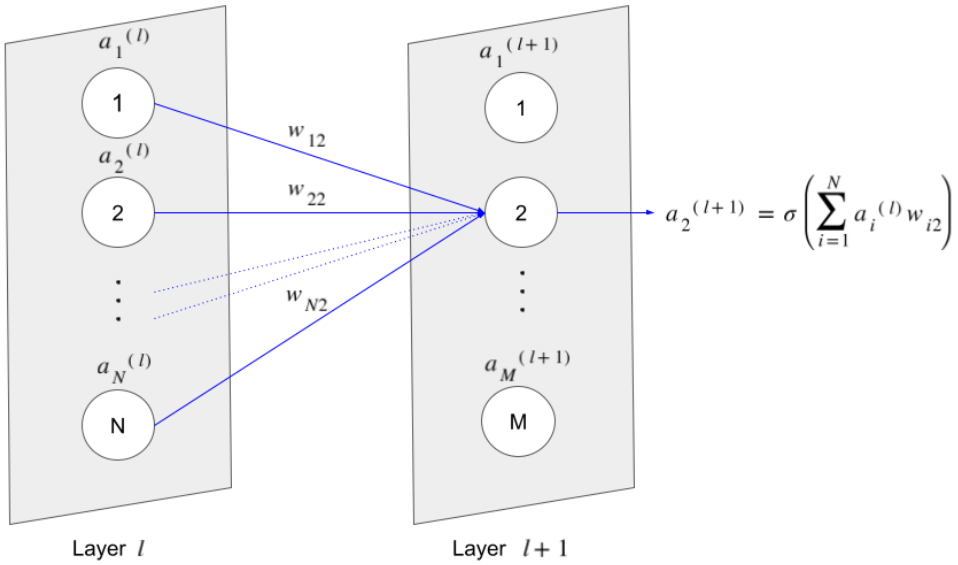


Figure 3.2: Units and layers of an FCN. For brevity, arrows and weights are only drawn for $a_2^{(l+1)}$

$$f(x)_y = \frac{exp(Z_y(x))}{\sum_{y' \in Y} exp(Z_{y'}(x))}$$

The FCNs are the plainest architectures for neural networks. In an FCN, all units are connected. This means that there is a weight between each unit of the FCN and its previous layer. They are very straightforward to construct. Nevertheless, the usage of these architectures for modeling sequences is not preferred due to the exponential number of trained parameters. Instead, when modeling sequences like sentences and documents, a common approach is to adopt RNNs. RNNs allow previous outputs to be used while having hidden states. This permits information to pass through the sequence. However, vanilla RNNs are not powerful enough to represent long-term dependencies (Bengio et al., 1994) and suffer from vanishing/exploding gradients (Pascanu et al., 2012). *Long Short-Term Memory* (LSTM) (Hochreiter & Schmidhuber, 1997) addresses the shortcomings of RNNs. The idea is to keep a cell state updated with the previous cell's state. This cell state is passed to the consequent cells to form a chain representing the document. More precisely, each cell corresponds to a token whose information will be shared with consequent tokens by the propagation of cell states. This approach is indeed very useful for long documents since news articles tend to be long and their

sentences contextually relevant. LSTM is usually used in different variations based on the same idea. For instance, a consequent study by Gers and Schmidhuber (2000) has extended LSTMs with *peephole connections*. LSTMs have been proven to deliver a good performance in NLP tasks such as *speech recognition* (Xiong et al., 2016). LSTMs perform well; however, they are being replaced with attention-based models due to long training times and large memory requirements during training.
One last thing to discuss is how the models are trained in terms of supervision. *Supervised* models are trained with label data. *Unsupervised* models work with unlabeled data and aims to find patterns in the data. An exciting setting is *semi-supervised* models. These models are often provided with small amounts of labeled data and large amounts of unlabeled data for training. There are two settings for semi-supervised learning. *Transductive* learning aims to predict unlabeled data, whereas *inductive* learning samples unlabeled data from the same distribution to infer the label (Gammerman et al., 1998).
To summarize, we first outlined the process of a text becoming a numeric vector for a model. Then we discussed the classification steps and recapped the formation of neural networks. Finally, we argued about more capable model architectures employed in language models. Next, we will introduce the attention mechanism and Transformer models.

3.1.2 Transformer Architecture

Transductive learning has been successfully utilized along with the encoder-decoder structure in many language tasks (Cho et al., 2014; Sutskever et al., 2014; Vaswani et al., 2017). Transduction is first proposed by Gammerman et al. (1998) to counteract the unlabeled data problem. In contrast to supervised learning, transductive learning does not require all data to be labeled. Instead, it utilizes the clustered behavior of data. Transductive learning assigns labels to unlabeled data using the gaps between different clusters and a small set of labeled data. Accordingly, Transformer models are transductive and use an encoder-decoder structure to achieve that.
Cho et al. (2014) proposed an encoder-decoder structure that takes into account the order of words. This encoder-decoder structure consists of one RNN as the encoder and one RNN as the decoder. The encoder maps an input sequence to a fixed-length vector, and the decoder maps this fixed-length vector to a target sequence. Transformer architecture adopts a similar approach that employs feed-forward and Multi-Head Attention layers in both encoder and decoder, which is illustrated in Fig. 3.3 with N=6 stacks of encoders and decoders.
In order to reduce sequential computation, CNNs have been adopted as building blocks that parallelly compute hidden representations for all input and output positions (Vaswani et al., 2017). Although aimed to reduce computation, the number of

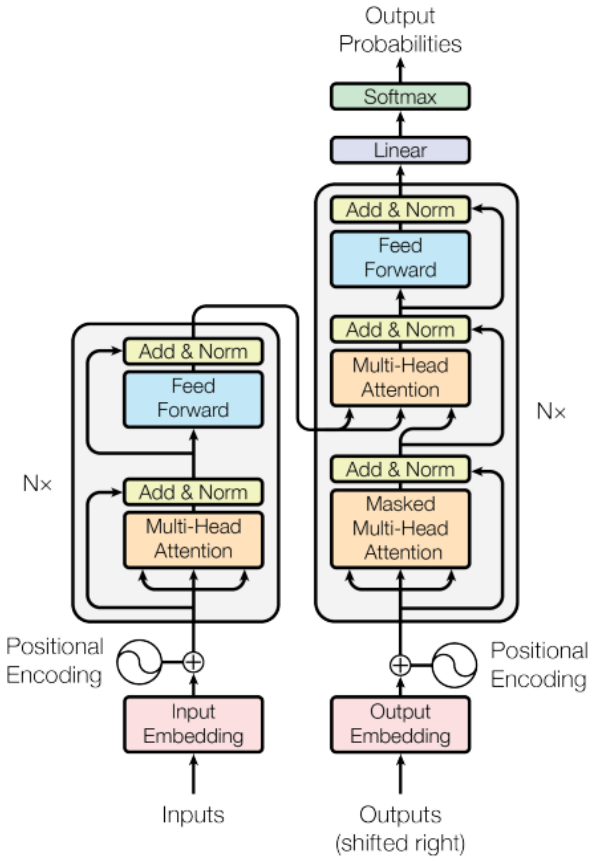


Figure 3.3: The Transformer Model Architecture (N=6). Figure obtained from (Vaswani et al., 2017)

operations to convey information from one random input or output to another increases linearly in ByteNet (Kalchbrenner et al., 2016) and logarithmically in ConvS2S (Gehring et al., 2017).
Contrary to CNNs, Transformers fix the number of operations by averaging attention-weighed positions, which decreases the effective resolution. However, this decrease in resolution is neutralized by using Multi-Head Attention (Vaswani et al., 2017). Initially suggested in the decoder of the model proposed by Bahdanau et al. (2014), an attention mechanism works similarly to human attention; it learns to put more importance on some words that convey the relevant information about the sentence. It does so by means of a context vector that depends on a sequence of *annotations*. An annotation h_i

for a word (or embedding) x_i contains information about the complete input sentence but with a focus on the words that are closer to the word x_i. The context vector c_i for word x_i is obtained as a weighted sum of all these annotations h_i (Bahdanau et al., 2014):

$$c_i = \sum_{j=1}^{|x|} \alpha_{ij} h_j.$$

The weight α_{ij} is obtained by applying softmax to associated energy e_{ij}, which is an output of the alignment model a. The alignment model a is a feed forward neural network that jointly learns with the rest of the system. More precisely, we compute these values as follows (Bahdanau et al., 2014):

$$\alpha_{ij} = \frac{e_{ij}}{\sum_{k=1}^{|x|} exp(e_{ik})}$$

where

$$e_{ij} = a(s_{i-1}, h_j)$$

with s_i representing the current and s_{i-1} the previous state of the model (Bahdanau et al., 2014). This is called *additive attention*.
For Transformer models, the authors define the attention function as *mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compability function of the query with the corresponding key.* (Vaswani et al., 2017). The Transformer model employs two different attention mechanisms, namely, *Scaled Dot-Product Attention* and *Multi-Head Attention*. Following (Vaswani et al., 2017), we denote that the input for the attention layers are matrices called queries $Q \in \mathbb{R}^{d_{model} \times d_k}$, keys $K \in \mathbb{R}^{d_{model} \times d_k}$, and values $V \in \mathbb{R}^{d_{model} \times d_v}$, with d_k as the number of keys or values and d_{model} being the model dimension. Scaled Dot-Product Attention computes the dot product of all queries $q_i \in Q$ with all keys K and scale the resulting weights with $\frac{1}{\sqrt{d_k}}$. After obtaining the softmax of the scaled weights, each weight is multiplied with the correspoding value to obtain attention values.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

The second attention mechanism, Multi-Head Attention, uses multiple attentions each of which uses a different learned linear projection of Q, K, V. Output of each of these attentions are then concatenated to obtain the final result. More precisely, it is computed as follows,

$$MultiHead(Q, K, V) = Concat(head_1, head_2, \ldots, head_h)W^O$$

where each $head_i$ is calculated as,

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

with projection for Q as $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, K as $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, Q as $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and lastly, $W^O \in \mathbb{R}^{hd_v \times d_{model}}$ (Vaswani et al., 2017).
We now outline each part of the Transformer model.

- *Encoder*: Consists of N=6 identical layers. Each of these layers have two sub-layers the first of which uses a Multi-Head Attention and layer normalization (Ba et al., 2016) along with a residual connection (He et al., 2015) around. The second sub-layer consists of a feed-forward layer and layer normalization as well as a residual connection around the feed-forward layer (Vaswani et al., 2017)
- *Decoder*: Same as the encoder, this part is also composed of N=6 layers. Additional to the previously discussed two sub-layers in encoder, decoder adopts a third sub-layer which computes the attention values over the output of encoder. As it was done for encoder, decoder also utilizes layer normalization at the end of each sub-layer as well as the residual connection (Vaswani et al., 2017).

Each of the feed forward networks in the sub-layers are position-wise, meanining that they are applied to each position separately and identically. These feed-forward networks use different parameters for each layer. The embeddings are obtained through transductive learning. Lastly, the positional encodings for input embeddings are calculated using sine and cosine functions of different frequencies (Vaswani et al., 2017). We will observe the effect of positionial encodings when explaining our model.
In order to lay foundations for the model we adopted, we have introduced its structure and mechanism. Next, we initially introduce details of BERT, then RoBERTa, and lastly DistilRoBERTa which is the base of our news content model.

3.1.3 Dataset and Model

We employ a fine-tuned version of case-sensitive Transformer model DistilRoBERTa[1] for our task of FND with news content. DistilRoBERTa is a distilled version of *A Robustly Optimized BERT Pretraining Approach* (RoBERTa) (Liu et al., 2019). It uses the same distillation procedure adopted for DistilBERT (Sanh et al., 2019) to *Bidirectional Encoder Representations from Transformers* (BERT) (Devlin et al., 2018). This distillation procedure

[1] https://huggingface.co/GonzaloA/distilroberta-base-finetuned-fakeNews

is referred to as *knowledge distillation* and it compresses a model (Buciluâ et al., 2006) - the teacher - by means of training a smaller model - the student - to reproduce the same behaviour (Hinton et al., 2015). In our case, the teacher is RoBERTa and the student is DistilRoBERTa. First, in order to examine properties of RoBERTa, we discuss BERT in detail.
As the name suggests, the model architecture of BERT is a multi-layer bidirectional Transformer model. BERT uses BookCorpus (Y. Zhu et al., 2015) and English Wikipedia as training dataset, with two training objectives, *Masked Language Modeling* (MLM) and *Next Sentence Prediction* (NSP).
NSP procedure is a binary classification loss that predicts whether two segments (sequences of tokens) are consecutive in the original text. *Positive* and *negative* examples are sampled with equal probability in this process. Positive examples are created with taking the consecutive sentences from the text corpus, whereas negative examples are generated by pairing segments from different documents (Devlin et al., 2018; Liu et al., 2019).
MLM procedure applies the following for each sentence sampled from a document in the cumulative dataset.

- Mask 15% of the tokens.
- In 80% of the cases, replace the masked tokens with [MASK].
- In 10% of the cases, replace the masked tokens with a different random vocabulary token.
- In the remaining 10% of the cases, the masked tokens are left unchanged.

RoBERTa is an optimized version of BERT. It was pretrained longer using longer sequences. RoBERTa uses MLM as training objective, but not NSP. Contrary to BERT, RoBERTa keeps a dynamic masking pattern that changes in training. It is pretrained on the reunion of five datasets (three more datasets than BERT) that size up to 160 gigabytes (GB): BookCorpus (Y. Zhu et al., 2015), English Wikipedia ("English Wikipedia," 2022), CC-News (Nagel, 2016), OpenWebText (Radford et al., 2019), Stories (Trinh & Le, 2018). RoBERTa tokenizes texts using BPE with a vocabulary size of 50,000 and maximum sequence length (maximum number of tokens) as 512. The beginning and end of each document (news article) is marked with <s> and </s> respectively. With MLM as training objective and Adam (2014) as the optimizer, the model reaches better results than BERT. Additionally, it should be noted that these models are further trained for downstream tasks, thus we refer to training stage as pretraining to avoid any confusion. DistilRoBERTa has the same general architecture as RoBERTa but the number of layers are reduced by a factor of two, the *token-type embeddings* and the pooler are removed.

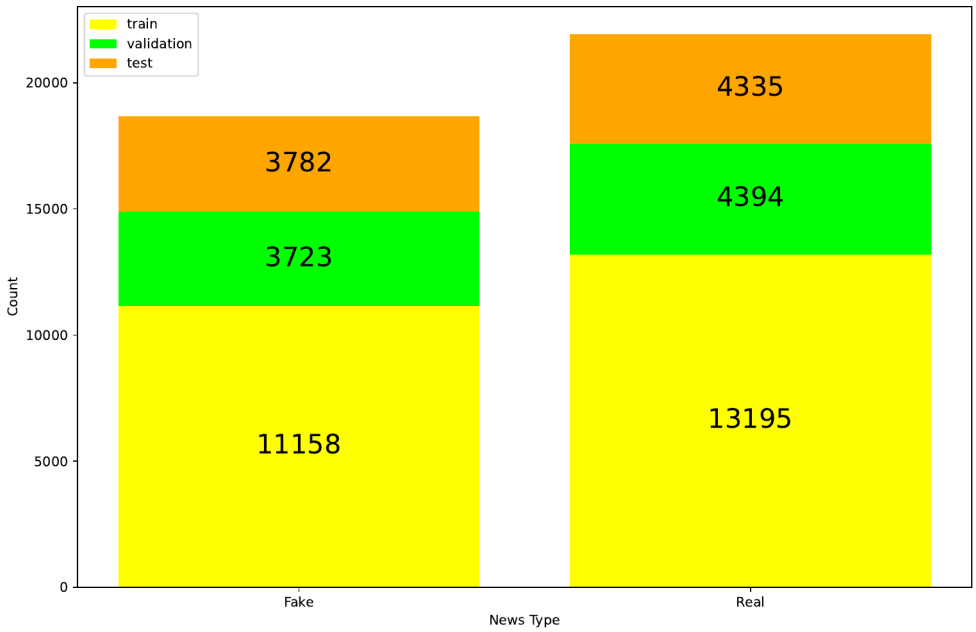
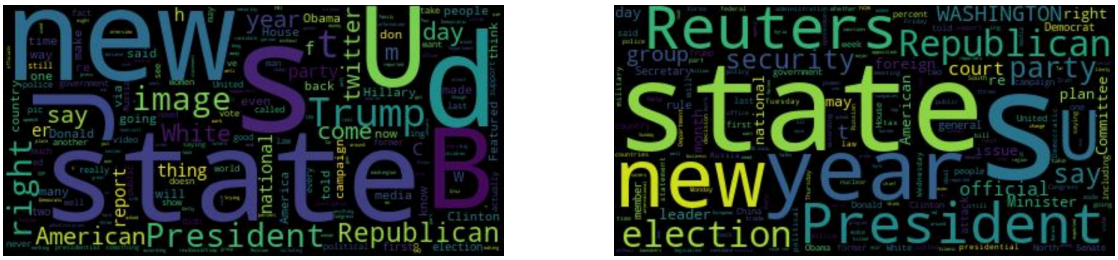


Figure 3.4: News content dataset distribution by label and train/validation/test split

Then DistilRoBERTa is initialized with layers from the teacher. The distillation is done with very large batches (Sanh et al., 2019). Using RoBERTa as a teacher, the student DistilRoBERTa is pretrained on OpenWebTextCorpus (Gokaslan & Cohen, 2019) a reproduction of OpenWebText (Radford et al., 2019).
Our fine-tuned DistilRoBERTa was trained on a dataset[1] curated from different sources. Although there exist better datasets and news content models, we opted for this particular model for two reasons. First, most SOTA news content models do not provide their code and dataset to reproduce results. Second, since Transformers are SOTA and this particular trained model provides us with not only the dataset but also with the train/test/validation splits which allows us to analyze its explanation.
Dataset. The news content dataset comprises of 40587 samples whose distribution of labels and train/validation/test splits are provided in Fig 3.4. The proportion of fake news is 46%, which is a fair distribution between real and fake news instances. The train/validation/test split is the common practice 60%/20%/20%. We analyze 500 most frequently occurring tokens in the dataset using a WordCloud (Oesper et al., 2011) visualization for all samples of real and fake news separately in Fig 3.5. From the visualization, we can observe that samples from both datasets contain the words "new",

[1]https://huggingface.co/datasets/GonzaloA/fake_news

"state", "President", "Republican". In fact, 500 most frequent tokens from fake news samples and real news samples share 63% of tokens. Furthermore, we observe that one of the most frequent token is "Reuters" in real news instances. In fact, 95.93% of real news samples contain the the token coalition "(Reuters)". It is crucial to note that the idea of analyzing the frequency of keywords was initially formed once we started explaining our model. We will share how we draw this insight in the following section.



