MANIFEST
A Human-centric Explainable Approach for Fake News Spreading Detection

Master Thesis

Candidate
Orestis Lampridis

Supervisor
Athena Vakali

March 2021
MANIFEST: a huMAN-centric explaInable approach for FakE news Spreading deTection

Orestis Lampridis

12th March 2021
Abstract

Fake news spreading is an ever increasing phenomenon which is deeply tied with the involvement of humans as they tend to adopt, circulate and fall for misinformation stories. However, there has not been significant work in the literature on the role that the human factor has and what characteristics of human users correlate with the diffusion of fake news. Such research would be valuable to better understand and combat misinformation patterns in human dominated platforms such as social networks. Recent work has shown that behavioral user profiling leads to promising results in identifying fake news spreaders. In spite of this, no in-depth analysis has been made to figure out the exact features that correlate with fake news spreading behavior. This work suggests an explainable human-centric approach on detecting fake news spreading behavior by building a fake news spreader classifier, utilizing the psychological characteristics of human users and applying state of the art explanation techniques. Experimentation demonstrates that our model achieves promising results at detecting fake news spreaders by utilizing user characteristics while also offering explanations of which of those characteristics contribute to the fake news spreading behavior.

In addition, to the best of our knowledge, this is the first work that aims to provide a fully explainable setup that evaluates fake news spreading based on users credibility applied to public discussions on Twitter threads and aiming for a comprehensive way to combat fake news circulation. The way we approach this is by utilizing the predictions made by the fake news spreader classifier built before on the users that took part in the discussion of a specific twitter thread and then by learning an interpretable linear model in that space. The explanations consist of example-based explanations and word feature importance. Quantitative evaluation shows that the linear model is able to accurately imitate the predictions of the more complex fake news spreader classifier with a high accuracy. Qualitative evaluation shows that the explanations are reasonable and intuitive and could prove fruitful for combating the propagation of fake news.

**Keywords:** fake news, social networks, user profiling, machine learning, explainable AI
Εκτεταμένη περίληψη

Η εξάπλωση παραπληροφόρησης και ψεύτικων ειδήσεων, γνωστή επίσης και με τον όρο "fake news", είναι ένα συνεχώς αυξανόμενο φαινόμενο που συνδέεται βαθιά με τη συμμετοχή του ανθρώπινου στοιχείου καθώς οι άνθρωποι είναι οι πρωταγωνιστές των ειδήσεων που τείνουν να εισβάλουν, να κυκλοφορούν και να πιστεύουν ανυπόστατες φήμες. Πράγματι, πρωτόγονα φαινόμενα συμβαίνουν από την κυκλοφορία ψεύτικων ειδήσεων που κλιμακώνονται στην πανδημική εποχή μας. Η διάδοση ψεύτικων ειδήσεων COVID-19 έχει οδηγήσει σε απώλειες ανθρώπινων ζωών λόγω ψεύδων πληροφοριών που κοινοποιούνται σχετικά με πιθανές θεραπείες. Επίσης, περίπου το 17% των Αμερικανών βρέθηκε να πιστεύει ότι η πανδημία COVID-19 σχεδιάστηκε σκόπιμα από ισχυρούς ανθρώπους. Ταυτόχρονα, οι χρήσεις της τεχνολογίας και της επικοινωνίας συνετάστηκαν από ισχυρούς ανθρώπους. Το πρόβλημα υπάρχει στα κοινωνικά δίκτυα (twitter, facebook, reddit etc.) τα οποία προτιμώνται από τους χρήστες με όλο και αυξανόμενο ρυθμό καθώς αποτελούν μια πολύ εύκολη διέξοδο πληροφόρησης σε σχέση με πιο παραδοσιακούς τρόπους πληροφόρησης. 

Για να αντιμετωπιστεί το πρόβλημα έχουν τεθεί διάφορες κατηγορίες τεχνικών όπως διασταυρωτές πληροφοριών, ταυτοποίηση βάσης μοντέλοποίησης γράφων και προβλέψεις βάσης μοντέλων μηχανικής μάθησης. Το συντριπτικό ποσοστό της βιβλιογραφίας που χρησιμοποιεί μοντέλα πρόβλεψης με μηχανική μάθηση ασχολείται με τον εντοπισμό ψεύτικης πληροφορίας στο επίπεδο της δημοσίευσης (κείμενο). Ωστόσο, μέχρι σήμερα δεν έχει γίνει σημαντική συμπεριφορά προφίλ χρηστών σχετικά με το ρόλο που έχει ο ανθρώπινος παράγοντας και ποια χαρακτηριστικά των ανθρώπων συσχετίζονται με τη διάδοση ψεύτικων ειδήσεων. Μια έρευνα θα ήταν πολύτιμη για την καλύτερη κατανόηση και καταπολέμηση των παραπληροφορίων στα κοινωνικά δίκτυα και προσδιορισμό των ακριβών χαρακτηριστικών των ανθρώπων στα κοινωνικά δίκτυα. Οι λίγες έρευνες που έχουν γίνει δείχνουν ότι η ανάλυση συμπεριφοράς προφίλ χρηστών οδήγησε σε υποσχέσεις αποτελέσματα στον εντοπισμό ψεύτικων που διαδίδονταν ψεύτικες ειδήσεις. Ωστόσο, δεν έχει γίνει στην βιβλιογραφία κάποια εκτεταμένη ανάλυση για να προσδιοριστούν τα ακριβή χαρακτηριστικά
που σχετίζονται με τη συμπεριφορά χρηστών που διαδίδουν αυτές τις ψεύτικες ειδήσεις. Αυτή η δουλειά προτείνει μια επεξηγούμενη άνθρωπο-κεντρική προσέγγιση για τον εντοπισμό διάδοσης ψεύτικων ειδήσεων δημιουργώντας έναν ταξινομητή που εντοπίζει τους χρήστες που εμπλέκονται στη διάδοση ψεύτικων ειδήσεων. Για αυτόν το σκοπό χρησιμοποιούνται τεχνικές επεξεργασίας φυσικής γλώσσας για να εξαχθούν τα ψυχολογικά και κοινωνικά χαρακτηριστικά των ανθρώπινων χρηστών (προσωπικότητα, φύλο, συναισθήματα, ανάλυση χρήσης λέξεων και χαρακτηριστικά αναγνωσιμότητας). Έπειτα δοκιμάστηκαν διάφοροι αλγόριθμοι μηχανικής μάθησης ώστε να βρεθεί ο πιο αποδοτικός. Για τον συσχετισμό των χαρακτηριστικών των χρηστών με τη συμπεριφορά διάδοσης ψεύτικες πληροφορίες χρησιμοποιούνται δύο τεχνικές επεξήγησης state-of-the-art στο πεδίο της ερμηνεύσιμης τεχνητής νοημοσύνης, το ELI5 και το SHAP.

Πιεραματική αξιολόγηση που διεξάχθηκε δείχνει ότι το μοντέλο μας επιτυγχάνει υποσχόμενη αποτελεσματικότητα στον μοντέλο χρηστών που διαδίδουν ψεύτικες πληροφορίες όταν συγκρίνεται με αντίστοιχες πρόσφατες δουλειές στην προσέγγιση η οποία αξιολογεί τη διάδοση ψεύτικων ειδήσεων με βάση την αξιοπιστία των χρηστών που συμμετέχουν σε δημόσιες συζήτησεις στο Twitter και στη συνεχίζουν σε έναν ολοκληρωμένο τρόπο καταπολέμηση της ψεύτικης διάδοσης. Ο τρόπος με τον οποίο γίνεται αυτό είναι με την εξέλιξη των προβλέψεων των χρηστών που αξιολογούνται σε ένα επίπεδο παραπληροφόρησης καταγράφοντας τις επιρροές και τις αναμεσιστικές καθώς και τη συγκεκριμένη χρήση γλώσσας είναι χαρακτηριστικά που διαδίδονται στη συμπεριφορά διάδοσης ψεύτικων ειδήσεων.

Επιπλέον, εξ όσων γνωρίζουμε, αυτό είναι η πρώτη δουλειά που στοχεύει στην παροχή μιας πλήρως επεξηγούμενης προσέγγισης η οποία εξελίχθηκε με βάση την εξέλιξη των προβλέψεων των χρηστών που συμμετέχουν σε δημόσιες συζήτησεις στο Twitter και στη συνεχίζουν σε έναν υπολογισμό τρόπο καταπολέμησης της κυκλοφορίας ψεύτικων ειδήσεων. Ο τρόπος με τον οποίο γίνεται αυτό είναι με την εξέλιξη των προβλέψεων των χρηστών που συμμετέχουν σε δημόσιες συζήτησεις στο Twitter και στη συνεχίζουν σε έναν υπόλοιπο τρόπο καταπολέμησης της κυκλοφορίας ψεύτικων ειδήσεων.
Πειραματική αξιολόγηση έγινε σε δύο κύρια επίπεδα: ποσοτική και ποιοτική. Η ποσοτική αξιολόγηση υποδεικνύει ότι το γραμμικό μοντέλο είναι σε θέση να μιμείται με υψηλή ακρίβεια τις προβλέψεις του πιο περίπλοκου ταξινομητή που εντοπίζει τους χρήστες που διαδίδουν παραπληροφόρηση, γεγονός που υποδηλώνει ότι η υπόθεση ότι ένας πλαστογράφος θα συνεχίσει να διαδίδει ψεύτικες ειδήσεις όταν απαντά σε άλλους χρήστες και το αντίστοιχο για τους πραγματικούς διανομείς ειδήσεων ισχύει. Η ποιοτική αξιολόγηση δείχνει ότι οι εξήγησες είναι λογικές και διαισθητικές και ότι θα μπορούσαν να αποδειχθούν καρποφόρες για να βοηθήσουν έναν χρήστη να καταλάβει αν ένα tweet που διαβάζει περιέχει ή όχι ψεύτικες πληροφορίες και γενικότερα να βοηθήσουν στην καταπολέμηση της διάδοσης ψευδών ειδήσεων στα κοινωνικά δίκτυα.

Λέξεις Κλειδιά: παραπληροφόρηση, κοινωνικά δίκτυα, ανάλυση προφίλ χρηστών, μηχανική μάθηση, ερμηνεύσιμη τεχνητή νοημοσύνη
Acknowledgements

Firstly, I would like to give thanks to my supervisor prof. Athena Vakali and PhD Student Dimitra Karanatsiou for allowing me to venture into my own ideas and for the feedback and suggestions they gave me along the way. I would also like to thank all my MSc classmates and especially those we worked on projects together for all the hard but also good times we had through our journey while completing this master program. I wish them the best in their future endeavors. Lastly, I would like to thank my family, friends and colleagues for their help, tolerance and overall support during this challenging time.
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.3</td>
<td>Feature engineering</td>
<td>58</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Fake News Spreader classifier</td>
<td>61</td>
</tr>
<tr>
<td>3.2.5</td>
<td>Explanations</td>
<td>61</td>
</tr>
<tr>
<td>3.3</td>
<td>Creation and annotation of tweet post-replies dataset</td>
<td>63</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Dataset collection and details</td>
<td>63</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Creating the User replies dataset</td>
<td>63</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Dataset transformation and annotation</td>
<td>64</td>
</tr>
<tr>
<td>3.4</td>
<td>Training of interpretable white box model</td>
<td>67</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Linear model</td>
<td>67</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Final explanation</td>
<td>69</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Fidelity</td>
<td>71</td>
</tr>
<tr>
<td>4</td>
<td>Experimentation and results</td>
<td>72</td>
</tr>
<tr>
<td>4.1</td>
<td>Hyper-parameter tuning</td>
<td>72</td>
</tr>
<tr>
<td>4.2</td>
<td>Predictive performances</td>
<td>74</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Gender classifiers</td>
<td>74</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Fake News Spreader classifiers</td>
<td>75</td>
</tr>
<tr>
<td>4.3</td>
<td>Explanations</td>
<td>78</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Global level</td>
<td>79</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Partial dependency plots</td>
<td>82</td>
</tr>
<tr>
<td>4.4</td>
<td>MANIFEST evaluation</td>
<td>85</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Fidelity evaluation</td>
<td>85</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Homophily evaluation</td>
<td>87</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Qualitative evaluation</td>
<td>88</td>
</tr>
<tr>
<td>5</td>
<td>Conclusions</td>
<td>95</td>
</tr>
<tr>
<td>5.1</td>
<td>Limitations and Future Work</td>
<td>96</td>
</tr>
</tbody>
</table>
List of Figures

26
2.2 Historical conspiracy theories and share of people that believe in them. Source: [3] .............................. 29
2.4 The pipeline of what explainable machine learning aims to accomplish [78] ................................. 43
2.5 How LIME is able to approximate the neighborhood of the instance we want to explain (the bold red cross) and generate synthetic neighbors ............................................................... 49
3.1 Architecture depicting phases A and B ............................... 54
3.2 Architecture depicting the final phase C ............................... 55
3.3 ELI5’s feature importance on the trained fake news spreader classifier ................................................. 62
3.4 PD plot that shows how the feature capitalized_count affects the final prediction ............................... 62
3.5 Example of Linear Regression ........................................ 68
3.6 The differences between linear regression and logistic regression ................................................. 69
3.7 An example of how fidelity can be calculated given the predictions of a complex and a simple model ............................... 71
4.1 The feature importance of the top 20 features as given by ELI5 .... 79
4.2 The feature importance of the top 20 tabular features as given by SHAP ........................................ 80
4.3 The summary plot given by SHAP for the tabular features .... 81
4.4 Global explanations for the textual features as given by ELI5 and SHAP ........................................ 82
4.5 Textblob polarity score partial dependency plot ............ 83
4.6 Hashtag count partial dependency plot ............................. 84
4.7 Tone partial dependency plot ......................................... 84
4.8 Slang count partial dependency plot ................................. 85
4.9 Accuracy curves of the logistic regression model  . . . . . . . . .  87
List of Tables

2.1 Overview of the related work for the task of fake news detection . 51
4.1 The performances of the gender classifiers after performing the random search. ................................. 74
4.2 The performances of the gender classifiers after performing grid search. ................................................... 75
4.3 The performances of the fake news spreader classifiers with only tabular features after performing the random search. ................. 76
4.4 The performances of the fake news spreader classifiers with only tabular features after performing the grid search. .................... 76
4.5 The performances of the fake news spreader classifiers for phase C with both text and tabular features after performing the random search. ...................................................... 77
4.6 The performances of the fake news spreader classifiers for phase C with both text and tabular features after performing the grid search. 77
4.7 The performances of the final picked fake news spreader classifiers for phase A, including only text features (left), only tabular features (center) and phase C with both text and tabular features (right). ....................................................... 77
4.8 The performance of the final picked fake news spreader classifier (GB) compared with the performance of the top contenders in the pan-clef challenge .............................................................. 78
4.9 Fidelity average for both datasets and both vector representations. The higher the better. ................................. 86
4.10 Homophily evaluation for both datasets. The higher the better. ............................................................... 88
4.11 The explanation for 3 given tweets of the us_election dataset classified as fake news spreading. ..................... 89
4.12 The explanation for 3 given tweets of the us_election dataset classified as real news spreading. ....................... 90
4.13 The explanation for 3 given tweets of the covid dataset classified as fake news spreading. ............................ 92
4.14 The explanation for 3 given tweets of the covid dataset classified as real news spreading. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
Chapter 1

Introduction

With the sharp rise of social networks, all different kinds of social media users are able to access, share, and discuss virtually any kind of information. Meanwhile, due to the ease and comfort of online social media, more and more individuals seek to keep up to date on the state of the news via social networks. In 2020, over 3.96 billion individuals were effectively using online social media worldwide, an increment of 10.9% from the previous year’s 3.48 billion. Back in 2015, there were just 2.07 billion individuals using social media. This shows an impressive general increase of over 92% in only a five year time-span.\(^1\)

However, lurking in the shadows of social networks lies a vast amount of deception, misinformation and disinformation, including the widely familiar term known as fake news [48]. At the same time, citizen journalism, the idea of citizens assuming a functioning part in the process of investigating, collecting, sharing, and spreading information and news, is also on the rise [12]. Even though citizen journalism is a great step forward with regards to democratic values it is also a process that tends to promote fake news instead of just advancing a more educated populace [24].

The rising spread of fake news in society and the effects they have can be profound. This is evident from several societal and political situations (e.g. fake news US 2016 and 2020 elections and the COVID-19 pandemic [3, 8, 71, 99]). The fake news impact has become really critical. Unprecedented phenomena occur by the circulation of fake news which escalate in our pandemic era. COVID-19 fake news spreading has even led to the loss of human lives due to the false information shared regarding potential treatments [40]. Also, about 17% of Americans were found to believe that the COVID-19 pandemic was intentionally planned by powerful people\(^2\). Fake news regarding the result of the US 2020 elections, includ-

\(^{1}\)https://backlinko.com/social-media-users
\(^{2}\)https://m.naftemporiki.gr/story/1627485/discussing-covid-19-conspiracy-theories
ing voter fraud have also led individuals to unprecedented actions such as taking arms and storming the US capitol; an action that threatened the very democracy of the United States. Thus, the need for effective fake news detection methods is a critical demand which requires efficient responses.

Meanwhile, the recent years have seen a quick expansion in the utilization of machine learning models in a wide scope of fields, including fake news detection. A big part of those models are *black boxes*, in the sense that their inner working and the rationale behind their choices for a given scenario are not clearly justifiable. This reality, alongside recent European law guidelines [33, 58] have given birth to the rising field known as *explainable machine learning*. Explainable machine learning aims to make the inner logic of the so-called "black box" algorithms understandable, or create simpler similar ones that are interpretable.

### 1.1 Topic Description

For tackling the fake news detection problem a variety of different techniques have been proposed. They can be grouped into two large categories based on the method they use to attack the problem. First and foremost, there are those that treat the problem as a natural language processing problem and use natural language features by focusing solely on the content of the text. Second, there are those that take advantage of the social setting encompassing the news post and use various different features stemming from knowledge graphs or metadata surrounding the news post [90]. There are also hybrid methods that combine features and qualities of both categories.

The vast majority of the literature is concerned with locating fake news texts in the post-level (i.e. by placing emphasis on the content of the news post) [2, 37, 70, 73, 83, 106, 109]. However, there is not adequate study on identifying fake news at the user level [91]. The consideration of users roles and profiles is important since earlier related work has shown that behavioral human-centric variations can largely impact fake news origination, spreading, and virality [17]. Recent research has also shown that taking into account the user’s engagement to a news article leads to promising results (i.e. the comments they post on a news article or the replies they leave on a Twitter post) [85, 92].

An initial work in the incorporation of user profiles in the fake news detection task was made by Shu et al. in [91] in which they try to find the correlation between the user profile characteristics and whether that person is a fake news spreader or a real news spreader. However, this work uses fully labeled datasets and uses statistical analysis methods in order to uncover the correlations. A step forward

---

CHAPTER 1. INTRODUCTION

was made in [93] where machine learning interpretable models were used to find the feature importance of various features related to the user profiles of fake news spreaders.

The findings in [93] also suggest that personality has minimal discriminative power in locating whether a post contains fake news or real news which is contradictory with other related work. Such is the case in [31] where they find that using personality traits significantly improves the accuracy of detecting whether a news post contains false news or not.

Attempts to incorporate explainable machine learning techniques in fake news detection and aiding in the process of understanding why a machine learning model assigns a specific label in a particular news post have been made recently. DE-FEND [92] utilizes the comments made by the users in the fake news post in order to offer the final explanation while GCAN [55] merely uses words found in the source text. Finally, a review that utilizes SHAP [56] in order to find the most important features already proposed in various previous works by selecting them uniformly at random was made in [76]. However, no previous work made a comprehensive effort to provide a human understandable approach that would inform users about their or others’ susceptibility to fake news.

1.2 Thesis Contribution

To approach the fake news detection problem at a real setting, we decide to focus on Twitter as it is a platform rich with fake news [6, 17] and also a great source for extracting information regarding user characteristics.

In this work, firstly a fake news spreading detection approach is proposed, by exploiting the user profiles of individuals that spread real news and false news to extract features pertaining to their psychological characteristics. By using these features, a fake news spreader classifier is built that when given a Twitter user can label them as a fake or real news spreader. Differently from the methods focusing on the user-level mentioned before, we have an added goal to find the traits that characterize fake news spreaders (i.e. what are the characteristics of users that spreads fake news?). This is made possible by using state-of-the-art explanation techniques such as SHAP [56] and ELI5.

Afterwards, we design a novel framework for detecting suspicious users and misinformation elements on public discussions such as Twitter threads in a fully explainable and human-comprehensive set up. To do this, we take influence from other related work which utilizes the replies posted by other users on the initial tweet that needs to be classified [37, 85, 92]. The proposed work is then based on the next two assumptions:
• Assumption 1: Individuals marked as fake news spreaders have a high enough chance of leaving a reply in another individual’s post which will potentially contain fake news.

• Assumption 2: Individuals marked as real news spreaders have a high enough chance of leaving a reply in another individual’s post which will potentially contain real news.

The inspiration behind the above is that since to build the fake news spreader classifier we analyse the last 100 tweets that a user posted on Twitter to classify them as real or fake news spreaders then it is very likely that their next post (the reply they leave on another post) will also follow the trend of the classification made. The veracity of these assumptions are evaluated at the end of our approach. With these assumptions in mind, we shift our focus to utilizing the replies of Twitter posts. From each reply of a Twitter post, we extract its user, apply the fake news spreader classifier on that user and use that label to link that to the replies they leave to the original Twitter post.

Afterwards, using those labels, we build an interpretable linear model that is trained only on the text of the replies given to an initial post. The main intuition behind this idea is that every news post forms a form of topic where the initial poster along with all users that replied on that post discuss that specific topic. Thus, by correlating the texts that fake news spreaders reply with and the texts that real news spreaders reply with, we can offer an explanation in the form of which words in the original post gravitate to which side of the discussion (i.e. feature importance). This way, aside from just tagging a tweet as fake news, we also show to the end-user which words contribute to which label and aid him in making the final decision. Aside from the feature importance, we also choose to offer the user the two closest replies in meaning (w.r.t cosine similarity of the text’s vector representation) from each label as example based explanations in order help the user better make his decision.

The final output when given as input a tweet with replies consists of the label of the user that left the tweet under inspection, as given by the fake news spreader classifier, the top words found in the input tweet’s text along with their weights as assigned by the linear model, which show how they influenced the prediction and finally the closest replies from each label. All these render our approach an extensional method that is not meant to directly detect fake news but rather aims for a higher generality in helping the user understand the factors that push the classification to either side and help them make the decision by themselves rather than merely tagging a post as fake news or not.

The research contributions of this work are summarized as next:
We propose a novel approach for explaining the predictions of fake news spreaders classifiers called MANIFEST which takes into advantage Twitter threads by exploiting user replies given to the original news post along with their labels as predicted by the trained fake news spreader classifier in order to build a neighborhood around that post. In that neighborhood, which is semantically close (since users replying to a post are likely talking about the content of that post), a linear model classifier is used in order to predict the class of the initial post. The explainable part first includes the word feature importance highlight, which is offered to the user as an explanation based on the words contained in the original post that show which words in the original post contribute to which outcome. Afterwards, the two closest replies from each class are offered to the user as example based explanations so the end-user can see what the other users are saying and understand better the decision behind the classification. Both explanation forms are meant to help the user understand why the tweet they are reading potentially contains fake or real news.

1.3 Thesis Structure

This thesis is structured into the following chapters:

- **Chapter 2 - Literature Review.** In this chapter, all the topics that are under inspection in this research are thoroughly explored by resorting to related literature. We initially present an introduction to the research field of fake news detection, including an introduction to fake news, the types of fake news
available and ways proposed to detect them by various researchers. This includes various features that can be extracted from the fake news sources as well as methods that successfully use these features in order to make the final classification. Thereafter, relevant work done in the field of psychology is analyzed to shed some light behind the spreading of fake news. Afterwards, the rising field of explainable machine learning is analyzed in depth with a brief introduction to machine learning in general. Finally, tying it all together the need for explainable fake news detection will be discussed.

• **Chapter 3 - Design and methodology.** In this chapter, we depict the overall design of our approach to deal with effectively detecting fake news and providing explanations by presenting it into three separate stages. At that point, we dive into more specialized insights about the implementation details of our methodology.

• **Chapter 4 - Experimentation and results.** Following from the previous chapter, here we explore various experiments that helped shape the final implementation details along with the various results that were derived from the experimentation done on our final model.

• **Chapter 5 - Discussion.** Here, we generally discuss our main findings regarding fake news spreading which includes how vulnerable individuals are at believing, supporting and spreading misinformation.

• **Chapter 6 - Conclusions.** Lastly, in this chapter, we present our final thoughts stemming from our assessment of the results found in the experimentation. We also recommend conceivable future enhancements and future work possible along with limitations of the current state of this work.
Chapter 2

Background - Literature Review

In this chapter, we present an extensive literature review which enlights several aspects of this thesis topics. This spans various research areas including fake news detection, psychological background importance and the relevant key themes of this research.

2.1 Fake News Detection

Fake news detection is a fast growing research area that has been attracting the attention of researchers and general public alike. Interest has been put on the different types of fake news and methods to detect them, by exploring the most relevant features and applying effective algorithms.

2.1.1 Introduction

The Internet is undoubtedly one of the greatest sources of information ever created. As we live in the digital era and there is an ever-increasing amount of our time spent online, most people choose to get informed on the current state of the news by using the web, either with the usage of various social media networks and platforms (e.g. Facebook, Reddit, Twitter, LinkedIn, etc.), or via news blogs and even online newspapers. This comes in contrast with using traditional news outlets (e.g. newspapers, television, magazines, billboards, posters, websites catered specifically for sharing the news) to stay informed. This change can be attributed on the main principle on which social networks were built on.

Young adults aged 18-24 years old are found to be online almost 8 hours a day. Over 85% of them are also found using another device while watching television with most of them using social media or reading the news on that device.¹

¹https://www.digitalmarketingcommunity.com/researches/digital-vs-
Meanwhile, millennials and especially Gen Z use social media as their sole source for getting informed on the state of the news. Furthermore, 44% of Gen Z check their social media accounts at least once in the span of an hour. Undoubtedly these generations make up a large part of the active online population in social networks as Gen Z alone constitutes almost 32% of the global population. Thus, news outlets need to take the characteristics of these generations into account in order to promote their articles more effectively.

Indeed, it is far less time consuming getting informed online via social media and there is also the capability to easily share what users read with others such as friends, relatives or even strangers. These online platforms also offer the users the capability of commenting on these types of news and offering their opinions on them to other users. As people consume these types of “easy” news in social media they also lay the ground for the spreading and circulation of fake news.

But what exactly constitutes fake news? There is currently no prevalent definition as to what exactly characterizes the nature of fake news. A general consensus is that the term fake news is a form of news that consists of the deliberate spreading of misinformation either through traditional news outlets or online social media [3, 49]. Another categorization can be made on whether the information being shared has the intent to harm or not as seen in Figure 2.1.

It can become apparent that the exact definition easily becomes blurry as different researchers give different definitions and there are many of them currently proposed [30]. A definition that is often used in the relevant literature is given by Shu et al. in their extensive survey on fake news detection [90] and is as follows:

**Definition 1.** Fake news is a news article that is intentionally and verifiably false.

Gelfert [30] also made an extensive research on all the available definitions and came up with the following definition by trying to capture most of the distinctive features given in all the previous ones.

**Definition 2.** Fake news is the deliberate presentation of (typically) false or misleading claims as news, where the claims are misleading by design.
2.1.2 History and Evolution of Fake News Spreading

The term fake news has recently gained popularity with the 2016 US presidential elections and by US president Donald Trump during his presidential campaign and after his election. President Donald Trump uses the term to usually describe the negative criticism given to him by the media regardless if they hold truth or not, which is a wrong usage of the term by our definitions [51]. However, fake news are not a new concept. An example can be traced all the way back to 1835 and involves the “Great Moon Hoax”. During that time the New York Sun had written a variety of articles about the discovery of life on the moon [3]. Nowadays, with the rise of the web and especially social media, fake news are finding new channels of spreading to many more individuals in a far easier manner [96].

The intent behind the creation and spreading of fake news can be seen as two-fold. On the one hand, there is a financial motivation. Clickbait headlines and stories with false substance that draws in the attention of individuals are more likely to go viral and thus earn far more advertising revenue from increased readership due to their outrageous nature. This naturally benefits the advertisers and increases traffic. An analysis made by BuzzFeed reinforces this behavior as it showed that the top fake news election stories about the 2016 US presidential election had higher engagement than the top real election stories in Facebook [95].

On the other hand, there is an ideological motivation. Fake news fabricators often want to discredit other persons, entities or organizations in order push forward their own agenda or that of people they want to advocate for and have financial or
CHAPTER 2. BACKGROUND - LITERATURE REVIEW

political gains. [3, 96]. Fake news spreading is also linked with post-truth politics [82] which can become highly apparent in the recent years considering the immense amount of fake new stories surrounding the 2016 US presidential elections and the UK Brexit.

It can thus come into view that the rise of fake news is the result of easy access to advertisement revenue online, increased political polarization and the rising popularity of social platforms. Governments have also been implicated in fabricating and promoting fake news. Instances of that involve Russia’s involvement particularly during 2016 US presidential elections [71] and Turkey during the rising tensions with Greece and the migrant crisis.

Examples of fake news are numerous and a few will be given next. On December 4, 2016, an armed man entered a pizza shop in Washington, DC. He was determined that the pizza shop was part of a large chain consisting of an underground child sex ring which was run by then presidential nominee Hillary Clinton. The man fired shots in the ceiling of the shop when his demands were not met. Luckily, nobody was injured or killed and the conspiracy which was dubbed “Pizzagate” was later debunked by experts [80]. The man was later arrested for firing gunshots and charged appropriately. Another example is when the minister of defense of Pakistan tweeted a hostile response to a false report that Israel had threatened Pakistan with nuclear weapons [32]; an event that could have led to conflict between the two nations.

Coronavirus misinformation and fake news stories have also plagued the social media during the COVID-19 pandemic. According to a study, people trying supposed products that were falsely labeled as cures for the disease such as drinking highly-concentrated alcohol or methanol have killed at least 900 people while also hospitalized about 5800 [40]. Another prevalent conspiracy theory is that 5G networks are contributing to the spread of the virus. This has led to many protests in the UK where 5G towers are taken down by civilians, operating in the hopes of stopping this spread. There is also a theory that suggests there are 5G microchips inside masks that help spread the virus and track individuals that wear them. A research done in the USA suggests that around 17% of Americans believe that the COVID-19 pandemic was intentionally planned by powerful people. A correlation in the education level of those believing this theory was also found, indicating

---

9 https://m.naftemporiki.gr/story/1627485/discussing-covid-19-conspiracy-theories
that people with lower educational level tend to believe this more than people who have a higher education.

A recent example of fake news concerning Greece and created by Turkey was during the Evros migrant crisis. Turkish media spread various fabricated fake news about Greece killing migrants at the border. Turkish media also made claims that the Greek coast guard illegally pushed back a boat filled with immigrants and caused the death of an infant child. The news of the child’s death were true, however the exact circumstances of how it happened were false and intentionally misleading. The incident of the death took place in another boat several hours later and was the cause of the illegal migrants themselves trying to sink their own ship to try and secure a SR (search and rescue) by the coast guard of Greece [20]. The issue was made even worse when Turkish users and bots got on Twitter and relayed these fake news as real news.

The above examples show us the gravity of the situation when dealing with the spreading of fake news. In the Pizzagate incident in particular, human lives were at stake due to a false statement made about Hillary Clinton. Fake news are one of the main threats to modern human society according to the World Economic Forum (WEF) [22]. In the COVID-19 cases, material damages were done to 5G towers which amounted to a large amount of capital loss. More specifically, due to the mask conspiracy theories, many individuals refuse to wear masks thus aiding in the spreading of the virus. It can become apparent thus, that necessary action to detect and stop fake news is of utmost importance. Further conspiracy theories observed over the years and the share of Americans that believe in them are shown in Figure 2.2.

Why do people believe fake news and not real news? Isn’t it easy to differentiate between real and fake news for most individuals with some critical thinking? There are several reasons that this is not always the case. Firstly, there is the issue of the echo chamber effect. An echo chamber is an environment in which individuals only read information or opinions that reflect and reinforce their own. Essentially, people tend to follow other like-minded people while ignoring everyone that has differing beliefs. Indeed, individuals take in information which is recommended by friends or relatives which have similar characteristics (same echo chamber) as them in their social networks [74]. This results in the formations of highly polarized clusters in which social homogeneity is the leading actor for the diffusion of information [22].

An interesting finding given by Vosoughi et al. [102], is that fake news travel significantly faster and broader than real news, which means that they reach much more people in a shorter time-frame. They also find that the effects are even more pronounced for fake political news than other types of fake news. In addition, fake news have a higher degree of novelty, which leads to the conclusion that people
Figure 2.2: Historical conspiracy theories and share of people that believe in them. Source: [3]

Tend to prefer sharing novel information. There is also a difference in the reaction in the comments to real and false news. False content reaction contains emotions such as surprise, disgust or fear, while real content contains joy, trust or sadness. The increase in the spreading could be attributed to the novelty that fake news include, as well as the more intense emotional reaction given to them by users reading them when compared to real news.

Before moving forward to analysing the different kinds of fake news, it is worth differentiating between fakes news and personal bias. There are news of which there is no real objective truth, only people who have the belief that their opinion is the absolute truth. As an example, someone could write in a news article that Italy is a very organized country with a very high standard of living, and another person might say the completely opposite (e.g. the country is a bureaucratic mess and completely disorganized). Both of these statements hold truth albeit a highly biased one and therefore a conclusion cannot be made as to whether one or the other is fake.

From all the above, we can conclude that Twitter is one of the most used plat-
forms in which fake news stories are circulated and diffused. This is why, in this work, we have decided to focus on the detection of fake news on Twitter.

2.1.3 Types of Fake News

Fake news can appear in various shapes and forms depending on their characteristics. These can range from conspiracy theories, clickbaits and selective editing, to unintentional reporting errors, hoaxes [45], satire [38], rumors [106] and others. In this section we will offer various taxonomies of fake news found in the literature that try to categorize the different kinds of fake news.

According to Claire Wardle of First Draft News, a project founded in 2015 by the Open Society Foundations, Twitter, Facebook and several other philanthropies to combat the spreading of fake news and fight misinformation in the web, there are seven types of fake news [105].

• **Satire or parody.** This type of fake news usually drains information from real content and has the purpose of containing irony and absurdity for entertainment purposes. It usually does not cause harm as it is rarely misinterpreted as fact.

• **False connection.** Also known as clickbait titles, these kinds of news are based on attracting viewership for revenue via a false connection between the headline, images or captions which are irrelevant to the content. An example of this is when reading a headline that states that a celebrity has died but once opened there is no mention of this celebrity.

• **Misleading content.** This type of fake news uses false or misleading information with some drops of truth in order to get ahead or bring down another individual or entity. It is often used in politics by rival parties to bring down each other.

• **False context.** False context or selective editing includes information that someone said but relayed out of the context of the conversation to make what was said sound shocking. Similarly to false connections, it is often used by journalistic sites in order to gain traction.

• **Impostor content.** Impostor content is a type of news that comes from a fake or non-existent source which pretends to be a real news outlet which usually tries to push a specific agenda.

• **Manipulated content.** Manipulated content contains information that is genuine but has the intent of deceiving its readers by telling a story different that the one assumed by the headline.
• **Fabricated content.** Novel and completely false content that is fabricated with the intent to deceive its readers and create distress.

Rubin et al. [84], make a more succinct categorization by placing different kinds of fake news into three distinct categories. These include: **serious fabrications** in the same context as above, **hoaxes** (i.e. providing false information via, for example, social media with the intent to be picked up by traditional news websites) and **satire** (i.e. humorous news items that make a play on genuine news by containing sarcasm and ridicule).

### 2.1.4 Detecting Fake News

So far, we have introduced what fake news are, their characteristics, the purpose behind their existence, why they are so important to detect and stop and the different kinds of fake news available. Moving forward, we will introduce the task of fake news detection and review the different methods that are used for detecting fake news in the literature. A definition often used for fake news detection is given in [90].