(a) Fake news

(b) Real news

Figure 3.5: WordCloud visualiations for most frequently occurring 500 tokens in samples of both classes.

Attention dropout probability	0.1
Batch size	16
Epochs	3
Hidden layer activation function	GELU (Hendrycks & Gimpel, 2016)
Layer normalization factor (ϵ)	1e-05
Learning rate	2e-05
Maximum position embeddings	514
Number of attention heads	12
Number of layers	6
Vocabulary size	50265
Weight Decay	0.01

Table 3.2: Hyperparameters of the news content model.

Training. Huggingface provides us with a *TextClassificationPipeline* which allows to feed the dataset from Huggingface directly (Wolf et al., 2020). This pipeline includes the tokenization process and the embedding layer before feeding it to our classifier. Our news content classifier has 6 layers, 12 attention heads and hidden size of 768 that totals up to 82M parameters (RoBERTa has 125M parameters). The vocabulary size is 50265 including special tokens. The model also uses a technique called *dropout* which is a regularization technique that drops some connections between units of layers based on a probability value (Srivastava et al., 2014). All model hyperparameters are defined in 3.2.
Note that maximum position embeddings parameter also considers document start and end tokens when reporting its length. When feeding tokens x^{tok} to the model, the start (<s>) and end (</s>) tokens are added by the model pipeline, so this leaves us with

Accuracy	Precision	Recall	F1 score
98.85%	98.82%	99.03%	98.92%

Table 3.3: The performance metrics for news content model.

maximum sequence length of 512, i.e., $|x| = 512$. Note that in Figure 3.1 the length of x is not 512 since the paddings are masked out before feeding the input x to the model. We only illustrated values for tokens that exist, as pads will receive a value that has no effect for the calculation of the output, such as 0.

Performance evaluation. Model is trained for 3 epochs with a batch size of 16, and Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e - 08$) as the optimizer. The performance metrics for the model are provided in Table 3.3. To evaluate, the model performs very well on the dataset. Only 93 samples out of 8117 are classified incorrectly in the test split. Out of that 93 samples, 51 of them are fake news instances and the rest are real news instances. Although success this high does not necessarily suggest the model has learned its task. It might have found a simple pattern between fake and real news instances, and base its prediction on this trivial feature. This is why explaining a model is crucial. We can inspect our model's behavior under different cases using explanation techniques.

We have laid out the foundations for the model we used to understand our model's learning mechanism. We reported the characterstics, training details, and performance of the news content model in this section. Our news content model seems to have learned its task very well. However, if this model was to be deployed for FND, then we need to see its performance in different settings, such as against adversarial approaches. We will explain our model by adopting methods from the coalitional game theory.

3.2 Explaining News Content Models

Although there exist models that can detect fake news with high accuracy such as our model, the reasons behind this performance is seldom investigated. There could be a number of reasons from dataset characteristics to wrong model architecture. Just by looking at our model's performance on a single dataset, one should not trust the model's spectacular performance, instead look at the reasons to understand why this is happening. Is model actually learning to distinguish between classes or is it learning some other thing that we do not want? The explanation of this model can also tell the model's type in terms of style-based FND approaches. Does our model capture features based on objectivity or deception? In order to answer questions like these and more, Lundberg and Lee, 2017 (2017) proposes a unified framework in which any model (except GNNs) can be explained.

We start this section by describing how Shapley Additive exPlanation (SHAP) framework helps explain a model. After examining SHAP framework, we utilize it for our model and examine the reasons behind its performance by conducting several experiments.

3.2.1 SHAP Framework

SHAP framework aims to explain a prediction of a single input instance x by computing Shapley values from coalitional game theory. These values correspond to the contribution of each input feature to the prediction made for x (Molnar, 2022). Specifically, we are interested in SHAP's ability to capture local explanations for our news content model. Fortunately, SHAP provides explanation frameworks for a variety of models, including DNNs. In order to handle DNNs, SHAP improves methods from DeepLIFT to produce DeepSHAP. However, DeepSHAP is not compatible with Transformer models. Thus, we use a model-agnostic explanation model provided in the SHAP framework, namely the partition explainer. We begin with introducing Shapley values which are then converted to Owen values via feature coalitions. We will examine how Owen values and Shapley values are connected.
As before, let us denote our news content classifier to be explained as f, and its input as x. We focus on local explanations built for a single prediction $\hat{y} = f(x)$ based on one input x. An explanation model g employs a simplified input x', which is the output of a mapping function $h_x(x') = x$ that maps x to x'. Each $h_x(x')$ is defined individually for each x. If two simplified inputs are similar $x' \sim z'$ then local methods aim to guarantee the approximation $g(z') \approx f(h_x(z'))$ (Lundberg & Lee, 2017).
The general framework is built upon *additive feature attribution methods*. These methods *have an explanation model that is linear of binary variables* (Lundberg & Lee, 2017).

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i$$

where $z' \in \{0,1\}^M$, M denotes the number of simplified input features, and $\phi_i \in \mathbb{R}$ feature relevance score of feature i (Lundberg & Lee, 2017). Note that ϕ_0 corresponds to the base value, and can be used as an indicator to detect the bias in a model. Additive feature attribution methods have three desired properties. The first, local accuracy states that *the explanation model $g(x')$ matches the original model $f(x)$ when $x = h_x(x')$* (Lundberg & Lee, 2017). The second property is missingness and it states that the features not included in the input should have no impact (Lundberg & Lee, 2017). The last property, consistency describes that if a simplified feature $x'_i \in x'$ have a greater impact in model

f' than the model f, then the feature x_i should be assigned at least the same or a higher value for its contribution (Lundberg & Lee, 2017).
The computation of each ϕ_i involves a similar process to the calculation of classic Shapley regression values (Lipovetsky & Conklin, 2001):

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!}[f_x(z') - f_x(z' \setminus i)]$$

where $z' \setminus i$ implies $z'_i = 0$, $|z'|$ is the number of non-zero elements in z', and $z' \subseteq x'$ denotes the set of vectors z' whose non-zero elements are a subset of non-zero elements in x' (Lundberg & Lee, 2017).
Given a hierarchy of features that define feature coalitions, the partition explainer recursively computes Shapley values through them and these Shapley values result in Owen values (Owen, 1977). Owen values are not the same as Shapley values, but if the coalition consists of the whole sentence or individual tokens, then Owen values and Shapley values become equal. A study (Casajus, 2009) also shows that Shapley values are the expected Owen values for all symmetric distributions on all partitions. Owen values will help us detect token groups (if any exist) that play a significant role in the prediction of a model. Moreover, Owen values are computationally very cheap which makes them more attractive.
We have summarized the foundations of the SHAP framework. As recommended in the source code documentation of SHAP[1], we use the partition explainer for our Transformer language model. Next, we will examine Owen values in action with our news content classifier.

3.2.2 Explaining News Content Classifier

Having introduced the required concepts, we now utilize them to explain our model using some real and fake news instances from the dataset. We adopt the partition explainer in order to understand how our Transformer model works. We specifically try to find its shortcomings so that we can decide whether it is a good idea to use this FND in real cases.
Initial Analysis. We took a fake news example from the train split of the dataset and analyzed which tokens are deemed most important. The explanation provided by the partition explainer for our first fake news instance is illustrated in Fig. 3.6. We can quickly observe that the base value is too high. In a normal setting, the base value should be around 0.5 for a binary classifier. Apart from a high base value, the model is actually capturing sensible features. For example, it can detect a non-formal narrative using words like "ride", "hood", "The genius protestors". Interestingly, punctuations

[1] https://github.com/slundberg/shap/

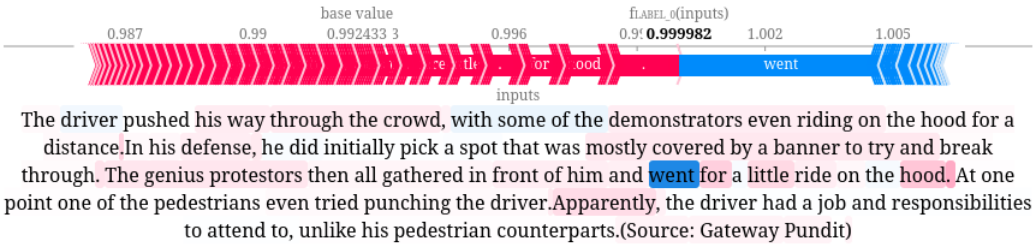


Figure 3.6: The explanation provided by SHAP partition explainer for our first fake news example. Example obtained by randomly selecting from our train split.

such as ".", "," and ". "were given particularly high importance even though they should not contribute to the context of the news that much.

We explain one more fake news instance from train split to see if the model behaves the same for that instance as well. The example is illustrated in Fig 3.7. We can quickly observe that the model was able to capture non-formal speech again. We further examine how the tokens and coalitions are constructed. We see that "not a U." and "S. citizen." are separate coalitions. This implies that the tokenizer evaluated "U." and "S." as separate tokens. This issue might have occurred due to the punctuations between "U" and "S". Typically, one would expect "U.S." to be tokenized as one token, but the tokenizer failed to capture this. Following this issue, we realized that the tokenizer splits capitalized words into smaller units, which suggests the tokenizer is not capable of understanding most capitalized words. This might not be good for our model. In addition, the base value is again too high.

We need to investigate why we obtain such high base values. Thus, to understand what our model thinks of the real news, we explain a real news instance from the train split. The example is shown in Fig. 3.9. We can easily see that the maximum share of Owen values go to three tokens: "(", "Reuters", ")". This suggests that our model has learned to distinguish news by source. In order verify this intuition, we run one more real news example through the partition explainer. The explanation for the second example is illustrated in Fig. 3.10.

For the second example, we observe a coalition of tokens "CA", "IRO", " ", "(", "Reuters", ")". This coalition is given the highest Owen value of 0.48. The distribution of Owen values among tokens is illustrated in Fig. 3.11. We can see that each token in the coalition was assigned almost equal Owen values. In two real news examples, the base value is really low. Again, one would expect around 0.5 for a binary classification setting. **Sensitivity Analysis.** Recall from 3.1.3, we reported that around 95% of real news instances contained the text "(Reuters)". And also we expressed our concerns

Figure 3.7: The explanation for the second fake news example. Observe that there are several coalitions between tokens such as "But on Friday he", "stopped telling The Big Lie" etc.

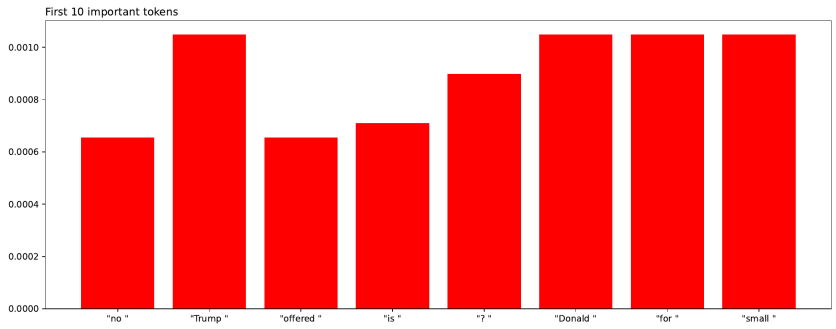
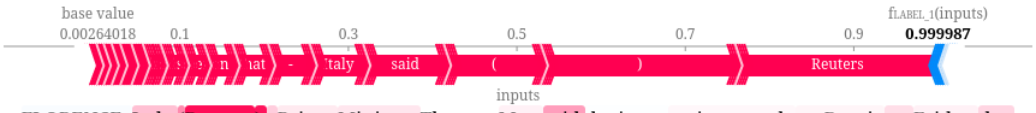
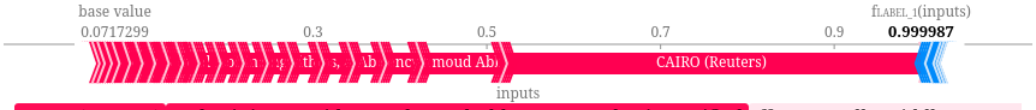


Figure 3.8: The bar plot for the Owen values of the tokens in the second fake news example. Since the news articles are long texts, we used a bar plot to visualize important tokens' Owen values.



FLORENCE, Italy (Reuters) - Prime Minister Theresa May said during a major speech on Brexit on Friday that membership of the European Economic Area would not suit Britain after it leaves the EU and would lead to friction.

Figure 3.9: The explanation for the first real news example.



CAIRO (Reuters) - Palestinian President Mahmoud Abbas on Monday intensified efforts to rally Middle Eastern countries against U.S. recognition of Jerusalem as Israel s capital, setting up talks with Arab leaders beginning in Cairo. Abbas met with President Abdel Fattah al-Sisi of Egypt, which has been a key broker in past peace talks with Israel and between fighting Palestinian factions. Abbas and Sisi agreed to continue high-level coordination and to use the widespread rejection of the U.S. move to maintain the rights of the Palestinians, a statement from the Egyptian presidency said. Palestinian Presidency Spokesman Nabil Abu Rdainah said there would be an Arab move to preserve the rights of the Palestinian people soon. Abbas will later head to Istanbul to give a speech at a meeting of the Organization of Islamic Cooperation (OIC). Arab states condemned U.S. President Donald Trump s Jerusalem decision last week, and vowed to press international bodies to take action against it. The Arab Parliament held an emergency meeting on the issue in Cairo on Monday, state news agency MENA reported. Sisi, after a separate meeting with visiting Russian President Vladimir Putin, said the U.S. moving its Israel embassy to Jerusalem would have dangerous effects on peace and security in the region . Abbas met the head of the Arab League in Cairo on Monday, local media reported. Arab foreign ministers held an hours-long emergency meeting at the weekend and vowed to seek a U.N. Security Council resolution rejecting the U.S. move, but gave few details on other measures they would take. The Palestinians hope for concrete action. Daring Palestinian and Arab decisions are required in the coming stage, which is very important, Abbas s spokesman Nabil Abu Rdainah told Palestinian official news agency WAFA. World powers have said the U.S. move will impede peace efforts in the decades-long Israeli-Palestinian conflict as anger spreads across the region. The Trump administration says it is still committed to the peace process. Abbas will not meet Mike Pence during the U.S. Vice President s visit to the region later this month, Palestinian Foreign Minister Riyad al-Maliki said on Saturday. Egypt s top Muslim and Christian religious leaders also said they would not meet Pence. Egypt has also brokered reconciliation deals between Abbas s Fatah party and Gaza-based Islamist group Hamas, which called for a new uprising against Israel last week. The planned handover of control of Gaza to the Fatah-dominated Palestinian Authority under the latest deal hit another delay on Sunday, with a Fatah official blaming obstacles without elaborating. Israel launched fresh air strikes in Gaza on Saturday in response to rocket fire from the enclave, where it fought a war in 2014 which killed more than 2,000 people, most of them civilians.

Figure 3.10: The explanation for the second real news example.

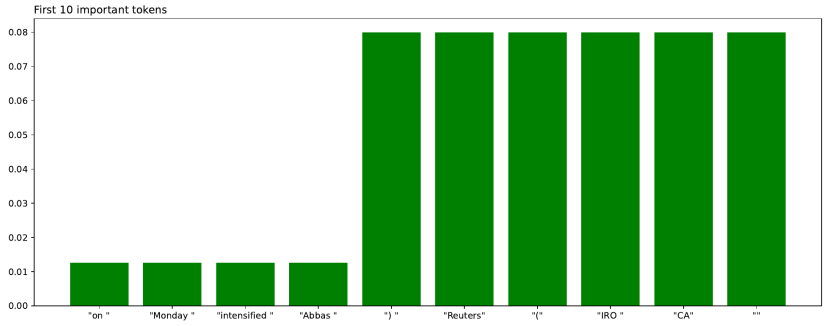


Figure 3.11: The bar plot of the 10 highest Owen values and their tokens for the second real news example.

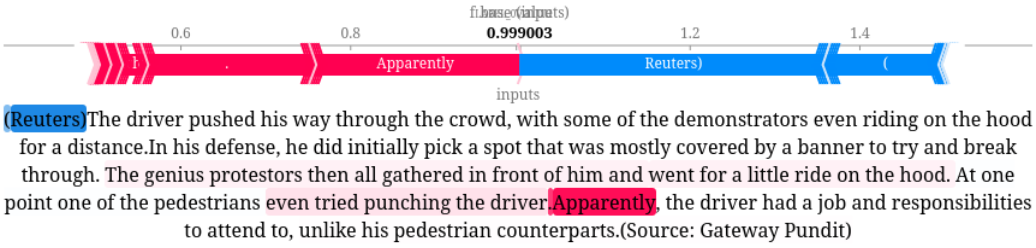


Figure 3.12: The explanation for the perturbed first fake news instance. This instance was predicted fake with more than 0.99 probability. We observe a very high base value as well. We see the negative effect of the token group "(Reuters)" on the force plot. Our intuition might be correct, however we might need a better alteration.

about model basing most of its prediction on a couple tokens. Now, let us consider that this model is deployed to detect fake news for a social platform. Once an adversarial realizes this weakness of the model, then they could simply append "(Reuters)" to the beginning of their fake news and simply evade the detection. For a concrete example, see Fig. 3.12.

When we observe Fig. 3.12, we can easily see that the adverse effect that (" and "Reuters" create is alleviated by assigning more Owen values to "Apparently" and ".". The model is still sure that this is a fake news instance. We might be required to adopt more complex adversarial approaches.

Knowing that the token group "(Reuters)" is never at the beginning of a real news article, we adopt more sophisticated changes in order to discover weaknesses of the model. We append a random string before "(Reuters)" and add the text " - " afterwards then append our fake news instance after this. This input sensitivity experiment shows

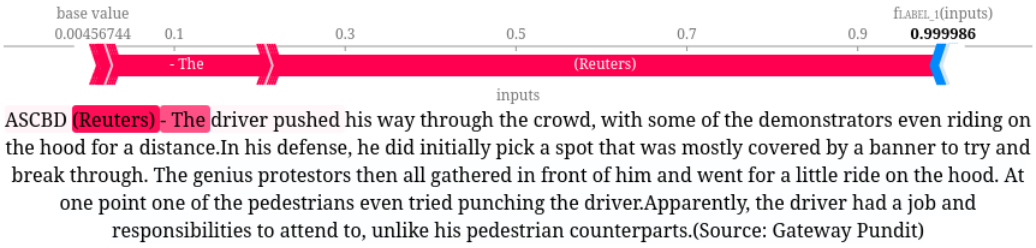


Figure 3.13: The explanation for the better perturbed first fake news instance. This instance was predicted real with over 0.99 probability. We can observe the dramatical probability change just by adding a couple words.

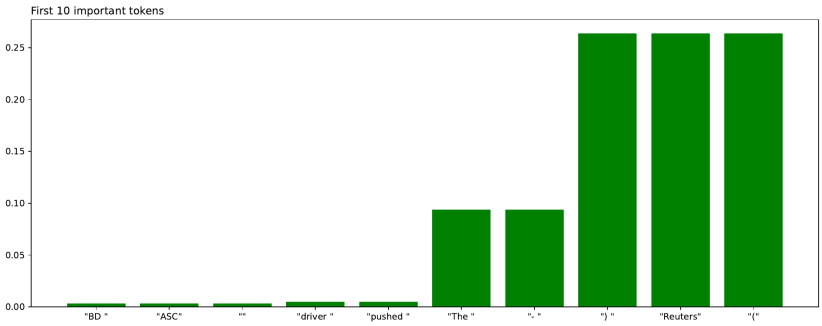


Figure 3.14: The bar plot for the Owen values of better perturbed first fake news instance. Even a non-sense text like "ASCBD" is assigned one of the highest importance score.

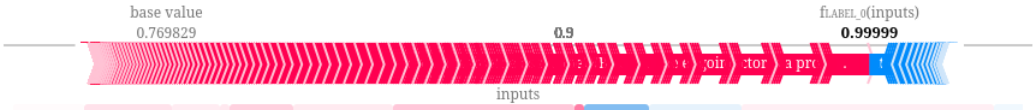
that our intuition was right for this instance. The explanation for this perturbation example is in Fig. 3.13.

The effect of model architecture. Now we can see the positional encodings' and attention mechanism's effect here. "ASCBD" is not a word and it is tokenized as "ASC" and "BD" (can be seen in the x-axis of Fig. 3.14), with both tokens being assigned relatively large Owen values. Moreover, when the position of "(Reuters)" changes from initial to a later position, the model directly changes its mind. One more thing that comes to mind is the length of the text, thus, we also applied the same technique to the second fake news example whose feauture relevance scores are better distributed. We got the same result for the second fake news example as well. We also observed a similar behavior when explaining other training instances. We also tried using characters instead of character sequences, and removing the space between "(Reuters)" and the initial character, the result was the same. This strongly suggests the model bases its predictions based on the existence and position of the token group "(Reuters)".

Accuracy	Precision	Recall	F1 score
98.92%	99.14%	98.84%	98.99%

Table 3.4: The performance metrics for the retrained news content model.

If we also consider the characteristics of DistilRoBERTa, we can see that the knowledge distillation process should be employed after the model is trained so that it can be deployed easily. We recommend to train a RoBERTa model with the cleaned dataset, then adopt the distillation process. This way more knowledge can be retained in the model.
We also conducted experiments with unseen data, but the results suggested the same outcome: clean the dataset and train again. Thus, we did not share the results from our unseen data experiment, as the results were the same with the ones we shared. In order to see the model's success without the upper hand the existence of "(Reuters)", we need to train our model with the same hyperparameters in Table 3.2, and the dataset should be changed such that the source part from the real news instances are removed.
Therefore, we train our model by eliminating the issue from the dataset. We report the performance of our model in Table 3.4. The performance stays the same, but what did our model learn this time? To comparatively illustrate this, we ran the same examples we have introduced again, and share the force plot of the second fake news example in Fig. 3.15. Now that we have a lower base value for fake news, we can observe patterns caught by model in more depth. First, the high Owen values for the coalitions of contractions with the missing apostrophe, such as "MSNBC s", "doctor s", "Let s", "there s", "O Donnell" and "isn t" is the first pattern we observe in this instance. Second, full stops still get a relatively high Owen value. Moreover, even though we decreased the base value of fake news, it is still high. To get the full understanding by comparing the explanation results, we also explain our second real news instance for our retrained model. The force plot of this explanation is shown in Fig. 3.15. We directly see the high base value, compared to Fig. 3.10, and also a better distribution of Owen values among the tokens and coalitions. The same detection for the lack of apostrophe is also captured in this explanation. Coalitions like "Israel s", "Trump s", "Abbas s", "President s", and "Egypt s" have a large negative Owen value. This suggests spelling mistakes such as leaving out the apostrophe is captured as informal by our model, even though these punctuations are removed in the tokenizer. There can be another reasons behind this behavior of the model, and can be uncovered by using explanation methods for other instances as well. We have seen that even though a model performs well this doesn't mean that this model has learned the task. Since our news content model tries to distinguish between fake and real news using the patterns provided in the



Donald Trump is a liar. This is pretty well-established by now. But on Friday he stopped telling The Big Lie, the one that allowed him to rise to political prominence and the one upon which so much of his initial support was based: Barack Obama is not a U.S. citizen.For years, Trump has harangued the President over his birth certificate, passport, college transcripts, and anything else the billionaire could use to delegitimize the first black President of the United States. The 2016 nominee was, in effect, the leader of the birthers, a movement he did not start but gladly latched onto, fueled, and drove forward as he and his cohorts ineffectively but annoyingly attacked Obama.Trump offered no apology. In fact, he spent only about 20 seconds admitting that President Obama was born in the United States, period. That was one lie. But what if Trump were to retract a lie per day? Could he do it? Would he even live long enough to do so? MSNBC s Lawrence O Donnell tackled the issue on Friday. Pointing out that Trump, via his doctor s shady letter The Donald allowed Dr. Oz to look at, was willing to lie about a single inch adding one inch to his actual height to narrowly avoid being classified as medically obese. Let s just assume Donald Trump is much healthier than he looks. Let s even assume he s taller and thinner than he really is, O Donnell says, deciding to estimate that Trump would live longer than his father. If Trump lived until 95 years old, he will be around for another 9,131 days. If Trump were to retract one lie per day for the rest of his life, he can retract 9,131 lies, O Donnell points out, but there s a problem. In retracting The Big Lie, he told two more: that Hillary Clinton started the birther rumors, and that Trump stopped them through his racist badgering of President Obama. If he does it the way he did it today, the lies of Donald Trump will live after him because as he retracted one lie today he told two more, O Donnell says. I think we know that Donald Trump isn t going to retract a lie every day. I think we know that Donald Trump is going to continue to lie every day. Then he brought it home: I think we know that there is nothing too small for Donald Trump to lie about, including that extra inch of height he decided he needed this week. He will lie about an inch. I think we know there s nothing too big for Donald Trump to lie about, nothing too big, nothing too solemn He will lie about anything. And so I think we know that long after he s gone, long after his name has disappeared from buildings, the lies of Donald Trump will live after him. Watch it below:Featured image via screengrab

Figure 3.15: The explanation of the second fake news instance for the retrained model. Observe the lower base value and better distribution of Owen values.

Figure 3.16: The explanation of the first fake news instance for the retrained model. Observe the lower base value and better distribution of Owen values.

dataset, it is crucial to have a dataset that will not create the bias as we observed. We used explanation techniques to uncover this issue with our model. We have seen the significance of adopting explanation methods to understand the behavior of our model. Thus, we have fulfilled **RO2** for news content models.
Also to fulfill **RO3** for the partition explainer, we report the understandability of the explanations provided by partition explainer. Although we tried to explain long articles, the explanations were understandable, and the interactive behavior that allows to view Owen values within coalitions helped us understand the impact of each token individually. With the help of bar plots, we were able to see the importance distribution between the first 10 tokens. This allowed us to pinpoint the issue with the model even though it was performing very well. By utilizing simple visualization techniques, we were able to understand what the model has learned from the dataset. We used Python 3.7 (van Rossum, 1995), PyTorch 1.11.0 (Paszke et al., 2019), Transformers 4.19.2 (Wolf et al., 2020), SHAP 0.40.0 (Lundberg & Lee, 2017), and Matplotlib 3.5.1 (Hunter, 2007) for our implementations, visualizations and experiments.
News content is highly related with the veracity of the news, nonetheless, we might need more information to distinguish fake and real news. In the next chapter, we introduce mixed approaches for FND in which we utilize news content data as well as social context data. Having this fusion of information proves to be very effective as shown in Dou et al., 2021, models that use news content with social context are more successful than the models that utilize social context only.

4 Mixed Approaches for Fake News Detection

4.1 Mixed Approaches

As discussed in Section 2.1, social media's interconnected nature leads to the fast dissemination of fake news. When a news piece is shared, it propagates through social media by means of friendship networks. These networks can be exploited to gather information about how fake news pieces spread. Moreover, users' historical information can prove effective when analyzing whether a news piece is real. This assumption stems from the psychological facts we discussed in Section 2.1. To recap, if a user is sharing fake news most of the time, then it is likely that the following news piece they will share will also be fake. Usually, fake news pieces are shared most within echo chambers, which gives rise to the quick spread of fake news.
There exist many different approaches to social context models. However, creating a dataset for social context models is a challenging task, as the amount of data to be collected can grow exponentially. Thus, we selected a dataset that provides us with the social context information and news content. A graph dataset for our mixed approach model, *User Preference-aware Fake Detection* (UPFD) (Dou et al., 2021), builds a propagation tree for each news using the users who retweeted the news. Nevertheless, in order to build a model that takes graphs as input, we can no longer rely on standard deep learning approaches because the graph data has a different structure. Thus, graph data is challenging to represent in the euclidean domain. Graphs hold node information as well as edge information between nodes, allowing to store rich relational data (Zhang et al., 2018).
In this section, we lay out the foundations of graphs and GNNs. Then we investigate UPFD, and report our findings. Lastly, we introduce our choice of mixed approach model and share its performance from our experiments.

4.1.1 Overview of Graphs

In this section, we introduce definitions to cover graphs and different types of GNNs. However, we do not discuss each model type in detail due to GNNs' extensive back-

ground. Thus, we do not provide a notation for this section, but we give mathematical definitions when required.
Graphs are an example of *non-euclidean* data, meaning that in contrast to *euclidean* data, they can represent complex relations and entities more accurately than one or two-dimensional representations. Non-euclidean data used in GDL can be grouped into grids, graphs, groups, geodesics, and gauges also called the 5G's of *Geometric Deep Learning* (GDL) (Bronstein et al., 2021). We are interested in one G only, graphs.

Definition 4.1.1 (*Geometric Deep Learning (GDL)*)**.** A class of deep learning that aims to build models that can learn to predict on the non-euclidean domain.

GDL is an extensive field covering various techniques that can be applied to the non-euclidean domain. Due to its complex nature, we do not provide a rigorous background on GDL. For a detailed background on GDL, we refer the reader to an extensive study on GDL (Bronstein et al., 2021). We only discuss parts related to GNNs that we utilized. First, we define a graph.

Definition 4.1.2 (*Graph*)**.** A graph is a non-euclidean type of data that represents the relations between entities.

From the perspective of individual node connections, graphs can be categorized into two classes, *directed* and *undirected*. Directed graphs have direction information in their edges, i.e., the information flows strictly from one node to another. On the other hand, undirected graphs do not have this limitation. The information flow is bidirectional. Since we are only interested in undirected graphs like UPFD, we give preliminaries for an undirected graph.
Following standard notation on graphs, we define an undirected graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ as the set of nodes and $\mathcal{E}$ as the set of edges in a graph $\mathcal{G}$. We say that an edge (v_i, v_j) exists between two nodes v_i and v_j. Moreover, from the perspective of a graph's connectedness, two types of graphs exist: *cyclic* or *acyclic*. Simply put, cyclic graphs have cycles in them, i.e., they allow at least one node v_i to have a series of different edges that creates a cycle back to node v_i. In contrast, acyclic graphs do not contain cycles. A concrete example for undirected acyclic graphs is trees, which strictly have one edge between two nodes.
When we look at graphs from node and edge type, we classify graphs as *homogeneous* and *heterogeneous* graphs. Homogeneous graphs have nodes and edges of the same type, whereas heterogeneous graphs have different types of nodes and edges. An example of a homogeneous graph is a social network in which the nodes are users, and the edges represent the friendship between two users. If we modify this social network into a more detailed version in which we choose to represent another relationship between users, such as colleagueship, then we would have a heterogeneous graph because there

would be two types of edges in the graph.
Finally, if we consider graphs from a temporal aspect, we observe two kinds of graphs, *static* and *dynamic*. Static graphs stay the same over time. Their topology or features do not change. In contrast, dynamic graphs' features and topology vary over time, thus making time an important factor when working with dynamic graphs.

4.1.2 Graph Neural Networks

GNNs are neural networks that can take graphs as input and produce predictions at three levels: *node-level*, *edge-level*, and *graph-level*. Node level tasks include node classification, node regression, and node clustering. Node classification aims to categorize nodes into classes. Node regression deals with predicting continuous value for each node. Lastly, node clustering aims to group nodes into several clusters. Edge-level tasks consist of edge classification and link prediction. Edge classification tries to categorize an edge, and link prediction seeks to find whether there is an edge between two nodes. Graph-level predictions are graph classification, graph regression, and graph matching. In all these settings, the model needs to learn graph representations (J. Zhou et al., 2018).
In our experiments, we used a model that uses a convolutional layer called GraphSAGE (Hamilton et al., 2017). We also worked with a convolutional attention layer called *Graph Attention* (GAT) (Veličković et al., 2017). Nevertheless, due to the length of experiments and insight collection, we opted not to report our findings from our study with GAT. In order to cover the background of GNNs, we outline both along with many others. We first examine the general framework.
J. Zhou et al., 2018 defines three modules in a generic GNN model: the *propagation module*, *sampling module*, *pooling module*. The propagation module deals with information propagation between nodes so that the aggregated information can represent features and topology of the graph. Propagation modules usually employ a *convolution operator* or *recurrent operator* to aggregate information from neighbors of nodes. These aggregated values are then used to update the representation of the nodes, edges, and graph. Additionally, these modules employ a skip connection that helps to alleviate the over-smoothing problem by collecting previous representations of nodes. A sampling module is used when the input graph is too large to handle propagation and before the propagation module. Lastly, pooling modules are employed when extracting information to represent high-level graphs.
We will primarily deal with the propagation module and the operations within. We do not discuss recurrent operations on graphs as their background is extensive and out of the scope of this thesis. We refer the reader to an extensive study by J. Zhou et al. (2018) for a mathematical background on convolutions and other operations in

the spectral domain. We will outline the behavior of convolutions on graphs in order to give sufficient knowledge for the models we used.

Convolutions on graphs. The idea of convolution operators is to generalize convolutions from another domain to spectral domain. In general there are two types of convolutional operations: *spectral* and *spatial*.