**Definition 3.** Given the social news engagements $E$ among $n$ users for news article $a$, the task of fake news detection is to predict whether the news article $a$ is a fake news piece or not.

Automatic fake news detection is a problem that can also be viewed more broadly under the category of deception detection, a task that has been around for quite some time and for which natural language processing (NLP) has been shown to be effective at solving it [59]. The problem however, becomes more profound with fake news detection as the content of fake news posts is usually shorter and often contains more complex language structure (e.g. political language) which makes it harder for NLP based approaches alone to solve [104].

For solving the fake news detection problem a variety of methods have been proposed which can be broadly put in two large categories. Firstly, there are those that are based on treating the problem as a NLP problem by using natural language features, in which the content of the fake news is extracted appropriately and used in various ways for revealing whether a fake news post contains false or true information.

Secondly, there are those that are based on the social setting surrounding the news post such as the characteristics of the users that engaged with the post or the structure of the network, in which the problem is transformed into a graph problem and various features such as metadata or knowledge graphs can be used. There are also hybrid approaches that utilize methods from both categories that are effective. Most of the approaches in both categories use various machine learning techniques to solve the fake news detection problem.
2.1.5 Features for Fake News Detection

2.1.5.1 Natural Language Features

As already mentioned, the first approach to create fake news detection models requires the use of natural language features. These features need to be extracted from the fake news content and they are usually taken from the source (i.e. the author of the news article), the headline or the body text. If available visual information can also be used with appropriate transformations (i.e. images). Features that have been used effectively in the literature are the following:

- **Linguistic based.** Here the text of the news post is leveraged by transforming it into vector representations using simple bag of words approaches by counting the frequency of words using n-grams or tf-idf. Another possibility is using part-of-speech (POS) tagging or named entity recognition (NER).

- **Readability.** Here, the text of the news post is leveraged by counting the number characters, or words, the characters per word, the hashtags or punctuation and other similar lexical based features.

- **Contextual based.** In this approach the text is used by transforming it into a vector representation that can hold the context of the words (e.g. word2vec, GloVe, etc.). Appropriate n-grams extensions can also be used as well as topic models such as the latent Dirichlet allocation (LDA) for the extraction of the topics).

- **Psycholinguistic cues.** In this category the LIWC lexicon [67] is typically used to represent the text as psycholinguistic processes. These can signal emotions such as sadness, anger or positive reinforcement. Work done on deception detection has also showcased LIWC as a valuable tool.

- **Syntax.** Here, syntax stands for structures of deeper language. This technique is approached with the usage of context free grammar (CFG) by transforming the sentences into a set of rules that can describe the deeper language structure (e.g. noun and verb phrases) of the text in the news post under inspection. This feature has had a high success for the task of deception detection [27]. However, it does not prove very useful on its own when applied in the detection of fake news.

2.1.5.2 Social Setting Features

The second category of approaches is based on the social setting surrounding the news post. Features that have been used effectively in this category are as follows:
• **User Characteristics.** These are features characterize the users that have engaged with the news post. They are taken from their user profiles and can either be explicit or implicit. Explicit features include number of followers, number of followees, whether the user is verified or not among others. Implicit features include personality, age and gender.

• **Temporal.** This category of approaches includes the analysis of the temporal aspect that a news post receives. This includes the user engagement and could be the response or reaction that a news post receives in a social network context (e.g. analysing the sentiment or emotion that a given news post receives and use that to make a decision on whether the original post is true or false). Temporal patterns that can be modeled as propagation paths can also be included in this category. Temporal models take into account the temporal changes of a news post and capturing the propagation elements via tree-like structures.

• **Network Structure.** The last category involves the transformation of the problem into a social network analysis problem and analyzing the network’s structure using graph theory. Many different kinds of networks are possible. A most common example is the friendship network, an undirected network which shows when two people are friends with each other. The diffusion of fake news can also be analysed with this approach.

### 2.1.6 Methods for Fake News Detection

In the previous sub-section we analyzed the variety of features that can be extracted from various fake news posts or the social setting surrounding them through a variety of different ways. In this section, we will showcase various methods that have been used in the literature in combination with these features can be used with to detect fake news. These include fact checking, predictive modeling and machine learning, propagation, network structure and credibility based methods.

#### 2.1.6.1 Fact checking

Fact checking is a manual process in which a piece of information (in our case a news post) undergoes a process of verification to establish its authenticity and truthfulness. It can be used before a news post is released in public internally to check for its correctness or after it has been released usually by external parties [34]. It is a technique also adopted for the task of fake news detection. Three distinct categories can be found which will be analyzed next. These include expert, crowdsourcing and automatic fact checking [108].
• **Expert.** In this category, experts in the field of the news post under inspection are handed with the news post and asked about its veracity. Usually an extremely accurate method but incredibly expensive as there are nowhere near enough experts to catch up with the real time fake news dissemination and apply fact checking.

• **Crowdsourcing.** In order to combat the expensive nature of using experts an idea is to instead use a large amount of people that are handed the news post and asked about its correctness. This kind of fact checking relies on the wisdom of the crowd mentality. Even though it is more capable to handle larger amount of fake news the reliability of the results is under question as different people can have differing opinions regarding the veracity of a news post.

• **Automatic.** Even though crowdsourcing fact checking can handle a larger amount of fake news than expert fact checking both of these methods are not able to cope with the huge influx of fake news that are continuously uploaded online. For this reason, an automatic fact checking is deemed of utmost importance. Automatic fact checking is usually based in using natural language features already introduced as well as knowledge graph features. Usually natural language features are used to find whether a news post needs to be checked. Afterwards, these news posts are compared with external sources such as knowledge bases to establish their veracity [18].

### 2.1.6.2 Predictive Modeling and Machine Learning

Features extracted as analyzed in the last sub-section can be used after appropriate pre-processing and transformations in order to train various machine learning models and used to predict whether a future post contains fake news or not.

Approaches in this category can be found using SVMs as in [107], naive bayes classifiers as in [66] or logistic regression as in [83]. These methods have sufficient levels of accuracy but the disadvantage is that the models can usually only successfully predict posts when used in the same topic as the data they were initially trained on (e.g. politics, economy, religion, social issues, celebrity gossip, immigration etc.). Reis et al. did an extensive research on using various features already presented in the literature to train various classifiers including k-Nearest neighbors (KNN), naive bayes classifiers (NB), random forests (RFs), Support Vector Machines (SVMs) and forest ensembles (XGB). They found that the best performing models were the RF and XGB classifiers, statistically tied with about 0.85 AUC [77]. Unsupervised machine learning methods like clustering can also be used to differentiate between real and fake news albeit with not that high level of accuracy [19].
CHAPTER 2. BACKGROUND - LITERATURE REVIEW

2.1.6.3 Propagation

The dissemination of information regarding a news post alongside its temporal aspect can be captured by modeling the activity of the users that interacted with that post (e.g. retweeted, liked) via propagation based structures (e.g. trees). These propagation structures can capture elements such as the breadth or depth that the news post has travelled or the number of users at a specific time frame that it has reached.

Features from propagation structures can be extracted and then fed into machine learning algorithms such as naive bayes classifiers, decision trees and support vector machines (SVMs) [17, 106]. In this line, Del Vicario et al. train various ML classifiers for identifying fake news content on social media by extracting features involving the polarization of the article using its propagation characteristics [23].

A different direction is taken by Finn et al. in [28], where the authors create a tool which for a tweet under inspection shows the user similar tweets and various propagation based stats so the user can decide with critical thinking whether the story they are reading is fake or not. Similarly, in [97], they aim to minimize the spread of fake news by stopping the propagation in the network. To accomplish this, they leverage users’ flags by first evaluating their flagging accuracy using a novel algorithm.

Instead of manual feature extraction, deep learning methods can also be used in order to learn the representation of the propagation paths automatically; a task known as representation learning [57]. As another possibility, propagation paths can be modeled like a time series (a sequence of points listed by the order of time) and the problem can therefore be transformed into a time series classification problem. Afterwards, recurrent neural networks (RNNs) or convolutional neural networks (CNNs) can be used to build a time series classifier [52].

2.1.6.4 Network Structure

Similarly to propagation methods, graph structures can be appropriately constructed using network techniques. These usually include the following types of networks; homogeneous, heterogeneous and hierarchical [110].

• **Homogenous.** Homogenous networks are characterized by the fact that all nodes and edges have a single function (i.e. all nodes represent users). A homogenous network based fake news detection approach was proposed in [109]. In the work of Zhou and Zafarani, each user sharing a news post represents a node and whether they are following each other is represented by an edge. These networks can then be analyzed at the node level, community level or the whole network. Indeed, homogenous based analysis of
fake news spreading has shown that data within specific narratives create homogenous networks with similar patterns [22].

- **Heterogenous.** In contrast with homogenous networks, in heterogenous networks different kinds of nodes and edges are possible (e.g. nodes can represent news posts, users, publishers etc.). The analysis of the relationship between different types of nodes can bare some fruitful results for the detection of fake news.

- **Hierarchical.** In hierarchical networks the elements that constitute the network can combine and form higher hierarchies. For example a hierarchy on Twitter could be a tweet containing a news post combined with all the retweets that were made about that posts combined with all the comments made in all the retweets. Features could be extracted from these networks similarly to the propagation method and fed to ML algorithms to make the final classification.

### 2.2 The Role of People in Fake News Spreading

The credibility of the sources that spread fake news along with the credibility of the users that share them can be useful at predicting whether the news post they share consists of false information or not as well as the analysis of the credibility of the comments that are found on the respective news post [17].

Even though there has been a thorough investigation in the literature regarding the task of fake news detection, this role of the people (i.e. social media users) that spread the fake news have has not been explored as much. We believe that the social media users are a fundamental pillar at the spreading of fake news as people can be the igniting spark for sharing and re-posting a fake or real news story and that their psychological characteristics can play an important part in their spreading behavior.

Only recently there has been a shift of attention in the literature with trying to utilize the user profiles and psychological patterns of social media users in order to classify them as fake or real news spreaders [10]. More specifically, Shu et al. [91] try to find correlations between the user profile characteristics and whether that person is a fake news spreader or a real news spreader. A step forward by the same authors was done in [93] where machine learning white box models where used to find the feature importance of various features related to the user profiles of fake news spreaders. More recently, Giachanou et al. [31] have shown that personality combined with contextual information in the form of word embeddings has a higher predictive power than using only linguistic information at classifying fake news spreaders.
How can we successfully predict the socio-psychological characteristics of users in social media and the role they play in the diffusion of fake news? This section will explore the theoretical background behind utilizing behavioral and socio-psychological patterns that characterize and are linked with fakesters profiling.

2.2.1 Classifying Fakesters

In [1], it has been shown that users that have a lower credibility have a higher chance of spreading fake news than more reliable users. Credibility in this case is defined as how trustworthy the user is. Users with low credibility include both users that publish fake news intentionally with malicious intent as well as users that are vulnerable to them and reproduce them. On the other hand, users with high credibility are more trustworthy. These observations can also be magnified with the inclusion of the echo chamber effect.

Zhou and Zafarani [110] make a concise classification between social media users which include malicious users (i.e. fake news spreaders) and normal users (i.e. non fake news spreaders). Malicious users also often include bots. However, not all of these bots are malicious as some of them are made for good purposes such as re-posting real news. Indeed, Vosoughi et al. [102] claim that social bots spread false and real news at a similar rate, which means that fake news spread faster and more broadly because real people have a higher likelihood of sharing them. Nonetheless, advances in bot detection [100] have made it possible to detect the malicious social bots that spread misinformation. Another category of malicious users are the so called “trolls” which tend to spread misinformation in order to create confusion and mayhem among vulnerable users for their own personal satisfaction.

2.2.2 Socio-psychological Background

The problem remains however at detecting the real human beings that spread fake news (disregarding bots with human-like behavior as it is out of the scope of this work). Analyzing the psychology behind misinformation and fake news can shed some light into the underlying patterns that these users follow and could aid with their successful detection.

Kumar and Geethakumari [44] analyse concepts in cognitive psychology that enable the vulnerability of users in social networks to misinformation. They argue that the innate beliefs of human users have a vital impact in tolerating news they read and that the radical polarization found in the internet today is a result of this part of human psyche. They also believe that preventing the spread of fake news in the first place is better than detecting them. This could be done by arming
them with the knowledge to make more accurate decisions when reading news and before deciding to retweet or share what they receive.

Moving forward, Lewandowsky et al. [50] attribute the diffusion of false information as a result of the cognitive process which is based on the evaluation of how truthful the information they read is perceived as. According to their work, the acceptance of a piece of information as factual is much a more often occurring phenomenon compared to the challenging and the use of critical thinking to decide on the veracity of a claim.

They came up with four factors that users take into account when assessing the veracity of any piece of information. These include the following:

- **Consistency.** When reading a new piece of information, is it compatible with their previous beliefs?

- **Coherency.** Is the information presented in a coherent manner and makes plausible sense?

- **Credibility.** How credible is the source that shares this information? Can they be trusted?

- **Acceptability.** The power of the herd. What is the general consensus on this piece of information according to other users? Do most believe or not the claims that are made?

In [69], the authors argue that as we live in the social media age and fake news can be shared extremely easily in social networks, it is undeniable that the exposure of individuals to misinformation has increased dramatically when compared to the past. The so called “illusory truth effect” is given as a possible culprit to explain the psychology behind why people fall for fake news. According to the authors, prior exposure to fake news stories vastly increases the likelihood that false statements would be perceived as factual by individuals. The only case that this isn’t true is when the information transmitted includes completely implausible statements (e.g. “the earth has the shape of a pyramid”).

As already stated, the COVID-19 pandemic has also brought upon many fake news stories. These can be analyzed to shed some insights behind the psychology of misinformation. Indeed, in [8] various observations were made by inspecting how users reacted to them. These include that commonly held beliefs without evidence supporting them (e.g. "COVID-19 can be cured with ginger and turmeric") gained significant traction. Also, false scientific interpretation was prevalent (the use of the drug hydroxychloroquine as treatment to COVID-19 soon as it was made known in news led the increase of its usage and also to side effects). Finally, as
shown in previous studies they also signify the importance of the power of the herd and prior exposure as a vehicle that reinforces fake news diffusion.

All of the above studies agree that misinformation acts similarly to a virus in the sense that after it starts spreading it is too late to stop it. Thus, effective measures need to be taken to stop fake news from affecting individuals in the first place before they have the chance of spreading [111]. However, it has also been shown that merely tagging suspicious stories as containing disputed claims does not effectively combat this problem [69]. The aim should also be to arm the individuals with the appropriate knowledge and explanations when reading this information to help them make an appropriate decision on their own [44].

2.2.3 User Characteristic Models

As already stated in the previous sections, research has shown that the user characteristics of the individuals play an important role in the spreading of fake news [31, 91, 93]. These characteristics include the gender, the personality, the psycholinguistic features of the user and the sentiment and emotion found in their text.

Since online networks offer a wide variety of features including language, emotion and various other metadata (e.g. followers, followees, tweets), related work has used them to build predictive models to infer their personality and gender. The most used personality model is the one that assigns the personality of the user to one of Big Five personality traits.

With regards to gender, related work done in the prediction of gender on Twitter has successfully used linguistic features (tf-idf) as in [21], contextual features (BERT) as in [15] while others investigate the usage of various readability features, including the length of words, the number of punctuation marks and emojis as in [43]. In line with previous research, in this work, we have decided to use a mixture of linguistic and readability features.

For the extraction of psycholinguistic features the same approach as mentioned in section 2.1.5.1 is used and namely with the usage of the LIWC lexicon.

Lastly, for the analysis of the sentiment of a given text, state of the art approaches include VADER [39] and TextBlob [53]. VADER is a tool that was created especially for the analysis of sentiment found on social media texts. TextBlob is a more general natural language processing tool that also contains features about sentiment analysis.
2.3 Explainable Machine Learning

In this section we will explore the needs for explainable fake news detection and related work done in this area. Firstly, we will present a brief introduction to the field of machine learning and state the needs for interpretability and explainability that exists in machine learning today which gave birth to explainable machine learning. Moving forward, we will analyze the literature involving explainable machine learning in more detail. In closure, we will review a few explainable fake news detection frameworks in the literature.

2.3.1 Needs for Interpretability in Machine Learning

Applications of machine learning today can be found in various applications of our everyday life, including song recommendations in Spotify, movie recommendations in IMDB or Netflix, friend recommendations in social networks (e.g. Facebook, Instagram, LinkedIn), which products to buy on Amazon or Ebay, automatic image recognition (e.g. friends in Facebook photos). Other applications include fake news detection, hate speech identification, self driving cars, bots that can play games or video games better than humans (e.g. chess, checkers, Dota 2), personalized ways to treat various diseases (e.g. cancer, COVID-19) and even aiding in the discovery of new stars and galaxies in the cosmos [42].

Machine learning approaches can have many different categorizations depending on who you ask. Typically, a division can be made into three broad sub-categories. These include supervised learning, unsupervised learning and reinforcement learning [89].

- **Supervised Learning.** In this category, there is a given set of data for which the outcome is already supplied (i.e. a labeled dataset). In machine learning terminology, this outcome of interest is called a class. The focus is to train a model on the data given and then predict the class for new unseen data. In supervised learning, we can find two sub-categories. **Classification** is the task of prediction when the output is categorical such as classifying between “hate speech” and “neutral speech” in a piece of text or between “red”, “green” and “blue” in an image. In the first case where we have only two possible classes, the problem is known as binary classification whereas in the second case the problem is called multi-class classification. Moving deeper, we can also find the problem of multi-label classification [98] which involves pieces of data that can be attributed to more than one class. For example, when classifying the genre of a movie, more than one genre could be applied in the same movie (drama, science-fiction, adventure etc.). **Re-
gression is the prediction if the output is continuous or numerical such as “Dollars” or “weight”.

- **Unsupervised Learning** is another category that most often includes two separate sub-categories; clustering and association rules. Contrary to supervised learning, here there is no given label for our data. Thus the algorithms need to find their own associations within the training data. Typical applications involve anomaly detection, clustering of items based on their similarity and association rule mining for locating item-sets that are frequently found together.