The first, spectral approaches, are based on graph signal processing and defines its convolutional operators in the spectral domain (Shuman et al., 2013). Spectral methods initially transform a graph signal x to the spectral domain by the graph Fourier transform $\mathcal{F}$, then the convolution operation is applied. The output from convolution is transformed back with the inverse Fourier transform $\mathcal{F}^{-1}$. Now we summarize the mathematics behind this approach in order to convey the UPFD classifier's characteristics. Graph Fourier and inverse graph Fourier transform are defined as,

$$\mathcal{F}(x) = U^T x$$

$$\mathcal{F}^{-1}(x) = Ux$$

where U is the eigenvalue matrix of normalized graph Laplacian $L = I_N - D^{\frac{-1}{2}} A D^{\frac{-1}{2}}$ with I_N the identity matrix of dimension N, D as the degree matrix and A as the adjacency matrix of the graph. L is a real symmetric positive semi-definite matrix which helps us with factorization $L = U\Lambda U^T$ where Λ denotes the diagonal eigenvalue matrix (J. Zhou et al., 2018). Now that we can convert our graph signal x to spectral domain and back, following Mallat (1999) and J. Zhou et al. (2018), we can define the convolution operation $\star$ on x.

$$\begin{aligned} g \star x &= \mathcal{F}^{-1}(\mathcal{F}(g) \bigodot \mathcal{F}(x)) \\ &= U(U^T g \bigodot U^T x) \end{aligned}$$

where $\bigodot$ stands for element-wise multiplication and $U^T g$ is the convolution filter in the spectral domain. This can be further simplified into the basis function of spectral methods:

$$g_w \star x = U g_w U^T x$$

where g_w is a diagonal learnable matrix. Essentially, all spectral methods use a convolutional filter g_w but their choice of design creates a variety of approaches that are built upon each other. We only discuss the ones necessary for the UPFD classifier. The main idea of modern convolutional operators come from approximating g_w with k-th order Chebsyshev polynomials. GCN adopts $k = 1$ to avoid overfitting. Moreover, it introduces a renormalization trick to solve the exploding/vanishing gradient problem (Kipf & Welling, 2016). Further works like AGCN (Li et al., 2018) have employed a similar

approach. Additionally, GCN is employed to an extent in spatial approaches.
The second, spatial approaches, focus on the graph structure. They define convolutions directly on graph topology. Spatial approaches can be grouped into *basic*, *attention-based*, and *framework*. Basic spatial approaches define convolution operations on the neighborhoods of different sizes. Neighborhoods are defined based on nodes as follows $\mathcal{N}_v$ for a node v. There exist several basic spatial approaches, such as the diffusion CNN (DCNN) (Atwood & Towsley, 2016), which describes the neighborhhood for nodes using transition matrices, and the learnable GCN (LGCN) (Gao et al., 2018), which utilizes CNNs as aggregators by applying max pooling on neighborhhood matrices.
Another example, GraphSAGE uses an inductive learning approach to sample then aggregate features from a node's local neighborhhood. GraphSAGE uniformly samples a fixed-size collection of neighbors then aggregates this collection to produce embeddings for the graph (Hamilton et al., 2017). More precisely, let us assume that we have learned the parameters of K aggregator functions denoted as AGG_k, and a set of weight matrices W^k where $\forall k \in \{1, \dots K\}$. k can be referred to as layer or search depth. Then for each k we go through all nodes $v \in \mathcal{V}$ and apply an aggregation then an update. Concretely, at each k and at each v, let h_v^{k-1} denote the hidden state of a node v at layer $k-1$. Following the same notation, we refer to the hidden state of a node's neighborhood $\mathcal{N}_v$ at layer k as $h_{\mathcal{N}_v}^k$. GraphSAGE formalizes its aggregation and update operation at layer k as:

$$h_{\mathcal{N}_v}^k = \text{AGG}_k(\{h_u^{k-1}, \forall u \in \mathcal{N}_v\})$$

$$h_v^k = \sigma(W^k[h_v^{k-1}||h_{\mathcal{N}_v}^k])$$

where $||$ denotes concatenation of two vectors, $[h_v^{k-1}||h_{\mathcal{N}_v}^k]$ concatenated vector and σ a non-linear activation function. GraphSAGE employs three different aggregators: *mean aggregator*, *LSTM aggregator*, and *pooling aggregator*. Mean aggregator collects sampled neighborhood information and takes the element-wise mean of the vector set $\{h_u^{k-1}, \forall u \in \mathcal{N}_v\}$. The inductive version also includes previous layer representation of node v, h_v^{k-1}. LSTM aggregator uses an LSTM to collect neighborhood information. LSTMs are more expressive than mean aggregators. However, since LSTMs are not permutation invariant, i.e., they process their inputs sequentially, they need a modification to work with unordered sets. Finally, pooling aggregator first independently feeds each neighbor's vector to an FCN, then applies a pooling operation on the output.
The third, *attention-based approaches*, utilizes the same attention mechanism introduced in 3.1.2 by following Bahdanau et al., 2014. A recent work Graph Attention Networks (GAT) (Veličković et al., 2017) proposes to integrate attention into the propagation step by computing the hidden state for node v at layer k. The last convolutional spatial approaches cover general frameworks that aims to integrate multiple different models

into one framework. A mixture model MoNet was proposed by Monti et al., 2016 (2016), which is a spatial framework for unifying models such as GCN (Kipf & Welling, 2016), DCNN (Atwood & Towsley, 2016) and many more in non-euclidean domain.
One more thing we need to investigate is graph sampling in order to cover everything that was utilized in this thesis. GNN models suffer from an issue called *neighbor explosion* which stems from having massive sizes of supporting neighbors from all previous layers as the number of layers increases J. Zhou et al., 2018. Likewise, as graph size increases, we will encounter the same problem. Sampling is used to solve this issue and it can be done on three levels: *node sampling*, *layer sampling*, and *subgraph sampling*. Node sampling creates a subset using the neighborhood set $\mathcal{N}_v$ of the node's v. As discussed previously, GraphSAGE (Hamilton et al., 2017) employs node sampling. Layer sampling takes a different approach and it selects a subset of nodes for aggregation. Lastly, subgraph sampling deals with the graph as a whole. One approach is to use clustering algorithms to obtain these subgraphs (Chiang et al., 2019). Another is to sample nodes and edges from graph to create a subgraph (Zeng et al., 2019).

4.1.3 Dataset and Model

Our choice of a dataset, UPFD (Dou et al., 2021), utilizes the dataset FakeNewsNet (Shu et al., 2018). From FakeNewsNet, the authors of UPFD build a graph dataset by collecting user preference information from historical posts and social context data with Twitter Developer API (Twitter, 2022). They further employ textual embedding techniques to encode historical posts and news content. We shall discuss how these operations are conducted in detail. UPFD has two preparation levels: *endogenous preference encoding* and *exogenous context extraction*. After preparation, a technique called *information fusion* is applied using GNNs. The framework is shown in Fig 4.1
Endogenous preference encoding deals with users' historical posts and news content using the news content and social engagement data in FakeNewsNet. With social engagement data, the authors collect 200 historical tweets of each user who have retweeted the news in FakeNewsNet. Summing up to almost 20 million tweets, this collection procedure of historical tweets for each FakeNewsNet news instance is as follows (Dou et al., 2021):

- Collect user ids who have retweeted.
- For each user, collect the 200 most recent tweets.
- For inaccessible (suspended or deleted) users, use randomly sampled tweets from accessible users who engage in the same news piece. This step also increases the effectiveness of the exogenous encoder.

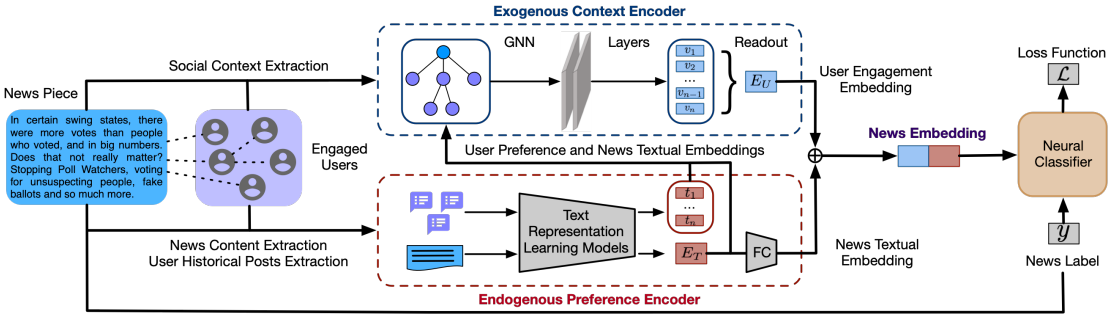
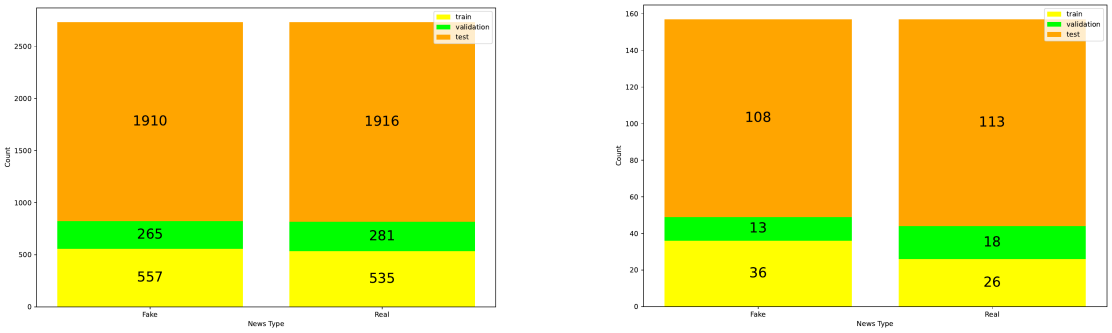


Figure 4.1: The framework of UPFD. Image obtained from (Dou et al., 2021)

- Finally, remove special characters such as "@" and URLs before applying textual techniques.

Now that we have news content and users' historical information, we encode these texts using text representation learning techniques. Still following Dou et al., 2021, the authors define two different settings for creating endogenous preference encoding. The first setting, *spacy*, deals with obtaining the representation of users' historical data and news content with spaCy (Honnibal & Montani, 2018) pretrained 300-dimensinal vectors for 680K words. For each user's historical post (and similarly for each news piece), if a word appears in the text, we include the vector for that word for the final calculation in which all included vectors are averaged. As a result, we obtain a 300-dimensional vector for each user's historical information and news piece. The second setting, *bert*, uses the BERT-Large model to encode news and historical user data. Using bert-as-a-service (Xiao, 2018), the authors encode the news content with a maximum input sequence length of 768. The authors opt for a different setting to encode historical information since 200 tweets cannot be processed as a single document due to BERT's maximum input sequence length limitation of 768. It is important to note that the paper of UPFD (Dou et al., 2021) states the maximum input sequence length as 512. However, in the dataset itself, the maximum sequence length is 768 (Team, 2022). Keeping in mind that tweets are usually short texts compared to news pieces, the authors use a maximum input sequence length of 16 for each historical tweet, then average all 200 tweets to obtain the final representation of users' historical data. This representation is of the same dimension as the root news node. Note that these text representation vectors are node features of a hierarchical tree-structured graph whose root node represents the news content, and the children of the root node represent the users who have retweeted.

Exogenous context extraction uses retweet information to build the previously mentioned hierarchical tree. The authors adopt a similar procedure used by Han et al.



(a) Gossipcop (b) Politifact

Figure 4.2: UPFD dataset distribution by label and split (train: 20%, val: 10%, test: 70%)

(2020),Monti et al. (2016), and Shu et al. (2020). In detail, the authors define a news piece as v_1 and the set of users who retweeted v_1 as $\{v_2, \ldots, v_n\}$ ordered by time. The rules are defined for this process to cover edge cases as well:

- Estimate that a news piece propagates to user v_i from user v_j with the latest timestamp, if any user from the set $\{v_j | j \in \{2, \ldots, n\} \setminus \{i\}\}$ retweeted the news piece before the user v_i. This conservative assumption is based on the fact that the latest tweets are presented to the user first in the Twitter app (Dou et al., 2021).
- If user v_i does not follow any users in the set $\{v_1, \ldots, v_n\} \setminus i$, i.e., all retweeters and the news source except the user itself, the authors approximate the spreader of that news piece to the user v_i as the user with most followers in the set. This approximation is based on the phenomenon that the tweets from accounts with more followers have a higher probability of being retweeted or viewed.

After this procedure, we have our hierarchical tree for each piece of news obtained from FakeNewsNet, which employs two sources: Politifact and Gossipcop. Thus, we evaluate these datasets separately within UPFD. The distribution of the instances with respect to train/validation/test split and label is provided in Fig. 4.2.
Information fusion is achieved via GNNs. As discussed in 4.1.2, GNNs can encode graphs by maintaining the node features and structural information. From this point on, we need to utilize the models we introduced with a classification setting. Classification is done for each graph representing the diffusion tree of the news piece (Dou et al., 2021). We adopted two models from this work's ablation study, the first one is based on GraphSAGE (Hamilton et al., 2017), and classification is handled with an FCN. We use BERT as the encoder since the best performance is obtained via that according to the experiments by Dou et al. (2021) and the experiments we conducted. We call this

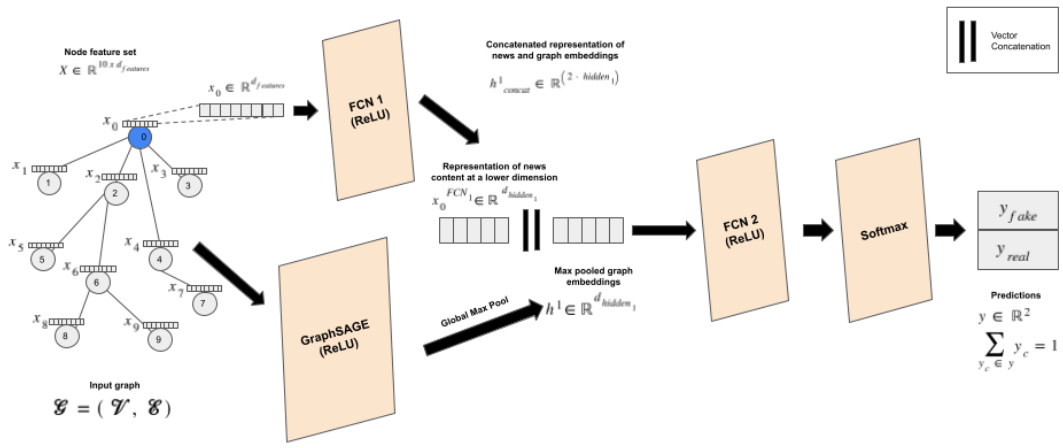


Figure 4.3: UPFD classifier model pipeline. We used an example input graph with 10 nodes. $d_{hidden_1} = 128$ stands for hidden layer dimension of FCN 1, accordingly, $d_{hidden_2} = 128$ for the hidden dimension of FCN 2, and $d_{feature} = 768$ for feature dimension.

model the UPFD classifier. It is illustrated in Fig 4.3.
After obtaining the graph encodings, the authors suggest concatenating a representation of news content with graph embeddings. More precisely, the feature vector of the root node is fed to an FCN to obtain a lower-dimensional representation which is then concatenated with graph embeddings. The concatenated vector is then fed to another FCN and the softmax layer for classification. With the setting illustrated in Fig 4.3, the optimizer as Adam with default β_1 and β_2 and hyperparameters provided in Table 4.1, we obtain 95.49% and 83.16% accuracy from Gossipcop and Politifact averaged by ten runs, respectively. All other statistics are provided in Table 4.2.
From the model architecure, we can easily see that the UPFD classifier's social context part is propagation-based and utilize a homogeneous network. On the other hand, the news content part is unknown, but it can be uncovered using explanation techniques for GNNs. Before we explain the UPFD classifier, we outline of the features of UPFD. To recap, UPFD is constructed with the news pieces and social context data obtained from FakeNewsNet (Shu et al., 2018). Shu et al. (2020) utilizes the data from FakeNewsNet to create two levels of propagation networks: *macro-level* and *micro-level*. For each

Activation function	ReLU (Nair & Hinton, 2010)
Batch size	128
Epochs	35
Learning rate	0.01
Hidden layer 1 size	128
Hidden layer 2 size	128
Weight decay	0.01

Table 4.1: Hyperparameters of UPFD classifier.

Dataset	**Accuracy**	**Precision**	**Recall**	**F1 score**
Gossipcop	95.49%	94.95%	96.17%	95.50%
Politifact	83.16%	84.05%	82.19%	83.25%

Table 4.2: UPFD classifier performance metrics averaged over 10 runs.

level, Shu et al. define and statistically test the significance of a set of features. We want to see if some of those features are adopted by the UPFD classifier. Briefly, macro-level networks are constructed to track the diffusion of a news piece via retweets, whereas micro-level propagation networks are built using the replies to retweets. UPFD is the macro-level propagation network. Thus our focus is on the features of the macro-level propagation network.
The features are macro-level propagation networks are categorized into two: *structural* and *temporal*. Structural features capture the patterns in the global dissemination of a news piece (Shu et al., 2020). These features include:

- (SF_1) *Tree Depth*: Indicates how far the news piece has spread through the propagation network (Shu et al., 2020).
- (SF_2) *Number of nodes in macro-network*: Indicates how many users have shared the news piece (Shu et al., 2020).
- (SF_3) *Maximum outdegree*: Helps reveal the tweet/retweet with the most influence in the propagation network (Shu et al., 2020).
- (SF_4) *Number of cascades*: The number of tweets that post the original news article (Shu et al., 2020).
- (SF_5) *Depth of node with maximum outdegree*: Indicates steps of propagation required for a news piece to be retweeted by an influential user (node) whose post is retweeted by more users than any other users' retweet (Shu et al., 2020).

- (SF_6) *Number of cascades with retweets*: Indicates the number of tweets that shared the original news article and retweeted at least once (Shu et al., 2020).
- (SF_7) *Fraction of cascades with retweets*: Corresponds to SF_6/SF_4 (Shu et al., 2020).
- (SF_8) *Number of bot users retweeting*: Captures the number of bot users that retweeted that news piece (Shu et al., 2020).
- (SF_9) *Fraction of bot users retweeting*: Corresponds to SF_8/SF_2 (Shu et al., 2020).

First of all, the feature set is not restricted to these features. One can derive more features from this dataset. We will investigate some of these features in order to understand whether we can capture similar features in our explanations. We do not have access to the information of bot users in UPFD. Thus we drop the analysis of features SF_8 and SF_9. In addition, we did not analyze temporal features, as obtaining them from the explanation proved to be a long process. We now discuss which features' average values are found to be significant in the statistical tests conducted by Shu et al. (2020). The first structural feature SF_1 is found to be consistently different for real and fake news instances in both datasets, Gossipcop and Politifact, under the statistical t-test (Shu et al., 2020). Concretely, this finding indicates that the tree depth of the fake news pieces is larger than that of real news pieces. Next, another simple feature SF_2 is also shown to be consistently different. To specify, fake news instances have fewer nodes than real news instances. Although helpful, without additional information, this feature is too broad. Thus, we look at the next structural feature SF_3. It is found by Shu et al. (2020) that this feature is consistently different for fake and real news in Gossipcop under t-test. More precisely, it is found that the maximum outdegree is bigger in fake news instances of Gossipcop. Considering this observation, we now proceed to more complex features.
When we look at features that analyze more subtle properties of the dataset, such as SF_5 and SF_7, it is reported that these features are also consistently different for fake and real news instances from both datasets Politifact and Gossipcop (Shu et al., 2020). According to the study, SF_5 is discovered to be deeper in fake news instances, i.e., the finding suggests fake news pieces reach the users from another user who retweeted the original fake news instance. The other feature SF_7 investigates how many of the first retweeters of the original post were retweeted. The t-test tells us that this value is different in real and fake news instances in both datasets. Specifically, the value of SF_7 is higher for fake news than for real news.
We have introduced the dataset and the model we have employed for FND in mixed approaches. We will analyze the model's behavior on the dataset. We adopt GNNExplainer (Ying et al., 2019) as our explanation tool and try to uncover the patterns

the model has learned. We shall also discuss some limitations we encountered while conducting these experiments.