- **Reinforcement Learning** aims in the maximization of a reward function with the usage of a set of actions in a predefined environment. It differs from the previous methods because there is no training data given as input so the model must make a decision on its own about what to do on a given task. Applications can be found in traffic lights control [5], in robotics [41], in games [94] and in self driving cars.

A fourth category can be classified as **Semi-supervised Learning**. These approaches fall somewhere in-between supervised and unsupervised learning. The way they work is by using a mix of both labeled and unlabeled data during the training phase. Thus, it can lead to models with a much higher predictive power than using solely unlabeled data.

There are many different machine learning models available in order to solve different kinds of problems. In the literature, models such as random forests (RFs), support vector machines (SVMs) and deep neural networks (DNNs) are often referred to as *black boxes*. This definition is used to showcase that one can’t really inspect how the algorithm is accomplishing what it is accomplishing or the decision process behind why it came to a certain conclusion. This contrasts with the term of *white boxes*, which refers to models which are transparent and their inner workings can be easily understood. Such is the case for algorithmic approaches such as linear models, decision trees, decision rules and k-nearest neighbor.

Due to the needs for interpretability that exist in various machine learning models, research on explainable machine learning has increased dramatically over the last years [36, 61, 64, 81, 88]. Such an intense interest can be attributed to the idea of creating effective black box classifiers with explanation methods in order to provide high performance and explainability simultaneously. Explainable machine learning aims to make black box models interpretable or create similar white box models which are understandable.

[10]https://towardsdatascience.com/interpretability-vs-accuracy-the-friction-that-defines-deep-learning-dae16c84db5c
Why not just trust a highly accurate black box machine learning model if it performs very well? Why do we need to know why it made a certain decision. Why should it even matter if the black box model has nearly perfect accuracy?

In certain events, it is of high importance to have the ability to know why a machine learning model makes a certain decision. The importance of this can be seen especially in highly sensitive fields such as medicine and self-driving cars. A potentially wrong decision made by the model, even if it is extremely rare could very well lead to the loss of a human life. Such was the case when a self-driving Uber car hit and took the life of a pedestrian woman in Arizona [103]. Uber self-driving cars use black box models with no explainability and thus no reasoning could be given as to why this decision was taken. Explainability of machine learning models has also become a mandatory requirement for software developers and engineers as it is a valuable tool for aiding in the debugging process of software engineering [25].

Another prevalent problem that exists in the machine learning pipeline is that of bias [26]. Black box models could potentially hide prejudice and injustice including but not limited to racism, sexism or historical prejudice in their logic (i.e. marking an individual as unsuitable for a bank loan due to their racial background). Such models can thus accentuate stereotypes and be harmful to basic human-rights. Explainability could help uncover these biases in the hopes of building models that rely on fairness with a final aim to not exclude any individual or social groups based on discrimination. This also ties in with the idea of fairness in machine learning which states that the outcome of interest should not correlate with variables that
represent the sensitive traits of individuals under inspection (e.g. gender identity, sexual orientation, racial background etc.) [11].

Finally, the European Union recently introduced a new regulation called Right to Explanation in the General Data Protection Regulation (GDPR) [33, 58] as an attempt to deal with the potential problems stemming from the rising importance of machine learning algorithms. As a result, as of 2018 all European Union citizens are entitled to know why a decision was taken on their behalf (e.g. why their loan was rejected). For the above reasons, it is understandable that interpretability is not only important, but it is necessary for the advancement of machine learning as a trustworthy, responsible and open data science.

Before moving forward, it is worth noting that explainability is not a holy grail that is always required. Interpretability is needed only if a machine learning model has an important impact usually in sensitive fields. This impact can be economical (e.g. the loss of capital in a business) or social (e.g. the loss of human life). Let’s say an individual wanted to build a black box machine learning model to classify animals based on a dataset containing images. In this case it wouldn’t be so important if the model made a mistake, nor would he be forced to give to anyone an explanation as to why an animal was classified as a cat if it was a dog. Adding explainability to this model could be a second step to help this individual debug their model. However, if the individual worked in a business and the core principle of the business was to classify animals given images, then the business could lose money and customers whenever the model makes a mistake. Thus the business and hence the model would owe an explanation to a potential customer as to why a wrong decision was made in the specific scenario. Lastly, the idea that we don’t always need to have interpretability is also interesting as usually, when we try to make models explainable we tend to lose some predictive power in the post-hoc model. Therefore, the goal is to find the golden ratio between this trade-off; offering explainability and the subsequent loss in terms of accuracy. This essentially comes down to offering accurate explanations.

By inspecting Figure 2.4, we can transfer the medical environment pipeline to
CHAPTER 2. BACKGROUND - LITERATURE REVIEW

our own problem of fake news spreader detection. Given a fake news spreader detection model and by having its prediction along with the feature used (e.g. number of followers, personality type, slang count in her tweets etc.), we could then apply an explanation technique to find out which of these features played the biggest part in the final classification.

2.3.2 Interpretability and Explainability

Before moving forward it is helpful to differentiate between the terms explainability and interpretability as their usage is not synonymous. Interpretability can be seen as a component that several machine learning models inherently have. Thus the terms interpretable models and white box models can be used interchangeably.

Explainability can be seen as an attempt to make the inner workings of a black box model understandable. This is usually done by using a post-hoc model which is used to explain the underlying model [25, 86].

Christoph Molnar on the other hand, decides to use the terms explainability and interpretability interchangeably in his book about interpretable machine learning [63]. Despite this, he supplies us with two interesting definitions about the concept of what it means to have a model with interpretability.

Definition 4. Interpretability is the degree to which a human can understand the cause of a decision.

Definition 5. Interpretability is the degree to which a human can consistently predict the model’s result. The higher the interpretability of a model, the easier it is for someone to comprehend why certain decisions or predictions have been made. A model is better interpretable than another model if its decisions are easier for a human to comprehend than decisions from the other model.

In this work the terms explainability and interpretability are considered separated and tied in a sense to black box models and white box models respectively. Explainability is used to make black box models understandable and thus interpretable while interpretability is an attribute that certain machine learning models (white boxes) have thus making them understandable by default. By using explainability methods we can say that we can attain interpretability also in black box models.

For better analyzing the interpretability of various different machine learning models, we will hereby recognize a set of dimensions which aid in understanding the attributes of the interpretability of a given model. [25].

- Global vs Local Interpretability. When a model offers global interpretability it means that we are able to understand it completely and as a whole
(i.e. when we are able to get explanations for a model in a catholic way). On the other side, local interpretability is observed when a model is only partially interpretable. We are not able to explain the model completely but we are able to offer explanations for part of the model (i.e. when we want to give explanations for a single instance).

- **Model Agnostic vs Model Specific.** Model agnostic techniques are those that are able to offer explanations for a given model regardless of the type of the underlying model whereas model specific are those that are designed only for a particular class of machine learning models (e.g. only for deep neural networks or only for support vector machines). In the majority of cases, model agnostic methods are only able to offer explanations for a specific type of data (e.g. tabular, image, text).

Prior to investigating the various sorts of techniques utilized in explainable machine learning we need to have a comprehension of the distinctive information types that are available from a machine learning point of view. The most used data types that can be found in the literature are the following:

- **Tabular data.** Contains data with either numerical or categorical values.
- **Image data.** Contains data that represent images. For usage in machine learning tasks features from these images need to be extracted. This is usually done by transforming them into various vector representations. An example of this transformation from the original image space into the vector space can be handled by image descriptors, which give use the capability to describe images by turning them into vectors. Two well known types of descriptors include color descriptors and shape descriptors. Color descriptors are based on the idea of describing an image using only its coloration while shape descriptors are based on describing images by the usage of the various shapes found in them [9, 54].
- **Textual data.** Data that contains natural language texts. As in the image case, the textual data needs to be transformed to a vector representation. This can be done by a variety of different approaches such as the bag-of-word model, the TF-IDF model or more advanced techniques such as learning the representation of words with the aim of capturing the relationship of the meaning between the words. For example words such as cat and dog, which have a similar meaning, should have a small distance from each other with respect to their representation, while words like cat and green should have a large distance as they describe different concepts. Examples include the word2vec model [60] and the GloVe model [68].
Of course, there are also different types of data in the machine learning landscape such as sequences, time series, graphs, networks etc. However, these types of data are not as studied in the literature as potential input for explainable machine learning models [36]. The most used types of data are either tabular, image or text.

It can be apparent that depending on the types of data that is used, a human user can gain different levels of interpretability by observing them. Arguably, the types of data that should be the most interpretable by humans should be images and texts, as they are found and used the most in their everyday life.

2.3.3 Explaining Black Box Models

In the literature attempts for explainable machine learning can be broken down in two categories. These include Reverse Engineering and Design of Explanations [36]. In the first approach the aim is to enter the black box and unravel its inner workings. Usually the predictions made by the black box model are given and the task is to provide explanations for them. In most cases, the dataset that was used to train the black box model is unknown and thus we try to replicate its predictions to offer explanations. In the second approach we build a white box model on a given dataset that is able to provide explanations directly to the end-user.

2.3.3.1 Reverse Engineering

Reverse engineering includes two main categories. These are Model Explanation, Outcome Explanation [36].

• Model Explanation. The model explanation category consists of providing global interpretability for the black box model under inspection, by using a transparent white box model that is able to approximate the decisions of the black box models and thus capture its inner logic. The task is to create an interpretable white box model which is very similar to the black box model and the goal is to make the output of the white box model be almost the same as the black box model. The global surrogate method belongs in this category of approaches and will be explained further in the following section.

• Outcome Explanation. The outcome explanation category consists of giving explanations in the form of local interpretability for the given black box model. This approach takes into advantage the fact that it is not necessary to replicate and explain the whole logic of the black box, rather a small neighborhood around the instance that we want to explain. The task is to give an explanation by using an interpretable white box model which is created by using the black box and data in the neighborhood of the instance.
we want to explain. LIME, ANCHORS, LORE, LIONETS and XSPELLS are examples of various outcome explanation methods found in the literature [35, 47, 62, 78, 79]. This category of approaches, that utilizes the local neighborhood approach, has become the most used and studied in the recent years for the explanation of black box models. In the literature, these models are also often called \textit{local explainers} or \textit{local explanators}.

2.3.3.2 Design of Explanations

Design of explanations essentially boils down to the usage of solely white box models instead of black box models. It consists of providing a local or global interpretable model directly. Examples of white box models include decision trees from which local or global explanations can be obtained in the form of decision rules by following the paths of the decision tree. Linear models are also appropriate for giving both local and global explanations by plotting the weights assigned by the model for a given instance or as a whole. State of the art approaches in this category use highly accurate rule models with low complexity [4, 46].

Recently, there has been a criticism on approaches attempting to open the black models by using post-hoc models (i.e. reverse engineering). Rudin in [86] is against the recent surge of explainable machine learning which rather than trying to create models that are inherently interpretable, use another model to be able to explain the first black box model. Going against the tide, she claims that explanations provided by these models are mostly unreliable and misleading as a post-hoc model is never able to capture the inner logic of a black box model perfectly. Instead, there has been a push to promote the usage of white box models (i.e. design of explanations). Ultimately, both methods have advantages and arguably it is not always possible to create white box models when faced with more complex problems.

2.3.4 Global Surrogate Method

The global surrogate method is a model agnostic explainable machine learning method which can offer global or local interpretability and belongs to the model explanation of reverse engineering category. It works by training an interpretable model (e.g. a decision tree) in order to approximate the predictions made by a black box model (e.g. a random forest). This way, we are able to have a simpler model that can mimic the more complex black box model but also offer interpretation and the ability to draw conclusions about the underlying black box model.

In this class of approaches, the following steps are usually followed. Firstly, the same dataset that was used for training the black box model is selected. For the selected dataset, we get the predictions by using the black box model. Then
we select an interpretable model (linear model, decision tree, etc.) and we train that model on the dataset we have selected and its predictions made by the black box model. To evaluate how well the surrogate model replicates the predictions of the black, we use the fidelity measure [36]. Generally, we don’t care how well the surrogate model performs on the true labels of the dataset, but on how well it replicates the black box. In the end, we can interpret the simpler interpretable model we have created [63].

This work takes advantage of the global surrogate method in order to create a simpler model out of a more complex one but capture it’s inner logic. In more details, after creating a fake news spreader classifier by taking into account user level characteristics taken from each user’s Twitter profile (by analyzing their 100 latest tweets) and using that to classify the users responding to a tweet as potential fake news spreaders or real news spreaders, we create a simpler linear model which takes as input only the text the user leaves and as label the outcome of the fake news spreader classifier for that user. Since a user has been classified as a fake news spreader for their last 100 tweets then there is a high enough chance that the reply they will leave in another individual’s post will also contain fake news. The opposite also holds true for real news spreaders.

Thus, by matching the content of the text with the label given by the fake news spreader classifier, we have effectively gone from a user level classification to a post level classification and we are now able to use the features that are left in the text to build a model only on text data and provide simpler explanations involving textual features.

2.3.5 Local Interpretable Model-Agnostic Explanations

Local Interpretable Model-Agnostic Explanations, most commonly known as LIME is a state of the art model agnostic technique which offers local interpretability and introduced in [78]. As one of the most popular and most used starting techniques in the explainable machine learning landscape today, LIME belongs to the outcome explanation category of problems. This means that it provides a local interpretable model that is able to explain the black box predictions with understandable human terms but only a specific instance.

It works by creating synthetic data randomly in the neighborhood of the instance that we want to explain, assigning appropriate weights to them based on their proximity to the instance and training a local linear model to assign weights to the features that contribute in the prediction in the instance’s neighborhood. The intuition that LIME is based on is that it is much easier and more efficient to approximate a black box model with the usage of a simpler model in the instance’s neighborhood than trying to approximate the model globally (i.e. global surrogate method).
LIME approach has a favorable position over others because whilst it is model agnostic it is also data agnostic, meaning that it can work on problems with different kind of data (tabular, data, text). It should be noted that under the hood different algorithmic procedures are used when different data kinds of data are supplied. This is important because usually model agnostic methods only work for a specific kind of data.

![Figure 2.5: How LIME is able to approximate the neighborhood of the instance we want to explain (the bold red cross) and generate synthetic neighbors.](image)

Despite its high success, LIME is criticized in the literature as being highly unstable and that is understandable as it is to a high degree based on randomness. A different state of the art approach that is able to combat this weakness of LIME is SHAP [56] and does so by using Shapley values and relating game theory to explanations. Shapley values show the average contribution each feature has when predicting in different scenarios. It has a much more solid mathematical background than LIME and for this reason it is argued to have more accurate explanations. The drawback when compared to LIME, is that it is much slower in the computation. SHAP has the capability of outputting various global and local explanations.

This work draws inspiration by the way LIME works. When given a sentence to explain, LIME generates a neighborhood around that sentence in order to then train a linear model in that neighborhood. The neighborhood of LIME contains sentences generated synthetically. In the case of text classification, LIME works by randomly removing n-grams from the initial text given to explain and then uses all that synthetic data to build a linear model classifier to predict the class of the original sentence. In the case of fake news detection on Twitter, given a tweet post and all the replies posted beneath it, we essentially already have a small neighborhood which by default would be semantically close to the initial post (since users replying to a post are talking about the content of that post). This means that we don’t need to generate a synthetic neighborhood since the replies given beneath a
post form a much better semantically close neighborhood. Thus, we can introduce a linear model directly by training it on the replies of an initial post.

2.3.6 Explainable Fake News Detection

As seen in the previous sub-sections, explainable and interpretable machine learning is blooming and incorporating its techniques into the fake news detection task should be an important step forward. As discussed already, early detection of fake news spreading to stop further dissemination is of utmost importance. Simply, baptizing a suspicious piece of text as fake news has not been effective. Thus, there is high need for having proper information and clarifications when individuals read a piece of text to help them reach their final decision. This is exactly where explainable machine learning ties in with fake news detection [44, 69, 111].

Indeed, some attempts to incorporate explainable machine learning techniques can be found in the literature. Recently, dEFEND [92] and GCAN [55] have been proposed in aiding in the process of understanding why a machine learning model assigns a specific label in a particular news post by utilizing DNNs and various co-attention networks. dEFEND utilizes the comments made by the users in the fake news post in order to offer the final explanation, while GCAN doesn’t take into account the comments and merely uses words found in the source text. Finally, a review that utilizes SHAP [56] in order to find the most important features already proposed in various previous works by selecting them uniformly at random and ranking them based on their SHAP values was made in [76].

2.4 Overview of Related Work

An overview of the related work found in the literature that has been analyzed in this work can be seen in Table 2.1. We can see that our approach MANIFEST is one of the few approaches that offers both explainability and targets the user level. Contrasted, with the related works that also offer explainability (albeit these approaches focus on the post level), the key difference between MANIFEST, DEFEND [92] and GCAN [55] is that MANIFEST relies on a linguistic representation for its textual features while DEFEND and GCAN use contextual features. The main issue with contextual features (e.g. word2vec embeddings) is that it is very hard to understand how the features were obtained as a result of the conversion of unstructured text data into structured set of data. In a sense, the dimensionality of the features is reduced thus the ability to interpret these dimensions and trace back to the original features is lost. The way [92] and [55] are able to attain explainability and make sense of these features is by using DNNs and incorporating an extra layer with attention mechanisms [7].
Contextual features wouldn’t work well with MANIFEST because of two reasons. First, our approach does not rely on DNN structures and thus it would be impossible to trace back the most important features if we used word2vec embeddings (i.e. in the case of offer the most important words for the classification). Second, even if we disregard the above restriction, word2vec is a model that needs a considerably larger amount of time to be trained in order to convert a single text into a vector when compared to a bag of words representation. Given that a user would not want to spend a considerable amount of time waiting for the explanations for a single post to be generated this was deemed as not an acceptable trade-off.

A solution to the large computation time would be to train the word2vec model before-hand in a large amount of tweets and replies and then apply it to each Twitter thread individually. However, the problem here is that we would lose important information regarding the semantic similarity that exists in each of the neighborhoods created by the specific twitter threads (e.g. some tweets could talk about very specific events that the word2vec model would not have been pre-trained on).

Contrasted with the only other work at the user level that also offers explanations [93], the difference is that the explanations as to which features contribute the most in the prediction in their work are by offering solely global explanations by
using feature importance. MANIFEST offers both global explanations and local explanations for specific instances by utilizing user engagement.
Chapter 3

Design and methodology

In this chapter, we explore the design and methodology of our original proposal for a huMAN-centric explainable approach for Fake news Spreading deTection (MANIFEST) to detect fake news spreading and also offer explanations in the form of examples and feature importance to the end-user.

The architecture of the overall MANIFEST is split into three different stages or phases with each phase containing distinct important and clear laid out steps. These phases include the following: phase A: Training of Fake News Spreader classifier, phase B: Creation and annotation of tweet post-replies dataset, and phase C: Training of interpretable white box model on each tweet post-replies data pair. From now on they will be simply called phase A, B, and C for simplicity purposes.

The reasons these phases were divided as such is due to the following. Phase A can be seen as a typical machine learning text classification task with the added component of the application of state of the art explanation techniques in the end of the pipeline. Phase B contains the creation of two new datasets to be used in the final stage by using the Twitter API, twint and tweepy and finally annotating it with the classifier made in phase A. Phase C finally involves the training of a linear model for each twitter thread contained in the datasets from phase B and the extraction of the explanations.

3.1 Overall architecture

Before expanding on each phase in the following sub-sections, a brief simple step by step procedure done on each phase will be given next. On Figure 3.1 a general schema of the architecture of phases A and B is presented. Briefly, phase A (Training of Fake News Spreader classifier) includes the following:

1. The preparation of the fake news spreader dataset in the appropriate form.
2. The extraction of features from each Twitter user profile.

3. The training of various classifiers with the usage of the extracted features.

4. The testing of how well the classifier works on unseen data.

5. Picking of the best performing classifier and saving it for subsequent phases.

6. In-Depth analysis of the features contained on the picked classifier via various explainable machine learning techniques.

Respectively, phase B (Creation and annotation of tweet post-replies dataset) includes the following:

1. The collection two datasets with the usage of the Twitter API, containing tweets related to two separate topics.

2. The usage of various tools to get for each tweet gathered above the replies made to that tweet and the creation of an tweet post-replies dataset.

3. Filtering of this dataset

4. Getting the user corresponding to each tweet in that dataset.

5. Feature extraction done on this user, in the same manner that was done in phase A.

6. Feeding the data to the classifier trained in phase A and getting the corresponding labels (whether the user that left that comment or post is a real or fake news spreader).
7. Collectively, end up with a new annotated dataset containing tweet post and replies along with the label given by the fake news spreader classifier.

Finally, Figure 3.2 depicts the schema of the architecture of the final phase C (Training of interpretable white box model on each tweet post-replies data pair).

1. Selecting only the text and label taken from the dataset created in phase B.
2. Iteration of each tweet post-replies data pair one by one.
3. Transformation of the text gained above into a latent representation.
4. The training of a linear model in that latent space.
5. Getting the word feature importance for the classification made by the linear model.
6. Getting the two closest replies in the latent space for each label as example based explanations.
7. Evaluating the outcome by comparing the outcome of the classification made by the fake news spreader classifier with the classification made by the linear model (fidelity).

![Figure 3.2: Architecture depicting the final phase C](image)

In the following sections, the methodology along with the technical details of the implementation of the architecture of MANIFEST will be explored. However, any experimentation done to choose for example the best type of classifier or features will not be explored here but in the following chapter.

Each of the following sections covers one of the phases already introduced. A few technical details before we continue to the next section. The programming language that was used for our implementation was Python 3.8.6 along with the help of several python libraries of which the main ones can be grouped in the following categories:
• **sklearn and xgboost.** For the training of the various machine learning classifiers and other ML related tasks.

• **pandas and numpy.** For data storage and manipulation.

• **lime, shap, pdpbox and eli5.** For producing various explanations in the local and global level.

• **twint, tweepy, requests.** For communicating with the Twitter API and collecting the various datasets needed.

• **nltk.** For aiding in the pre-processing of the text.

• **textblob, vaderSentiment and emoji.** For analysing the sentiment of texts in the feature extraction procedure.

• **pickle.** For storing and reusing ML models.

### 3.2 Training of Fake News Spreader classifier

The first phase of our architecture can be characterized as a standard machine learning procedure pipeline along with the application of various tools that are able to offer local and global explanations on the final trained classifier. More details for each step of phase A will be given next.

#### 3.2.1 Fake News Spreader dataset

As with any machine learning task, an appropriate dataset was needed in order to train our algorithms. To facilitate all the requirements that were needed for our task we had to find a dataset that was based on Twitter and that had information regarding the user profiles of these Twitter users. The dataset that covered our needs is the Profiling Fake News Spreaders on Twitter dataset [75], from now on simply fake news spreaders dataset. This dataset is hosted in zenodo and was designed to facilitate the annual pan-clef challenge regarding author profiling. The purpose of the challenge was to identify potential fake news spreaders with the end goal of preventing the propagation of these news from the dangerous users to others on social media.

In this specific pan-clef challenge, it was shown that simple linguistic features (ngrams) usually outperformed more complex features (e.g. contextual features such as bert embeddings or sentiment features). Overall, a mix of different features was proposed including readability, sentiment, emotion and personality in the different works proposed to solve this challenge. However, no explanations
were offered on the results (e.g. Is there any significant difference between the personality features among the two classes?).

The dataset contains the last 100 tweets of 300 users on Twitter. 150 of the users belong to the class real news and 150 of the users belong to the class fake news. The format in which the dataset was supplied contained 301 files of which 300 of these where the xml files containing the 100 tweets of each user along with the user’s id as the name of the file. The final file contained the ground truth along with the user id corresponding to every user. To create the ground truth the authors annotated the dataset manually with the help of fact checking websites such as PolitiFact and FactCheck.org.

To use this dataset and feed it into our machine learning classifiers appropriate transformations were deemed necessary. We needed a unified representation of all the data, where each line corresponded to one user, contained all the texts and finally the ground truth for that user. A python script was thus implemented that parsed the xml files and was able to create our final desired csv file. As per our specifications, this file contained the following columns: user_id, tweet_text (all 100 tweets for the given user_id concatenated) and ground_truth.

3.2.2 Text pre-processing

As with any machine learning task dealing with text, there has to be a way to clean the data. The transformation of unstructured text into a structured set of data is not a straight-forward task and the areas of text mining and natural language processing offer a wide variety of different tools and approaches to face the irregularities which materialize in texts containing natural language.

The pipeline of text pre-processing consists of a number of stages that aim to further improve the results of the various machine learning algorithms. In this work, the following steps were used for the pre-processing of the text data used.

- Remove links
- Remove usernames
- Remove punctuation
- Lowercasing
- Remove special Twitter words such as ”rt”, ”via” and ”amp”
- Remove consecutive non-ASCII characters
- Replace contractions with their full word counterparts (e.g. ”ain’t” becomes ”is not”, ”aren’t” becomes ”are not”, etc.)
- Remove all non alphanumeric values
• Stopword removal
• Tokenization
• Removal of words that are less than 2 characters, as they are deemed insignificant.

The above steps of text pre-processing are used in some form in various different stages of our work. In some places all of the above steps are used while in others some of the information needed to be preserved thus some steps were skipped (e.g. removal of emojis, URLs, etc.).

3.2.3 Feature engineering

As already explored in the literature review of the previous chapter there is a wide variety of features that can be exploited and help in the creation of accurate classifiers. The features that were chosen in this work can be broadly put in the following categories:

• Linguistic: The transformation of the raw text data into a vector representation. In this work, the Term Frequency–Inverse Document Frequency or TF-IDF vectorizer was used. A maximum of 1000 features were used to create the vectors.

• Readability: Various features that give statistical information regarding the text or how the text is read. Includes the following features: average word count per tweet, emoji count, slang count, capitalized count (e.g. "Apple", "Woof", "Dog" but not "APPLE" etc.), full capitalized count (e.g. "APPLE", "WOOF"), retweets count, user mentions count, hashtags count, URL count. In total 9 features.

• Sentiment: Contains features regarding the emotion behind the text such as anger, fear, joy and sadness. Also contains the number of words in negation (e.g. "not", "no") as well as the sentiment given by the the python libraries VADER and TextBlob vader compound score and textblob polarity score. In total 7 features.

• Psycholinguistic: Contains features returned by the LIWC lexicon. More specifically, the four summary variables which include the ‘Analytic’, ‘Clout’, ‘Authentic’ and ‘Tone’ features. A total of 4 features

• Personality: Features describing the personality of the user based on the language they use. Contains features such as avoidance, anxiety along with the big 5 personality trains which include extroversion, conscientiousness, openness, neuroticism, agreeableness. A total of 7 features.
• **Gender**: A single feature with the gender of the user (male or female) as predicted based on the language they use. In total, 1 feature.

Tying in with the types of data for interpretable models as shown in sub-section 2.3.2, these features can be further put into the categories of textual data such as in the case of the linguistic features and tabular data such as the case for the readability, sentiment, psycholinguistic, personality and gender features. Since various explanation methods work differently under the hood when given different kinds of data (text and tabular in our case), we had to create two separate models, one which contains only the tabular data (all features minus the linguistic), to draw the explanations from and one that contains all of the data combined. More information on this and also how all the features were calculated will be given next.

### 3.2.3.1 Linguistic

Following all the steps detailed in the pre-processing sub-section 3.2.2, the text is cleaned. Afterwards, we are ready to transform our text into vectors. A tf-idf vectorizer from sklearn.feature_extraction.text was used with the following parameters: max_features=1000, ngram_range=(1, 3), min_df=0.01, max_df=0.90. The parameter ngram_range=(1, 3) denotes that we take into consideration not only unigrams, but also bigrams and trigrams, as has been shown in related work to be highly effective [10]. This way, we also give value to sets of words that tend to appear together frequently. The parameter min_df removes tokens that exist with too low frequency. For example, a value of 0.01 signifies to not take into consideration tokens that appear in less than 1% of all the documents. On the contrary, max_df removes tokens that exist with too high frequency. For example, a value of 0.90 means not to take into consideration tokens that appear in more than 90% of the documents.

### 3.2.3.2 Readability

To extract the readability features, the text had to stay on its original form. This means that absolutely no pre-processing was done. Most of the features here were extracted by using regular expressions. The slang count was computed with the help of a slang dictionary found on webopedia which contains 119 slang words used on Twitter along with abbreviations

### 3.2.3.3 Gender

Similarly to the tf-idf case, the problem here is a simple binary text classification problem. We have a dataset that contains the Twitter user ids of the users and

---

a label denoting their gender as ground truth.

Using the tweepy module to communicate with the Twitter API, a script was implemented that for each user id, fetched the last 100 of their tweets. Afterwards, all of the tweets were combined to create a new dataset which now contained the user id, the text of all the 100 tweets combined and the gender of the respective user.

Following the footsteps of previous research as showcased in 2.2.3, a mix of readability and linguistic features were then extracted from the texts. Afterwards, these features were normalized in the [0,1] range and were then fed into various machine learning algorithms. The way the best one was chosen will be explained in the following chapter. After, the best is found it is saved using pickle for later use. The tf-idf vectorizer used to transform the text was also saved using pickle as the same tf-idf features as the one the model was trained on need to be used to predict on unseen data.

In order to find the gender of a new Twitter user, the same features need to be extracted from their profiles and then fed onto the saved vectorizer and model.

### 3.2.3.4 Sentiment

In this category we have the emotion (anger, fear, joy and sadness) the sentiment as given by VADER and TextBlob as well as the negation count.

For the emotion, the python library emoji is used which transforms the emojis into a corresponding text. Then, an emoji lexicon is used that assigns to each emoji values for the anger, fear, joy and sadness. Then for each emoji in a text these values are added up and the final score is calculated.

The way TextBlob works is by weighing all the words by a score. This score is calculated with the usage of a sentiment lexicon. The final values for a sentence are determined by using a weighted average of all the scores of a word in a sentence. TextBlob can return the polarity and subjectivity of given sentences. In this work, we only utilize the polarity which is a value lying between the [-1,1] range with -1 characterizing a negative sentiment and 1 characterizing a positive sentiment.

Similarly, VADER’s compound score is processed by adding the valence scores of each word in a sentence again by using a lexicon. Afterwards, these values are normalized to lie between the [-1,1] range with -1 being the most negative sentiment and 1 being the most positive.

Finally, with all the steps described above the final number of features in all 6 categories is 1028 (1000 textual features and 28 tabular features).
3.2.4 Fake News Spreader classifier

Having completed the feature engineering stage, we are now ready to train our machine learning algorithm. To find the model that would best fit our dataset we decided to test a variety of different predictive models suitable for text classification, including support vector machines, naive bayes, logistic regression, random forest and gradient boosting.

Each of the above model also has multiple hyper-parameters that need fine-tuning. To find the best hyper-parameters a set of them was selected for each model and then with the use of RandomizedSearchCV, which searches across all the different combinations that we have set, the best combination of them was found. In the case of the KNN algorithm only one hyper-parameter exists; namely how many neighbors to take into account. For this reason, GridSearchCV was used instead.

The performance of the model was measured using cross validation with 10 folds. The metrics that were chosen were accuracy, precision, recall and f1-score. After finding the best performing ML algorithm (by measure of the accuracy), the classifier was fitted on all the dataset for having a more accurate final model. The number of folds was chosen to be 10 as a trade-off between the high execution time and higher accuracy in building the model in contrast to not using any folds. The final fitted model was saved using pickle.

3.2.5 Explanations

For producing the explanations for the fake news spreader classifier, a variety of state of the art methods in the field of explainable machine learning were explored. These include ELI5\(^2\), PDP [29], LIME [78] and SHAP [56]. A few words for each of these methods will be given next.

ELI5, short for Explain like I am 5 years old, is a Python module developed to aid in the machine learning debugging process. The way it does that is by producing intuitive explanations by providing the feature importance for the model’s predictions in a clear and straightforward way. The explanations created can represent both the local and the global view of the model.

Partial dependence plots, in short PDP, are able to give a visualization of the impact that a feature can have on the resulting predictions. In other words, they show how different values of a single feature affects the final prediction. PDP explanations can be characterized as global since they describe the behavior of the feature in the model as a whole.

LIME and SHAP have already been introduced in the previous chapter. In contrast with PDP, LIME offers explanation only on the local level by trying to

\(^2\)https://github.com/eli5-org/eli5
 approximate the behavior of the black box in the neighborhood of the instance we want to explain. SHAP is able to offer both global and local explanations in the form of feature importance, partial dependence plots, force plots and others. As mentioned in the previous chapter, SHAP is more stable than LIME because of the more solid background it carries as it relates game theory with the creation of the explanations and doesn’t rely on randomness as LIME does in the synthetic data generation.
3.3 Creation and annotation of tweet post-replies dataset

Phase A has been completed and our final model is saved in pickle form ready to be passed on to phase B. This section involves the collection of our data from Twitter along with the creation of our novel dataset which contains the initial tweet of a user along with all the replies or comments given to that initial tweet via appropriate transformations.

3.3.1 Dataset collection and details

For collecting the data needed from Twitter, the Twitter Streaming API was used. The information gathered was then stored away in a MongoDB database, as they are a very attractive choice for efficient data storage and retrieval.

A script was created that implements a stream listener class that streams the information as they are posted on Twitter dependent on a particular subject and saves them to a MongoDB collection of documents. The Tweepy module was utilized for aiding in the implementation of the stream listener and pymongo for easier communication with MongoDB.

The tags that we chose to analyze involved the United States 2020 presidential elections (using the hashtags USElections and Elections2020) and the COVID-19 pandemic (using the hashtags coronavirus, covid, covid19) two extremely hot topics at the time of this research that as stated in the previous chapters are plagued with the circulation of fake news stories. The tweets were only streamed in the English language by applying an appropriate filter. In the end, our collections in MongoDB for the topic US elections reached 1.7 million documents and for the topic COVID-19 2.8 million documents.

3.3.2 Creating the User replies dataset

So far, we have collected all the tweets for our two topics. Now we need to create the post-replies dataset. All we need to get are the replies assigned to those tweets to create datasets that involve the whole conversation threads. Twitter API doesn’t support the returning of replies to tweet posts directly. Newly introduced in the Twitter API 2 there is an option to get the conversation_id of a tweet when given its tweet id. The conversation_id is the same for all tweets that belong in the same thread which means that the original post along with all of its replies will share the same conversation id.

Thus to find all replies to an original Twitter post all that needs to be done is firstly get the conversation_id of that post and then search through all replies made
to that user_id and then cross-check if the conversation_id of those replies is equal to the original tweet’s conversation_id. Finally, store the tweet ids of the original tweet and of all the replies.

Twitter API only allows the search of tweets directed to a particular user for only the last week (7 days). To overcome this limitation, the twint module is used which has no restriction on how far back the tweets can be scraped. In order to make the amount of replies that had to be searched fewer a time filter of 5 days had to be used. This means that the script searches all the replies done to a specific user from the date and time their tweet was posted up to exactly 5 days after.

A self-evident issue that arises is that the vast majority of tweet posts contain no replies. When this happens the tweet gets removed from the procedure and not saved in the final post-replies dataset. To try and combat this issue and also save some computational time, we made the assumption that users with few followers would generally not have sufficient replies on their posts. Thus, an additional filter was added and if a user had less than 5000 followers, then his tweet was skipped before searching through all the user replies directed at him through twint.

The above approach relies on several different external services (Twitter API, tweepy, twint) to get the required information and thus appropriate measures had to be taken in the case any of them faced issues so that our script would not crash. For this reason, various exception handling techniques had to be implemented.

Finally, our tweet post-replies dataset is created and is a json file of the form:

```json
{
    initial_tweet_id1 : [reply_id1 ,... , reply_idn],
    initial_tweet_id2 : [reply_id1 ,... , reply_idm],
    
    initial_tweet_idn : [reply_id1 ,... , reply_idk]
}
```

### 3.3.3 Dataset transformation and annotation

Our twitter post-replies dataset has been prepared. However, it only contains the ids of the tweets and the replies. Appropriate transformations are needed in order to have all the necessary information to perform the feature extraction for our users. For this task, the Twitter API was used to obtain the tweet text and user id of the given tweet id. The tweepy module was used to also obtain the last 100 tweets of a user given their user id.