4.2 Explaining Graph Neural Networks

Except GNNs, many neural networks can be explained using various tools such as SHAP, LIME, LRP etc. GNNs make their predictions by aggregating information from nodes, edges and graph topology. There can be cases where an edge is plays a significant role in the node prediction only if it is considered with another edge (Duvenaud et al., 2015). Therefore, we can not model separate contributions of node, edge, and graph features in a linear manner.

4.2.1 GNNExplainer

GNNExplainer is a post-hoc model-agnostic explainer tool that aims to provide information about the effects of node features, edges, and graph structure on the prediction. More precisely, let us denote the node v's computation graph as $\mathcal{G}_c(v)$, the binary adjacency matrix as $\mathcal{A}_c(v) \in \{0,1\}^{n \times n}$ with n: the number of nodes, and the feature set as $X_c(v) = \{x_j | v_j \in \mathcal{G}_c(v)\}$. Then we say that a GNN model f learns a conditional distribution $P_f(Y|\mathcal{G}_c, X_c)$ with $Y \in \{1, \ldots, C\}$ as a random variable that stands for class probabilities of nodes. Then in probabilistic terms, the prediction is denoted as $\hat{y} = f(\mathcal{G}_c(v), X_c(v))$ which suggests that the prediction depends on three parts. Therefore, GNNExplainer takes into account the GNN model f, the graph information $\mathcal{G}_c(v)$, and the node features $X_c(v)$ (Ying et al., 2019).
Simply put, GNNExplainer finds a subgraph $\mathcal{G}_S(v) \in \mathcal{G}_c(v)$ and node feature subset $X_S = \{x_j | v_j \in \mathcal{G}_S\}$ that are important for a given prediction $\hat{y}$. The subgraph that explains the prediction best is obtained by maximizing the *mutual information MI* between the prediction Y and the pair of subgraph and subset of node features $(\mathcal{G}_S, X_S)$ (Ying et al., 2019).

$$\max_{\mathcal{G}_S} MI(Y, \mathcal{G}_S, X_S) = H(Y) - H(Y|G = \mathcal{G}_S, X = X_S)$$

where $H(Y)$ is the marginal entropy of Y and fixed for a trained GNN, and $H(Y|G = \mathcal{G}_S, X = X_S)$ stands for the conditional entropy. Since $H(Y)$ is fixed, the authors reformulate the objective function as a minimization function that looks for a subgraph $\mathcal{G}_S$ that minimizes the uncertainty of the GNN model f while using a subset of node features X_S, thus maximizing the probability of $\hat{y}$. However, this formulation brings computation issues as large dense graphs can have massive amounts of candidate

explanation subgraphs for a prediction $\hat{y}$. Thus, the authors enforce a constraint on the fractional adjacency matrix $A_S \in [0,1]^{n \times n}$ of the subgraphs which are now treated as random variables $\mathcal{G}_S \sim \mathcal{G}$. With convexity assumption and upper-bound from Jensen's inequality, the minimization problem is rewritten as (Ying et al., 2019),

$$\min_{\mathcal{G}} = H(Y|G = \mathbb{E}_{\mathcal{G}}[\mathcal{G}_S, X = X_S]).$$

Finally, to estimate $\mathbb{E}_{\mathcal{G}}$, the authors employ mean-field variational approximation and decompose $\mathcal{G}$ into a multivariate Bernoulli distribution. This allows to estimate the expectation in terms of mean-field approximation, thus obtaining A_S whose (i,j)-th element denotes the expectation of the existince of the edge (v_i, v_j). The final objective is defined as (Ying et al., 2019),

$$\min_{M} - \sum_{c=1}^{C} \mathbb{1}[y = c] \;\; log(P_f(Y = y \mid \mathcal{G} = A_S \bigodot \sigma(M), X = X_S))$$

where $M \in \mathbb{R}^{n \times n}$ is the mask to learn, $\bigodot$ is element-wise multiplication, and σ is the mapping function of mask to values to interval $[0,1]^{n \times n}$. For large graphs however, a threshold might be required to remove low values in the mask. We will propose two approaches to effectively handle this only for explanations of UPFD classifier (Ying et al., 2019).
Besides the edge mask M, as mentioned previously, GNNExplainer also learns a node mask $F \in \{0,1\}^{d}_{feature}$ through a binary feature selector that is learned by maximizing MI in the first equation. More formally, the explanation $(\mathcal{G}_S, X_S)$ is optimized together to maximize MI:

$$\max_{\mathcal{G}_S, F} \;\; MI(Y, (\mathcal{G}_S, F)) = H(Y) - H(Y \mid \mathcal{G} = \mathcal{G}_S, \; X = X_S^F)$$

where $X_S^F = \{x_j^F | v_j \in \mathcal{G}_S\}$ and $x_j^F = [x_{j,t_1}, \ldots, x_{j,t_k}]$ for $F_{t_i} = 1$ denotes the node features that are not masked out by F (Ying et al., 2019) . So far, the explanation method is defined for node classification setting. However, when the task is at graph-level, the method needs to be extended. Ying et al. (2019) suggest to unionize the adjacency matrices A_S for all nodes in the graph to be explained. The aggregation of node embeddings, however, restricts the explanation to a subset where the resulting subgraph must be connected. Thus, the threshold for the edge mask is set to a very low value or zero. This allows for a larger connected subgraph as an explanation (Ying et al., 2019).

4.2.2 Explaining UPFD Classifier

In Section 4.1, we have introduced our UPFD classifier, a GNN that uses GraphSAGE (Hamilton et al., 2017) as the graph encoder, two FCNs and softmax for classification of news propagation trees. We now investigate the reasons behind the model's

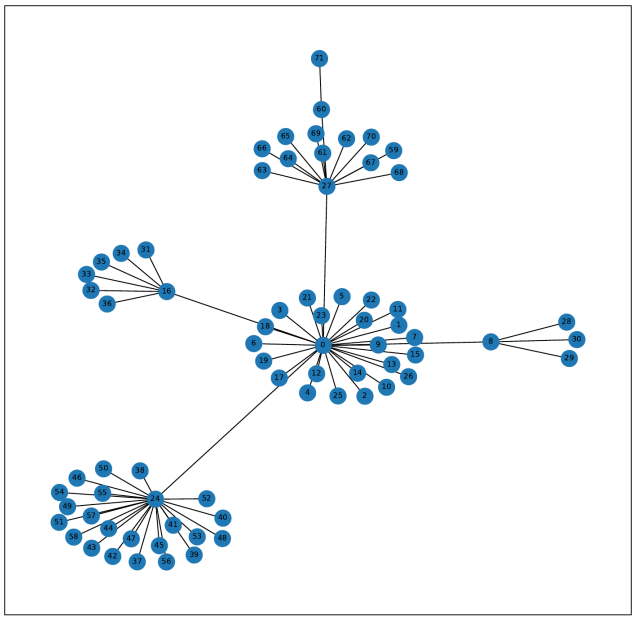


Figure 4.4: A fake news instance in UPFD-Politifact that exhibits echo chamber behavior. 0-th node represents the root node with news embeddings. We will refer to this fake news example as the *echo chamber example*.

performance on the UPFD-Politifact dataset by employing GNNExplainer as our explainer. Note that this model was trained only on the UPFD-Politifact dataset, thus data in UPFD-Gossipcop is unseen to this model. The explanations we obtain for our classifier's performance on UPFD-Politifact dataset might provide us with useful information that enables us to improve the model.

First, to clarify how the explanations should be interpreted, let us consider an example where we obtain a single explanation $(\mathcal{G}_S, X_S)$ for a graph (news propagation tree). We interpret the edge masks M of $\mathcal{G}_S$ as the importance scores of edges between nodes. Therefore, an edge (v_i, v_j) with a mask value m_{ij} lower than a fixed *threshold* will be dropped, effectively dropping the connected nodes v_i and v_j if (v_j, v_i) does not exist or also have a smaller value than the mask value m_{ji}.

We examine whether our classifier learned a similar intution that we discussed in 2.1.1. To recap, fake news tend to create an "echo chamber" effect. In the echo chamber example in Fig. 4.4, it is trivial to observe some echo chambers created by nodes 24, 27, 16, 8 from biggest to smallest respectively. Furthermore, the initial spreaders of this example are the followers of the publisher of root node. We look for features, introduced in 4.1.3, that are captured by the UPFD classifier. First, let us observe the

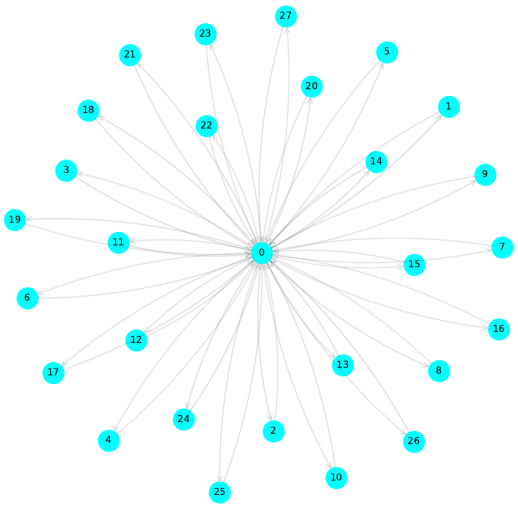


Figure 4.5: The explanation provided for the UPFD classifier and the echo chamber example for its root node. This instance was predicted as fake news with a probability of 0.9912. Contains 28 nodes.

explaining subgraph $\mathcal{G}_S$ in Fig 4.5 for the echo chamber example.
Even though UPFD has undirected edges, we have obtained an explanation with bidirectional directed edges. Undirected edges are interpreted as bidirectional edges, this is why GNNExplainer provided us with two edges per node pair. Additionally, there exist self-loops for each node. Self-loops are edges from and to the same node (v_i, v_i). The edge mask values for edges indicate the importance of the messages passed between two nodes. In order to understand the behavior of the UPFD classifier in terms of the root node, we look at the incoming edges to the root node in the explanation. Furthermore, the transparency of the edges are an indicator for the magnitude of the edge mask value. However, this can not be clearly observed in this example.
On the other hand, when we inspect the node feature mask F, we can understand which features are deemed important by the UPFD classifier for a given node. We are particularly interested in the root node, and the explanation associated to it. Since the root node consists of news content embeddings, intuitively, we can consider the root node explanation as the important subset of news content embeddings. However, we were not able to test this intuition due to (i) the size of data needed to be downloaded to construct FakeNewsNet (Shu et al., 2018) whose news content and social context data was utilized in UPFD, (ii) insufficient information on Dou et al., 2021 to reproduce the UPFD framework, (iii) large maximum input sequence (768) for BERT model to

process the news content on a local computer. Thus, we continue our analysis without considering node feature mask from now on.

Interpreting the Explanation. As one would expect, the UPFD classifier thinks that the the cascaders (initial retweeters) are the most important users (nodes) for the prediction of fake for this instance. Only considering the echo chamber effect, one would expect higher edge mask values for the spreaders v_8, v_{16}, v_{24}, and v_{27}. Accordingly, we observe larger importance scores on edges that connect these nodes to the root node. The edge (v_0, v_{13}) was assigned the highest edge mask value with 0.1371. The edges (v_0, v_8), (v_0, v_{16}), (v_0, v_{24}), and (v_0, v_{27}) were assigned 0.1081, 0.0951, 0.1222, and 0.1181, respectively. While the reasons behind the maximal importance score assignment to edge (v_0, v_{13}) remains unclear, we observe that the biggest echo chamber's connection to the root node has a high edge mask value, which suggests that the UPFD classifier might be able to utilize the echo chamber effect with success for prediction. However, we need to observe similar behavior in other fake news instances that exhibit echo chambers.

We also examined the highest edge mask valued five edges to understand which edges are important for the UPFD classifier. Before we proceed, it should be noted that the subgraph generation does not take into account where the maximal edge mask values reside, instead it tries to capture the biggest subgraph with the highest mutual information between the original graph and the subgraph with respect to the prediction.

Edge Mask Analysis. Proceeding our analysis for the five largest values in our edge mask provided by GNNExplainer, we observed that the second highest value belongs to the edge (v_{24}, v_{41}), i.e., an echo chamber connection. The third one belongs to a self-loop (v_{44}, v_{44}) and the fourth one is for the edge (v_{44}, v_{24}). We can interpret the edge mask value for the self-loop as the significance that the UPFD classifier puts on the node v_{44}. The edge mask value for edge (v_{44}, v_{24}) confirms our interpretation for node v_{44}, because this is the first instance for an important edge that is directed from a retweeter to its source. Thus, the user v_{44} and their historical data should be investigated. The fifth largest edge mask value is assigned to the edge (v_{27}, v_{62}) which is another propagation edge in an echo chamber. The distribution of the edge mask values are given in Fig. 4.6

Visiting back the features we introduced, we observe that the feature SF_7 is captured in this explanation. To elaborate, we see that the edges to the cascade node with the highest number of retweets were deemed important by the explainer. This implies that the UPFD classifier captured an important feature for this instance. Although, in order to get this interpretation, one would require a data scientist. A domain expert or an end user is likely to either misinterpret or not understand the explanation provided by GNNExplainer. Furthermore, some of the most important edges were not even included in the visualization. This certainly decreases the understandability of the

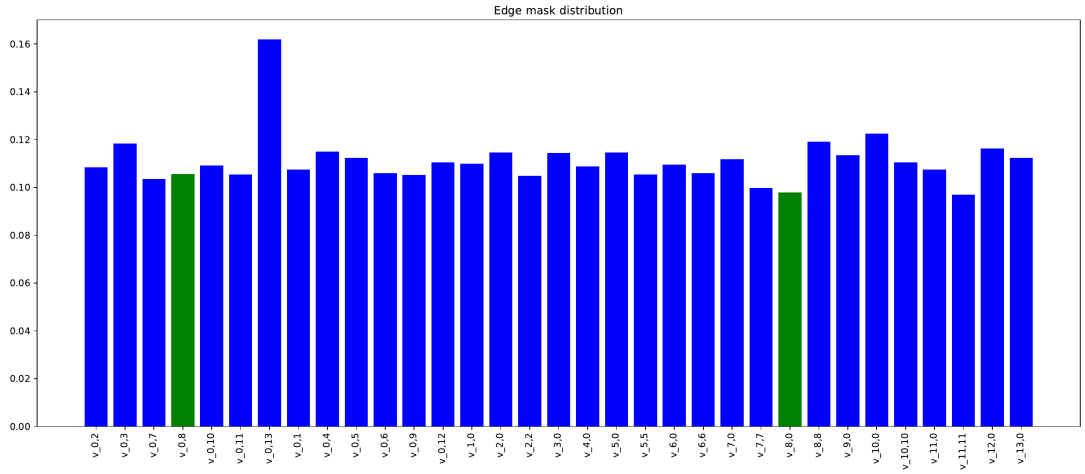


Figure 4.6: Edge mask distribution of the subgraph provided for echo chamber example explanation. The green bars represent the edge mask values of echo chamber connections.

explanations.

Sensitivity Analysis. We also analyzed the faithfulness of the explanations to the UPFD classifier. We took the explanation subgraph we provided in 4.5, restored its node features and fed it to the UPFD classifier to see if the learned structure is still predicted as fake with high probability. In fact, the UPFD classifier predicted the subgraph as fake with 0.9915, which is higher than the prediction probability of the original input. Overall, we say that the UPFD classifier is faithful to this instance, and GNNExplainer can indeed reduce noise in a graph as Ying et al. (2019) claim.

We proceed our analysis of this fake news instance with an experiment. In this experiment, we remove the news content and historical information of children nodes, both in the original graph and the explanation subgraph to see their effect on the prediction. We provide the results in Table 4.3.