Firstly, we load the json file saved in the previous procedure. Because a machine learning algorithm (see also next section) requires a minimal amount of data
in order to be trained and have sufficient results, we decided to keep only tweets that have at least 10 replies. This means that all tweets with fewer than 10 replies were removed. At the same time, because there are many requests made to the twitter API and it imposes the limit of 300 requests per each 15 minutes time window, a decision was made to also remove all tweets that have more than 200 replies.

With the above filtering, we end with 1365 tweets that have more than 10 comments and less than 200 for the us elections dataset and 308 tweets that have more than 10 comments and less than 200 in the covid dataset.

Iterating through all the remaining tweet ids and then all the replies, we finally create a new json file rich with useful information. This includes the tweet id of the initial tweet, the user id of the user that posted the tweet, the text of the tweet that was posted, the text of the last 100 tweets of the user that posted the tweet and all the replies given to the initial tweet in the same form.
For better understanding, the following schema was used for the json file:

```json
[
  {
    'tweet_id': initial_tweet_id,
    'user_id': initial_user_id,
    'tweet_text': initial_tweet_text,
    'text': initial_100_tweets,
    'replies': [
      {
        'tweet_id': initial_tweet_id,
        'user_id': initial_user_id,
        'tweet_text': initial_tweet_text,
        'text': initial_100_tweets,
      },
      ...
    ]
  },
  ...
  {
    'tweet_id': initial_tweet_id,
    'user_id': initial_user_id,
    'tweet_text': initial_tweet_text,
    'text': initial_100_tweets,
    'replies': ...
  }
]
```

The final step in phase B is to annotate all the users contained in the resulting...
json file by using our already trained fake news spreader classifier from phase A. This procedure is fairly simple as all the information needed is already saved in the json file. All that needs to be done is iterate through each user, extract the same features that were also extracted in phase A from their 100 most recent tweets and then feed them into the fake news spreader classifier. Finally, the classifiers returns the appropriate labels for each user and we are able to match that label with the text they left on their reply.

It should be noted that we also find the label for the user that left the initial tweet aside from the users that replied to it. However, we don’t use that label as the final classification returned by MANIFEST but rather as an evaluation measure. The reason why and how this is done will be explained with more detail in phase C.

3.4 Training of interpretable white box model

Phase A and phase B have been completed and we have all the necessary information to continue onto the final phase C, train an interpretable linear model and offer to the end-user the final classification along with explanations in the form of feature importance. Finally, we are able to evaluate the performance of our model by using the fidelity measure.

3.4.1 Linear model

Before we go into more technical details regarding the implementation, some background regarding linear models will be explored. Linear models are widely used models that work by learning a linear function by using the features that are given as input. Linear models exist both for classification and for regression problems. More specifically, when working with regression problems, the function a linear model has to learn is:

\[ y = w_0 * x_0 + w_1 * x_1 + ... + w_p * x_p + b \]

Here, \( x_0, x_1 \) to \( x_p \) show for a single data point the number of features. The parameters \( w \) and \( b \) are the ones that the model needs to learn. The output that is given by the algorithm is given by \( y \) in the equation. In simple words, \( y \) is just a weighted sum of the input features \( x_0 \) to \( x_p \), alongside the weights given by the respective weights \( w_0, w_1 \) to \( w_p \) and by adding \( b \) which denotes the intercept [65].

Many different ways exist to find the parameters \( w \) (slope) and \( b \) (intercept), based on the input data points. All these different ways to calculate them give us the different kinds of linear model algorithms. One of the most widely used is the algorithm known as linear regression, an example of which is given in the following figure.
Classification problems can also be solved with linear models. However, rather than calculating the weighted sum of the input features, as is done in regression, this time we set the predicted value to be zero. Afterwards, if the linear function gives us a value which is less than zero, then the algorithm predicts that it belongs to the negative class and likewise if it is greater than zero, then it is predicted as belonging to the positive class. The function that describes a linear model for classification is the following:

\[ y = w_0 x_0 + w_1 x_1 + \ldots + w_p x_p + b = 0 \]

The above general rule for making predictions using linear models for classification is common among all such algorithms. As in the regression case, the differences between the various algorithm lies about the calculation of the values of \( w \) and \( b \). The most commonly used algorithm in this case is called logistic regression.

The fact that linear models offer linearity is also their greatest advantage as it makes the prediction procedure simple and allows the interpretation of the learnt model. This why linear models see such widespread use in other more sensitive fields than machine learning such as medication, human sciences, brain research, and numerous other quantitative examination fields. For instance, in medicine, it isn’t simply imperative to anticipate the clinical result of a patient, but to additionally evaluate the impact of the medication and simultaneously take age, sex and various different features into account in an interpretable manner [63].

In logistic regression in particular, we can use the weights learnt by the linear model in order to find out the features that have the highest impact in the classi-
Figure 3.6: The differences between linear regression and logistic regression

A global level interpretation of this model would be to show the weights (the coefficient of the features or otherwise feature importance) that is learned by the model and assigned to each feature. To get the local level interpretation all we have to do is multiply the vector fed as input with the weight vector. This way we get the weights of the features contained in the instance we want to explain (as features that don’t exist in the instance we want to explain would have a value of zero in the input vector). The final classification made by the linear model is of course the summation of all these weights multiplied by their features and finally by adding the learnt intercept. Thus, by observing the values of the weights for each feature we can understand which class that feature contributes to and to what extent. Lastly, the intercept, also called bias, shows the expected value for the prediction when all other features equal to zero.

3.4.2 Final explanation

So far, we have the text of each tweet along with the label given to that tweet. We have explained how the linear model works and how it can be interpreted. By adhering to the global surrogate method introduced in section 2.3.4, what we aim to do is approximate the predictions of the fake news spreader classifier by using and training a simpler linear model solely on the text that a user classified as fake or real news spreader left. After the linear model learns the logic of the more complex fake news spreader classifier then we can interpret the results.

In other words, for each item in the json file we train a linear model with input features the texts of all the replies left on the initial tweet and ground truth the labels assigned to the users that left this particular reply by the fake news spreader classifier. After the linear model is trained we use it to predict the value of the initial tweet text and afterwards interpret the classification. In the end the label given by the linear model is compared with the label given by the fake news spreader classifier for the initial user for evaluation (see following sub-section).
To feed the text into the linear model we need first to transform it into some sort of latent representation. For the purposes of this work, we decided to use two different latent representations and compare their results. The first one is based on the bag of words (BOW) as implemented by the CountVectorizer in sklearn which counts the frequency of words using unigrams. The second one is the term frequency–inverse document frequency (TF-IDF) procedure which scores words by ranking how important they are in a document with respect to the whole corpus as implemented by the TfidfVectorizer in sklearn. Before doing that, we also apply all the steps of the pre-processing as described in 3.2.2. For the implementation of the logistic regression the version from sklearn was used.

It should be noted that in some cases, all the replies given to an initial Twitter post might happen to all have the same label showcasing some form of homogeneity. Logistic regression as implemented by sklearn is not programmed to handle such a case and crashes. When this happens, MANIFEST procedure assigns as label for the initial post, the label that all the replies share. In such cases since no model is trained no explanation can be offered. But since all the replies seem to share the same label it can be argued that the particular post would be really polarizing (either very likely fake news or very likely real news).

In other cases, there could be high class imbalance between the fake news spreaders and the real news spreaders that the logistic regression classifier is called to model. In such cases, the logistic regression could learn a highly biased model by having a skewed intercept. To combat this issue, we used the balanced mode when creating the logistic regression model which uses the values of y to adjust the weights to a proportion which is inverse to that of the class frequencies found in the training set and thus combating the class imbalance issue.

Finally, we need to present to the user the two closest replies from each class. We have already converted the original post and the replies to a vector representation (BOW or TF-IDF) to train the linear model. To present to the user the two closest replies, we calculate the cosine similarity between the initial tweet’s vector representation and all the replies’ vector representation. Afterwards, all replies are ranked on descending order based on that cosine similarity and the two that have the highest similarity from each class are chosen.

These replies are offered to the user as example-based explanations. These explanation strategies offer specific occurrences of the dataset to help clarify the logic of ML models. As a rule, example-based explanations work better if the features that the model has been trained on can be further utilized to convey more context. This means that data in the form of text or images excel at this category while for tabular data it is more difficult to address such information in a concise manner. Example-based explanations help people better understand ML models as well as the data that was used to train the ML model under inspection.
3.4.3 Fidelity

A common practice in the field of explainable ML to evaluate how well a white box model learns to imitate a black box is the fidelity measure [63]. It basically checks whether the white box model’s predictions agree with those of the black box. A definition can be found in [36].

**Definition 6.** The degree in which the simpler model under inspection can be used to precisely approximate the predictions of a more complex model.

As already stated in the literature review chapter and more specifically according to the global surrogate method, it is often valuable to create a new interpretable model that tries to mimic or approximate a more complex black box model.

Fidelity shows the ability of an interpretable model to mimic the inner logic of a more complex model. Fidelity is similar to the well known metric of accuracy in the sense that it is calculated with the same evaluation techniques, with the sole distinction being that it is calculated concerning the result of the black box model it is approximating rather than the ground truth label.

Thus, treating as ”ground truth” the output that the black box model (fake news spreader classifier) gave us we check how well the white box model (linear model) performs and if the final performance is good, it means that we have indeed made a successful simpler model.

Figure 3.7: An example of how fidelity can be calculated given the predictions of a complex and a simple model
Chapter 4

Experimentation and results

4.1 Hyper-parameter tuning

The ML algorithms that were chosen to perform the various classification tasks were the following:

- K Nearest Neighbors (KNN)
- Logistic Regression (LR)
- Multinomial Naive Bayes (NB)
- Support Vector Machine (SVM)
- Random Forest (RF)
- Gradient Boosting (GB)

To find the best hyper-parameters for them, RandomizedSearchCV and GridSearchCV are used from sklearn.model_selection module. The differences between the two is that in grid search, the algorithm searches through all possible combinations given in the hyper-parameter values and trains a new model for each whereas in random search combinations are randomly sampled. Since the above machine learning algorithms all have different hyper-parameters, a different set of them to search was chosen.

For all the algorithms tested, initially a random search was run to sample from an initial set of hyper-parameters. Afterwards a grid search was run close to the parameters returned by the random search to get even more refined results.

For the K nearest neighbors specifically, there is only one hyper-parameter that can be tuned which is the number of neighbors to consider. The values that are searched in this context are in the [1,100] range, by using only grid search and sampling 70 of them at random.
For all random searches performed with the RandomizedSearchCV, the following hold true:

- scoring = 'accuracy', the metric that is compared.
- cv = 3, the number of folds for cross validation.
- n_iters = 50, how many samples are taken, trade-off between computational time and quality of results.
- random_state=42.

For the logistic regression algorithm the following hyper-parameters were considered for the randomized search parameter grid:

- C, with possible values: [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0].
- multi_class, with possible values: ['ovr'].
- penalty, with possible values: ['l2'].
- solver, with possible values: ['liblinear', 'newton-cg', 'sag', 'saga', 'lbfgs'].

For the naive bayes algorithm the following hyper-parameters were considered:

- alpha, with possible values: [0, 1].
- fir_prior, with possible values: [True, False].

For the SVM algorithm the following hyper-parameters were considered:

- 'C': [0.0001, 0.001, 0.01],
- 'degree': [1, 2, 3, 4, 5],
- 'gamma': [0.0001, 0.001, 0.01, 0.1, 1, 10, 100],
- 'kernel': ['linear', 'rbf', 'poly'],
- 'probability': [True, False]

For the RF algorithm the following hyper-parameters were considered:

- 'n_estimators': 5 randomly sampled in the [200,1200] range,
- 'max_features': ['auto', 'sqrt'],
- 'max_depth': 5 randomly sampled in the [20,120] range,
- 'min_samples_split': [2, 5, 10],
CHAPTER 4. EXPERIMENTATION AND RESULTS

74

• 'min_samples_leaf': [1, 2, 4],
• 'bootstrap': [True, False]

Finally, for the gradient boosting algorithm the following hyper-parameters were considered:

• 'n_estimators': 5 randomly sampled in the [200,1200] range,
• 'max_features': ['auto', 'sqrt'],
• 'max_depth': 5 randomly sampled in the [20,120] range,
• 'min_samples_split': [2, 5, 10],
• 'min_samples_leaf': [1, 2, 4],
• 'learning_rate': [0.1, 0.5],
• 'subsample': [0.5, 1.0]

4.2 Predictive performances

4.2.1 Gender classifiers

For predicting the gender, all the machine learning algorithms mentioned in section 4.1 were utilized. Initially, we present the results after training the models with the random search. To test the performances of our models we split our data into 80% training data (to perform the random and grid search) and 20% testing data to test the final model on completely unseen data. In Table 4.1 the performance of the six gender classifiers after performing the random search is shown, in terms of accuracy, precision, recall and f1-score, while in Table 4.2, the same metrics are shown after performing the grid search.

<table>
<thead>
<tr>
<th></th>
<th>KNN</th>
<th>NB</th>
<th>LR</th>
<th>SVM</th>
<th>RF</th>
<th>GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy</td>
<td>0.60</td>
<td><strong>0.71</strong></td>
<td>0.69</td>
<td>0.65</td>
<td>0.67</td>
<td><strong>0.71</strong></td>
</tr>
<tr>
<td>precision</td>
<td>0.68</td>
<td><strong>0.71</strong></td>
<td>0.71</td>
<td>0.67</td>
<td>0.69</td>
<td><strong>0.71</strong></td>
</tr>
<tr>
<td>recall</td>
<td>0.52</td>
<td><strong>0.71</strong></td>
<td>0.67</td>
<td>0.63</td>
<td>0.65</td>
<td>0.70</td>
</tr>
<tr>
<td>f1-score</td>
<td>0.41</td>
<td><strong>0.71</strong></td>
<td>0.66</td>
<td>0.62</td>
<td>0.65</td>
<td>0.70</td>
</tr>
</tbody>
</table>

Table 4.1: The performances of the gender classifiers after performing the random search.

We can see in Table 4.1 that the best performing gender classifiers are the naive bayes and gradient boosting algorithms with similar performances with NB leading
slightly in the recall and f1-score metrics. Next, we will perform a more extensive grid search to see if we can improve on these baseline models. Since KNN and NB don’t have any more parameters to search through they will be excluded from this search.

<table>
<thead>
<tr>
<th></th>
<th>LR</th>
<th>SVM</th>
<th>RF</th>
<th>GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy</td>
<td>0.69</td>
<td>0.54</td>
<td>0.67</td>
<td>0.69</td>
</tr>
<tr>
<td>precision</td>
<td>0.71</td>
<td>0.27</td>
<td>0.70</td>
<td>0.69</td>
</tr>
<tr>
<td>recall</td>
<td>0.67</td>
<td>0.5</td>
<td>0.65</td>
<td>0.68</td>
</tr>
<tr>
<td>f1-score</td>
<td>0.66</td>
<td>0.35</td>
<td>0.64</td>
<td>0.67</td>
</tr>
</tbody>
</table>

Table 4.2: The performances of the gender classifiers after performing grid search.

After performing the more extensive grid search, we can see in 4.2 that we don’t manage to find a better model when it comes to predict the unseen data compared to the ones before. Since the best performing model is the NB one, that is the one picked to predict the gender of the users in the fake news spreader dataset.

4.2.2 Fake News Spreader classifiers

For building the fake news spreader classifier, all the machine learning algorithms from section 4.1 were chosen except from the KNN one, as it didn’t have satisfactory performance in the gender prediction task and is generally not supported for high dimensional input features as is the case here. Before feeding the features as described in 3.2.3 into all the classifiers, a scaler was applied to normalize the values of all the input features in the [0,1] range as algorithms such as the logistic regression and naive bayes are unable to work with negative values.

As in the gender case, to find the best hyper-parameters we used 80% of the dataset and then tested the models with the best hyper-parameters on the rest of the 20%. To have ever more accurate results, after finding the best hyper-parameters for each algorithm, we performed a cross validation evaluation by using 10 folds on the whole dataset in which we used as metrics the mean values of accuracy, precision, recall and f1-score as taken from each fold.

Firstly, the fake news spreader classifier with the tabular features is trained. As stated in the previous chapter, this classifier is the one used to draw the explanations from regarding phase A. The second classifier, that contains both text and tabular features and will be used in phase C is trained afterwards. The metric that we used to pick our best model was chosen to be the accuracy and in the case of a tie, the f1-score (which is the weighted average of recall and precision).
Table 4.3: The performances of the fake news spreader classifiers with only tabular features after performing the random search.

Table 4.3 shows us that the best classifier is the RF with 0.72 accuracy, with the GB classifier following second with 0.7 accuracy. LR and SVM follow after and NB is the least accurate with an unsatisfactory performance of 0.59 accuracy. It is worth noting that NB had the best recall among all classifiers, while having the worst precision. Moving forward, we will see if we can improve on these classifiers by using a more targeted grid search to further improve the hyper-parameters. NB will be excluded since it doesn’t have any more hyper-parameters to tune.

Table 4.4: The performances of the fake news spreader classifiers with only tabular features after performing the grid search.

In Table 4.4 we can see the performances of our classifiers after performing the grid search. The performances didn’t change significantly and the best performing is the RF once again. Thus, the RF classifier is picked out of all the algorithms to move to the last step of phase A and draw the explanations. The exact same procedure is followed for the classifier which considers only text features. The performance of the best performance found for both classifiers when considering only text and only tabular features is found in Table ??.

Next, we train the fake news classifier with both tabular and textual features.

The results of all the algorithms can be seen in Table 4.5. The best performing algorithm is the GB with 0.73 accuracy. It is interesting that all the other algorithms scored 0.71 accuracy but all with slightly variable results for the other metrics. Next, we will see if we can improve the hyper-parameters of these models even further.