It is trivial to observe in Table 4.3 that the UPFD classifier's confidence decreases as we remove news content for the echo chamber example. Interestingly, when we remove users' historical information from all nodes except root node, the prediction probability almost does not decrease. This suggests UPFD classifier takes into account the news content and structural information more than the historical information. This behavior can also be observed from the UPFD classifier's architecture in Fig. 4.3. The skip connection and concatenation of the news content in the root node appears to be doing what is expected of it. In fact, the authors of UPFD (Dou et al., 2021) showed that with a vanilla version of UPFD classifier, i.e., without skip connection and concatenation,

Operation	$p_{original}$	$p_{subgraph}$
No change	0.9912	0.9915
Remove news content	0.8328	0.8018
Remove historical information	0.9812	0.9809
Remove all node information	0.4940	0.4940

Table 4.3: Model prediction probabilities for input perturbation experiment. $p_{original}$ denotes prediction probability for the original input. Analogously, $p_{subgraph}$ refers to the prediction probability for the explanation subgraph.

the performance drops. This finding of ours also goes in parallel with their finding. Furthermore, when we remove all node information, the UPFD classifier can no longer tell fake from real. Thus, without node information, just depending on the structure of the echo chamber example for prediction turns out to be a bad idea. We need the embeddings for the news.
We illustrate more examples in order to understand if the statistically significant features were captured in other instances. First, it is nontrivial to look for SF_1 in the explanations as the tree depth will be decreased significantly. Most of the time the tree depth in the explanations we observed were one. Thus, we skip this feature analysis. We look for the effect of feature SF_2 in the explanations. We compare our echo chamber example in Fig. 4.4 with a real news example in Fig. 4.7. The echo chamber example has 72 nodes whereas the real news example has 59 nodes. Now let us observe the explanation provided for the real news instance in Fig. 4.8.
The explanation subgraph of the real news example contains more nodes than the explanation subgraph for the echo chamber example. This might imply the effect of SF_2 on the explanations. In other words, explanations capture the importance of more nodes for the real news example. Although, this should be tested with other real and fake news instances to be verified. This instance has some nodes that exhibit an echo chamber effect. Still, we can easily observe that some of these nodes were also retweeted, a behavior not observed in the echo chamber example.
Initial Improvement Suggestions for GNNExplainer. During our experiments, we realized that some explanations are hard to read, particularly, the ones with too many nodes and edges. Including the explanation provided in Fig. 4.8, it is hard to distinguish edges. In order to understand the importance of edges, and simplify the explanation subgraph, we employed three simple techniques that uses a threshold value. Then we apply this threshold value to the edge mask. This operation sets the values less than or equal to the threshold to zero. The first approach was straightforward, take the maximum and the minimum value of the edge mask values. Then, set the threshold to a

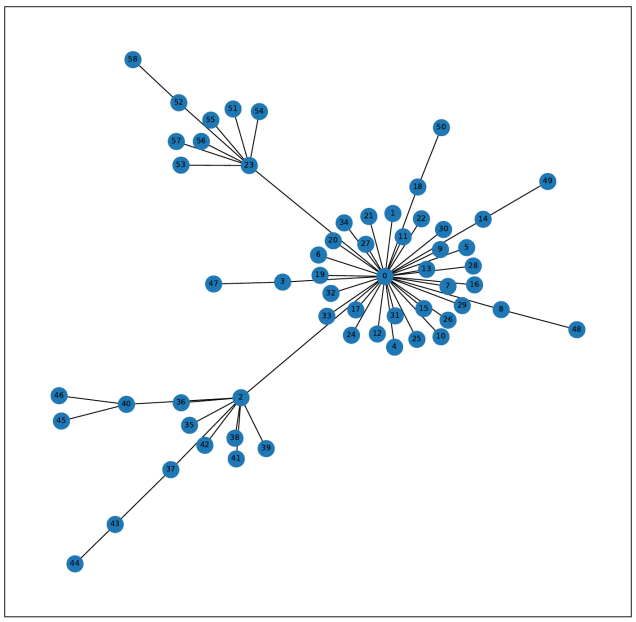


Figure 4.7: A real news instance from UPFD-Politifact. We shall refer to this example as the *real news example*. It was predicted as real news with probability 0.9956

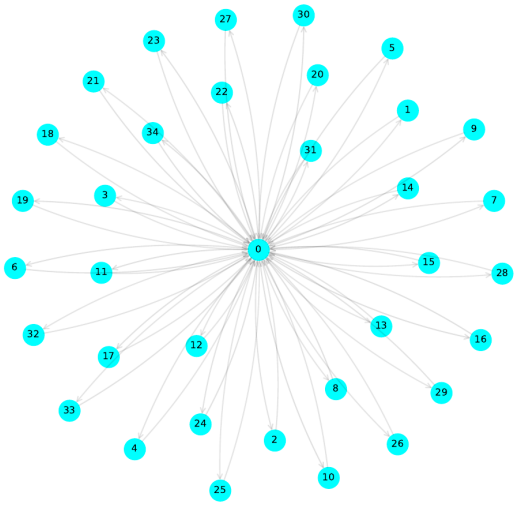
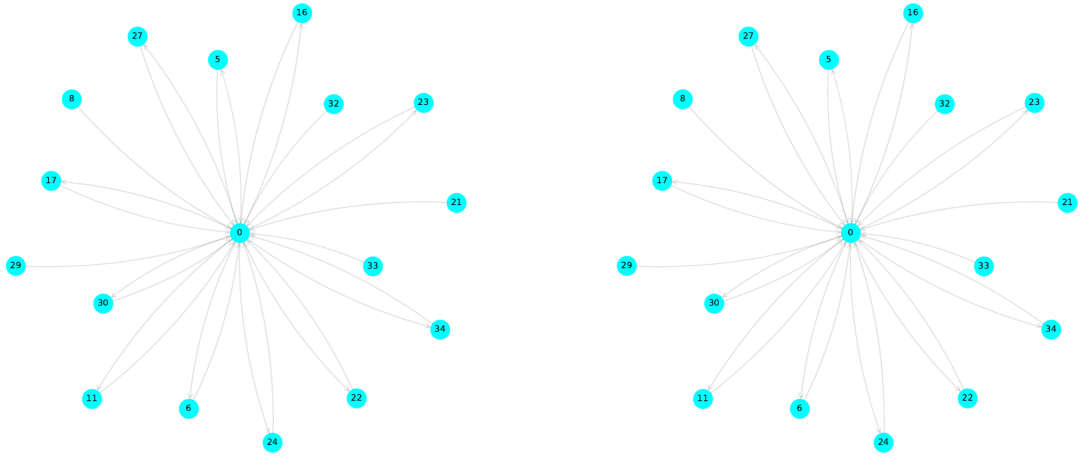


Figure 4.8: Explanation of the real news example from UPFD-Politifact. Contains 35 nodes.



(a) Median (b) Mean

Figure 4.9: Threshold methods for subgraph explanations of the real news example. In this case median and mean provides the same output.

value less than the maximum but higher than the minimum by manual estimation, and observe the changes in the explanation subgraph as we decrease the threshold value. This approach was too much manual work, so we came up with a more systematic approach. The second approach is to get the mean of all edge mask values for the current instance to get a threshold value. This way we obtain less edges in our initial explanation, but keeping an important set of edges to understand their importance. Our third approach is to take the median of the edge mask set, and use it as a threshold that applies the same effect as the averaging approach. There were no big differences between mean and median method. However, we opted for median as it was able to capture more edges for the explanations we worked with compared to the mean method. We illustrate the explanations with threshold methods mean and median in Fig 4.9.

We can extend our method to an interactive plot in which we are able to change the threshold value dynamically. Furthermore, we can use the edge mask values provided for the edges in the subgraph to apply average and mean thresholding methods. We leave this for future work.

Introducing Unseen Data. We analyzed the predictions made for one real and one fake news instances that were in the test split. This time, we randomly sampled one fake and one real news from the UPFD-Gossipcop dataset, which is unseen data for the UPFD classifier that was trained on the UPFD-Politifact dataset. We did not generate

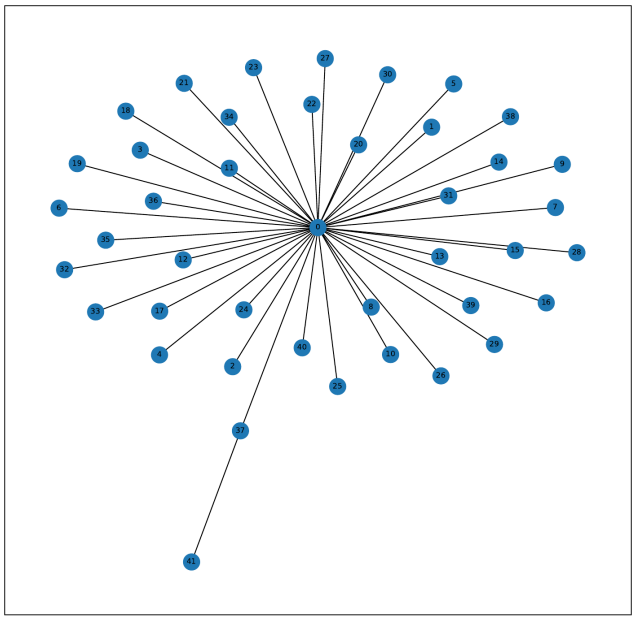


Figure 4.10: A fake news example from UPFD-Gossipcop. We introduced this instance as an unseen data to the UPFD classifier, and it predicted this fake news piece as real with a probability of 0.9984.

a new graph instance as obtaining the news piece, creating embeddings for it, and collecting its social context information is a cumbersome process without this data prepared beforehand.

Our unseen fake news example from Gossipcop is illustrated in Fig. 4.10. It is obvious that this fake news piece does not display echo chamber patterns, in fact, structurally it looks more like a real news piece. This may be why the UPFD classifier mispredicted this instance. Let us further dive in to understand this. There can be a variety of reasons. It can be because of the difference in the news style in both datasets. Politifact houses a set of political news whereas Gossipcop houses a massive set of celebrity news. So the difference of textual content could be a reason. Another reason can be the small number of training instances for the set. However, we were not able to gather suggestive information from the explanation thus we did not include the explanation. One way to understand whether the UPFD classifier responds to echo chamber structures is to create these echo chambers by adding children nodes to some of cascade nodes (initial retweeters), and feed this to the UPFD classifier. Thus, initially, we added 20 nodes with no information in them, and selected nodes v_{37} and v_{39} arbitrarily to be the spreaders. The resulting graph is illustrated in Fig. 4.11. When the fed the perturbed sample to the

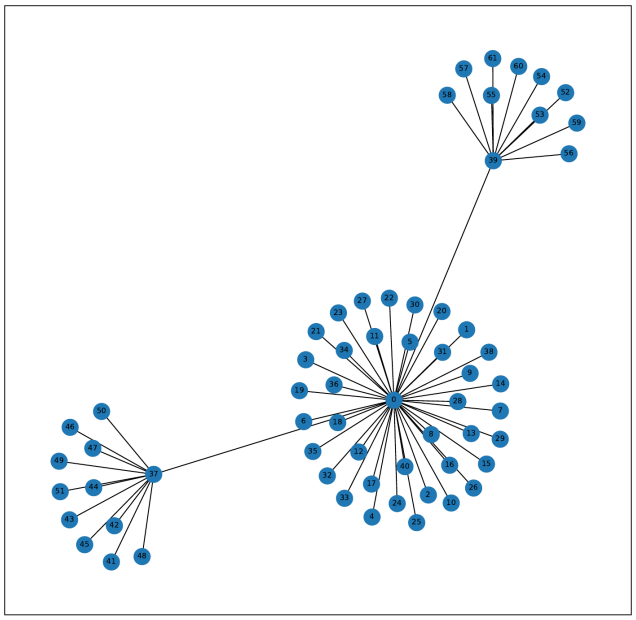
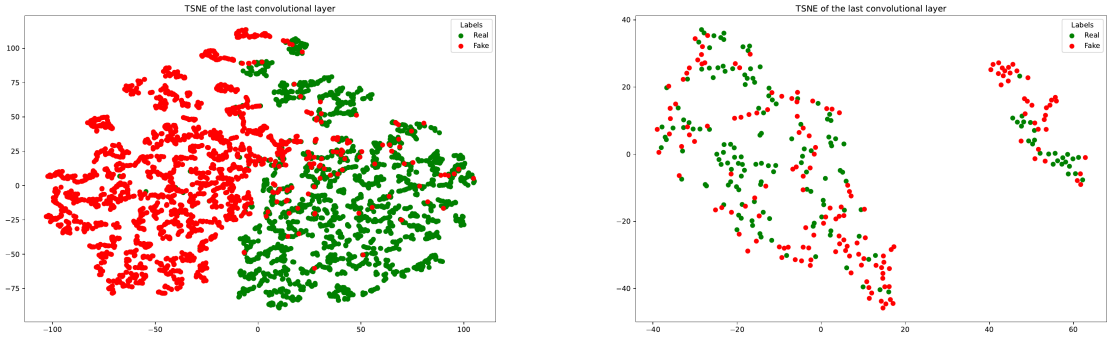


Figure 4.11: The perturbed version of the fake news example from UPFD-Gossipcop. We added echo-chamber like structures to nodes v_{37} and v_{39}.

UPFD classifier, the increase in prediction probability for fake news slightly increased. So we also sampled some historical information from a randomly obtained fake news instance that is different than the one we are analyzing. We embedded the new nodes with randomly obtained users' historical information. Then we fed this instance to the UPFD classifier, but the change in probability was too small that it can be omitted. Thus, for this instance, our experiment suggests that echo chamber structures does not necessarily increase the probability for fake news.

Latent Space Analysis. We apply t-SNE (van der Maaten & Hinton, 2008) using default settings provided by scikit-learn (Pedregosa et al., 2011) to see if GraphSAGE is able to distinguish fake and real news before its outputs are concatenated with news content in the root node and then fed to the FCN. The visualization for this procedure is provided in Fig 4.12. Clearly GraphSAGE is able to distinguish fake and real news instances better in Gossipcop dataset, but it fails to make this discirimination with Politifact dataset. This can be due to the low amounts of instances in the Politifact dataset.

Target Audience, XAI Goals, and Final Improvement Suggestions. It is clear that understanding the UPFD classifier's behavior locally with just a visualization is not sufficient for an end user to understand why a particular news piece was predicted as real or fake. However, given enough resources, one can reverse engineer the process



(a) Gossipcop (b) Politifact

Figure 4.12: t-SNE visualizations for max pooled GraphSAGE on UPFD.

in the UPFD framework, which would allow for the analysis of node feature masks of news content and historical information embeddings. That would certainly increase the trustworthiness and informativeness of the UPFD classifier. We also suggest to employ interactive visualizations when working with GNNExplainer since the explanations can get very complex. Explanations produced by GNNExplainer should be investigated by a data scientist who will simplify the explanation for a domain expert or an end user. For UPFD, this simplification process can include aggregation of bidirectional edge mask values so that they can be represented as an undirected edge. Moreover, these undirected edges can be color-coded such that each color represents a value above a certain threshold which can be obtained using the methods we suggested. Also for GNNs with language models, the node features should be easily mappable to the textual content in such a way that we can create a variation of force plots provided by SHAP framework.

To sum up, we have analyzed the explanations produced by GNNExplainer for the UPFD classifier which was trained on UPFD-Politifact dataset. We have observed some of the statistically significant features' existence in the explanations. We have also addressed shortcomings of GNNExplainer, suggested simple yet effective methods to reduce the complexity of an explanation. All experiments were conducted using Python 3.7 (van Rossum, 1995), PyTorch 1.11.0 (Paszke et al., 2019) and PyTorch Geometric 2.0.4 (Fey & Lenssen, 2019). All visualizations were obtained through the use of NetworkX 2.6.3 (Hagberg et al., 2008), and Matplotlib 3.5.1 (Hunter, 2007). The detailed library usage and the implementation of all models are provided in our github repository [1].

[1] https://github.com/sersery35/Explainability_of_FND_Models/tree/main/GNNFakeNews

5 Conclusion

Automated FND is becoming a necessity as more people obtain their news from social media or the web. While the source of a news piece is vital for its authenticity, history observed that even the most trustworthy sources could share fake news. The dissemination of fake news leads to more misinformed people, creating a worldwide issue. Many researchers have tried to build FND systems in various settings to tackle this problem. With the evolution of ML systems, FND systems became automated. Nevertheless, automated FND is still challenging due to the ever-changing format of fake news, malicious accounts, outdated or missing datasets, and the lack of reproducible code. The research community should put in more effort to tackle the issue of fake news.
Moreover, all the FND works we have observed reported their performance for their model, but they did not attempt to utilize explanation methods to interpret the model's behavior. To our best knowledge, this is the first work that attempts to explain FND models. We strongly encourage the researchers to adopt explanation methods for their models to understand their models' behavior. As we observed in 3.2.2, we could trick the model into predicting a fake news instance as real even though the model has seen that fake news instance in training. Adversarial examples, explanation methods, and input sensitivity analysis should be incorporated into the model development process to ensure that the model will function as intended. Particularly for black-box models like DNNs or GNNs, previously stated procedures are essential. Otherwise, one cannot completely understand whether the model has learned the actual task.
We listed our research objectives in Chapter 1, and now, we show that we have fulfilled all of them. The first, **RO1**, is satisfied in Chapter 2, in which we discussed explanation methods for DNNs and GNNs. We provided the intuition behind the most popular explanation methods available for FND. For all models that are not graph-based, we can easily use tools from the SHAP framework; for GNNs, we utilize GNNExplainer.
For the second objective, we have shown that a model can be deceived with several perturbations to the input. So we adopted the partition explainer to help us uncover this fallacy. We showed that this fallacy was closely related to the model's architecture and the contents of the dataset. We then retrained the news content model so that we could analyze it without the features that were causing the issue. We obtained a similar performance with better base values, which suggests that the model learned to distinguish between the news by looking at more complex features. Thus, **RO2** is

fulfilled in 3.2.2.
For the last objective, **RO3**, our suggestions are twofold. The first is for the explanations from the partition explainer. We suggested the usage of bar plots of Owen values for long texts in 3.2.2. The second is for the explanations from GNNExplainer. We proposed in 4.2.2 to use median or mean for edge masks where the explanation is too complex. We discuss the understandability and structure of explanations regarding the input data. We recommend producing explanations in the same form as the input, which would increase the understandability of the explanations. Although, this is a complicated task considering GNNExplainer attempts to explain all non-euclidean data, thus providing an explanation for every type. For this shortcoming for tree-structured datasets, we suggest aggregating the edge mask values of the edges between the same nodes so that the explanation will resemble the input. Furthermore, we suggest the adoption of interactive plots for GNNExplainer so that one can observe the changes in edges and nodes as they interact with the threshold.
In addition to the research objectives, we showed that utilizing social context data with news content data can increase performance by sensitivity analysis. We also discussed how to interpret explanations from GNNExplainer, analyzed its explanations' faithfulness, and observed the news content's importance for a prediction done by a GNN model. We stressed in 4.2.2 the importance of the availability of the news content in UPFD to utilize node feature masks, which can then be illustrated with the text input similar to the way the SHAP framework does. Thus, we could not analyze the importance per token in a news piece because of the limitations we enumerated in 4.2.2. Furthermore, incompatibility issues of DeepSHAP with Transformer models, outdated datasets, and the availability of the datasets were other limitations. For instance, our attempt to construct FakeNewsNet from the authors' code took more than two months to download only around 80% of the dataset. Therefore, we were not able to analyze the UPFD classifier's behavior in depth. Fortunately, the incompatibility issue of DeepSHAP was mitigated by the model-agnostic partition explainer.
The research community took a particular interest in FND, and new studies are considering these limitations in their designs. Lately, the discussion of model aging and early fake news detection has gained momentum. As we discussed in 2.1.1, fake news tends to spread too fast. Hence, a model that can detect fake news pieces before they reach their peak dissemination is necessary. Otherwise, due to the psychologica tendencies of humans, we will not be able to alleviate the adverse effects of fake news. The ever-changing nature of fake news content gives rise to the necessity of FND systems with continual learning, which Han et al. (2020) discuss. We encourage a similar approach for FND to create robust automated systems.
Undertaking these limitations should be the next step in future research. Without well-structured and informative datasets, developing a competitive model is not pos-

sible. Specifically, rich datasets like UPFD should provide more information about their process so that more insight can be gained from the dataset and explanations. Therefore, as a final word, we draw attention to the importance of the availability of datasets and reproducible studies and urge researchers to be more contributing on this matter.