In Table 4.7, we can see the results of all the algorithms after the grid search. The best performing algorithms are now the GB and the RF with same values in
CHAPTER 4. EXPERIMENTATION AND RESULTS

<table>
<thead>
<tr>
<th></th>
<th>NB</th>
<th>LR</th>
<th>SVM</th>
<th>RF</th>
<th>GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy</td>
<td>0.71</td>
<td>0.71</td>
<td>0.71</td>
<td>0.71</td>
<td>0.73</td>
</tr>
<tr>
<td>precision</td>
<td>0.72</td>
<td>0.72</td>
<td>0.76</td>
<td>0.71</td>
<td>0.74</td>
</tr>
<tr>
<td>recall</td>
<td>0.71</td>
<td>0.71</td>
<td>0.64</td>
<td>0.70</td>
<td>0.70</td>
</tr>
<tr>
<td>f1-score</td>
<td>0.71</td>
<td>0.71</td>
<td>0.69</td>
<td>0.71</td>
<td>0.72</td>
</tr>
</tbody>
</table>

Table 4.5: The performances of the fake news spreader classifiers for phase C with both text and tabular features after performing the random search.

<table>
<thead>
<tr>
<th></th>
<th>LR</th>
<th>SVM</th>
<th>RF</th>
<th>GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy</td>
<td>0.73</td>
<td>0.71</td>
<td>0.73</td>
<td>0.73</td>
</tr>
<tr>
<td>precision</td>
<td>0.75</td>
<td>0.80</td>
<td>0.73</td>
<td>0.75</td>
</tr>
<tr>
<td>recall</td>
<td>0.69</td>
<td>0.57</td>
<td>0.71</td>
<td>0.70</td>
</tr>
<tr>
<td>f1-score</td>
<td>0.71</td>
<td>0.66</td>
<td>0.72</td>
<td>0.72</td>
</tr>
</tbody>
</table>

Table 4.6: The performances of the fake news spreader classifiers for phase C with both text and tabular features after performing the grid search.

the accuracy and f1-score. The GB was picked as our final model because it had slightly higher f1-score (that can’t be seen with this 2 point decimal representation). It is worth noting that the performance of the LR also increased, while the SVM stayed about the same.

In the final table below, we compare between the models made with only text features, only tabular features and the model made with both tabular and textual features.

<table>
<thead>
<tr>
<th></th>
<th>RF (text)</th>
<th>RF (tabular)</th>
<th>GB (tabular+text)</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy</td>
<td>0.70</td>
<td>0.72</td>
<td>0.73</td>
</tr>
<tr>
<td>precision</td>
<td>0.70</td>
<td>0.72</td>
<td>0.75</td>
</tr>
<tr>
<td>recall</td>
<td>0.71</td>
<td>0.73</td>
<td>0.70</td>
</tr>
<tr>
<td>f1-score</td>
<td>0.69</td>
<td>0.72</td>
<td>0.72</td>
</tr>
</tbody>
</table>

Table 4.7: The performances of the final picked fake news spreader classifiers for phase A, including only text features (left), only tabular features (center) and phase C with both text and tabular features (right).

We can see that the models have varying levels of performance depending on the feature combination used. The performance of the model with only textual features (tf-idf) achieves the worst performance. The performances of the other two models (tabular only and text+tabular) is about the same, with the GB which considers both tabular and textual features slightly higher. These findings are im-
important because it means that even though we don’t use textual features in the RF model that will be used to draw the explanations, we don’t have a significant loss in accuracy.

Next up, we compare the performance of our best model (GB) with other approaches that took part in the PAN-CLEF challenge [10]. In total 65 different approaches were proposed in this challenge. It should be noted that the challenge involved taking into account both the English and Spanish aspect of the dataset. Since we only took part in the English language, we compare our results with the accuracy of the contender approaches on only the English language. It should also be noted that the results of the performance in the PAN-CLEF challenge were calculated on a different test dataset that was not available to us. Thus, we compare our 10-fold validation accuracy with the results given by the organizers of the challenge for the top performers. The results are shown in Table 4.8.

<table>
<thead>
<tr>
<th>approach</th>
<th>accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  [13]</td>
<td>0.75</td>
</tr>
<tr>
<td>2  [72]</td>
<td>0.735</td>
</tr>
<tr>
<td>3  deborjavalero20 ¹</td>
<td>0.73</td>
</tr>
<tr>
<td>3  This work</td>
<td>0.73</td>
</tr>
</tbody>
</table>

Table 4.8: The performance of the final picked fake news spreader classifier (GB) compared with the performance of the top contenders in the pan-clef challenge

By observing Table 4.8 we can see that the approach proposed in this work is ranked 3rd with a tie out of all the 65 teams that participated in this challenge, which denotes that our learnt model has a very satisfactory performance.

4.3 Explanations

In this section, the explanation techniques as specified in the previous chapter will be applied and their results will be discussed. We will review the results of the explanations on the global level by utilizing ELI5 and SHAP and afterwards we will plot the partial dependencies of the most important features. For the global explanations, we will first offer the figures for the tabular features and then for the textual features.

¹No citation could be found
4.3.1 Global level

Explanations in the global level can offer highly useful insights as they show us how the model "thinks" as a whole. In other words, we can see in detail which are the most important features of a model when it makes a prediction after taking into account all cases. The feature importance for the tabular features of the fake news spreader classifier can be seen in Figure 4.1 and Figure 4.2.

<table>
<thead>
<tr>
<th>Weight ± Standard Error</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.030 ± 0.0042</td>
<td>Clout</td>
</tr>
<tr>
<td>0.027 ± 0.0065</td>
<td>avg_word_count</td>
</tr>
<tr>
<td>0.023 ± 0.0073</td>
<td>openness</td>
</tr>
<tr>
<td>0.022 ± 0.0116</td>
<td>hashtags_count</td>
</tr>
<tr>
<td>0.018 ± 0.0161</td>
<td>capitalized_count</td>
</tr>
<tr>
<td>0.018 ± 0.0090</td>
<td>neuroticism</td>
</tr>
<tr>
<td>0.018 ± 0.0108</td>
<td>anxiety</td>
</tr>
<tr>
<td>0.017 ± 0.0115</td>
<td>Authentic</td>
</tr>
<tr>
<td>0.016 ± 0.0065</td>
<td>textblob_polarity_score</td>
</tr>
<tr>
<td>0.015 ± 0.0080</td>
<td>extraversion</td>
</tr>
<tr>
<td>0.014 ± 0.0065</td>
<td>agreeableness</td>
</tr>
<tr>
<td>0.013 ± 0.0060</td>
<td>conscientiousness</td>
</tr>
<tr>
<td>0.012 ± 0.0080</td>
<td>avoidance</td>
</tr>
<tr>
<td>0.012 ± 0.0053</td>
<td>vader_compound_score</td>
</tr>
<tr>
<td>0.010 ± 0.0098</td>
<td>url_count</td>
</tr>
<tr>
<td>0.010 ± 0.0094</td>
<td>user_mentions_count</td>
</tr>
<tr>
<td>0.009 ± 0.0078</td>
<td>full_capitalized_count</td>
</tr>
<tr>
<td>0.009 ± 0.0154</td>
<td>Tone</td>
</tr>
<tr>
<td>0.008 ± 0.0033</td>
<td>joy</td>
</tr>
</tbody>
</table>

Figure 4.1: The feature importance of the top 20 features as given by ELI5.

We can see that the top 20 features have quite many differences in the 2 techniques. ELI5 places the analytic feature as the top feature with the clout feature a close second, while SHAP places the textblob polarity score first and the hashtag count as a close second. The two methods however share many similarities considering that both of the 2 top features are in the other’s top 10. In the top 12 features, they also share the same 9 features, albeit in slightly different positions.

In Figure 4.3, we can see SHAP’s summary plot. In this plot we can see the top 20 features as given by the feature importance but also inspect individually how each feature affects the predictions. Each point represents one instance and the coloration of the points show the intensity of the value from blue which means low to red which means high. As our model was trained by using [0,1] as classes (0 is real news spreader; 1 is fake news spreader), the term negative here means that the feature is contributing to the prediction of real news spreaders while positive
to the prediction of fake news spreaders.

By inspecting the plot, we can make some interesting conclusions. For example, high values of Textblob’s polarity score, affect the prediction negatively (contributes to the “real news spreader” class) while low values affect the prediction positively (contributes to the “fake news spreader” class). The same thing can be said for Tone and vader compound score. This makes sense as all of these three features essentially capture the sentiment of the text and whether it is positive or negative. This means that negative sentiment implies someone is a fake news spreader while positive sentiment implies the opposite. The rest of the LIWC features, namely Clout shows us that users with high Clout contribute to fake news spreading behavior, while users with lower Clout contribute to real news spreading behavior. Lower levels of Analytic also contribute to real news spreading behavior but high levels of Analytic don’t have a significant effect towards any label. Finally, low levels of the Authentic feature denote fake news spreading behavior and high values denote real news spreading. The final sentiment feature which is sadness, shows that high sadness has a slight impact towards real news spreading while the opposite, i.e. low sadness has an even slighter impact towards fake news.
With regard to the readability features, when hashtag count is low it doesn’t affect the prediction but when it is high it seems to affect the prediction negatively, which means that real news spreaders generally use many hashtags. High amount of user mentions denote real news spreading behavior while low amount of user mentions don’t steer the prediction either way. Meanwhile, high levels of capitalized count generally denote fake news spreading behavior while low levels real news spreading. For the retweets count and the emoji count, one can draw the conclusion that high levels contribute to real news spreading while low levels have neutral effects.

Looking at the personality features, we can deduce that users with high levels of anxiety, agreeableness and openness generally contribute negatively to the prediction, while lower levels of these features contribute positively. No significant impact on either side can be seen by looking at the conscientiousness, neuroticism
or avoidance features.

Next, we will apply the same techniques for the classifier created with only the textual features. The results can be seen Figure 4.4.

![Figure 4.4: Global explanations for the textual features as given by ELI5 and SHAP](image)

By inspecting Figure 4.4, we can once again deduce that the two approaches have many similarities as in the top 14 features given by SHAP the 5 are also in the top 9 of ELI5. These are a lot of common features considering there are 1000 of them in total. It should be noted that in the case of textual explanations the results cannot be condensed in the same manner as in the tabular explanations as in this case we have 1000 features. Thus, for a human to analyze all of them and make comprehensible insights is not an easy task. The methods above just showed the top 20 features out of those 1000. ELI5 also didn’t yield satisfactory results as it only applied weight to 10 features out of all the 1000, which is questionable.

### 4.3.2 Partial dependency plots

Partial dependency plots show the effect each of the feature has on the final prediction. The PD plots for the 4 most important features according to SHAP will be shown next. SHAP was chosen as it is considered to be more stable than ELI5. The PD Plots for all 28 features are all available in the github repo and can be analyzed there in more depth.

In all of the plots below the y axis shows the the effect that the feature under inspection has on the final prediction. The red line is at the value of 0 in the x
axis and denotes that the effect is completely neutral. Values less than 0 have a negative effect on the prediction, while values higher than 0 have a positive effect. As in the summary plot given by SHAP, the term negative here means that the feature is contributing to the prediction of real news spreaders while positive to the prediction of fake news spreaders. The x axis denotes the value that the feature can take. Since, the features were normalized to the [0,1] range before training the classifier, all features will have these as min and max values.

Figure 4.5: Textblob polarity score partial dependency plot

In Figure 4.5, we can safely assume that the polarity score as given by Textblob starts with a neutral effect and then at the value of about 0.4 starts having a negative effect (contributes to the “real news spreader” class) on the prediction that keeps increasing and level out at about 0.6 and thereafter stays negative. This could be explained due to the fact that values that are greater than 0.5 in TextBlob (since we normalized the values) are considered to reflect positive texts and real news spreaders generally write in a more positive manner.

By inspecting Figure 4.6, we can deduce that the hashtag count feature has a consistent and stable negative effect (contributes to the “real news spreader” class) on the prediction regardless of its value. The only exception is when there are very few or none hashtags which then renders the feature neutral.

Moving forward, in Figure 4.7, we can see that the tone feature doesn’t have a big overall neutral effect (doesn’t contribute much to either the “real news spreader” or the ”fake news spreader” class). We can however see that lower values of tone have a slightly positive impact, while higher values have a negative impact. This
Figure 4.6: Hashtag count partial dependency plot

Figure 4.7: Tone partial dependency plot
makes sense, as the tone feature built by LIWC contains negative and positive emotion in the [0,1] range after normalization. Negative emotion thus could be attributed to fake news spreading while positive emotion to real news spreading.

![Figure 4.8: Slang count partial dependency plot](image)

Last but not least, by analyzing Figure 4.8, we can safely assume that the existence of only a few slang words has a positive impact on the prediction. When the slang count is 0, or really close to 0 the effect of the feature is neutral. Surprisingly, as the slang count increases the effect on the fake news spreader also slightly decreases (although the effect is still present).

### 4.4 MANIFEST evaluation

In this section, we evaluate the performance of the whole MANIFEST procedure with regards to phase C. In more detail, we present the results of the fidelity evaluation, the homophily evaluation and finally we give examples of output from our model as qualitative evaluation.

#### 4.4.1 Fidelity evaluation

As established in section 3.4.3, we use fidelity to evaluate our approach with the comparison of the classifications made on the initial post by the fake news
spreader classifier and the classifications made when using the linear model in both our datasets. Fidelity can show us to what extent a simpler model is able to accurately replicate a more complex model. For this reason, it is an appropriate measure to evaluate if the linear model is able to accurately classify a piece of text as possibly containing fake news or not.

To have a more fair calculation for the fidelity, we excluded the initial posts that had the same label in all the replies (e.g. A tweet was replied to 10 times, and all 10 individuals that left those replies are labeled by the fake news spreading classifier as fake news spreaders) as the logistic regression as implemented by sk_learn is not able to handle a case when there is only 1 class in the training data. After this filtering procedure, we are left with 1122 posts for the us_election out of the total 1365 posts and 240 posts out of the total 303 posts for the covid dataset.

We calculate the fidelity for the rest of the sentences for both datasets (US elections and Covid) and for both latent representations (BOW and TF-IDF) as explained in 3.4.2. Fidelity is 1, if linear model and fake news spreading classifier agree on the same classification; 0 otherwise. Finally, we computed the average for all sentences in each dataset. The results can be seen in Table 4.9.

<table>
<thead>
<tr>
<th>dataset</th>
<th>fidelity</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BOW</td>
<td>TF-IDF</td>
</tr>
<tr>
<td>us_elections</td>
<td>0.85</td>
<td>0.88</td>
</tr>
<tr>
<td>covid</td>
<td>0.70</td>
<td>0.71</td>
</tr>
</tbody>
</table>

Table 4.9: Fidelity average for both datasets and both vector representations. The higher the better.

More specifically, in the US Elections Dataset, for the BOW vector representation, 957 out of the 1122 total posts were classified as having the same label by both the fake news spreader classifier and the linear model and respectively for the TF-IDF representation 992 out of the 1122 total posts. For the Covid dataset, for the BOW representation, 168 posts out of the total 240 were found to have the same classification and for the TF-IDF representation, 170 out of the 240 total sentences.

With the above taken into account, we can come to the conclusion that the simpler linear model is able to accurately predict the same label with very high success in the us_election dataset and with lower success for the covid dataset. For both datasets however, it can become apparent that the simple linear model learns to successfully imitate the more complex fake news spreader classifier on both datasets. Looking at the different vector representations, the TF-IDF representation outperforms the BOW approach in all cases, which means it is a more appropriate way to represent the sentences in the whole MANIFEST procedure.
Next up, we are going to produce the accuracy curves for the logistic regression model. These curves include the Receiver operating characteristic (ROC) curve and the precision-recall curve. The ROC curve is a useful tool that helps in visualising the trade-off between the TPR (true positive rate) and FPR (false positive rate) by adjusting the different thresholds for making a classification and calculating the TPR and FPR for each threshold. The AUC (are under the ROC curve) can summarize the performance of a model with values between 0.5 which denote a no skill classifier and 1.0 denotes a perfect classifier. In other words, the closer the ROC curve reaches the top left corner the better the performance of the model.

Similarly to the ROC Curve, the precision-recall curve calculates the values of precision and recall for different thresholds. In this case, a no skill classifier is denoted with a horizontal line with precision equal close to 0. On the other hand, a perfect classifier would have maximum precision and recall and would be placed in the top-right corner. Thus, the closer the curve is to the top-right corner the better the performance of the model.

We can see in Figure 4.9, that our logistic regression model has an overall good performance with respect to learning the fake news spreader classifier as the curve for the ROC curve tends to reach close to the top left corner and respectively for the precision recall curve as it tends to reach the top right corner.

### 4.4.2 Homophily evaluation

For the rest of the sentences that were excluded from the fidelity evaluation, a different form of evaluation was conducted to evaluate whether the echo chamber effect holds true (when all the users replying have the same label, then the author of the initial post also has the same label). When all the replies hold the same label, MANIFEST applies to the initial post the label of the replies. The accuracy metric...
below checks whether that label is the same as the label given to the initial post by the fake news spreader classifier. The accuracy is 0 if the labels are different and 1 if the labels are the same. The results for both datasets can be seen below by taking the average of the accuracy calculated over all sentences.

<table>
<thead>
<tr>
<th></th>
<th>us_elections</th>
<th>covid</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy</td>
<td>0.98</td>
<td>0.97</td>
</tr>
</tbody>
</table>

Table 4.10: Homophily evaluation for both datasets. The higher the better.

In more detail, for the US Elections Dataset, 224 sentences out of 228 (about 0.98 or 98% accuracy) had the same label while for the Covid dataset 61 sentences out of a total of 63 (about 0.97 or 97% accuracy) sentences had the same label.

### 4.4.3 Qualitative evaluation

In qualitative evaluation, we present the explanations as given by MANIFEST on 12 input tweets (6 from us _election dataset and 6 from the covid dataset). Each time, the first 3 will belong to the fake news spreading classification and the latter 3 will belong to the real news spreading classification. For all examples, we will showcase the selected examples of real news and fake news replies and the top features along with their weights as assigned by the linear model (including the bias/intercept). Since in the fidelity evaluation, the TF-IDF vector representation was shown to outperform the BOW vector representation we choose to offer all results for the TF-IDF vector representation.

In Table 4.11, we can see how MANIFEST returns the explanations for three given tweets for the us _election dataset. Firstly, we can see the tweet that is under investigation. Afterwards, we have the two closest latent replies to the original tweet with the opposite label (real news spreaders replies) and afterwards the two closest from the opposite label (fake news spreaders replies). Lastly, the top features that contributed to the classification along with their weights are offered on descending order, including the bias assigned by the linear model on the top. Both of these tweets have been assigned as fake news by the fake news spreader classifier.