List of Figures

List of Tables

Bibliography

Adams, S. H. (2002). Communication under stress: Indicators of veracity and deception in written narratives.

Addawood, A., Schneider, J., & Bashir, M. (2017). Stance classification of twitter debates: The encryption debate as a use case. *Proceedings of the 8th International Conference on Social Media & Society*. https://doi.org/10.1145/3097286.3097288

Afroz, S., Brennan, M., & Greenstadt, R. (2012). Detecting hoaxes, frauds, and deception in writing style online. *2012 IEEE Symposium on Security and Privacy*, 461–475. https://doi.org/10.1109/SP.2012.34

Agrawal, A. (2016). Clickbait detection using deep learning. *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, 268–272. https://doi.org/10.1109/NGCT.2016.7877426

ALDayel, A., & Magdy, W. (2021). Stance detection on social media: State of the art and trends. *Information Processing & Management*, *58*(4), 102597. https://doi.org/https://doi.org/10.1016/j.ipm.2021.102597

Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, *31*(2), 211–36. https://doi.org/10.1257/jep.31.2.211

Allen, J., Arechar, A. A., Pennycook, G., & Rand, D. G. (2021). Scaling up fact-checking using the wisdom of crowds. *Science Advances*, *7*(36), eabf4393. https://doi.org/10.1126/sciadv.abf4393

Angwin, J., Larson, J., Kirchner, L., & Mattu, S. (2016). Machine bias.

Antunes, P., Herskovic, V., Ochoa, S. F., & Pino, J. A. (2008). Structuring dimensions for collaborative systems evaluation. *ACM Comput. Surv.*, *44*(2). https://doi.org/10.1145/2089125.2089128

Arras, L., Montavon, G., Müller, K.-R., & Samek, W. (2017). Explaining recurrent neural network predictions in sentiment analysis. https://doi.org/10.48550/ARXIV.1706.07206

Asch, S. E., & Guetzkow, H. (1951). Effects of group pressure upon the modification and distortion of judgments. *Groups, leadership, and men*, 222–236.

Ashforth, B. E., & Mael, F. (1989). Social identity theory and the organization. *The Academy of Management Review*, *14*(1), 20–39.

Atwood, J., & Towsley, D. (2016). Diffusion-convolutional neural networks. *Advances in neural information processing systems*, *29*.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, 722–735.

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. https://doi.org/10.48550/ARXIV.1607.06450

Bachenko, J., Fitzpatrick, E., & Schonwetter, M. (2008). Verification and implementation of language-based deception indicators in civil and criminal narratives. *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, 41–48.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., & Müller, K.-R. (2010). How to explain individual classification decisions. *Journal of Machine Learning Research*, *11*(61), 1803–1831.

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

Balmas, M. (2014). When fake news becomes real: Combined exposure to multiple news sources and political attitudes of inefficacy, alienation, and cynicism. *Communication Research*, *41*(3), 430–454. https://doi.org/10.1177/0093650212453600

Bansal, T., Belanger, D., & McCallum, A. (2016). Ask the gru: Multi-task learning for deep text recommendations. *Proceedings of the 10th ACM Conference on Recommender Systems*. https://doi.org/10.1145/2959100.2959180

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, *58*, 82–115. https://doi.org/https://doi.org/10.1016/j.inffus.2019.12.012

Barrón-Cedeño, A., Elsayed, T., Nakov, P., Da San Martino, G., Hasanain, M., Suwaileh, R., Haouari, F., Babulkov, N., Hamdan, B., Nikolov, A., Shaar, S., & Ali, Z. S. (2020). Overview of checkthat! 2020: Automatic identification and verification of claims in social media. In A. Arampatzis, E. Kanoulas, T. Tsikrika, S. Vrochidis, H. Joho, C. Lioma, C. Eickhoff, A. Névéol, L. Cappellato, & N. Ferro (Eds.), *Experimental ir meets multilinguality, multimodality, and interaction* (pp. 215–236). Springer International Publishing.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Trans Neural Netw*, *5*(2), 157–166.

Bennetot, A., Laurent, J.-L., Chatila, R., & Díaz-Rodríguez, N. (2019). Towards explainable neural-symbolic visual reasoning. https://doi.org/10.48550/ARXIV.1909.09065

Berinsky, A. J. (2017). Rumors and health care reform: Experiments in political misinformation. *British Journal of Political Science*, *47*(2), 241–262.

Bessi, A., & Ferrara, E. (2016). Social bots distort the 2016 us presidential election online discussion. *First Monday*, *21*(11).

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, *3*(null), 993–1022.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching word vectors with subword information. *CoRR*, *abs/1607.04606*.

Braşoveanu, A. M. P., & Andonie, R. (2019). Semantic fake news detection: A machine learning perspective. In I. Rojas, G. Joya, & A. Catala (Eds.), *Advances in computational intelligence* (pp. 656–667). Springer International Publishing.

Breiman, L., Friedman, J., Stone, C., & Olshen, R. (1984). *Classification and regression trees*. Taylor & Francis.

Brewer, P. R., Young, D. G., & Morreale, M. (2013). The Impact of Real News about "Fake News": Intertextual Processes and Political Satire. *International Journal of Public Opinion Research*, *25*(3), 323–343. https://doi.org/10.1093/ijpor/edt015

Bridle, J. (1989). Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In D. Touretzky (Ed.), *Advances in neural information processing systems*. Morgan-Kaufmann.

Bronstein, M. M., Bruna, J., Cohen, T., & Veličković, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. https://doi.org/10.48550/ARXIV.2104.13478

Buciluâ, C., Caruana, R., & Niculescu-Mizil, A. (2006). Model compression. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 535–541. https://doi.org/10.1145/1150402.1150464

Burrell, J. (2016). How the machine 'thinks': Understanding opacity in machine learning algorithms. *Big Data & Society*, *3*(1), 2053951715622512. https://doi.org/10.1177/2053951715622512

Byrne, R. M. J. (2019). Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 6276–6282. https://doi.org/10.24963/ijcai.2019/876

Caruana, R., Kangarloo, H., Dionisio, J. D., Sinha, U., & Johnson, D. (1999). Case-based explanation of non-case-based learning methods. *Proc AMIA Symp*, 212–215.

Casajus, A. (2009). The shapley value, the owen value, and the veil of ignorance. *International Game Theory Review (IGTR)*, *11*, 453–457. https://doi.org/10.1142/S0219198909002431

Castelvecchi, D. (2016). Can we open the black box of ai? *Nature*, *538*, 20–23. https://doi.org/10.1038/538020a

Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on twitter. *Proceedings of the 20th International Conference on World Wide Web*, 675–684. https://doi.org/10.1145/1963405.1963500

Cauchy, L. A. (1847). M´ethode générale pour la résolution des systémes d'équations simultanées. *Compte Rendu á l'Académie des Sciences*.

Chen, Y., Conroy, N. K., & Rubin, V. L. (2015). News in an online world: The need for an "automatic crap detector". *Proceedings of the Association for Information Science and Technology*, *52*(1), 1–4. https://doi.org/https://doi.org/10.1002/pra2.2015.145052010081

Cheng, J., Bernstein, M., Danescu-Niculescu-Mizil, C., & Leskovec, J. (2017). Anyone can become a troll: Causes of trolling behavior in online discussions. *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 1217–1230. https://doi.org/10.1145/2998181.2998213

Chiang, W.-L., Liu, X., Si, S., Li, Y., Bengio, S., & Hsieh, C.-J. (2019). Cluster-GCN. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. https://doi.org/10.1145/3292500.3330925

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. https://doi.org/10.48550/ARXIV.1406.1078

Cinus, F., Minici, M., Monti, C., & Bonchi, F. (2022). The effect of people recommenders on echo chambers and polarization. *Proceedings of the International AAAI Conference on Web and Social Media*, *16*(1), 90–101.

Conroy, N. K., Rubin, V. L., & Chen, Y. (2015). Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology*, *52*(1), 1–4. https://doi.org/https://doi.org/10.1002/pra2.2015.145052010082

Daelemans, W. (2010). Pos tagging. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of machine learning* (pp. 776–779). Springer US. https://doi.org/10.1007/978-0-387-30164-8_643

Dan, V., Paris, B., Donovan, J., Hameleers, M., Roozenbeek, J., van der Linden, S., & von Sikorski, C. (2021). Visual mis- and disinformation, social media, and democracy. *Journalism & Mass Communication Quarterly*, *98*(3), 641–664. https://doi.org/10.1177/10776990211035395

Datta, A., Sen, S., & Zick, Y. (2016). Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. *2016 IEEE Symposium on Security and Privacy (SP)*, 598–617. https://doi.org/10.1109/SP.2016.42

Denil, M., Demiraj, A., & de Freitas, N. (2014). Extraction of salient sentences from labelled documents. *CoRR*, *abs/1412.6815*.

DePaulo, B. M., Charlton, K., Cooper, H., Lindsay, J. J., & Muhlenbruck, L. (1997). The accuracy-confidence correlation in the detection of deception [PMID: 15661668]. *Personality and Social Psychology Review*, *1*(4), 346–357. https://doi.org/10.1207/s15327957pspr0104_5

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, *abs/1810.04805*.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. https://doi.org/10.48550/ARXIV.1702.08608

Dosilovic, F. K., Brcic, M., & Hlupic, N. (2018). Explainable artificial intelligence: A survey. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 0210–0215. https://doi.org/10.23919/MIPRO.2018.8400040

Dou, Y., Shu, K., Xia, C., Yu, P. S., & Sun, L. (2021). User preference-aware fake news detection. *CoRR*, *abs/2104.12259*.

Du, J., Xu, R., He, Y., & Gui, L. (2017). Stance classification with target-specific neural attention. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 3988–3994. https://doi.org/10.24963/ijcai.2017/557

Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A., & Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. https://doi.org/10.48550/ARXIV.1509.09292

Edwards, L., & Veale, M. (2017). Slave to the algorithm? why a 'right to an explanation' is probably not the remedy you are looking for. *Duke law and technology review*, *16*, 18–84.

ElBoghdady, D. (2013). Market quavers after fake ap tweet says obama was hurt in white house explosions.

English wikipedia. (2022).

Escalante, H. J., Escalera, S., Guyon, I., Baro, X., Gucluturk, Y., Guclu, U., & van Gerven, M. (2018). *Explainable and interpretable models in computer vision and machine learning* (1st). Springer Publishing Company, Incorporated.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., & Yates, A. (2005). Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, *165*(1), 91–134. https://doi.org/https://doi.org/10.1016/j.artint.2005.03.001

Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2016). The rise of social bots. *Commun. ACM*, *59*(7), 96–104. https://doi.org/10.1145/2818717

Fey, M., & Lenssen, J. E. (2019). Fast graph representation learning with PyTorch Geometric. *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Fisher, M., Cox, J. W., & Hermann, P. (2016). Pizzagate: From rumor, to hashtag, to gunfire in d.c.

Foster, V. S. (2016). The great moon hoax. In *Modern mysteries of the moon: What we still don't know about our lunar companion* (pp. 11–44). Springer International Publishing. https://doi.org/10.1007/978-3-319-22120-5_2

Gage, P. (1994). A new algorithm for data compression. *C Users J.*, *12*(2), 23–38.

Galton, F. (1907). Vox populi (the wisdom of crowds). *Nature*, *75*(7), 450–451.

Gammerman, A., Vovk, V., & Vapnik, V. (1998). Learning by transduction. *In Uncertainty in Artificial Intelligence*, 148–155.

Gao, H., Wang, Z., & Ji, S. (2018). Large-scale learnable graph convolutional networks. *CoRR*, *abs/1808.03965*.

Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. https://doi.org/10.48550/ARXIV.1705.03122

Gers, F., & Schmidhuber, J. (2000). Recurrent nets that time and count. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, *3*, 189–194 vol.3. https://doi.org/10.1109/IJCNN.2000.861302

Ghanem, B., Rosso, P., & Rangel, F. (2018). Stance detection in fake news a combined feature representation. *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 66–71. https://doi.org/10.18653/v1/W18-5510

Gokaslan, A., & Cohen, V. (2019). Openwebtext corpus.

Goodman, B., & Flaxman, S. (2017). European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, *38*(3), 50–57. https://doi.org/10.1609/aimag.v38i3.2741

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Comput. Surv.*, *51*(5). https://doi.org/10.1145/3236009

Gunning, D., & Aha, D. (2019). Darpa's explainable artificial intelligence (xai) program. *AI Magazine*, *40*(2), 44–58. https://doi.org/10.1609/aimag.v40i2.2850

Guo, Z., Schlichtkrull, M., & Vlachos, A. (2022). A Survey on Automated Fact-Checking. *Transactions of the Association for Computational Linguistics*, *10*, 178–206. https://doi.org/10.1162/tacl_a_00454

Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring network structure, dynamics, and function using networkx. In G. Varoquaux, T. Vaught, & J. Millman (Eds.), *Proceedings of the 7th python in science conference* (pp. 11–15).

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. *CoRR*, *abs/1706.02216*.

Han, Y., Karunasekera, S., & Leckie, C. (2020). Graph neural networks with continual learning for fake news detection from social media. https://doi.org/10.48550/ARXIV.2007.03316

Hancock, J. T., Curry, L. E., Goorha, S., & Woodworth, M. (2007). On lying and being lied to: A linguistic analysis of deception in computer-mediated communication. *Discourse Processes*, *45*(1), 1–23. https://doi.org/10.1080/01638530701739181

Handelman, G. S., Kok, H. K., Chandra, R. V., Razavi, A. H., Huang, S., Brooks, M., Lee, M. J., & Asadi, H. (2019). Peering into the black box of artificial intelligence: Evaluation metrics of machine learning methods [PMID: 30332290]. *American Journal of Roentgenology*, *212*(1), 38–43. https://doi.org/10.2214/AJR.18.20224

Hanselowski, A., PVS, A., Schiller, B., Caspelherr, F., Chaudhuri, D., Meyer, C. M., & Gurevych, I. (2018). A retrospective analysis of the fake news challenge stance-detection task. *Proceedings of the 27th International Conference on Computational Linguistics*, 1859–1874.

Harding, L. (2012). Putin seen behind bars in spoof video.

Hassan, N., Li, C., & Tremayne, M. (2015). Detecting check-worthy factual claims in presidential debates. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 1835–1838. https://doi.org/10.1145/2806416.2806652

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. https://doi.org/10.48550/ARXIV.1512.03385

Hearst, M., Dumais, S., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their Applications*, *13*(4), 18–28. https://doi.org/10.1109/5254.708428

Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). https://doi.org/10.48550/ARXIV.1606.08415

Herlocker, J. L., Konstan, J. A., & Riedl, J. (2000). Explaining collaborative filtering recommendations. *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, 241–250. https://doi.org/10.1145/358916.358995

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. https://doi.org/10.48550/ARXIV.1503.02531

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*, 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735

Honnibal, M., & Montani, I. (2018). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, *5*(2), 1.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: With applications in r* (1st ed.). Springer.

Jin, Z., Cao, J., Jiang, Y.-G., & Zhang, Y. (2014). News credibility evaluation on microblog with a hierarchical propagation model. *2014 IEEE International Conference on Data Mining*, 230–239. https://doi.org/10.1109/ICDM.2014.91

Jin, Z., Cao, J., Zhang, Y., & Luo, J. (2016). News verification by exploiting conflicting social viewpoints in microblogs. *Proceedings of the AAAI Conference on Artificial Intelligence*, *30*(1). https://doi.org/10.1609/aaai.v30i1.10382

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, *28*, 11–21.

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. *CoRR*, *abs/1607.01759*.

Kahneman, D., & Tversky, A. (1979). Prospect theory: Analysis of decision under risk. *Econometrica*, *47*, 263–291.

Kalchbrenner, N., Espeholt, L., Simonyan, K., Oord, A. v. d., Graves, A., & Kavukcuoglu, K. (2016). Neural machine translation in linear time. https://doi.org/10.48550/ARXIV.1610.10099

Kiesel, J., Mestre, M., Shukla, R., Vincent, E., Adineh, P., Corney, D., Stein, B., & Potthast, M. (2019). SemEval-2019 task 4: Hyperpartisan news detection. *Proceedings of the 13th International Workshop on Semantic Evaluation*, 829–839. https://doi.org/10.18653/v1/S19-2145

Kim, B. (2015).

Kim, B., Glassman, E. L., Johnson, B., & Shah, J. A. (2015). Ibcm: Interactive bayesian case model empowering humans via intuitive interaction.

Kim, B., Rudin, C., & Shah, J. (2015). The bayesian case model: A generative approach for case-based reasoning and prototype classification. https://doi.org/10.48550/ARXIV.1503.01161

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. https://doi.org/10.48550/ARXIV.1412.6980

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. https://doi.org/10.48550/ARXIV.1609.02907

Kowalski, P. A., & Kusy, M. (2018). Sensitivity analysis for probabilistic neural network structure reduction. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(5), 1919–1932. https://doi.org/10.1109/TNNLS.2017.2688482

Kuhn, M., & Johnson, K. (2013). Applied predictive modeling.

Kwon, S., Cha, M., Jung, K., Chen, W., & Wang, Y. (2013). Prominent features of rumor propagation in online social media. *2013 IEEE 13th International Conference on Data Mining*, 1103–1108. https://doi.org/10.1109/ICDM.2013.61

Lakkaraju, H., Bach, S. H., & Leskovec, J. (2016). Interpretable decision sets: A joint framework for description and prediction. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1675–1684. https://doi.org/10.1145/2939672.2939874

Lapuschkin, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, *10*, e0130140. https://doi.org/10.1371/journal.pone.0130140

Lazer, D. M. J., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., Metzger, M. J., Nyhan, B., Pennycook, G., Rothschild, D., Schudson, M., Sloman, S. A., Sunstein, C. R., Thorson, E. A., Watts, D. J., & Zittrain, J. L. (2018). The science of fake news. *Science*, *359*(6380), 1094–1096. https://doi.org/10.1126/science.aao2998

Li, R., Wang, S., Zhu, F., & Huang, J. (2018). Adaptive graph convolutional neural networks. *Proceedings of the AAAI conference on artificial intelligence*, *32*(1).