Looking at the first tweet, we can see that the author is making appalling remarks on both presidential candidates by comparing them to a ”clown” and a ”gaffe-prone plagiarist”, while also accusing the son of one of corruption. Afterwards he denotes of choosing the lesser of two evils as president Trump because he brings peace between Israel and three Arab countries. The first real news reply comes in conflict with the author by asking him where he sees the peace when Israel has annexed Palestinian lands, while the second doubts the claims made about
Table 4.11: The explanation for 3 given tweets of the us_election dataset classified as fake news spreading.

<table>
<thead>
<tr>
<th>tweet</th>
<th>real news spreaders replies</th>
<th>fake spreaders replies</th>
<th>top features</th>
<th>weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>In a choice between a clown and a gaffe-prone plagiarist tarred by his son’s alleged corruption, Trump deserves the vote for at least one reason: bringing peace between Israel and three Arab countries, with two more in the bag.</td>
<td>Peace between 3 Arab countries and Israel? So where does Israel’s annexation of Palestinian land fit into this ‘peace’?</td>
<td>Bringing Israel and these Arab countries together is a miracle for me.</td>
<td>bias -0.085</td>
<td></td>
</tr>
<tr>
<td>Alleged you said not proven so why bring him in and trump he is racist, and baffling buffoon... so clown is better</td>
<td></td>
<td>Absolute clown</td>
<td>bringing 0.112</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trump -0.099</td>
<td></td>
</tr>
<tr>
<td>Has anyone in history campaigned more intensely in more places in a short amount of time than @realDonaldTrump? This guy doesn’t need the money nor power. He’s doing it because he loves the USA!</td>
<td>The only thing Trump loves is himself, power, attention, and money. He certainly has no love or respect for America. Patriots are voting him out on Tuesday.</td>
<td>He’s doing it for money and power</td>
<td>bias 0.056</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>doing 0.143</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>power 0.029</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>thankful and grateful for our president trump</td>
<td>loves -0.014</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>money 0.011</td>
<td></td>
</tr>
<tr>
<td>I’m soo scared trump will lose! As Biden as president will effect the whole world! Fingers crossed Americans make the right choice!</td>
<td>You’re right. Biden is the one that tells the world Democracy is broken in America and mainstream media controls elections.</td>
<td>Fingers crossed Trump will be re-elected. I’ve also a £20 bet on</td>
<td>bias -0.280</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trump 0.160</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>fingers 0.092</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>crossed 0.092</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>president -0.058</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

the son, stating that they are not proven. On the other hand, the first reply from the fake news replies seems to agree with the author and states that peace Israel and Arab countries would be a miracle. The second reply doesn’t really add much and just states ”absolute clown”. Looking at the top features we can see words with positive meaning such as ”peace”, push the classification towards the real news (since the real news class is 0), while negative words such as ”clown” push it towards fake news.

The author of the second tweet is admiring president Trump’s campaign and states that the ex-president doesn’t care about money or power. The two real news replies both disagree with the author and state the opposite. Looking at the top fake news reply, we can also see that there is the same belief that president Trump does it all for money and power. The second fake news reply instead praises president Trump. The top features show positive words such as love, contribute to the real news class, while negative words in the context of this tweet such as power and money contribute towards the classification of fake news.
The author of the final tweet that was classified as fake news spreading is saying he is scared that Trump will lose and urges Americans to make the right choice. The first real news reply has a sarcastic tone that makes fun of the author of the fake news post. He states that Biden is the one that is saying the mass media control the USA while those are exactly the beliefs of Trump. The two fake news replies selected are also highly interesting examples showcasing false information spreading behavior. The second one especially states that president Biden wouldn’t make it for more than 100 days in the white house and that he is a puppet of the far left. The top words correctly assign positive weight to the word trump, since in this context it is the object of the discussion and also assigns to the word president which could describe both candidates a slightly negative weight.

Next, we will review another set of tweets from the same dataset that were classified as real news spreading.

<table>
<thead>
<tr>
<th>tweet</th>
<th>real news spreaders replies</th>
<th>fake news spreaders replies</th>
<th>top features weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>Am I the only one who finds the whole ‘describing armed people who try to disrupt elections, intimidate voters, and threaten candidates with death’ as ‘Patriots’ a little bit too 1930s Germany?</td>
<td>Hitler also used the slogan &quot;make Germany great again&quot;</td>
<td>If Trump wins, it would give this lot a licence to autocracy in UK.</td>
<td>bias -0.537</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>people 0.112</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trump -0.099</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Germany -0.083</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Patriots 0.081</td>
</tr>
<tr>
<td>We don’t want to hear anything like voters suppression as it happened in the past again in the USA. By now, USElections2020 should be more fair than what happened in the past. We are watching aggressively from Nigeria and Africa.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Illusions of European strategic autonomy must come to an end: Europeans will not be able to replace America’s crucial role as a security provider.</td>
<td>It’s this weird situation where European strategic autonomy = spending more money, but relying dependent on USA also = spending more money.</td>
<td>Macron will be disappointed... he was dreaming of huge amount of arms sale to EU army project...</td>
<td>bias -0.804</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>end 0.137</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>strategic -0.024</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>autonomy -0.024</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>European -0.015</td>
</tr>
</tbody>
</table>

Table 4.12: The explanation for 3 given tweets of the us_election dataset classified as real news spreading.

In Table 4.12, we can see how MANIFEST returns the explanations for three given tweets for the us_election dataset classified as real news spreading. By ob-
serving the first tweet, we can see that the author is making the comparison that the period that the elections took place resembles with the beginning of nazi Germany in the 1930s. The real news replies make logical statements by drawing a parallel with president Trump’s “make America great again”. Looking at the fake news replies, both of these tweets share extremist views.

The author of the second tweet implies he is from Nigeria and states that he doesn’t want to hear about voter suppression in the US like it has happened in the past, which is indeed true that voter suppression cases have been reported. The two real news replies share a similar sentiment with the original author with the first one stating that there should be no violence in the elections and the second one wondering if voter suppression does really happen in western countries. The fake news replies are completely opposite with the first one stating “Democratic USA my ass” and the second one saying that voter suppression in the USA is completely implausible (despite the fact that there have being cases in the past).

Looking at the third and final tweet, the original author is stating the opinion that Europeans will not be able to replace the role that the US has as a security provider. The two real news replies poster seem to agree. The two fake news spreaders focus instead on Macron and how his dream of an EU army would not work if the EU was independent from the security of the US and that Macron should instead offer NATO membership to Russia to gain their security. The top features also shed some light as the word ”end” has a negative connotation (i.e. positive weight by referring to the end of the dreams of strategic autonomy that the EU has) and words like European, strategic and autonomy are instead contributing to the real news class.

Three examples for the covid dataset classified as fake news spreading are given in Table 4.13. The first example confuses coronavirus with electoral fraud, with reference to misinformation. Short answers from trusted users present the logical voice and reassure while from unreliable users opinions related to electoral fraud and other conspiracy theories are reported. MANIFEST explanation shows that references to the election result tend to push the categorization towards fake news. Looking at the top features, we can see that negative connotation words such as ”shit” and ”fraud” push the prediction towards fake news, while positive words such as ”kind” push it towards real news.

The second example criticises a public figure appointed in the covid task force by president Biden. Responses from credible users indicate either that these views are terrifying or they are trying to provide arguments to support the situation. On the contrary, susceptible users agree and magnify the criticism and follow other extremist views. The top words do not shed much light into the classification result in this example.

The final example refers to a piece of news that suggests that ice cream has been found containing traces of Covid-19; a rather odd statement that could be confused
CHAPTER 4. EXPERIMENTATION AND RESULTS

<table>
<thead>
<tr>
<th>tweet</th>
<th>real news spreader replies</th>
<th>fake news spreader replies</th>
<th>top features</th>
<th>weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newsmax host warns Democrats will “round up all the Republicans” into “COVID internment camps”. There’s so much talk about the “election fraud” rhetoric leading to the murderous mob, but this is the kind of shit that led up to it and it’s pumped out by conservative media all day every day</td>
<td>I feel as though so many Republicans are genuinely worried about this kind of stuff, and there’s no way to convince them that their lives are not in danger. The government isn’t taking over everything; no matter how hard you try</td>
<td>How many Dem election losses are blamed on “voter suppression”? A different way of saying election fraud.</td>
<td>bias</td>
<td>0.158</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>shit</td>
<td>0.079</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>election</td>
<td>0.077</td>
</tr>
<tr>
<td></td>
<td>I was thinking about the “government buildings” bit in the warnings. State health departments should get a head’s up</td>
<td>Just the Republican QAnon Enablers who PROMOTED Election Fraud, Sedition, treason and inciting a MINDLESS ZOMBIE Neo-Nazi Supremacist Mob to Insurrection. So if that means ALL REPUBLICANS, So be it...</td>
<td>kind</td>
<td>-0.059</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>fraud</td>
<td>0.049</td>
</tr>
<tr>
<td>The worst news of this week is that Biden has appointed Andy Slavitt to his covid task force. Slavitt is a REALLY bad guy. A charlatan, and a habitual liar and self-promoter who has spread terror for his own aggrandizement. Here is selling a mask he claims “deactivates” covid</td>
<td>how do people listen to that stuff its just 24/7 fear porn.</td>
<td>Most of the picks have been good so far. No one’s going to buy 1000. Every f*cking person Trump appointed was a piece of shit, so don’t flip out about one controversial pick from Biden and pull the “both sides are the same” bullshit I see many are doing on here.</td>
<td>bias</td>
<td>-0.098</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>biden</td>
<td>0.104</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>covid</td>
<td>-0.074</td>
</tr>
<tr>
<td></td>
<td>It seems that his remit is limited (temporary) and is confined to communications/messaging around vaccines and coordination in getting vaccines out. We will see how effective he will be at this. Giving even bad actors the benefit of the doubt for the moment.</td>
<td>Biden will pick a cabinet of the usual suspects and America will suffer.</td>
<td>appointed</td>
<td>0.051</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bad</td>
<td>-0.026</td>
</tr>
<tr>
<td>Ice cream ‘tests positive’ for Covid with 4,800 tubs affected as probe launched</td>
<td>Who tests ice cream.</td>
<td>Who eats ice cream in January?</td>
<td>bias</td>
<td>-0.618</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>tests</td>
<td>-0.089</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>positive</td>
<td>-0.059</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>tubs</td>
<td>-0.055</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>affected</td>
<td>-0.043</td>
</tr>
<tr>
<td>Even the ice cream is affected. At least there is no cases of affected alcohol</td>
<td>Cant people see, they want you to have covid, so they can get rid of the population, where are all the thousands of bodies and funerals going please? The government are liars. Besides has anyone seen the video of Sadiq Khan taking the vaccine with the cap on?</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.13: The explanation for 3 given tweets of the covid dataset classified as fake news spreading.

as fake news. Most texts are not very polarizing in this example with the most notable text being the one given by the second fake news spreader which contains and propagates extreme conspiracy theories. It should be noted that in this specific example there was a mis-classification between the fake news spreader classifier and the linear model. The classifier predicted "fake news" while the linear model "real news", which explains why all the weights have negative values. This is also interesting because it shows that sometimes when the linear model doesn’t agree with the fake news spreader prediction it would not necessarily mean that it was wrong. This could also suggest that the even though the fidelity is not perfect it could be better for the overall explainability as when the linear model doesn’t agree its reasoning can be explained.

Three examples for the covid dataset classified as real news spreading are given in Table 4.14. The first example relays something that the CDC stated. The real news spreader replies either agree with that statement or oppose it but with logical arguments. On the other hand both the fake news spreader replies have hints
CHAPTER 4. EXPERIMENTATION AND RESULTS

Table 4.14: The explanation for 3 given tweets of the covid dataset classified as real news spreading.

<table>
<thead>
<tr>
<th>tweet</th>
<th>real news spreader replies</th>
<th>fake news spreader replies</th>
<th>top features weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDC says more contagious coronavirus variant will increase the percentage of population immunity required to get the pandemic under control</td>
<td>Taking measures to reduce transmission now can lessen the potential impact of B.1.1.7 and allow critical time to increase vaccination coverage.</td>
<td>All they say is more contagious. Never more deadly.</td>
<td>bias -0.295</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>contagious 0.140</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>cdc 0.073</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>population -0.063</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>variant 0.055</td>
</tr>
<tr>
<td>What nonsense. If you get the more contagious version and that spreads, all it will do is enable the population to get infected and get past this virus quicker - developing natural immunity.</td>
<td>This variant could have been caused by Johnson’s deliberate and lethal herd immunity experiment - essentially Cummings/Johnson breeding a more deadly variant - both criminals should certainly now be prosecuted.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Why was Nicola Sturgeon not fronting her Covid update today, it’s unacceptable there are thousands of pensioners relying on her, not a substitute, it must be her.</td>
<td>I tuned into the Scottish &quot;government&quot; Covid update for the first time today to see what all the fuss was about.</td>
<td>Rumour is she is off to Ireland to spread mischief there.</td>
<td>bias 0.099</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>unacceptable 0.111</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nicola -0.049</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>today -0.040</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>update -0.021</td>
</tr>
<tr>
<td>San Francisco has led from the very beginning, it’s why we have one of the highest rates of testing per capita and one of the lowest death rates in the entire country. We’re ready to do it again with vaccines.</td>
<td>If you work in San Francisco as a healthcare provider, and are employed in San Francisco, but do not live there, can you still go to a SF vaccine site?</td>
<td>What percentage of vaccines received by San Francisco have not been used as of today?</td>
<td>bias -0.317</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ready -0.076</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>death -0.037</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>announcement -0.031</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>vaccines 0.021</td>
</tr>
<tr>
<td>We’re ready for this! Let’s do it!</td>
<td>Too bad SF, Bay Area and California have mucked up the rollout of vaccines. Can’t blame the feds for the incompetence of local, county and state officials.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

of conspiracy theories. Polarizing words such as "contagious" and "variant" are contributing towards fake news while more neutral words such as "population" contribute to the real news class.

The second example contains text that complains about a public figure not appearing on a programmed Covid update. The real news replies this time contain non relevant information while the second one sarcastically states that it’s the end of the world because she didn’t appear. The fake news spreader replies on the other hand both spread propaganda about the public figure. The top words also shed some light as "unacceptable" a word with negative connotation has the biggest contribution to the fake news spreading class.

The third and final example contains some motivational text about San Francisco’s good work at handling the pandemic thus far. The first real news spreader asks a general question (which would certainly not belong to fake news spreading behavior) while the second real news spreader replier agrees with the sentiment of the original poster. The fake news spreaders seem to completely disagree with the
original poster stating that San Francisco has done a horrid job either by hinting at unused percentage of the vaccines or by having messed up the roll-out of the vaccines.

To conclude, all the above examples in the us_elections and covid datasets are remarkable examples that show that MANIFEST is able to give a clueless reader insight as to why the initial tweet he is reading could potentially contain fake news by offering them the two closest replies from each class as well as the top features to aid in their judgement and help them better understand the reason behind the more complex fake news spreader classifier’s decision.
Chapter 5

Conclusions

In this thesis, MANIFEST was introduced, an approach that is able to offer example based explanations for helping a user make a decision on whether a tweet contains false information or not on Twitter, by utilizing the predictions made by a complex fake news spreader user classifier on the users that took part in the discussion, matching the text they left in their reply with that prediction and finally by learning a linear model in that new space. The explanations returned by MANIFEST consist of a set of example based replies that have the same label as the text under inspection, a set of replies that have the opposite label and also by the most important words that influenced the classification.

Based on the global explanation techniques applied on the classifier of phase A, an in-depth exploration of the social and psychological characteristics for both fake and real news spreaders was attained. Our outcomes suggest that negative sentiment polarity, higher levels of clout, lower levels of anxiety, agreeableness and openness as well as specific language use are traits associated with fake news spreading behavior.

Based on the fidelity evaluation made on the previous chapter we can deduce that the linear model is able to accurately imitate the predictions of the more complex fake news spreader classifier with a high accuracy which would suggest that the assumption that a fake news spreader will continue to spread fake news when replying to other users and the respective for real news spreaders holds true. Based on the homophily evaluation, we can deduce that the echo chamber effects holds true in Twitter thread conversations and thus it is reasonable to assign to an original post the post of all the replies if all of them hold the same label. Based on the qualitative evaluation, we come to the conclusion that the explanations offered in the form of the closest replies from each label along with the top words and their weights are reasonable and intuitive and could prove fruitful for aiding users better understand whether a post contains fake news or not.

It should be noted that MANIFEST is not targeted for fake news detection
but rather as an extra tool to help users understand why a post could contain fake news. Fake News detection systems can’t perfectly differentiate between real and false information since not even humans can but with the addition of explainable approaches they can aid the user in making a more educated final decision.

5.1 Limitations and Future Work

The performance of MANIFEST and consequently the quality of explanations relies strictly on the performance and accuracy of the fake news spreader classifier; a better classifier would lead to more accurate results and thus more accurate explanations.

Since the accuracy of the fake news spreader classifier created in the context of this work attains at best 0.73 accuracy it means that about 1 in 4 users will still be wrongly classified as potential fake news spreaders or the opposite. This has a direct impact on the quality of the linear model and in consequence in the final explanations.

In this line, a creation of a novel bigger fake news spreader dataset which also contains more rich information aside from the last 100 tweets regarding other explicit Twitter metadata (e.g. user description, user picture, number of followers, number of favorites etc.) would be of great first step to further improve the fake news spreader classifier.

More research could also be done to extract other valuable implicit features from the explicit features provided by Twitter. An example of such a feature would be the educational background of a user. By using explanation techniques there could also be some insight on whether that feature is important in the task of fake news spreading.

Another interesting research direction would be to experiment with different linear models when approximating the decisions of the more complex classifier in a specific twitter thread. A good first experimentation would be to use a linear regression model instead of a classification one (e.g. usage of the LASSO model). LASSO is a technique applied in linear regression models and is quite successful for the task of feature selection. The drawback however is that the number of features that LASSO returns depends on a regularization parameter.

Finally, a human evaluation of MANIFEST (e.g. via crowd-sourcing) would be definitively required for assessing the quality of the explanations.
Bibliography


97


[38] B. D. Horne and S. Adali. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news, 2017.