Liang, B., Li, H., Su, M., Li, X., Shi, W., & Wang, X. (2021). Detecting adversarial image examples in deep neural networks with adaptive noise reduction. *IEEE Transactions on Dependable and Secure Computing*, *18*(1), 72–85. https://doi.org/10.1109/tdsc.2018.2874243

Lindeman, M., & Aarnio, K. (2007). Superstitious, magical, and paranormal beliefs: An integrative model. *Journal of Research in Personality*, *41*(4), 731–744. https://doi.org/https://doi.org/10.1016/j.jrp.2006.06.009

Lipovetsky, S., & Conklin, M. (2001). Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, *17*(4), 319–330. https://doi.org/https://doi.org/10.1002/asmb.446

Lipton, P. (1990). Contrastive explanation. *Royal Institute of Philosophy Supplement*, *27*, 247–266. https://doi.org/10.1017/S1358246100005130

Lipton, Z. C. (2016). The mythos of model interpretability. https://doi.org/10.48550/ARXIV.1606.03490

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. https://doi.org/10.48550/ARXIV.1907.11692

Lopez-Paz, D., Nishihara, R., Chintala, S., Schölkopf, B., & Bottou, L. (2016). Discovering causal signals in images. https://doi.org/10.48550/ARXIV.1605.08179

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, *1*(4), 309–317. https://doi.org/10.1147/rd.14.0309

Lundberg, S., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. https://doi.org/10.48550/ARXIV.1705.07874

Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B. J., Wong, K.-F., & Cha, M. (2016). Detecting rumors from microblogs with recurrent neural networks. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 3818–3824.

Ma, J., Gao, W., Wei, Z., Lu, Y., & Wong, K.-F. (2015). Detect rumors using time series of social context information on microblogging websites. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 1751–1754. https://doi.org/10.1145/2806416.2806607

Magdy, A., & Wanas, N. (2010). Web-based statistical fact checking of textual documents. *Proceedings of the 2nd International Workshop on Search and Mining User-Generated Contents*, 103–110. https://doi.org/10.1145/1871985.1872002

Mallat, S. (1999). *A wavelet tour of signal processing*. Elsevier.

Mann, W. C., & Thompson, S. A. (1988). *Text - Interdisciplinary Journal for the Study of Discourse*, *8*(3), 243–281. https://doi.org/doi:10.1515/text.1.1988.8.3.243

McNee, S. M., Riedl, J., & Konstan, J. A. (2006). Being accurate is not enough: How accuracy metrics have hurt recommender systems. *CHI'06 extended abstracts on Human factors in computing systems*, 1097–1101.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. https://doi.org/10.48550/ARXIV.1301.3781

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *CoRR*, *abs/1310.4546*.

Miller, T. (2017). Explanation in artificial intelligence: Insights from the social sciences. https://doi.org/10.48550/ARXIV.1706.07269

Miller, T., Howe, P., & Sonenberg, L. (2017). Explainable ai: Beware of inmates running the asylum or: How i learnt to stop worrying and love the social and behavioural sciences. https://doi.org/10.48550/ARXIV.1712.00547

Mohammad, S. M., Sobhani, P., & Kiritchenko, S. (2016). Stance and sentiment in tweets. *CoRR*, *abs/1605.01655*.

Mohseni, S., Block, J. E., & Ragan, E. D. (2018). A human-grounded evaluation benchmark for local explanations of machine learning. https://doi.org/10.48550/ARXIV.1801.05075

Molnar, C. (2022). *Interpretable machine learning: A guide for making black box models explainable* (2nd ed.).

Montavon, G., Samek, W., & Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, *73*, 1–15. https://doi.org/https://doi.org/10.1016/j.dsp.2017.10.011

Monti, F., Boscaini, D., Masci, J., Rodolà, E., Svoboda, J., & Bronstein, M. M. (2016). Geometric deep learning on graphs and manifolds using mixture model cnns. https://doi.org/10.48550/ARXIV.1611.08402

Monti, F., Frasca, F., Eynard, D., Mannion, D., & Bronstein, M. M. (2019). Fake news detection on social media using geometric deep learning. *CoRR*, *abs/1902.06673*.

Mustafaraj, E., & Metaxas, P. T. (2017). The fake news spreading plague: Was it preventable? *CoRR*, *abs/1703.06988*.

Nagel, S. (2016). Cc-news.

Nair, V., & Hinton, G. (2010). Rectified linear units improve restricted boltzmann machines vinod nair. *Proceedings of ICML*, *27*, 807–814.

Nakamura, K., Levy, S., & Wang, W. Y. (2020). Fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection. *Proceedings of the 12th Language Resources and Evaluation Conference*, 6149–6157.

Newman, M. L., Pennebaker, J. W., Berry, D. S., & Richards, J. M. (2003). Lying words: Predicting deception from linguistic styles [PMID: 15272998]. *Personality and Social Psychology Bulletin*, *29*(5), 665–675. https://doi.org/10.1177/0146167203029005010

Newman, N., Fletcher, R., Robertson, C. T., Eddy, K., & Nielsen, R. K. (2022). Reuters institute digital news report 2022. *Digital News Report 2022*.

Nickerson, R. S. (1998). Confirmation bias: A ubiquitous phenomenon in many guises. *Review of General Psychology*, *2*(2), 175–220. https://doi.org/10.1037/1089-2680.2.2.175

Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J., & Sohl-Dickstein, J. (2018). Sensitivity and generalization in neural networks: An empirical study. https://doi.org/10.48550/ARXIV.1802.08760

Nyhan, B., & Reifler, J. (2010). When corrections fail: The persistence of political misperceptions. *Political Behavior*, *32*(2), 303–330. https://doi.org/10.1007/s11109-010-9112-2

Oesper, L., Merico, D., Isserlin, R., & Bader, G. D. (2011). Wordcloud: A cytoscape plugin to create a visual semantic summary of networks. *Source code for biology and medicine*, *6*(1), 7.

Olson, R. S., La Cava, W., Orzechowski, P., Urbanowicz, R. J., & Moore, J. H. (2017). PMLB: A large benchmark suite for machine learning evaluation and comparison. *BioData Mining*, *10*(1), 36.

Oneto, L., & Chiappa, S. (2020). Fairness in machine learning. In *Recent trends in learning from data* (pp. 155–196). Springer International Publishing. https://doi.org/10.1007/978-3-030-43883-8_7

Owen, G. (1977). Values of games with a priori unions. In R. Henn & O. Moeschlin (Eds.), *Mathematical economics and game theory* (pp. 76–88). Springer Berlin Heidelberg.

Palau-Sampio, D. (2016). Reference press metamorphosis in the digital context: Clickbait and tabloid strategies in elpais.com. *Communication & Society*, *29*, 63–79. https://doi.org/10.15581/003.29.2.63-79

Pariser, E. (2011). *The filter bubble: What the internet is hiding from you*. Penguin UK.

Pascanu, R., Mikolov, T., & Bengio, Y. (2012). On the difficulty of training recurrent neural networks. https://doi.org/10.48550/ARXIV.1211.5063

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. https://doi.org/10.48550/ARXIV.1912.01703

Paul, C., & Matthews, M. (2016). The russian "firehose of falsehood" propaganda model: Why it might work and options to counter it.

Pearl, J. (2009). *Causality: Models, reasoning and inference* (2nd). Cambridge University Press.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pennebaker, J. W., Booth, R. J., & Francis, M. E. (2007). Linguistic inquiry and word count (liwc2007).

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Pennycook, G., & Rand, D. G. (2019). Fighting misinformation on social media using crowdsourced judgments of news source quality. *Proceedings of the National Academy of Sciences*, *116*(7), 2521–2526. https://doi.org/10.1073/pnas.1806781116

Pennycook, G., & Rand, D. G. (2021). The psychology of fake news. *Trends in Cognitive Sciences*, *25*(5), 388–402. https://doi.org/https://doi.org/10.1016/j.tics.2021.02.007

Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., & Stein, B. (2017). A stylometric inquiry into hyperpartisan and fake news. *CoRR*, *abs/1702.05638*.

Qazvinian, V., Rosengren, E., Radev, D. R., & Mei, Q. (2011). Rumor has it: Identifying misinformation in microblogs. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 1589–1599.

Qi, P., Cao, J., Yang, T., Guo, J., & Li, J. (2019). Exploiting multi-domain visual information for fake news detection. *CoRR*, *abs/1908.04472*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners.

Rao, Z., He, M., & Zhu, Z. (2019). Input-perturbation-sensitivity for performance analysis of cnns on image recognition. *2019 IEEE International Conference on Image Processing (ICIP)*, 2496–2500. https://doi.org/10.1109/ICIP.2019.8803012

Read, M. (2016). Donald trump won because of facebook.

Reed, E. S., Turiel, E., & Brown, T. (2013). Naive realism in everyday life: Implications for social conflict and misunderstanding.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. https://doi.org/10.1145/2939672.2939778

Robbins, H., & Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, *22*(3), 400–407. https://doi.org/10.1214/aoms/1177729586

Rony, M. M. U., Hassan, N., & Yousuf, M. (2017). Diving deep into clickbaits: Who use them to what extents in which topics with what effects? *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, 232–239. https://doi.org/10.1145/3110025.3110054

Rubin, V., Conroy, N., & Chen, Y. (2015). Towards news verification: Deception detection methods for news discourse. https://doi.org/10.13140/2.1.4822.8166

Rubin, V., Conroy, N., Chen, Y., & Cornwell, S. (2016). Fake news or truth? using satirical cues to detect potentially misleading news. *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*, 7–17. https://doi.org/10.18653/v1/W16-0802

Rubin, V. L. (2010). On deception and deception detection: Content analysis of computer-mediated stated beliefs. *Proceedings of the 73rd ASIST Annual Meeting on Navigating Streams in an Information Ecosystem - Volume 47*.

Rubin, V. L., Chen, Y., & Conroy, N. K. (2015). Deception detection for news: Three types of fakes. *Proceedings of the Association for Information Science and Technology*, *52*(1), 1–4. https://doi.org/https://doi.org/10.1002/pra2.2015.145052010083

Rubin, V. L., & Lukoianova, T. (2015). Truth and deception at the rhetorical structure level. *J. Assoc. Inf. Sci. Technol.*, *66*(5), 905–917. https://doi.org/10.1002/asi.23216

Rubin, V. L., & Vashchilko, T. (2012). Identification of truth and deception in text: Application of vector space model to rhetorical structure theory. *Proceedings of the Workshop on Computational Approaches to Deception Detection*, 97–106.

Rudin, C. (2018). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. https://doi.org/10.48550/ARXIV.1811.10154

Salzberg, S. L. (1994). C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning*, *16*(3), 235–240. https://doi.org/10.1007/BF00993309

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. https://doi.org/10.48550/ARXIV.1910.01108

Santia, G., & Williams, J. (2018). Buzzface: A news veracity dataset with facebook user commentary and egos. *Proceedings of the International AAAI Conference on Web and Social Media*, *12*(1), 531–540.

Sapir, A. (1987). *Scientific content analysis (SCAN)*. Laboratory of Scientific Interrogation.

Sawer, P. (2020). 'deepfake' queen's speech: Channel 4 criticised for 'disrespectful' christmas message.

Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. https://doi.org/10.48550/ARXIV.1508.07909

Shapley, L. S. (1952). *A value for n-person games*. RAND Corporation. https://doi.org/10.7249/P0295

Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. https://doi.org/10.48550/ARXIV.1704.02685

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2018). Fakenewsnet: A data repository with news content, social context and spatialtemporal information for studying fake news on social media. https://doi.org/10.48550/ARXIV.1809.01286

Shu, K., Mahudeswaran, D., Wang, S., & Liu, H. (2020). Hierarchical propagation networks for fake news detection: Investigation and exploitation. *Proceedings of the International AAAI Conference on Web and Social Media*, *14*(1), 626–637.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. *SIGKDD Explor. Newsl.*, *19*(1), 22–36. https://doi.org/10.1145/3137597.3137600

Shu, K., Wang, S., & Liu, H. (2019). Beyond news contents: The role of social context for fake news detection. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 312–320. https://doi.org/10.1145/3289600.3290994

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., & Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, *30*(3), 83–98.

Silverman, C. (2016). This analysis shows how viral fake election news stories outperformed real news on facebook.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. In Y. Bengio & Y. LeCun (Eds.), *2nd international conference on learning representations, ICLR 2014, banff, ab, canada, april 14-16, 2014, workshop track proceedings*.

Smith, N. (2001). *Reading between the lines: An evauluation of the scientific content analysis techniques (scan)*. Home Office.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(56), 1929–1958.

Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, *41*(3), 647–665. https://doi.org/10.1007/s10115-013-0679-x

Sundararajan, M., Taly, A., & Yan, Q. (2016). Gradients of counterfactuals. https://doi.org/10.48550/ARXIV.1611.02639

Sunstein, C. R. (2001). *Echo chambers: Bush v. gore, impeachment, and beyond*. Princeton University Press.

Sunstein, C. R., & Vermeule, A. (2009). Conspiracy theories: Causes and cures*. *Journal of Political Philosophy*, *17*(2), 202–227. https://doi.org/https://doi.org/10.1111/j.1467-9760.2008.00325.x

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. https://doi.org/10.48550/ARXIV.1409.3215

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. https://doi.org/10.48550/ARXIV.1704.07506

Team, P. (2022). Upfd source code.

Thorne, J., & Vlachos, A. (2018). Automated fact checking: Task formulations, methods and future directions. *CoRR*, *abs/1806.07687*.

Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). FEVER: A large-scale dataset for fact extraction and VERification. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 809–819. https://doi.org/10.18653/v1/N18-1074

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288. https://doi.org/https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Trabelsi, A., & Zaiane, O. R. (2014). Finding arguing expressions of divergent viewpoints in online debates. *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, 35–43. https://doi.org/10.3115/v1/W14-1305

Trinh, T. H., & Le, Q. V. (2018). A simple method for commonsense reasoning. https://doi.org/10.48550/ARXIV.1806.02847

Tversky, A., & Kahneman, D. (1992). Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, *5*, 297–323.

Twitter. (2022). Twitter api.

Undeutsch, U. (1954). *Die entwicklung der gerichtspsychologischen gutachtertaetigkeit*. Hogrefe.

van der Maaten, L., & Hinton, G. E. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*, 2579–2605.

van Rossum, G. (1995). *Python tutorial* (tech. rep. CS-R9526). Centrum voor Wiskunde en Informatica (CWI). Amsterdam.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. https://doi.org/10.48550/ARXIV.1706.03762

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2017). Graph attention networks. https://doi.org/10.48550/ARXIV.1710.10903

Vicario, M. D., Bessi, A., Zollo, F., Petroni, F., Scala, A., Caldarelli, G., Stanley, H. E., & Quattrociocchi, W. (2016). The spreading of misinformation online. *Proceedings of the National Academy of Sciences*, *113*(3), 554–559. https://doi.org/10.1073/pnas.1517441113

Vlachos, A., & Riedel, S. (2014). Fact checking: Task definition and dataset construction. *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, 18–22. https://doi.org/10.3115/v1/W14-2508

Walker, M., Anand, P., Abbott, R., & Grant, R. (2012). Stance classification using dialogic properties of persuasion. *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 592–596.

Walker, M., & Matsa, K. E. (2021). News consumption across social media in 2021.

Wang, W. Y. (2017). "liar, liar pants on fire": A new benchmark dataset for fake news detection. *CoRR*, *abs/1705.00648*.

Watson, A. (2022). Usage of social media as a news source worldwide 2022.

Weir, W. (2009). In *History's greatest lies: The startling truths behind world events our history books got wrong* (pp. 28–41). Fair Winds Press.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., … Rush, A. (2020). Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Xiao, H. (2018). Bert-as-service.

Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M., Stolcke, A., Yu, D., & Zweig, G. (2016). Achieving human parity in conversational speech recognition. *CoRR*, *abs/1610.05256*.

Yang, F., Liu, Y., Yu, X., & Yang, M. (2012). Automatic detection of rumor on sina weibo. *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*. https://doi.org/10.1145/2350190.2350203

Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019). Gnnexplainer: Generating explanations for graph neural networks. https://doi.org/10.48550/ARXIV.1903.03894

Yu, B. (2013). Stability. *Bernoulli*, *19*(4). https://doi.org/10.3150/13-bejsp14

Yuan, X., He, P., Zhu, Q., & Li, X. (2017). Adversarial examples: Attacks and defenses for deep learning. https://doi.org/10.48550/ARXIV.1712.07107

Zang, C., Cui, P., & Faloutsos, C. (2016). Beyond sigmoids: The nettide model for social network growth, and its applications. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015–2024. https://doi.org/10.1145/2939672.2939825

Zeng, H., Zhou, H., Srivastava, A., Kannan, R., & Prasanna, V. (2019). Graphsaint: Graph sampling based inductive learning method. https://doi.org/10.48550/ARXIV.1907.04931

Zhang, Z., Cui, P., & Zhu, W. (2018). Deep learning on graphs: A survey. https://doi.org/10.48550/ARXIV.1812.04202

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2018). Graph neural networks: A review of methods and applications. https://doi.org/10.48550/ARXIV.1812.08434

Zhou, L., Booker, Q., & Zhang, D. (2002). Rod - toward rapid ontology development for underdeveloped domains. *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, 957–965. https://doi.org/10.1109/HICSS.2002.994046

Zhou, L., Burgoon, J. K., Nunamaker, J. F., & Twitchell, D. (2004). Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated communications. *Group Decision and Negotiation*, *13*(1), 81–106. https://doi.org/10.1023/B:GRUP.0000011944.62889.6f

Zhu, J., Liapis, A., Risi, S., Bidarra, R., & Youngblood, G. M. (2018). Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation. *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, 1–8. https://doi.org/10.1109/CIG.2018.8490433

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *arXiv preprint arXiv:1506.06724*.